ST. MARTIN'S ENGINEERING COLLEGE
An UGC Autonomous Institute
NBA & NAAC A+ Accredited
Dhulapally, Secunderabad-500 100
www.smec.ac.in

PROJECT REPORTS OF ECE

A

PROJECT REPORT

On

RECURSIVE BLOCK BASED KEYPOINT MATCHING COPY MOVE IMAGE FORGERY DETECTION

Submitted by

1)Ms. A. Sriya (17K81A0401)
2)Ms. A. Varshika (17K81A0402)
3) Ms. V. Bindu Priya (17K81A0456)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Dr. A. Anand

Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST.MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500100

NBA &NAAC A+ Accredited

JUNE 2021

St.MARTIN'S ENGINEERING COLLEGE
UGC AUTONOMOUS

Dhulapally, Secunderabad – 500100
NBA &NAAC A+ Accredited

BONAFIDE CERTIFICATE

This is to certify that the project entitled "Recursive Block Based Key Point Matching Copy Move Image Forgery Detection" is being submitted by **1. Ms. A. Sriya [17K81A0401], 2. Ms. A. Varshika [17K81A0402],**
3. Ms. V. Bindu Priya [17K81A0456] in partial fulfillment of the requirement for the award of the degree of **BACHELOR OFTECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Internal Guide	**Head of the Department**
Dr. A. Anand Department of Electronics and Communication Engineering	Dr. Hari Krishna Department of Electronics and Communication Engineering

Internal Examiner External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **A. SRIYA** WITH ROLL NO.**17K81A0401**, **A. VARSHIKA** WITH ROLL NO.**17K81A0402**, **BINDU PRIYA** WITH ROLL NO.**17K81A0456**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**RECURSIVE BLOCK BASED KEYPOINT MATCHING FOR COPY MOVE IMAGE FORGERY DETECTION**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of '**Bachelor of Technology in Department of Electronics and Communication Engineering**', session: 2017-2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project work entitled '**Recursive Block Based Key point Matching for Copy Move Image Forgery Detection**' is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

A. Sriya	A. Varshika	V. Bindu Priya
17K81A0401	17K81A0402	17K81A0456

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his valuable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to, **Dr. B. Hari Krishna**, **Head & Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept.project coordinator and our internal coordinator Dr. A. Anand Kumar, Professor,** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for along time to come.

We sincerely thank our class coordinator **Ms. A. Soumya**, **Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**,**Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

A. Sriya (17K81A0401) A. Varshika (17K81A0402) V. Bindu Priya(17K81A0456)

ABSTRACT

Images are used as legal proof in areas like forensic investigations. The authenticity of an image intended for investigation may adversely affect the result of such investigations, if the image is a manipulated one. In such fields, the detection of image forgery is very critical and sensitive. Among the prevailing methods, Block based methods divide the images into overlapping regular blocks and find the match between every block of the whole image. This method is found to be more accurate though computationally expensive. Whereas Keypoint based methods compute the key points of an image and find the match between those keypoints. If the image is forged by copy paste (splicing), there will be the highest number of keypoint matches between the corresponding regions. This method is found to be computationally efficient but the accuracy is less. The proposed method utilizes the advantages of both keypoint based and block-based forgery detection methods. We identify meaningful irregular blocks and the similarity of such blocks are measured using the number of matched SIFT keypoints. To identify whether the image is forged or not, an adaptive threshold is employed on the number of keypoint matches and judiciously decide whether to go for block based matching strategy or not for each block. We show that the proposed method achieves better detection rate without compromising the merit on the computational complexity of keypoint based forgery detection

TABLE OF CONTENTS:

LIST OF EQUATIONS-

LIST OF TABLES:

LIST OF FIGURES:

CHAPTER 1 : INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

- The authenticity of an image is crucial in many fields today. The most common ways to tamper a digital image is copy-paste forgery which is used to conceal objects or produce a non-existing scene.
- An image can be copied and pasted in two ways : Active approach and Passive approach.
- In Passive approach there are two more types of copying or forging images i.e., Independent and Dependent forgery. Independent forgery is defined as the methods that are used to detect the forgeries independent of forgery type but based on the artifact traces left during the process of re-sampling and due to lighting inconsistencies.
- This project deals with Dependent Forgery, to be more specific Copy paste (splicing) forgery.

1.2 OBJECTIVE OF THE PROJECT

The main objective of the project is to check whether the query image has been copy pasted into the test image i.e., whether a particular picture has been forged into another image.

1.3 SCOPE OF THE PROJECT

The main scope of the project is to identify the key points/features in the images which will help in determining whether the forgery is done or not.

1.4 INTRODUCTION OF PROJECT

In this day and age, digital image tampering has been made easy with widely available image editing softwares, such as Adobe Photoshop. The advancement of image editing softwares has reached a level such that image tampering can be done without degrading its quality or leaving obvious traces. This is alarming as images are now being presented as supporting evidence and historical records in various fields, such as in forensic investigation, law enforcement, journalistic photography and medical images. Moreover, in many instances, tampered images have appeared in the news or social media, such as the manipulated images of Iranian missile test released on July 9, 2008 by Sepah News, the official media arm of Iran's Revolutionary Guard. The tampered image, shown in Fig. 1 is aimed at exaggerating the country's military capabilities. The forgery is detected a day later when the same source released another image taken from the same angle at almost the same time, but with different content. The scientific community is also not spared from image tampering. stated that 20% of accepted manuscripts of the Journal of Cell Biology contains inappropriate figure manipulation. Hence, image tampering and detection have garnered substantial attention as manipulated images can be used to misrepresent their meaning with This paper is organized as follows: Section II presents the recent works on Copy-Move Forgery Detection, focusing on block based and keypoint based CMFD techniques. Section III presents the research methodology of the proposed CMFD technique, focusing on speeded up robust features (SURF) and Oriented FAST and rotated BRIEF (ORB) as the feature extraction methods and 2 Nearest Neighbour (2NN) and Hierarchical Agglomerative Clustering (HAC) as the feature matching method. Section IV presents the result of our proposed CMFD technique with a summary of the contribution and achievements of the work. Lastly, we conclude this work and provide potential future works.

1.5 INTRODUCTION TO CMFD:

CMFD is a non-intrusive detection technique where a portion of the image's content is copied and moved to another region of the same image. In general, tampered regions in copy move forgery can be in the form of an object, texture or wording. Manipulation of textures, such as ocean and grass fields can be used to hide details of an image and are generally harder to identify. Furthermore, copy move forgery can be associated with different geometric transforms, such as translation, rotation, scaling or multiple duplication of manipulated regions in a single image. Copy-move forgery detection can be segregated into two categories, namely block-based and keypoints-based CMFD techniques. Block-based techniques can be categorized into frequency transform, moment invariant, dimension reduction and intensity. In recent years, keypoints-based CMFD has garnered more attention.

1.6 KEYPOINTS-BASED CMFD TECHNIQUES

In keypoints-based CMFD, information of distinctive features (interest point) such as edge, blob and corners of an image are extracted and stored in the form of a feature vector. The feature vector is typically constructed using pixel information within a specific window of the interest point. With keypoints techniques, the preprocessing step of block division can be eliminated. Common keypoints-based CMFD techniques used are Scale Invariant Feature Transform (SIFT) and SURF. SIFT was initially introduced in the field of object recognition. SIFT is designed to be robust against scale and rotation and the algorithm performs well even with noise and changes in illumination. SIFT is used to detect salient points at different scales from the Difference of Gaussian (DoG) in scale space representation. Amerini et al. performed SIFT for CMFD and the implementation is tested on the MICC-F220. Compared to the work of Fridrich et al. and Popescu and Farid, CMFD with SIFT obtained a minimum of 10% increase in True Positive Rate (TPR) and reduction of computational time from 70.97s to 4.94s. Even though SIFT has successfully reduced the computational complexity of CMFD with the introduction of DoG, the number of feature vectors generated is still relatively high. This would impact the matching phase especially for images with high resolution. Hence, He et al. proposed the use of PCA to reduce the dimensionality of SIFT feature vectors. Apart from computational complexity, SIFT techniques also face difficulties in detecting flat copied regions. Mohamadian and Pouyan proposed the use of SIFT and Zernike moments. Zernike moments detection method can be performed on regions where SIFT features are not found. However, the approach would increase the computational time as both methods have to be applied on the image. SURF is the state of the art technique in CMFD . SURF was initially implemented in object recognition and adopted in CMFD due to its scale and rotation-invariant properties. Bo et al. demonstrated the use of SURF technique in the field of CMFD. The author extended Bay’s technique by increasing the length of the feature vector from 64 to 128. However, the experimental results were not shared in the paper. Mishra et al. [9] implemented CMFD with SURF and obtained a TPR of 70%. The concept of Hierarchical Agglomerative Clustering

(HAC) is introduced to perform grouping on the matched keypoints. The SURF and HAC combination was tested on the MICC-F220 database and achieved a TPR of 73%. Hashmi et al. [10] introduced a preprocessing step to CMFD by transforming images into dyadic wavelet domain (DyWT) before applying the SURF technique. The combination of SURF and DyWT is more robust and it is capable of sustaining various attacks. With the implementation, an increase in TPR of 76% is obtained. However, the authors did not discuss the tradeoff between computational time and performance improvement.

Image forgery detection is attracting the attention of scientists in computer vision, digital image processing, biomedical technology, investigation, forensics, etc. With popular and complicated technologies and powerful software tools in digital images, it is difficult to confirm if the image is original or not through naked eyes. This challenges researchers to develop algorithms and propose methods to detect the forgery in image. Upon the survey on IEEE and Elsevier, the number of publications on image forgery detection from 2000 increased rapidly in 2010 and more in the following years . An image can be faked by changing any characteristics including brightness, darkness or image parameters, or hiding information. Watermarking and digital signature are solutions of information security in which a security code is inserted in the image so these methods have information of a code and the original image. A question is asked in the case if there is no code or signature inserted or information of the original image, how to confirm its authenticity. Blind/passive techniques in which the detection is done in the tested image itself without any prior information are developed to solve the problem given.

According to blind/passive techniques are grouped into two kinds: copy-move and splicing. The copy-move is defined by cutting an image region and pasting it to another place in the same image while splicing is understood by cutting an image region and pasting it to a different image. Based on this classification, searching the regions having similar features in copy-move images or completely different regions in spliced images is the principle of forgery detection. Many techniques are proposed and used in this field but actually, they can solve only problems on copy-move or splicing separately. The dataset in the previous publications often consists of copy-move images or spliced images, not both in images. This paper proposes a method which can detect the forgery in images not only for copy-move or splicing but also for both. The literature review and proposed method are presented in part II and III. Simulation results and conclusions are shown in the following parts.

The technique of manipulating the characteristics of an image is image forgery. The most commonly used techniques are copy move, image splicing and steganography. Images are taken as legal proof in areas like forensic investigations. The authenticity of an image is very critical as a forged image can change the result of an investigation. With the advancement of image editing and manipulating software, image forgery has become an easy task. The forgery has become so advanced that it makes the detection of forged regions a difficult task. As a result there have been technological methods to prove the credibility of an image. Among different types of forgery, the copy-move method has developed so much that it has become very difficult to find it out at a glance. The method of copy-move forgery is to copy a part of the image and cunningly paste it to another part of the same image. Since the copied part of the image is pasted to the same image most of the image properties will be the same that makes detection difficult. One can rotate; scale, compress, and can even add noise to the forged region. To find out such forgery three methods are mainly used: Block-based methods, Key point based methods and combination of both block based and key-point based methods.

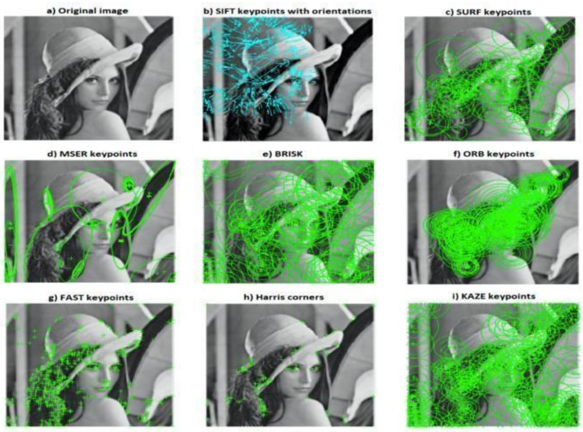


Figure 1.1Key Point Based Technique Example

1.7 BLOCK BASED CMFD TECHNIQUE

Block based method splits the image into overlapping blocks and applies a suitable technique to extract features on the basis of which the blocks are compared to determine similarity. Firstly the image is preprocessed i.e. Converted to grayscale.

Preprocessing is optional. Then the image is subdivided into overlapping blocks of pixels. For an image size of M × N and a block n size of bxb, the number of overlapped blocks is given by (M-b+1) x (N-b+1). On each of these blocks, a feature vector is extracted. After feature extraction matching is done. Feature vector depends on which feature has been used. Highly similar feature vectors are matched as pairs. Methods that are used for matching are lexicographic ordering on the feature vectors and nearest neighbor determination [9]. Any one from both can be used. The similarity of two features can be determined by different similarity criteria, e.g., the Euclidean distance. There are a number of algorithms that according to the features that are selected for the feature extraction.The block-based methods divide the images into overlapping regular blocks and find the match between every block of the whole image Some other methods are also used. Machine learning approaches are also used to detect forgery, like BusterNet. Various methods to detect forgery are compared. Block Based methods are more accurate but the segmentation of the image into overlapping blocks makes the method computationally expensive. Keypoint based methods identify the key points of an image and use it to find the copy-pasted forged region.

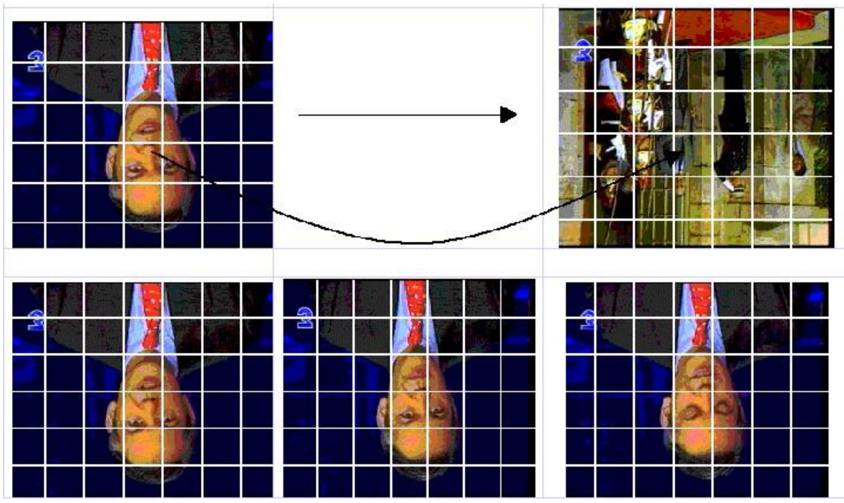


Figure 1.2 Block based technique example

The keypoint based methods differ in choosing the algorithms for detecting keypoints and the keypoint matching threshold for detecting whether the image is forged or not. Clearly the block based methods are computationally expensive as it requires an exhaustive search. A natural choice is to combine both the methods to achieve comparable accuracy of block based methods but without compromising on the computational advantage offered by keypoint matching based methods.

1.8 INTRODUCTION TO IMAGE PROCESSING

1.8.1 IMAGE

An image is a two-dimensional picture, which has a similar appearance to some subject, usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Figure 1.3 General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

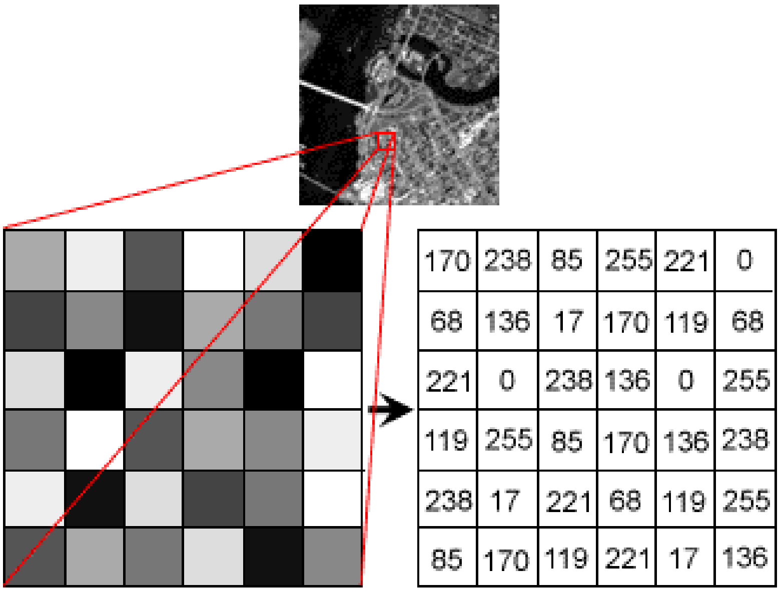


Figure 1.4 Image pixel

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
Transparency	Red	Green	Blue

Figure 1.5 Transparency image

1.8.2 IMAGE FILE SIZES:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

1.8.3 IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

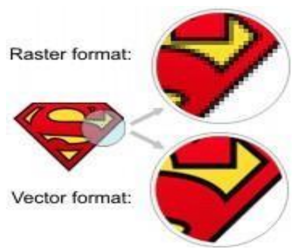


Figure 1.6 Resolution image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

1.8.4 RASTER FORMATS:

These formats store images as bitmaps (also known as pixmaps)

- **JPEG/JFIF:**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

- **EXIF:**

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

- **TIFF:**

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

- **PNG:**

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

- **GIF:**

GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

- **BMP:**

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

1.8.5 VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM:**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text.

All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

- **SVG:**

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

1.8.6 IMAGE PROCESSING:

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

1.8.7 FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:

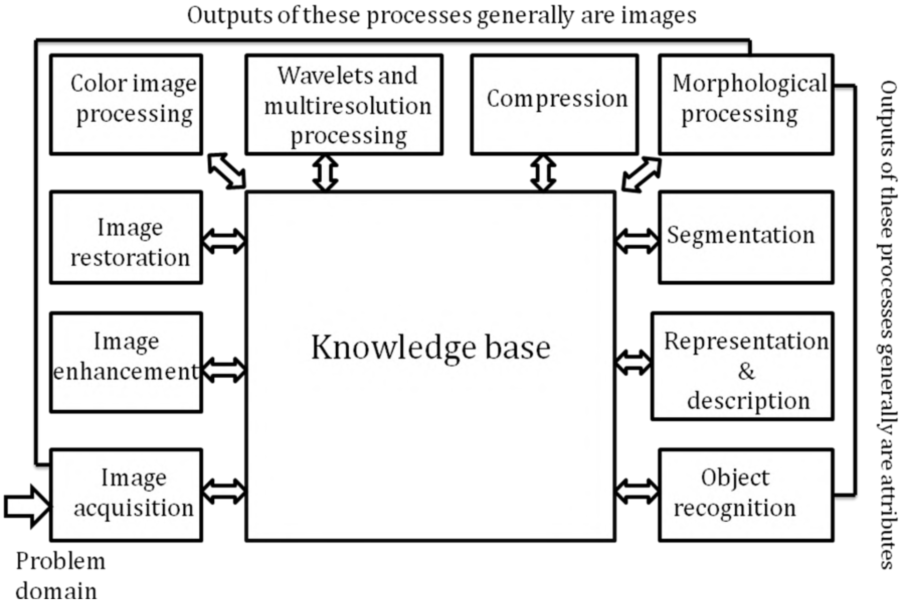


Figure 1.7 Image fundamental

1.8.8 IMAGE ACQUISITION

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Figure 1.8 Digital camera image

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Figure 1.9 Digital camera cell

1.8.9 IMAGE ENHANCEMENT

Image enhancementis among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of an interesting image. A familiar example of enhancement is when we increase the contrast of an image because “it looks better.” It is important to keep in mind that enhancement is a very subjective area of image processing.

Figure 1.10 Image enhancement

1.8.10 IMAGE RESTORATION

Image restorationis an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Figure 1.11 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, whereas removal of image blur by applying a deblurring function is considered a restoration technique.

1.8.11 COLOR IMAGE PROCESSING

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Figure 1.12 Colour & grayscale image

1.8.12 WAVELETS AND MULTI RESOLUTION PROCESSING

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform based image processing since the late1950’s, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

Figure 1.13 RGB histogram image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called Multiresolutiontheory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

1.8.13 COMPRESSION

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

1.8.14 MORPHOLOGICAL PROCESSING

Morphological processingdeals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Figure 1.14 Blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

1.8.15 SEGMENTATION

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

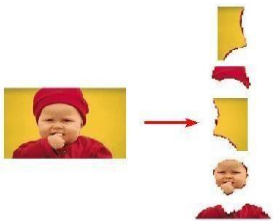


Figure 1.15 Image segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

1.8.16 REPRESENTATION AND DESCRIPTION

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

1.8.17 OBJECT RECOGNITION

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

1.8.18 KNOWLEDGEBASE

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an inter related to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

1.8.19 COMPONENTS OF AN IMAGE PROCESSING SYSTEM

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

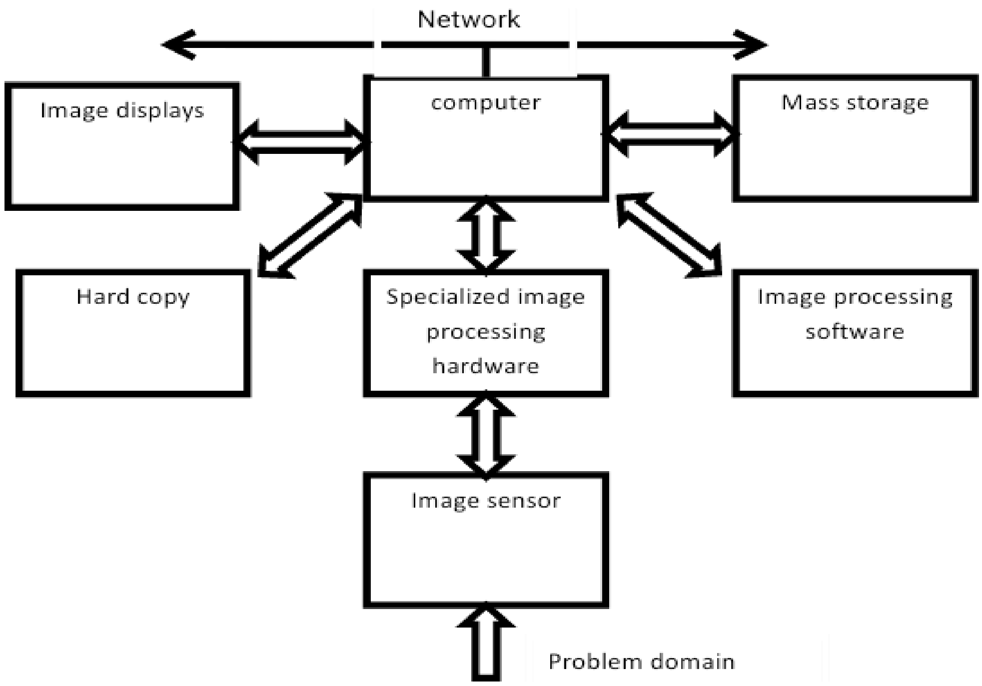


Figure 1.16 Components of image processing

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 1.24 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

- **Image sensors:**

With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

- **Specialized image processing hardware:**

Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

- **Computer:**

The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

- **Image processing software:**

Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

- **Mass storage:**

Mass storage capability is a must in image processing applications. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage forimage processing applications fall into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), Gbytes (meaning giga, or one billion, bytes), and Tbytes (meaning tera, or one trillion, bytes)

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig. 1.24. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in “jukeboxes” are the usual media for archival applications.

- **Image displays:**

Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are there requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

- **Hardcopy:**

Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

- **Network:**

Networking is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of optical fiber and other broadband technologies.

Color and texture are two low-level features widely used for image classification, indexing and retrieval. Color is usually represented as a histogram, which is a first order statistical measure that captures global distribution of color in an image One of the main drawbacks of the histogram- based approaches is that the spatial distribution and local variations in color are ignored. Local spatial variation of pixel intensity is commonly used to capture texture information in an image. Grayscale Co-occurrence Matrix (GCM) is a well-known method for texture extraction in the spatial domain. A GCM stores the number of pixel neighborhoods in an image that have a particular grayscale combination. Let I be an image and let p and Np respectively denote any arbitrary pixel and its neighbor in a given direction. If GL denotes the total number of quantized gray levels and gl denotes the individual gray levels, where, gl $\in$ {0, . . .,GL _ 1}, then each component of GCM can be written as follows:

$$\mathrm{gcm}(i,j) = \Pr((\mathrm{gl}_{\mathrm{p}}, \mathrm{gl}_{N_{\mathrm{p}}}) = (i,j))$$

Equation 1-1 Each component of GCM

gcm(i, j) is the number of times the gray level of a pixel p denoted by glp equals i, and the gray level of its neighbor Np denoted by glNp equals j, as a fraction of the total number of pixels in the image. Thus, it estimates the probability that the gray level of an arbitrary pixel in an image is i, and that of its neighbor is j. One GCM matrix is generated for each possible neighborhood direction, namely, 0, 45, 90 and 135.Average and range of 14 features like Angular Second Moment, Contrast, Correlation, etc., are generated by combining all the four matrices to get a total of 28 features. In the GCM approach for texture extraction, color information is completely lost since only pixel gray levels are considered.

To incorporate spatial information along with the color of image pixels, a feature called color correlogram has recently been proposed. It is a three dimensional matrix that represents the probability of finding pixels of any two given colors at a distance ‘d’ apart Auto correlogram is a variation of correlogram, which represents the probability of finding two pixels with the same color at a distance ‘d’ apart. This atproach can effectively represent color distribution in an image. However, correlogram features do not capture intensity variation Many image databases often contain both color as well as gray scale images. The color correlogram method does not constitute a good descriptor in such databases.

Another method called Color Co-occurrence Matrix (CCM) has been proposed to capture color variation in an image. CCM is represented as a three-dimensional matrix, where color pair of the pixels p and Np are captured in the first two dimensions of the matrix and the spatial distance ‘d’ between these two pixels is captured in the third dimension. This approach is a generalization of the color correlogram and reduces to the pure color correlogram for d = 1. CCM is generated using only the Hue plane of the HSV (Hue, Saturation and Intensity Value) color space. The Hue axis is quantized into HL number of levels. If individual hue values are denoted by hl, where

$$\mathrm{hl} \in \{0, \ldots, \mathrm{HL} - 1\}$$

Equation 1-2 Hue values set

then each component of CCM can be written as follows:

$$\mathrm{ccm}(i, j) = \Pr((\mathrm{hl_p}, \mathrm{hl}_{N_p}) = (i, j))$$

Equation 1-3 Each component of CCM

Four matrices representing neighbors at angles 0, 90, 180 and 270 are considered. This approach was further extended by separating the diagonal and the non-diagonal components of CCM to generate a Modified Color Co-occurrence Matrix (MCCM). MCCM, thus, may be written as follows: MCCM = (CCMD;CCMND)

Here, CCMD and CCMND correspond to the diagonal and off-diagonal components of CCM. The main drawback of this approach is that, like correlogram, it also captures only color information and intensity information is completely ignored.

An alternative approach is to capture intensity variation as a texture feature from an image and combine it with color features like histograms using suitable weights . One of the challenges of this approach is to determine suitable weights since these are highly application-dependent. In certain applications like Content-based Image Retrieval (CBIR), weights are often estimated from relevance feedback given by users.

While relevance feedback is sometimes effective, it makes the process of image retrieval user-dependent and iterative. There is also no guarantee on the convergence of the weight-learning algorithms. In order to overcome these problems, researchers have tried to combine color and texture features together during extraction proposed two approaches for capturing color and intensity variations from an image using the LUV color space. In the Single-channel Co-occurrence Matrix (SCM), variations for each color channel, namely, L, U and V are considered independently. In the Multi channel Co-occurrence Matrix (MCM), variations are captured taking two channels at a time – UV, LU and LV. Since the LUV color space separates out chrominance (L and U) from luminance (V), SCM in effect generates one GCM and two CCMs from each image independently. As a result, correlation between the color channels is lost

However, in MCM, the count of pairwise occurrences of the values of different channels of the color space is captured. Thus, each component of MCM can be written as follows:

mcmUV(i; j) = Pr((up; vNp) = (i; j))

mcmLU(i; j) = Pr((lp; uNp) = (i; j))

mcmLV(i; j) = Pr((lp; vNp) = (i; j))

Here, mcmUV(i, j) is the number of times the U chromaticity value of a pixel p denoted by up equals i, and the V chromaticity value of its neighbor Np denoted by vNp equals j, as a fraction of the total number of pixels in the image. Similarly, mcmLU(i, j) and mcmLV(i, j) are defined. One MCM matrix is generated for each of the four neighborhood directions, namely, 0, 45, 90 and 135.

Deng and Manjunath (2001)proposed a two-stage method called JSEG, which combines color and texture after image segmentation. In the first stage, colors are quantized to the required levels for differentiating between various regions of an image. Pixel values of the regions are then replaced by their quantized color levels to form a color map. Spatial variation of color levels between different regions in the map is viewed as a type of texture composition of the image.

Yu et al. (2002) suggested the use of color texture moments to represent both color and texture of an image. This approach is based on the calculation of

Local Fourier Transformation (LFT) coefficients. Eight templates equivalent to LFT are operated over an image to generate a characteristic map of the image. Each template is a 3 · 3 filter that considers eight neighbors of the current pixel for LFT calculation. First and second order moments of the characteristic map are then used to generate a set of features.

In this paper, we propose an integrated approach for capturing spatial variation of both color and intensity levels in the neighborhood of each pixel using the HSV color space. In contrast to the other methods, for each pixel and its neighbor, the amount of color and intensity variation between them is estimated using a weight function. Suitable constraints are satisfied while choosing the weight function for effectively relating visual perception of color and the HSV color space properties. The color and intensity variations are represented in a single composite feature known as Integrated Color and Intensity Co-occurrence Matrix (ICICM). While the existing schemes generally treat color and intensity separately, the proposed method provides a composite view to both color and intensity variations in the same feature. The main advantage of using ICICM is that it avoids the use of weights to combine individual color and texture features. We use ICICM feature in an image retrieval application from large image databases.

Early result on this work was reported in (Vadivel et al., 2004a). In the next section, we describe the proposed feature extraction technique after introducing some of the properties of the HSV color space. Choice of quantization levels for color and intensity axes, selection of parameter values and a brief overview of the image retrieval application

1.8.20 INTEGRATED COLOR AND INTENSITY CO – OCCURRENCE MATRIX

We propose to capture color and intensity variation around each pixel in a two-dimensional matrix called Integrated Color and Intensity Co-occurrence Matrix (ICICM). This is a generalization of the Grayscale Co-occurrence Matrix and the Color Co-occurrence Matrix techniques. For each pair of neighboring pixels, we consider their contribution to both color perception as well as gray level perception to the human eye. Some of the useful properties of the HSV color space and their relationship to human color perception are utilized for extracting this feature. In the next sub-section, we briefly explain relevant properties of the HSV color space. In the subsequent subsection, we describe how the properties can be effectively used for generating ICICM.

1.8.21 HSV COLOR SPACE

HSV Color space: Basically there are three properties or three dimensions of color that being hue, saturation and value HSV means Hue, Saturation and Value. It is important to look at because it describes the color based on three properties. It can create the full spectrum of colors by editing the HSV values. The first dimension is the Hue. Hue is the other name for the color or the complicated variation in the color. The quality of color as determined by its dominant wavelength. This Hue is broadly classified into three categories. They are primary Hue, Secondary Hue and Teritiary hue. The first and the foremost is the primary Hue it consists of three colors they are red, yellow and blue. The secondary Hue is formed by the combination of the equal amount of colors of the primary Hue and the colors of the secondary Hue which was formed by the primary Hue are Orange, Green and violet. The remaining one is the teritiary Hue is formed by the combination of the primary Hue and the secondary Hue. The limitless number of colors are produced by mixing the colors of the primary Hue in different amounts. Saturation is the degree or the purity of color. Then the second dimension is the saturation. Saturation just gives the intensity to the colors. The saturation and intensity drops just by mixing the colors or by adding black to the color. By adding the white to the color in spite of more intense the color becomes lighter. Then finally the third dimension is the Value. The value is the brightness of the color. When the value is zero the color space is totally black with the increase in the color there is also increase in the brightness and shows the various colors. The value describes the contrast of the color. That means it describes the lightness and darkness of the color. As similar to the saturation this value consists of the tints and shades. Tints are the colors with the added white and shades are the colors with the added black.

1.8.22 PROPERTIES OF HSV COLOR SPACE

Sensing of light from an image in the layers of human retina is a complex process with rod cells contributing to scotopic or dim-light vision and cone cells to photopic or bright-light vision (Gonzalez and Woods, 2002). At low levels of illumination, only the rod cells are excited so that only gray shades are perceived. As the illumination level increases, more and more cone cells are excited, resulting in increased color perception. Various color spaces have been introduced to represent and specify colors in a way suitable for storage, processing or transmission of color information in images. Out of these, HSV is one of the models that separate out the luminance component (Intensity) of a pixel color from its chrominance components (Hue and Saturation). Hue represents pure color, which is perceived when incident light is of sufficient illumination and contains a single wavelength. Saturation gives a measure of the degree by which a pure color is diluted by white light. For light with low illumination, corresponding intensity value in the HSV color space is also low.

The HSV color space can be represented as a Hexa cone, with the central vertical axis denoting the luminance component, I (often denoted by V for Intensity Value). Hue, is a chrominance component defined as an angle in the range [0,2p] relative to the red axis with red at angle 0, green at 2p/3, blue at 4p/3 and red again at 2p. Saturation, S, is the other chrominance component, measured as a radial distance from the central axis of the hexacone with value between 0 at the center to 1 at the outer surface. For zero saturation, as the intensity is increased, we move from black to white through various shades of gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the color represented by its hue. When saturation is near 0, all the pixels in an image look alike even though their hue values are different.

As we increase saturation towards 1, the colors get separated out and are visually perceived as the true colors represented by their hues. Low saturation implies presence of a large number of spectral components in the incident light, causing loss of color information even though the illumination level is sufficiently high. Thus, for low values of saturation or intensity, we can approximate a pixel color by a gray level while for higher saturation and intensity, the pixel color can be approximated by its hue. For low intensities, even for a high saturation, a pixel color is close to its gray value. Similarly, for low saturation even for a high value of intensity, a pixel is perceived as gray. We use these properties to estimate the degree by which a pixel contributes to color perception and gray level perception.

One possible way of capturing color perception of a pixel is to choose suitable thresholds on the intensity and saturation. If the saturation and the intensity are above their respective thresholds, we may consider the pixel to have color dominance; else, it has gray level dominance. However, such a hard thresholding does not properly capture color perception near the threshold values. This is due to the fact that there is no fixed level of illumination above which the cone cells get excited. Instead, there is a gradual transition from scotopic to photopic vision. Similarly, there is no fixed threshold for the saturation of cone cells that leads to loss of chromatic information at higher levels of illumination caused by color dilution. We, therefore, use suitable weights that vary smoothly with saturation and intensity to represent both color and gray scale perception for each pixel.

1.8.23 NON INTERVAL QUANTIZATION

Due to the large range for each component by directly calculating the characteristics for the retrieval then the computation will be very difficult to ensure rapid retrieval. It is essential to quantify HSV space component to reduce computation and improve efficiency. At the same time, because the human eye to distinguish colors is limited, do not need to calculate all segments. Unequal interval quantization according the human color perception has been applied on H , S ,V components.

Based on the color model of substantial analysis, we divide color into eight parts. Saturation and intensity is divided into three parts separately in accordance with the human eyes to distinguish. In accordance with the different colors and subjective color perception quantification, quantified hue(H), saturation(S) and value(V)

In accordance with the quantization level above, the H, S, V three-dimensional feature vector for different values of with different weights to form one dimensional feature vector and is given by the following equation:

$$G = Qs*Qv*H+Qv*s+V$$

Where Qs is the quantized series of S and Qv is the quantized series of V. And now by setting Qs = Qv = 3, Then G = 9H+3S+V

In this way three component vector of the HSV from one dimensional vector, which quantize the whole color space for the 72 kinds of the main colors. So we can handle 72 bins of one dimensional histogram. This qualification is effective in reducing the images by the effect of the light intensity, but also reducing the computational time and complexity.

1.8.24 IMAGE RETRIEVAL

Image retrieval is nothing but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords or the descriptions to the images so that the retrieval can be performed. Manual image annotation is time consuming, expensive and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc..,

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others. one can today generate volumes of images with content as diverse as family get-togethers and national park visits. Low-cost storage and easy Web hosting has fueled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchable image data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real-world image search system designers.

As far as technological advances are concerned, growth in Content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real world implications, applications, and constraints of the technology. Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future.

An image retrieval system designed to serve a personal collection should focus on features such as personalization, flexibility of browsing, and display methodology. For example, Google's Picasa system [Picasa 2004] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods. The notable few applications include an FPGA implementation of a color-histogram-based image retrieval system [Kotoulas and Andreadis 2003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 2003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 2002]

Discussion. Regardless of the nature of the collection, as the expected user-base grows, factors such as concurrent query support, efficient caching, and parallel and distributed processing of requests become critical. For future real-world image retrieval systems, both software and hardware approaches to address these issues are essential. More realistically, dedicated specialized servers, optimized memory and storage support, and highly parallelizable image search algorithms to exploit cluster computing powers are where the future of large-scale image search hardware support lies.

1.8.25 OVERVIEW OF TEXTURE

We all know about the term Texture but for defining it is a hard time. One can differentiate the two different Textures by recognizing the similarities and differences. Commonly there are three ways for the usage of the Textures:

Based on the Textures the images can be segmented To differentiate between already segmented regions or to classify them. We can reproduce Textures by producing the descriptions.

1.9 ORGANIZATION OF CHAPTERS

1.9.1 INTRODUCTION

This section provides information about the project such as overview, objective and scope of the project. Along with that, this chapter also includes information about the techniques that have been used or considered while working on the project.

1.9.2 LITERATURE SURVEY

This section provides the information about the existing methods on the detection of forgery with the help of which this project was modified.

1.9.3 PROJECT DESIGN

This section provides the information about the methodology that is applied in this project i.e., the design is discussed and also the software used to implement it.

1.9.4 PROJECT IMPLEMENTATION

This section provides the information about the algorithm used and discusses each and every step in the algorithm. It also contains the results of the project.

1.9.5 CONCLUSION AND FUTURE ENHANCEMENT

This section concludes the project along with the applications of it. This project can be enhanced further for better results and wide utility and the future enhancement chances are mentioned here.

CHAPTER 2 : LITERATURE REVIEW

As discussed in the introduction block-based methods divide images into overlapping regularly shaped blocks. In consecutive blocks a major portion of a block will get duplicated in the next block. Once the blocks are divided, each block is compared with every other block using block level features. Block based algorithms follow the same framework, with the difference in what feature and metric it uses for block matching. Fridrich et al. used quantized discrete cosine transform as a block feature. Popescu and Farid use a reduced dimension representation for image blocks using principal component analysis(PCA). Luo et al. use RGB color components along with direction information. Li et al. use SV vector as block feature by applying both singular value decomposition (SVD) and discrete wavelet transform(DWT). Mahdian and Saic used 24 blur invariant moments. Kang and Wei use singular values of a reduced rank approximation. Bayram et al. calculate the FourierMellin transformation of the image blocks. Wang et al. with different radii create a circle with the block center as the center and find the average intensity. Lin et al. calculate each block and sub-blocks the average gray result. Ryu et al. calculate Zernike moments of each block. BravoSolario and Nandi [calculate the information entropy of each block. The problem with this type of algorithm is that the complexity is high, that it won't be able to detect forgery, if the forged region is applied with some image transformation, and also because of the regular shape of the blocks this type of algorithm has less recall rate.

Keypoint based methods find the key points of the images. For each keypoint find the similarity with all other key points in the same image. If there is a similarity beyond the predefined similarity threshold, that particular keypoint region is suspected as forged. The two important methods to find the key points of the images are Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features(SURF) . SIFT is used to find the key points and find the similarity between the SIFT-vector. In and use SURF to find the key points. The matched keypoint region may not be the exact forgery region. There can be situations like more than one forgery happens on a single image and in such cases if there are no matched keypoints in some region then forgery can't be detected for that region. For such an algorithm the detection accuracy will be less compared to the block-based methods. Clearly, the complexity will be less as it avoids exhaustive searching and the recall rate of images with transformed copy move image patches are high.

Combined block-based and keypoint based methods are also used to detect the image forgery. These algorithms judiciously take the advantages of both block-based and feature keypoints based methods. In such an algorithm, the image is segmented into non-overlapping blocks. For each block, compute the key points. Then compute the similarity between the block by matching the key points. Based on the predefined similarity threshold the forged region is identified. The SIFT algorithm is used to find the feature keypoints and adaptive over-segmentation is used to divide the image into blocks.

2.1 CONCLUSION ON REVIEWS

The detection accuracy will get reduced when the copy-move happens in the same image block and also if the number of image keypoints is less. We propose an improved version of existing combined block-based and feature keypoint based methods for better accuracy without much increase in the complexity. The adaptive over-segmentation algorithm proposed by Pun et al. is used to divide the images into non-overlapping irregular blocks. For each block find the SIFT keypoints and calculate the number of matched keypoints. If the number of keypoint matches for two blocks are found to be greater than a precomputed threshold, a copy move is detected and the image is found to be forged. If there are no keypoint matches, the image can be treated as unaltered else we need to decide whether it's actually forged or not by considering additional measures. The strategy that we have adopted for this is to apply block-based methods only to a regular region around the matched keypoints instead of applying to the entire image to reduce the complexity.

CHAPTER 3 : PROJECT DESIGN

3.1 OVERVIEW OF THE DESIGN

In CMFD, the typical workflow consists of five main stages; image acquisition, image pre-processing, feature extraction, feature matching and visualization.

3.1.1 IMAGE ACQUISITION

In our proposed work, images are taken from the CMFD datasets or own images to evaluate the proposed CMFD method. Both datasets contain tampered images which have suffered different geometrical attacks, namely translation, different degrees of rotation and symmetric or asymmetric scaling or a combination of them. Table I illustrates the various geometrical attacks available in the MICC-F2000 database. The different attacks are represented by θ, SX and SY, where θ represents the degree of rotation, SX and SY represents the scaling factor in the x and y axis respectively.

3.1.2 IMAGE PRE-PROCESSING

Image pre-processing is generally performed to reduce the amount of redundant information in an image and to improve the computational efficiency in the following CMFD stages. In our work, the pre-processing operations consist of image RGB to grayscale conversion, image resizing and tampered region identification.

3.1.3 CMFD USING Oriented FAST AND Rotated BRIEF (ORB) :

Oriented FAST and Rotated BRIEF (ORB) [11] is a binary descriptor that uses oFAST keypoints and rBRIEF features for feature extraction. A typical binary descriptor consists of three main components, namely the sampling pattern, orientation compensation and sampling pairs. The ORB approach does not have a specific sampling pattern. However, it uses oFAST moments and rBRIEF learned pairs for its orientation compensation and sampling pairs respectively. With ORB, Rublee et al. [11] introduced the concept of intensity centroid to obtain orientation information of an interest point. As a descriptor, ORB is capable of producing uncorrelated feature vectors, which implies that each element of the feature vector contains unique information. Furthermore, ORB produces descriptors with high variance; this implies that the features respond differently to different inputs, making it more discriminative. 1) Oriented FAST keypoints: The ORB algorithm uses FAST detector to identify interest points within an image. FAST is a feature from accelerated segment test and it is the method of choice for locating interest points in real time systems. In order to obtain FAST keypoints, the intensity of the central pixel is compared with its neighboring pixels located on a circle of fixed radius.

After identifying a set of interest points, the interest points are sorted using Harris corner measure to select the top N points. Even though FAST has low computation cost, it is not scale or rotation invariant. Hence, in order to achieve scale invariance, a scale pyramid of the image is employed to generate FAST features at each level.

The proposed method converts the input image into grayscale and is then segmented in too many super pixels using SLIC. SLIC will divide the images into meaningful super pixels that will improve the recall rate. For each block SIFT key points are identified and will be used as the primary check for forgery detection. The keypoints are matched using simple Euclidean distance measure. The number of matched keypoints between the blocks is recorded and will be used to find a dynamic threshold for identifying the forged regions. Based on the matched keypoints and threshold the image is divided into 3 categories: Forged Not-Forged, and Forgery Suspected. If the number of matching is greater than zero and less than the threshold then it belongs to the Forgery Suspected category. For this type of image, apply block based method to a particular region around the matched keypoints instead of applying to the whole image to reduce the computational complexity. Finally, apply some morphological operations mentioned by Pun et al. to get the approximate forged region. Each of these step is explained in detail in following subsections. A. Preprocessing In the preprocessing step the image is segmented into meaningful image blocks. Traditionally in block based forgery detection methods images are divided into overlapping blocks of the same size and uniform structure. This approach will get better accuracy if the block size is small, but that will increase the complexity of that algorithm. To reduce the complexity without much drop in accuracy an adaptive over segmentation method has been proposed. For segmenting the images into blocks, simple linear iterative clustering (SLIC) algorithm is used. This algorithm will use the K-means clustering approach to segment the image into meaningful clusters. SLIC will generate k clusters for an image. Then combine the similar superpixel and segment the images into a meaningful region. This will give a more accurate forgery region and helps to reduce the complexity. The texture of the image can be used to decide cluster size. For a smooth textured image the segment size can be large and for an image having detailed texture the segment size should be small. Thus the initial size of a cluster is calculated based on the texture of that image. The initial size for SLIC is calculated using the energy distribution of the image proposed by Pun et al. The percentage of the low-frequency energy distribution of an image is calculated and if that is greater than 50% of the frequency energy then Eq.1 is used otherwise Eq.2 is used, Where H x W represents the height and width of the image. Note that as per the equations Eq.1 and 2 a smooth image gets a bigger block size and a detailed image gets a smaller block size. Sizeinitial = $\sqrt{H * W * 0.02}$ (1) Sizeinitial = $\sqrt{H * W * 0.01}$ (2) B. Keypoint Detection Segmentation will divide the images into meaningful superpixels. Each super pixel represents a block and for each block the keypoints are identified. Christlein et al. perform a comparative study between popular keypoint detection algorithms and proved that SIFT is a very good feature detector which are invariant to geometric transformations like scale, translation and rotation and are invariant to brightness and illumination. So in the proposed method we are using the SIFT algorithm for identifying the key points. Instead of finding key points of the whole image, we will find key points for each block and that will represent the block feature. The normalized 128-dimensional SIFT descriptor describes keypoints in terms of location, scale, and orientation.

3.2 DEFINE THE MODULES

3.2.1 INTRODUCTION TO MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modelling, simulation, and prototyping

- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In the industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available to include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

3.2.2 THE MATLAB SYSTEM

The MATLAB system consists of five main parts

- **Development Environment:**

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library:**

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language:**

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both ―programming in the small‖ to rapidly create quick and dirty throw-away programs, and ―programming in the large‖ to create large and complex application programs.

- **Graphics:**

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using the IMAGE PROCESSING toolbox.

3.2.3 GRAPHICAL USER INTERFACE (GUI)

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG file contains binary data that does not need to be parsed when the associated GUI-based M function is executed.

- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited and callback functions that are executed when a user interacts with GUI objects for example when a button is pushed.

To launch GUIDE from the MATLAB command window, type :

guide filename

Where filename is the name of an existing FIG-file on the current path. If the filename is omitted,

GUIDE opens a new (i.e., blank) window.

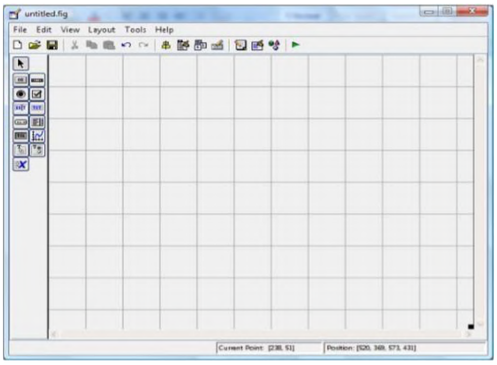


Figure 3.1 MATLAB procedure

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, pushbuttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

3.3 GETTING STARTED TO MATLAB

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativitycomputing power that will change the way you work.

3.3.1 INTRODUCTION.

- describes the components of the MATLAB system.

3.3.2 DEVELOPMENT ENVIRONMENT

- introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

3.3.3 MANIPULATING MATRICES

- introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

3.3.4 GRAPHICS

- introduces MATLAB graphic capabilities, including information about plotting data,annotating graphs, and working with images. 2.4.5 Programming with MATLAB
- describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

3.4 DEVELOPMENT ENVIRONMENT

3.4.1 INTRODUCTION

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

3.4.2 STARTING MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop. On a UNIX platform, to start MATLAB, type Matlab.

At the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup and reduce startup time in some situations.

3.4.3 QUITTING MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu on the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

3.4.4 MATLAB DESKTOP

When you start MATLAB, the MATLAB desktop appears, containing tools(graphical user interfaces) for managing files, variables, and applications associated with MATLAB. The first time MATLAB starts, the desktop appears as shown in the following illustration, although your LaunchPad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

3.4.5 DESKTOP TOOLS

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- LaunchPad
- Help Browser Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacsmagik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

LaunchPad

MATLAB's LaunchPad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the Help button in the toolbar, or type help browser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favourites tab - View a list of documents you previously designated as favourites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term on the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find Mfiles and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT file, which has a .mat extension. There are options for saving to different formats. To read in a MAT file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to run MATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop.

3.5 DIGITAL IMAGE PROCESSING

Background:

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An important characteristic underlying the design of image processing systems is the significant level of testing & experimentation that normally is required before arriving at an acceptable solution. This characteristic implies that the ability to formulate approaches &quickly prototype candidate solutions generally plays a major role in reducing the cost & time required to arrive at a viable system implementation.

3.5.1 DIGITAL IMAGE PROCESSING DEFINITION

An image may be defined as a two-dimensional function f(x, y), where x & y are spatial coordinates, & the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital images by means of a digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensors, so it is not surprising that images play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis& computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting& somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low-level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low- level process is characterized by the fact that both its inputs & outputs are images.

Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images. Finally higher- level processing involves "Making sense" of an ensemble of recognized objects, as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

3.5.2 ABOUT IMAGE

An image is represented as a two dimensional function f(x, y) where x and y are spatial co-ordinates and the amplitude of 'f' at any pair of coordinates (x, y) is called the intensity of the image at that point.

3.5.3 GRAY SCALE IMAGE

A grayscale image is a function I(xylem) of the two spatial coordinates of the image plane.

I(x, y) is the intensity of the image at the point (x, y) on the image plane.

I (xylem)takes non-negative values assume the image is bounded by arectangle[0, a] ·[0, b]I: [0, a] · [0, b] → [0, info)

3.5.4 COLOR IMAGE

It can be represented by three functions, R (xylem)for red,G (xylem)for green *and* B (xylem)for blue.

An image may be continuous with respect to the x and y coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate's values is called sampling. Digitizing the amplitude values is called quantization.

3.5.5 COORDINATE CONVENTION

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image f(x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size M X N. The values of the coordinates (xylem) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates.

In many image processing books, the image origin is defined to be at (xylem)=(0,0).The next coordinate values along the first row of the image are (xylem)=(0,1).It is important to keep in mind that the notation (0,1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Following figure shows the coordinate convention. Note that x ranges from 0 to M-1 and y from 0 to N-1 in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (xylem) the toolbox uses the notation (race) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate topples, (alb), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentation refers to the coordinates. Less frequently the toolbox also employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refers to rows. This is the opposite of our use of variables x and y.

3.6 IMAGE AS MATRICES

The preceding discussion leads to the following representation for a digitized image function:

$$
f(xylem) = \begin{matrix} f(0,0) & f(0,1) & \ldots\ldots\ldots.. & f(0,N-1) \\ f(1,0) & f(1,1) & \ldots\ldots\ldots\ldots & f(1,N-1) \\ . & . & & . \\ . & . & & . \\ f(M-1,0) & f(M-1,1) & \ldots\ldots\ldots\ldots\ldots\ldots & f(M-1,N-1) \end{matrix}
$$

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements. A digital image can be represented naturally as a MATLAB matrix:

$$
f = \begin{matrix} f(1,1) & f(1,2) & \ldots\ldots\ldots\ldots & f(1,N) \\ f(2,1) & f(2,2) & \ldots\ldots\ldots\ldots & f(2,N) \\ . & . & & . \\ . & . & & . \\ f(M,1) & f(M,2) & \ldots\ldots\ldots. & f(M,N) \end{matrix}
$$

Where f (1,1) = f(0,0) (note the use of a monoscope font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation f(p ,q) denotes the element located in row p and the column q. For example f(6,2) is the element in the sixth row and second column of the matrix f. Typically we use the letters M and N respectively to denote the number of rows and columns in a matrix. A 1xN matrix is called a row vector whereas an Mx1 matrix is called a column vector. A 1x1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using mono-scope characters. We use conventional Roman, italic notation such as f(x ,y), for mathematical expressions

3.7 READING IMAGES

Images are read into the MATLAB environment using function imread whose syntax is “ Imread (‘filename’) “

FORMAT NAME	DESCRIPTION	RECOGNIZED EXTENSION
TIFF	Tagged Image File Format	.tif, .tiff
JPEG	Joint Photographic Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	.xwd

Table 3-1 Image formats

Here filename is a spring containing the complete of the image file(including any applicable extension).For example the command line

>> f = imread (‘8. jpg’);

Reads the JPEG (above table) image chest x ray into image array f. Note the use of single quotes (‘) to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output If a semicolon is not included. MATLAB displays the results of the operation(s) specified in that line. Theprompt symbol (>>) designates the beginning of a command line, as it appears in the MATLAB command window.

3.8 DATA CLASSES

Although we work with integers coordinates the values of pixels themselves are not restricted to be integers in MATLAB. Table above list various data classes supported by MATLAB and IPT are representing pixels values. The first eight entries in the table are refers to as numeric data classes. The ninth entry is the char class and, as shown, the last entry is referred to as logical data class.

NAME	DESCRIPTION
Double	Double _ precision, floating_ point numbers the approximate.
uint8	unsigned 8_bit integers in the range [0,255] (1byte per element).
unit16	unsigned 16_bit integers in the range [0, 65535] (2byte per element).
unit32	unsigned 32_bit integers in the range [0, 4294967295](4 bytes per element).
int8	signed 8_bit integers in the range [-128,127] 1 byte per element).
int16	signed 16_byte integers in the range [32768, 32767] (2 bytes per element).
int32	Signed 32_byte integers in the range [-2147483648, 21474833647] (4 byte per element).
single	single _precision floating _point numbers with values in the approximate range (4 bytes per element).
char	characters (2 bytes per element).
logical	values are 0 to 1 (1byte per element).

Table 3-2Dataclasses

All numeric computations in MATLAB are done in double quantities, so this is also a frequent data class encounter in image processing applications. Class unit 8 also is encountered frequently, especially when reading data from storages devices, as 8 bit images are most common representations found in practice. These two data classes, classes logical, and, to a lesser degree, class unit 16 constitute the primary data classes on which we focus. Many ipt functions however support all the data classes listed in table. Data class double requires 8 bytes to represent a number uint8 and int 8 require one byte each, uint16 and int16 requires 2bytes and unit 32.

Int 32 and single required 4 bytes each. The char data class holds characters in Unicode representation. A character string is merely a 1*n array of characters logical array contains only the values 0 to 1,with each element being stored in memory using function logic or by using relational operators.

3.9 IMAGE TYPES

The toolbox supports four types of images:

1 .Intensity images;

2. Binary images;

3. Indexed images;

4. R G B images.

Most monochrome image processing operations are carried out using binary or intensity images, so our initial focus is on these two image types. Indexed and RGB colour images.

3.9.1 INTENSITY IMAGES

An intensity image is a data matrix whose values have been scaled to represent intentions. When the elements of an intensity image are of class unit8, or class unit 16, they have integer values in the range [0,255] and [0, 65535], respectively. If the image is of class double, the values are floating point numbers. Values of scaled, double intensity images are in the range [0, 1] by convention.

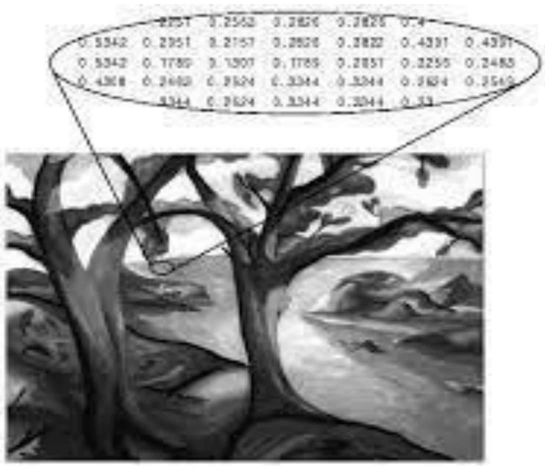


Figure 3.2 Intensity image example

3.9.2 BINARY IMAGES

Binary images have a very specific meaning in MATLAB.A binary image is a logical array 0s and1s.Thus, an array of 0s and 1s whose values are of data class, say unit8, is not considered as a binary image in MATLAB .A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, we create an array B using the statement.

B=logical (A)

If A contains elements other than 0s and 1s.Use of the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.

Using relational and logical operators also creates logical arrays.

To test if an array is logical we use the I logical function: islogical(c).

If c is a logical array, this function returns a 1.Otherwise returns a 0. Logical array can be converted to numeric arrays using the data class conversion functions.

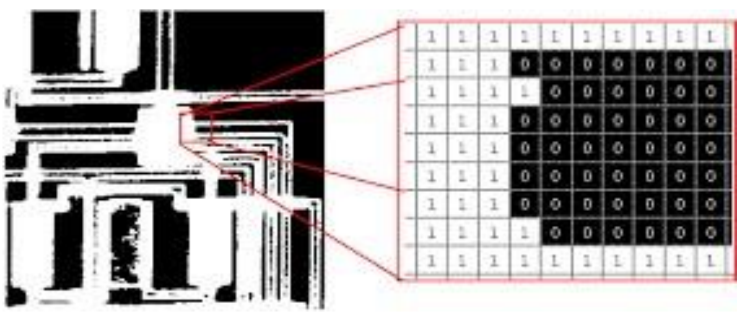


Figure 3.3 Binary image example

3.9.3 INDEXED IMAGES

An indexed image has two components: 1. A data matrix integer, x 2. A color map matrix, map

Matrix map is an m*3 arrays of class double containing floating point values in the range [0, 1].The length m of the map are equal to the number of colors it defines. Each row of map specifies the red, green and blue components of a single color. An indexed images uses "direct mapping" of pixel intensity values color map values. The color of each pixel is determined by using the corresponding value the integer matrix x as a pointer in to map. If x is of class double ,then all of its components with values less than or equal to 1 point to the first row in map, all components with value 2 point to the second row and so on. If x is of class units or unit 16, then all components value 0 point to the first row in map, all components with value 1 point to the second and so on.

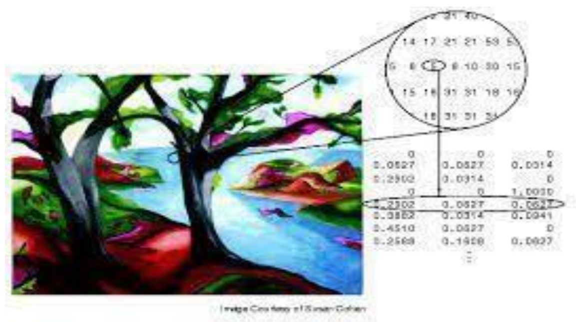


Figure 3.4 Indexed image example

3.9.4 RGB IMAGE

An RGB color image is an M*N*3 array of color pixels where each color pixel is triplet corresponding to the red, green and blue components of an RGB image, at a specific spatial location. An RGB image may be viewed as "stack" of three gray scale images that when fed in to the red, green and blue inputs of a color monitor

Produce a color image on the screen. Convention the three images forming an RGB color image are referred to as the red, green and blue components images. The data class of the components images determines their range of values. If an RGB image is of class double the range of values is [0, 1].

Similarly the range of values is [0,255] or [0, 65535].For RGB images of class units or unit 16 respectively. The number of bits use to represents the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8bit image, the corresponding RGB image is said to be 24 bits deep.

Generally, the number of bits in all component images is the same. In this case the number of possible color in an RGB image is (2^b) ^3, where b is a number of bits in each component image. For the 8bit case the number is 16,777,216 colors.

Figure 3.5 RGB image example

3.10 IMAGE FORGERY

3.10.1 DEFINITION OF IMAGE FORGERY

Image forgery means manipulation of the digital image to conceal some meaningful or useful information of the image. There are cases when it is difficult to identify the edited region from the original image. The detection of a forged image is driven by the need of authenticity and to maintain integrity of the image.

3.10.2 CLASSIFICATION OF IMAGE FORGERY TECHNIQUES

There are two kinds of techniques for image forensics: one is active protection, and the other is passive detection. Which again consist of many different methods, as shown in below figure:

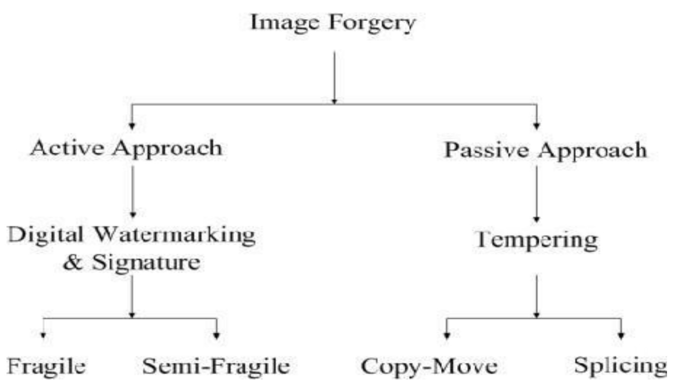


Figure 3.6 Classification of image forgery techniques

A. Active Approach

In this active approach, the digital image requires some kind of pre-processing such as watermark embedded or signatures are generated at the time of creating the image. However, in practice this would limit their application. Digital watermarking and signature are two main active protection techniques, as something is embedded into images when they are obtained. We can detect if the Image is tampered, if special information cannot be extracted from that obtained image. Watermarking is such a method of active tampering detection, as a security structure is embedded into the image, but most present imaging devices do not contain any watermarking or signature module and that are similar to the application of active protection. This structure is used for integrity evaluation in the sense that if any discrepancy is found with the structure then the image is tampered and an inverse analysis over the structure is done to locate tampered Regions of the image. In recent times, various schemes are proposed for providing security to the image, which is analogous to the concept of watermarking like, message authentication code, image hash, image checksum and image shielding as a counterpart to it.

B. Passive Approach

Passive image forensics is usually a great challenge in image processing techniques. There is not a particular method that can treat all these cases, but many methods each can detect a special forgery in its own way. The stream of passive tampering detection deals with analyzing the raw image based on various statistics and semantics of image content to localize tampering of image. Neither construct is embedded in the image and nor associated with it for security, as like active approaches and hence this method is also known as raw image analysis. The localization of tampering is solely based on image feature statistics. Hence, algorithms and methods of detection and localization of image based on passive tampering vary depending upon the type of security construct used. Nevertheless, passive tampering detection typically aims for localization of tampering on raw images.

3.10.3 TYPES OF IMAGE FORGERY

A. Image Retouching:

Image Retouching is considered as less harmful kind of digital image forgery than other types present. In case of image retouching original image does not significantly changes, but there is enhancement or reduces certain feature of original image. This technique is popular among magazine photo editors. This type of Image forgery is present in almost all magazine covers that would employ this technique to enhance certain features of an image so that it is more attractive. Actually, the fact is that such enhancement is ethically wrong.

Figure 3.7 Image forgery example

B. Image splicing or photomontage:

This technique for making forgery images is more aggressive than image retouching. Image splicing is fundamentally simple process and can be done as crops and pastes regions from the same or separate sources. This method refers to a paste-up produced by sticking together images using digital tools available such as Photoshop. In Image Splicing technique there is composition of two or more images, which are combined to create a fake image. Examples include several infamous news reporting cases involving the use of faked images. Fig below shows how to create forge Image; by copying a spliced portion from the source image into a target image, it is a composite picture of scenery which is forge image.

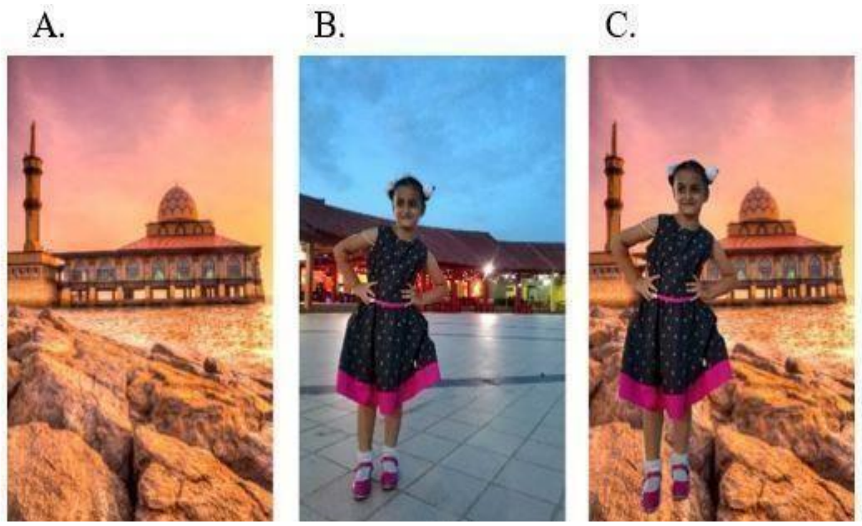


Figure 3.8 Image splicing example

C. Copy-Move Attack:

The copy move forgery is popular as one of the most commonly used kind of image tampering techniques. In this technique, one needs to cover a part of the image in order to add or remove information. In the Copy-Move image manipulation technique a part of the same image is copied and pasted into another part of that image itself. In a copy-move attack, the intention is to hide something in the original image with some other part of the same image . The example of the Copy-Move type is as shown in figure 3 below. The original image contains only three missiles and its Copy-Moved version on the right has four missiles.

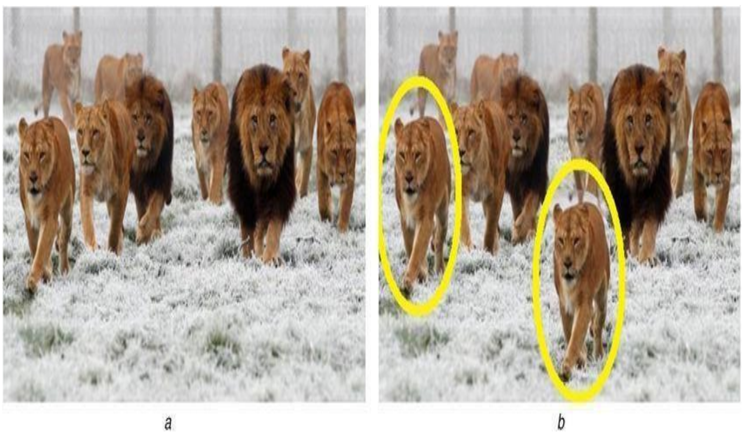


Figure 3.9 Copy move image example

CHAPTER 4 : PROJECT IMPLEMENTATION

4.1 IMPLEMENTATION STAGES

4.1.1 BLOCK DIAGRAM:

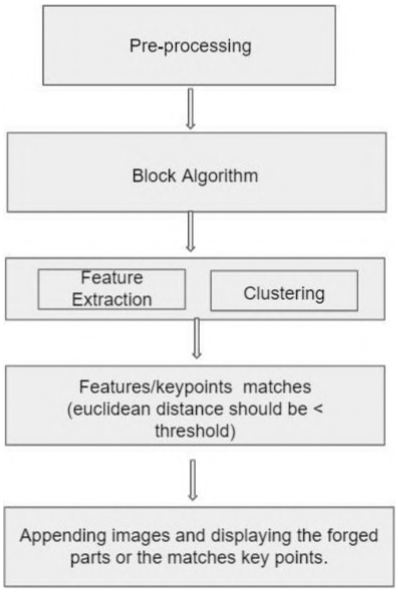


Figure 4.1 Block diagram of project

The above figure represents the block diagram of the project. Each and every block performs specific tasks or functions in order to obtain the required output.

The functions of each and every block is as discussed below:

PRE-PROCESSING-

This block is responsible for the pre-processes i.e., converting an RGB image into a Grayscale image for easier and effective results and also to make sure that all the images are of the same dimensions.

BLOCK ALGORITHM-

Application of block algorithm is nothing but dividing the image into required number of blocks (or frames) for effective search.

FEATURE EXTRACTION-

Features are extracted from the images using keypoint technique i.e., key points (or patches) are extracted in every block that was formed due to block based algorithm.

CLUSTERING-

After all the features/key points are extracted they are clustered into an array for better comparison process between the key points.

KEY POINT MATCHING-

Here all the key points are compared and depending upon the euclidean distances between the key points, matched key points are found and marked.

APPENDING IMAGES AND DISPLAY-

The input images are appended and the matched key points are connected for better understanding of the forged area.

4.1.2 PROGRAM ALGORITHM:

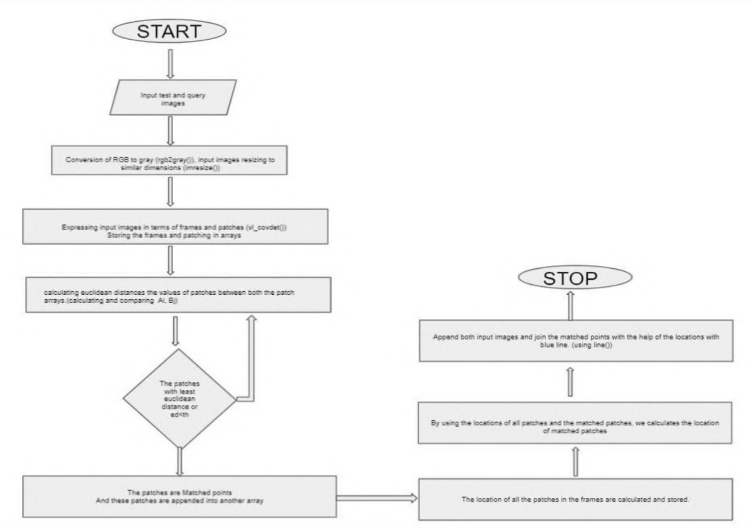


Figure 4.2 Program algorithm

The above figure is the algorithm of the project. The steps include:

STEP 1:

- Take two images as input, one is a query image and the other is a test image (where the test image is an image on which the query image has been copy pasted i.e., spliced image).

STEP 2:

- The default type of images that is available are RGB, so the images have to be converted from RGB to Grayscale for better processing using the rgb2gray() function.

- The images may be of different dimensions when initially considered so it is important to resize the images. It is done by using the imresize() function.

STEP 3:

- The images are now divided into frames and patches where frames are nothing but blocks that are considered in the images and patches are the key points which contain crucial information about the image.

- These frames and patches are then stored in the form of arrays for easier access.

- This entire process is carried out by the vl_covdet() function which is usually used to extract features in an image.

STEP 4:

- After extracting the features/key points, the matched key points have to be accessed or extracted.

- For identifying the matched points it's important to find the euclidean distances between the patches of both the images.

- The patches or key points which have the least euclidean distance between them or euclidean distance less than the given threshold between them will be considered as the matched key points.

- The distances will be calculated between each pair of patches repeatedly until all the patches or key points in the array are considered.

- All the matched points are then stored separately in an array.

STEP 5:

- The locations of all the points in the images are calculated and stored in variables again in the form of an array.

- The matched points locations are found by using that array where location of each pixel or feature is stored.

STEP 6:

- The pre-processed query image and test image are appended for better understanding of the result.

- The matched points are marked on both images as the locations of the points have been calculated in the previous step.

- The matched points are now connected using a line function with a blue straight line displaying the forged area in the image.

4.2 RESULTS

Figure 4.3 Query image

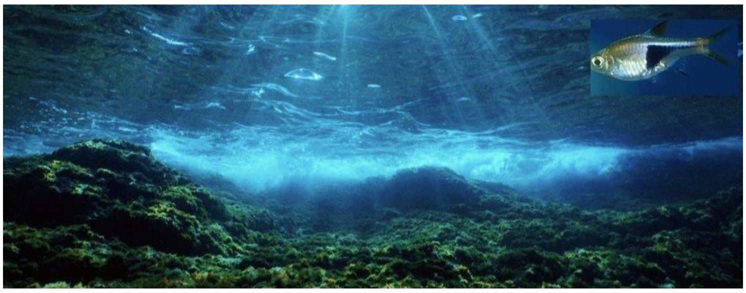


Figure 4.4 Test image or copy pasted image

In this project we considered two images: Fig 7.1 - Query image, Fig 7.2 - Test image.

The query image has been copy pasted into the test image which is considered as the forged image. Now we have to show the region of forgery.

Fig 7.3 shows the output of the project:

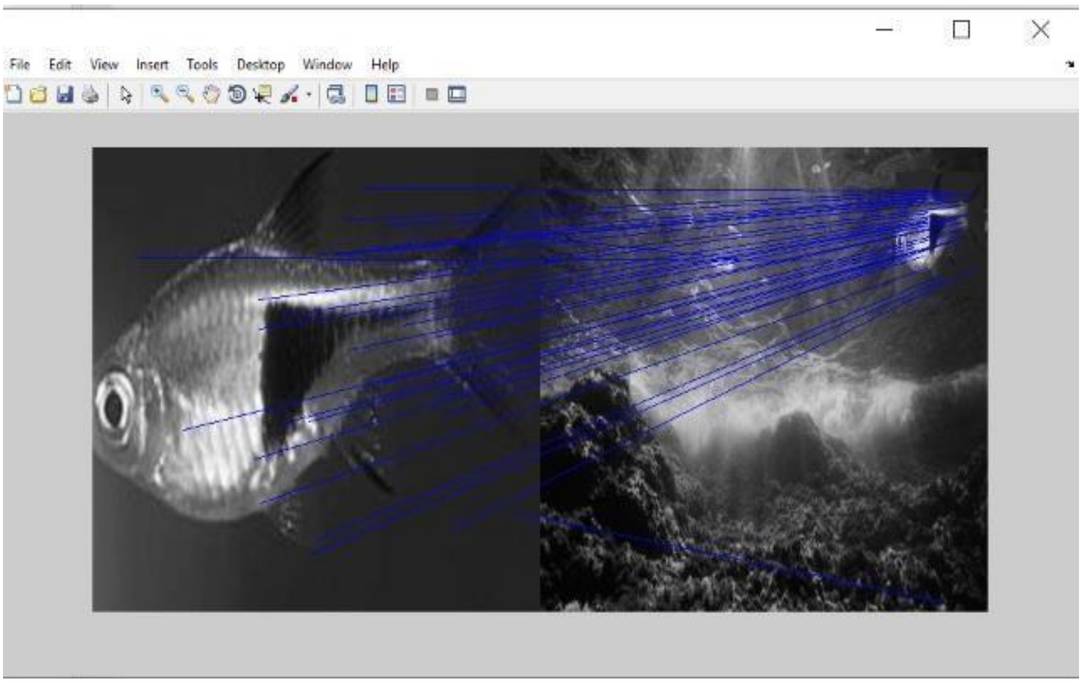


Figure 4.5 Project output

The figure shows blue lines connecting points in one image to another. These points are the matched key points where the Euclidean distance is least or less than the threshold. The output displays the forged area thus indicating that the query image has been copy pasted into the test image which shows a forged image result.

CHAPTER 5 : CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

We have proposed a copy paste detection technique that achieves high detection accuracy of block based methods without compromising the advantage and recall rate offered by Key point detection based methods. This was achieved by primarily applying block based detection strategy in forgery detection and by judiciously applying the key point based detection at each and every block. By using the advantages of both the techniques we were able to detect the forgery up to 96.67% accuracy. We believe that our research has moved a step forward for a false proof copy move forgery detection technique that aims ultimately to assist law enforcement agencies to accept images as a piece of evidence.

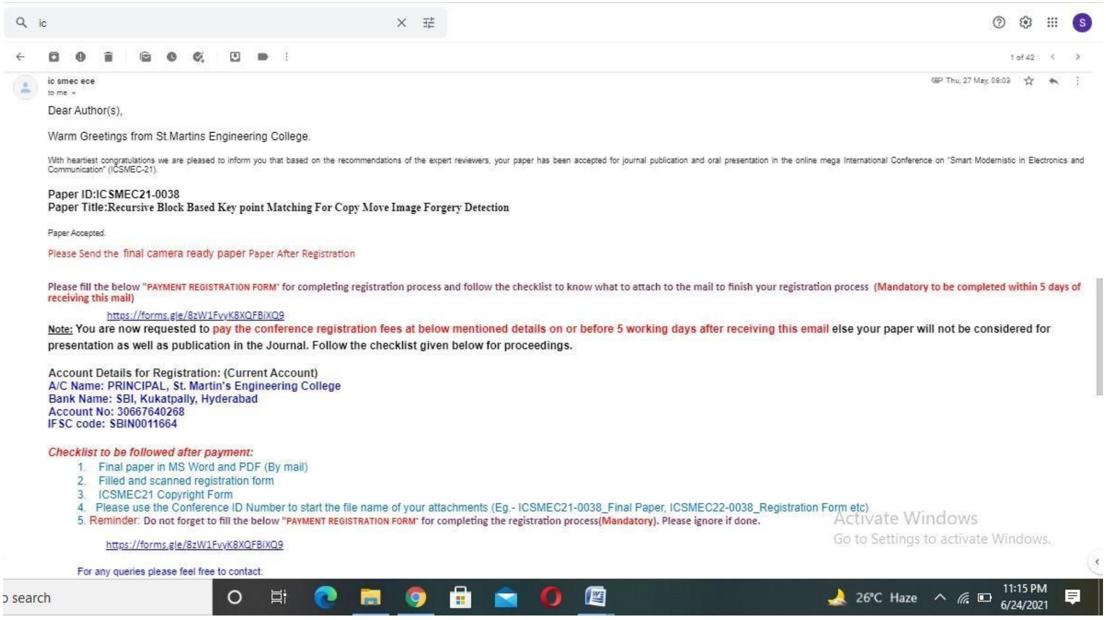
5.1.1 APPLICATIONS:

This technology can be used to verify the authenticity of an image.It can be used in situations where images are legal evidence and alsocan be used in forensic investigations.

5.2 FUTURE SCOPE:

- Future research in the field may concentrate on finding or identifying the forgery without the help of any query image and also to be able to find copy move forgery.
- It also may concentrate on recognizing profoundly compounded pictures. This method can further be developed for recognizing phony videos for investigation purposes.

PUBLICATION



ic smec ece
to me
Thu, 27 May, 08:03

Dear Author(s),

Warm Greetings from St.Martins Engineering College.

With heartiest congratulations we are pleased to inform you that based on the recommendations of the expert reviewers, your paper has been accepted for journal publication and oral presentation in the online mega International Conference on "Smart Modernistic in Electronics and Communication" (ICSMEC-21).

Paper ID:ICSMEC21-0038
Paper Title:Recursive Block Based Key point Matching For Copy Move Image Forgery Detection

Paper Accepted.

Please Send the final camera ready paper Paper After Registration

Please fill the below "PAYMENT REGISTRATION FORM" for completing registration process and follow the checklist to know what to attach to the mail to finish your registration process (Mandatory to be completed within 5 days of receiving this mail)

https://forms.gle/8zW1FvyK8XQFBiXQ9

Note: You are now requested to pay the conference registration fees at below mentioned details on or before 5 working days after receiving this email else your paper will not be considered for presentation as well as publication in the Journal. Follow the checklist given below for proceedings.

Account Details for Registration: (Current Account)
A/C Name: PRINCIPAL, St. Martin's Engineering College
Bank Name: SBI, Kukatpally, Hyderabad
Account No: 30667640268
IFSC code: SBIN0011664

Checklist to be followed after payment:

1. Final paper in MS Word and PDF (By mail)
2. Filled and scanned registration form
3. ICSMEC21 Copyright Form
4. Please use the Conference ID Number to start the file name of your attachments (Eg.- ICSMEC21-0038_Final Paper, ICSMEC22-0038_Registration Form etc)
5. Reminder: Do not forget to fill the below "PAYMENT REGISTRATION FORM" for completing the registration process(Mandatory). Please ignore if done.

https://forms.gle/8zW1FvyK8XQFBiXQ9

For any queries please feel free to contact:

The project has been approved by international conference that is being conducted by out college i.e., ICSMEC-21 for publication.

REFERENCES

[1] A Jessica Fridrich, B David Soukal, and A Jan Luka´s.ˇ "Detection of copy-move forgery in digital images". In: in Proceedings of Digital Forensic Research Workshop. 2003.

[2] Alin C Popescu and Hany Farid. Exposing digital forgeries by detecting duplicated image regions. Tech. rep. 2004.

[3] Weiqi Luo, Jiwu Huang, and Guoping Qiu. "Robust Detection of Region-Duplication Forgery in Digital Image". In: 18th International Conference on Pattern Recognition (ICPR'06). Vol. 4. 2006, pp. 746–749.

[4] G. Li et al. "A Sorted Neighborhood Approach for Detecting Duplicated Regions in Image Forgeries Based on DWT and SVD". In: 2007 IEEE International Conference on Multimedia and Expo. 2007, pp. 1750–1753.

[5] Babak Mahdian and Stanislav Saic. "Detection of copy–move forgery using a method based on blur moment invariants". In: Forensic Science International 171.2 (2007), pp. 180–189. ISSN: 0379-0738. DOI: https : / / doi . org / 10 . 1016 / j . forsciint . 2006 . 11 . 002. URL: http : / / www. sciencedirect . com / science / article / pii/S0379073806006748.

[6] X. Kang and S. Wei. "Identifying Tampered Regions Using Singular Value Decomposition in Digital Image Forensics". In: 2008 International Conference on Computer Science and Software Engineering. Vol. 3. 2008, pp. 926–930.

[7] S. Bayram, H. Taha Sencar, and N. Memon. "An efficient and robust method for detecting copy-move forgery". In: 2009 IEEE International Conference on Acoustics, Speech and Signal Processing. 2009, pp. 1053–1056.

[8] J. Wang et al. "Detection of Image Region Duplication Forgery Using Model with Circle Block". In: 2009 International Conference on Multimedia Information Networking and Security. Vol. 1. 2009, pp. 25–29.

[9] Junwen Wang et al. "Fast and robust forensics for image region-duplication forgery". In: Acta Automatica Sinica 35.12 (2009), pp. 1488–1495.

[10] Hwei-Jen Lin, Chun-Wei Wang, Yang-Ta Kao, et al. “Fast copy-move forgery detection”. In: WSEAS Transactions on Signal Processing 5.5 (2009), pp. 188–197.

[11] Seung-Jin Ryu, Min-Jeong Lee, and Heung-Kyu Lee. “Detection of copy-rotate-move forgery using Zernike moments”. In: International workshop on information hiding. Springer. 2010, pp. 51–65.

[12] Seung-Jin Ryu et al. “Rotation invariant localization of duplicated image regions based on Zernike moments”. In: IEEE Transactions on Information Forensics and Security 8.8 (2013), pp. 1355–1370.

[13] Sergio Bravo-Solorio and Asoke K Nandi. “Exposing duplicated regions affected by reflection, rotation and scaling”. In: 2011 IEEE International Conference onAcoustics, Speech and Signal Processing (ICASSP). IEEE. 2011, pp. 1880–1883.

[14] Irene Amerini et al. “A sift-based forensic method for copy–move attack detection and transformation recovery”. In: IEEE transactions on information forensics and security 6.3 (2011), pp. 1099–1110.

[15] Hailing Huang, Weiqiang Guo, and Yu Zhang. “Detection of copy-move forgery in digital images using SIFT algorithm”. In: 2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application. Vol. 2. IEEE. 2008, pp. 272–276.

[16] Xunyu Pan and Siwei Lyu. “Region duplication detection using image feature matching”. In: IEEE Transactions on Information Forensics and Security 5.4 (2010), pp. 857–867.

[17] Xu Bo et al. “Image copy-move forgery detection based on SURF”. In: 2010 International Conference on Multimedia Information Networking and Security. IEEE. 2010, pp. 889–892.

[18] BL Shivakumar and S Santhosh Baboo. “Detection of region duplication forgery in digital images using SURF”. In: International Journal of Computer Science Issues (IJCSI) 8.4 (2011), p. 199.

[19] C. Pun, X. Yuan, and X. Bi. “Image Forgery Detection Using Adaptive Oversegmentation and Feature Point Matching”. In: IEEE Transactions on Information Forensics and Security 10.8 (2015), pp. 1705–1716.

[20] Hrudya P et al. “Digital image forgery detection on artificially blurred images”. In: 2013 International Conference on Emerging Trends in Communication, Control, Signal Processing and Computing Applications (C2SPCA). 2013, pp. 1–5.

[21]P Subathro, A Baskar and D Senthil Kumar, "Detecting digital image forgeries using resampling by automatic Region of Interest (ROI)", ICTACT Journal on image and video processing, vol. 2, no. 4, 2012.

[22] Mary Saana, N J Sruthi, S Menon and G. Deepa, "Image Forgery Detection using Hash Functions", *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 1, 2019.

[23] Yue Wu, Wael Abd-Almageed and Prem Natarajan, "BusterNet: Detecting Copy-Move Image Forgery with Source/Target Localization", *The European Conference on Computer Vision (ECCV)*, Sept. 2018.

[24] Khurshid Asghar, Zulfiqar Habib and Muhammad Hussain, "Copy-move and splicing image forgery detection and localization techniques: a review", *Australian Journal of Forensic Sciences*, vol. 49, no. 3, pp. 281-307, 2017.

[25] David G Lowe, "Object recognition from local scale-invariant features", *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, pp. 1150-1157, 1999.

[26] Herbert Bay, Tinne Tuytelaars and Luc Van Gool, "Surf: Speeded up robust features", *European conference on computer vision*, pp. 404-417, 2006.

[27] Radhakrishna Achanta et al., "SLIC superpixels compared to state-of-the-art superpixel methods", *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274-2282, 2012.

APPENDICES

```
clc;
clear all;
close all;

run('toolbox/vl_setup')
database_dir = 'Database';

query_image = 'Query/1.jpg';

img = single(rgb2gray(imread(query_image)));
aa=[400 400];
img=imresize(img,aa);
num=zeros(1,2);
show = true;
%%
for l=1:1;
out_images = [database_dir'/' num2str(l) '.jpg'];

%
tic;



img_que=imread(out_images);
ifndims(img_que) == 3
img_que=single(rgb2gray(img_que));
else
img_que=single(img_que);
end
img_que=imresize(img_que,aa);

   [frames_query,sift_query] =
vl_covdet(img,'Method','DoG','EstimateOrientation',true);
```

```
    [frames_test,sift_test] =
vl_covdet(img_que,'Method','DoG','EstimateOrientation',true);
    threshold=2;
    [matches,x] = vl_ubcmatch(sift_query,sift_test,threshold);
%%%%%%%%%%% features
    loc1 = frames_query(1:2,:)';
    loc2 = frames_test(1:2,:)';

if(show)
im = appendimages(uint8(img),uint8(img_que));
        close all;
figure('Position', [256 256 size(im,2) size(im,1)]);
imshow(im);
        hold on;
        cols1 = size(img,2);

fori = 1:size(matches,2)

%matching features
            line([loc1(matches(1,i),1) loc2(matches(2,i),1)+cols1], ...
                 [loc1(matches(1,i),2) loc2(matches(2,i),2)], 'Color', 'b');
end
%         vl_plotframe(frames_query(:,matches(1,:)));
%         vl_plotframe(frames_test(:,matches(2,:))+repmat([cols1 0 0 0 0
0]',1,size(matches,2)))

        hold off;
%         pause;
end

toc;
end
```

A

PROJECT REPORT

On

Downlink User Selection For Massive MIMO-OFDM-IM Systems Using ZF Precoding

Submitted by

1)Mr. B.S.S. Prabhath (17K81A0407)
2)Mr. Ch. Rohith Kiran (17K81A0408)
3) Mr. C.S. Sai Teja (17K81A0411)

in partial fulfillment for the award of the degreeof

BACHELOR OF TECHNOLOGY
IN

ELECTRONICS AND COMMUNICATION

Under The Guidance of

Mr. S. Ravi Kumar

Associate Professor

DEPARTMENT OF ECE

ST.MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

BONAFIDE CERTIFICATE

This is to certify that the project entitled Downlink User Selection For Massive MIMO-OFDM-IM Systems Using ZF Precoding, is being submitted by 1. **Mr. B.S.S. Prabhath (17K81A0407), 2. Mr. Ch. Rohith Kiran (17K81A0408), 3. Mr. C.S. Sai Teja (17K81A0411)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

	Head of the Department
Mr. S. Ravi Kumar	Dr. B. Hari Krishna
Department of ECE	Department of ECE

Internal Examiner	External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **BSS. PRABHATH** WITH ROLL NO.**17K81A0407**, **CH. ROHITH KIRAN** WITH ROLL NO.**17K81A0408**, **C.S SAI TEJA** WITH ROLL NO**.17K81A0411**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**DOWNLINK USER SELECTION FOR MASSIVE-MIMO-OFDM-IM SYSTEMS USING ZF PRECODING**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com **| contact: 7330666881/82/83/84/86**

DECLARATION

We, the student of **Bachelor of Technology** in Department of Electronics and Communication', session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled Downlink User Selection For Massive MIMO-OFDM-IM Systems Using ZF Precoding is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

B.S.S. Prabhath (17K81A0407)
Ch. Rohith Kiran (17K81A0408)
C.S. Sai Teja (17K81A0411)

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr. P. Santosh Kumar Patra**, **our beloved Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr. B. Hari Krishna**, **Head &Professor**, the Dept. of Electronics and Communications for his/her advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We sincerely thank our project internal guide, Mr. S. Ravi Kumar, Associate Professor, Department of ECE and our class project coordinator Ms. A. Sowmya, Assistant Professor, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this project work and for their support and co- operation that is difficult to express in words.

1. B.S.S. Prabhath
2. Ch. Rohith Kiran
3. C.S. Sai Teja

// ABSTRACT

By the use of a large number of M antennas in the base station side, makes it possible to beamform different signals to different users, so the signals add up constructively at the desired user and destructively everywhere else, that is why massive MIMO-ODFM combing with beamforming is a promising technology for next-generation wireless communication network (5G).

Indeed, the performance of Massive MIMO systems is systematically dependent on users' selection method especially when the number of users is huge in the cell. In this project, we focus on the performance of such a system in terms of max rate using the zero-forcing precoding method. We use MATLAB software for the simulation process and the performance of the user selection techniques are observed. In the existing method the data rate is low and bit error rate of the system is high which is not ideal for 5G wireless communication network. In our project we overcome these limitations using MIMO-OFDM-IM system which is very ideal for 5G wireless communication network.

.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Orthogonal frequency division multiplexing (OFDM) has become the most popular multicarrier signalling form for high-speed wireless communications and has been included in many standards such as Long Term Evolution (LTE), IEEE 802.11 wireless local area network (WLAN) and digital video broadcasting (DVB). Due to its efficient implementation and robustness to the frequency selectivity, OFDM and its combination with multiple-input multiple-output (MIMO) system sun surprisingly appear as a strong alternative for 5G networks. OFDM with index modulation (OFDM-IM) is a recently proposed novel scheme that transmits information not only by M-ary constellation symbols but also by the indices of the active subcarriers which are activated according to the incoming information bits. Subcarrier index modulation techniques for OFDM have attracted considerable attention from researchers and have been investigated in some recent studies due to the interesting trade-offs they offer in error performance and spectral efficiency compared to classical OFDM systems. The bit error probability of OFDM-IM is analytically derived. The spectral efficiency of OFDM-IM is improved by selecting the active subcarriers in a more flexible way, where index modulation is applied for both in-phase and quadrature components of the subcarriers. The authors deal with the problem of selecting the optimal number of active subcarriers in OFDM-IM. More recently, OFDM-I Mis-combined with coordinate interleaving to achieve additional diversity gains. However, the combination of OFDM-IM and MIMO transmission techniques remains an open and interesting research problem.

In this study, we propose MIMO-OFDM-IM with index modulation (MIMO-OFDM-IM) as an efficient alternative multicarrier transmission scheme for 5G networks by combining MIMO and OFDM-IM transmission techniques. In the proposed scheme, each transmit antenna transmits its OFDM-IM frame as in Vertical Bell Labs layered space-time (V-BLAST) scheme, and at the receiver side, these OFDM-IM frames are separated and demodulated using a novel and low complexity minimum mean square error (MMSE) detection and log-likelihood ratio (LLR) calculation based detector. It is shown via computer simulations that the MIMO-OFDM-IM scheme achieves significantly better bit error rate (BER) performance than classical V-BLAST type MIMO-OFDM-IM for several

MIMO configurations. The rest of the letter is organized as follows. The system model of MIMO-OFDM-IM is presented, receiver structure of the MIMO –OFDM –IM scheme is given. Simulation results are provided. we propose MIMO-OFDM-IM with index modulation (MIMO-OFDM-IM) as an efficient alternative multicarrier transmission scheme for 5G networks by combining MIMO and OFDM-IM transmission techniques. In the proposed scheme, each transmit antenna transmits its OFDM-IM frame as in Vertical Bell Labs layered space-time (V-BLAST) scheme [12], and at the receiver side, these OFDM-IM frames are separated and demodulated using a novel and low complexity minimum mean square error (MMSE) detection and log-likelihood ratio (LLR) calculation based detector.

Orthogonal frequency division multiplexing (OFDM) has become one of the most common multicarrier transmission techniques for wideband wireless communications in recent years. Because of its benefits like efficient implementation and strength to frequency selective fading channels, OFDM has been enclosed in several standards like long term evolution (LTE), IEEE 802.11x wireless local area network (LAN), digital video broadcasting (DVB) and IEEE 802.16e-WiMAX. Considering the benefits of multiple-input multiple-output (MIMO) systems over single-antenna systems like improved rate and energy efficiency, the mixture of OFDM and MIMO transmission techniques seems like a robust difference for future wireless standards like 5G and beyond. OFDM with index modulation (OFDM-IM) can be a new multicarrier transmission method that has been planned as another to classical OFDM. Stimulating from the SM conception, in OFDM-IM, index modulation techniques are applied for the indices of the offered subcarriers of an OFDM system.

In the OFDM-IM method, only a set of available subcarriers are preferred as active consistent with the data bits, whereas the remaining inactive subcarriers are set zero. In different words, the data is transmitted not only by the information symbols selected from M-ary signal constellations, however additionally by the indices of the active subcarriers. Not like classical OFDM, the number of active subcarriers may be adjusted within the OFDM-IM scheme, and this flexibility within the system style provides a stimulating trade-off between error performance and spectral efficiency. Moreover, it's been shown that OFDM-IM has the potential to realize a more robust error performance than classical OFDM for low-to-mid spectral efficiency values. Due to its adjustable range of active subcarriers, OFDM-IM may be a potential candidate not just for high-speed wireless communications systems however additionally for machine-to-machine (M2M) communications systems that need low power consumption.

In statistics and signal processing, a minimum mean square error (MMSE) estimator is an estimation method that minimizes the mean square error (MSE), which is a common measure of estimator quality, of the fitted values of a dependent variable. In the Bayesian setting, the term MMSE more specifically refers to estimation with a quadratic loss function. In such a case, the MMSE estimator is given by the posterior mean of the parameter to be estimated. Since the posterior mean is cumbersome to calculate, the form of the MMSE estimator is usually constrained to be within a certain class of functions. Linear MMSE estimators are a popular choice since they are easy to use, easy to calculate, and very versatile. It has given rise to many popular estimators such as the Wiener–Kolmogorov filter and Kalman filter.

An M-ary transmission is a type of digital modulation where instead of transmitting one bit at a time, two or more bits are transmitted simultaneously. This type of transmission results in reduced channel bandwidth. This process is known as quadrature modulation.

M-ary Encoding. The word binary represents two-bits. This is the type of digital modulation technique used for data transmission in which instead of one-bit, two or more bits are transmitted at a time. As a single signal is used for multiple bit transmission, the channel bandwidth is reduced. An M-ary transmission is a type of digital modulation where instead of transmitting one bit at a time, two or more bits are transmitted simultaneously. This type of transmission results in reduced channel bandwidth. However, sometimes, two or more quadrature carriers are used for modulation. Multi-level modulation techniques permit high data rates within fixed bandwidth constraints. A convenient set of signals for M-ary PSK is

$\varphi i(t) = A \cos(\omega ct + \theta i), 0 < t \leq Ts,$

where the M phase angles are $\theta i = 0, 2\pi M , . . ., 2(M - 1)\pi M$.

For equiprobable ones and zeros, the PSD for M-ary PSK is S

$\varphi(\omega) = A 2 TsSa2(\omega - \omega c) Ts 2.$

The symbols, in this case, are of duration Ts, so the information (or bit rate) Tb satisfies Ts = Tb log2 M. The potential bandwidth efficiency of M-ary PSK can be shown to be FB B = log2 M bps/Hz.

All signals have the same energy Es over the interval (0, Ts), and each signal is correctly demodulated at the receiver if the phase is within $\pm\pi/M$ of the correct phase θi. No information is contained in the energy of the signal. A probability of error calculation involves analysing the received phase at the receiver (in the presence of noise) and comparing it to the actual phases. An exact solution is difficult to compute but for P< 10−3 an approximate probability of making a symbol error is P 2erfcs 2Es η sin2 π M, M > 2. If a Gray code is used, then the corresponding bit error is approximately Pbe/ log2 M. Stremler provides a table of the SNR requirements of M-ary PSK for fixed error rates. The results indicate that for QPSK (M = 4) has definite advantages over coherent PSK (M = 2) the bandwidth efficiency is doubled for only about a 0.3dB increase in SNR. For higher-rate transmissions in band-limited channels, the choice M = 8 is often used. Values of M > 8 are seldom used due to excessive power requirements. M-ary PSK requires more complex equipment than BPSK signalling. Carrier recovery is also more complicated. The requirement that the carrier is recovered can be mitigated by using a comparison between the phases of two successive symbols. This leads to M-ary differential PSK and is in principle similar to DPSK (which is differential PSK for M = 2). For large SNR the probability of error is P=2erfcs 2Es η sin2 π √ 2M. Thus differential detection increases the power requirements by the factor 0 = sin2 π/M sin2 (π/ √ 2M) For m = 4, the increase in required power is about 2.5dB, which may be justified by the saving in equipment complexity.

In telecommunications, orthogonal frequency-division multiplexing (OFDM) is a method of encoding digital data on multiple carrier frequencies. OFDM has developed into a popular scheme for wideband digital communication, used in applications such as digital television and audio broadcasting, DSL internet access, wireless networks, power line networks, and 4G mobile communications.

In coded orthogonal frequency-division multiplexing (COFDM), forward error correction (convolutional coding) and time/frequency interleaving are applied to the signal being transmitted. This is done to overcome errors in mobile communication channels affected by multipath propagation and Doppler effects. COFDM was introduced by Alard in 1986 for Digital Audio Broadcasting for Eureka Project 147. In practice, OFDM has become used in combination with such coding and interleaving, so that the terms COFDM and OFDM co-apply to common applications.

OFDM is a frequency-division multiplexing (FDM) scheme used as a digital multi-carrier modulation method. OFDM was introduced by Chang of Bell Labs in 1966. Numerous closely spaced orthogonal sub-carrier signals with overlapping spectra are emitted to carry data. Demodulation is based on Fast Fourier Transform algorithms. OFDM was improved by Weinstein and Ebert in 1971 with the introduction of a guard interval, providing better orthogonality in transmission channels affected by multipath propagation. Each sub-carrier (signal) is modulated with a conventional modulation scheme (such as quadrature amplitude modulation or phase-shift keying) at a low symbol rate. This maintains total data rates similar to conventional single-carrier modulation schemes in the same bandwidth.

Conceptually, OFDM is specialized frequency-division multiplexing (FDM) method, with the additional constraint that all subcarrier signals within a communication channel are orthogonal to one another.

In OFDM, the sub-carrier frequencies are chosen so that the sub-carriers are orthogonal to each other, meaning that cross-talk between the sub-channels is eliminated and inter-carrier guard bands are not required. This greatly simplifies the design of both the transmitter and the receiver; unlike conventional FDM, a separate filter for each sub-channel is not required.

The orthogonality requires that the sub-carrier spacing is Hertz, where T_U seconds is the useful symbol duration (the receiver-side window size), and k is a positive integer, typically equal to 1. This stipulates that each carrier frequency undergoes k more complete cycles per symbol period than the previous carrier. Therefore, with N sub-carriers, the total passband bandwidth will be $B= N*\Delta f$ (Hz).

The orthogonality also allows high spectral efficiency, with a total symbol rate near the Nyquist rate for the equivalent baseband signal (i.e. near half the Nyquist rate for the double-sideband physical passband signal). Almost the whole available frequency band can be used. OFDM generally has a nearly 'white' spectrum, giving it benign electromagnetic interference properties with respect to other co-channel users.

In radio, multiple-input and multiple-output, or MIMO, is a method for multiplying the capacity of a radio link using multiple transmission and receiving antennas to exploit multipath propagation. MIMO has become an essential element of wireless communication standards including IEEE802.11n (Wi-Fi), IEEE802.11ac (Wi-Fi), HSPA+ (3G), WiMAX (4G), and Long Term Evolution (LTE4G). More recently, MIMO has been applied to power-line communication for 3-wire installations as part of ITU G.hn standard and HomePlugAV2 specification.

At one time, in wireless, the term "MIMO" referred to the use of multiple antennas at the transmitter and the receiver. In modern usage, "MIMO" specifically refers to a practical technique for sending and receiving more than one data signal simultaneously over the same radio channel by exploiting multipath propagation. MIMO is fundamentally different from smart antenna techniques developed to enhance the performance of a single data signal, such as beamforming and diversity.

MIMO is often traced back to 1970s research papers concerning multi-channel digital transmission systems and interference (crosstalk) between wire pairs in a cable bundle: AR Kaye and DA George (1970), Branderburg and Wyner (1974), and W. van Etten (1975, 1976). Although these are not examples of exploiting multipath propagation to send multiple information streams, some of the mathematical techniques for dealing with mutual interference proved useful to MIMO development. In the mid-1980s Jack Salz at Bell Laboratories took this research a step further, investigating multi-user systems operating over "mutually cross-coupled linear networks with additive noise sources" such as time-division multiplexing and dually-polarized radio systems.

Methods were developed to improve the performance of cellular radio networks and enable more aggressive frequency reuse in the early 1990s. Space-division multiple access (SDMA) uses directional or smart antennas to communicate on the same frequency with users in different locations within a range of the same base station. An SDMA system was proposed by Richard Roy and Bjorn Ottersten, researchers at ArrayComm, in 1991. Their US patent (No. 5515378 issued in 1996) describes a method for increasing capacity using "an array of receiving antennas at the base station" with a "plurality of remote users."

Arogyaswami Paulraj and Thomas Kailath proposed an SDMA-based inverse multiplexing technique in 1993. Their US patent (No. 5,345,599 issued in 1994) described a method of broadcasting at high data rates by splitting a high-rate signal "into several low-rate signals" to be transmitted from "spatially separated transmitters" and recovered by the receive antenna array based on differences in "directions-of-arrival." Paulraj was awarded the prestigious Marconi Prize in 2014 for "his pioneering contributions to developing the theory and applications of MIMO antennas. His idea for using multiple antennas at both the transmitting and receiving stations – which is at the heart of the current high-speed WiFi and 4G mobile systems – has revolutionized high-speed wireless."

In an April 1996 paper and subsequent patent, Greg Raleigh proposed that natural multipath propagation can be exploited to transmit multiple, independent information streams using co-located antennas and multi-dimensional signal processing. The paper also identified practical solutions for modulation (MIMO-OFDM-IM), coding, synchronization, and channel estimation. Later that year (September 1996) Gerard J. Foschini submitted a paper that also suggested it is possible to multiply the capacity of a wireless link using what the author described as "layered space-time architecture."

Greg Raleigh, V. K. Jones, and Michael Pollack founded Clarity Wireless in 1996, and built and field-tested a prototype MIMO system. Cisco Systems acquired Clarity Wireless in 1998. Bell Labs built a laboratory prototype demonstrating its V-BLAST (Vertical-Bell Laboratories Layered Space-Time) technology in 1998. Arogyaswami Paulraj founded Iospan Wireless in late 1998 to develop MIMO-OFDM-IM products. Iospan was acquired by Intel in 2003. V-BLAST was never commercialized, and neither Clarity Wireless nor Iospan Wireless shipped MIMO-OFDM-IM products before being acquired.

A Massive MIMO system, shown in Figure 1, is a system comprising a relatively high number of M antennas (M> 10) at the base station [4, 2, 7]. This large number of radiating elements, each controlled numerically, will allow fine control of the overall radiation of the base station. Although quite suitable for point-to-point transmissions, this feature fact that this solution is often associated with multi-user techniques because of its potential to focus energy towards several particular positions of space. This characteristic is implemented by the precoding process, whose role is to exploit certain knowledge of the propagation channel to target multiple users and serve simultaneously. Each of them can then benefit from the totality bandwidth and get a very high bit rate.

Note that many studies consider single-antenna terminals [1, 5, 4] because all processing is performed at the base station, the user not having the information needed to perform post coding treatments. This focus on energy is often associated with the term energy efficiency which helps to make it a very appropriate solution for the future [3, 9]. Indeed, this focus results in a higher gain related to precoding without increasing the transmission power. In other words, a Massive MIMO system theoretically allows dividing the radio power necessary to obtain the same quality of service as a SISO system by a factor proportional to the number of antennas of the system (α M) [4,6]. Designed and properly used, a massive MIMO system can improve transmission throughput and reliability and provide a better compromise between energy use and signal spectral efficiency. Besides, a large-scale antenna system can eliminate the effects of small-scale fading and uncorrelated noise. Simple linear precoding and detection have been found to work optimally. The philosophy of Massive MIMO systems is therefore based on the use of a large number of transmit antennas to serve a limited number of users simultaneously.

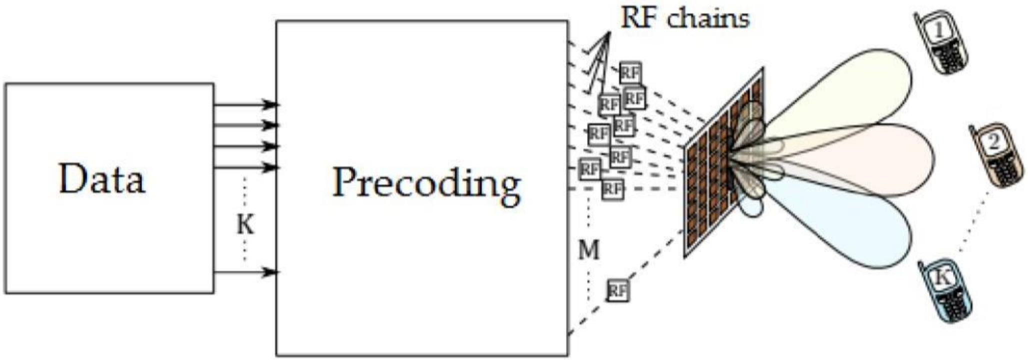


Figure 1.1 : Massive MIMO system

On the other hand, the number of users that can be supported simultaneously is closely dependent on the number of transmit antennas. Also, the performance of Massive MIMO systems is mainly related to the user selection method. In the literature, the problem of user scheduling in MIMO systems has been extensively studied. For example, the authors propose three heuristic scheduling algorithms for CDMA MIMO systems. In the massive context of MIMO, very few articles have addressed this problem. In, the authors propose to solve the problem of joint antenna selection and user planning, but in massive MIMO systems distributed under the constraint of capacity. The optimal strategy facing the problem of user scheduling is guaranteed by an exhaustive search (search by brute force), which generates a computationally prohibitive complexity.

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

Beixiong Zheng et al.“Multiple-Input Multiple-Output OFDM with Index Modulation: Low-Complexity Detector Design”, in this paper proposed two low-complexity detectors derived from the SMC theory for the MIMO-OFDM-IM system. The first proposed subblock-wise detector draws samples at the subblock level, exhibiting near-optimal performance for the MIMO-OFDM-IM system. The second proposed subcarrier-wise detector draws samples at the subcarrier level, exhibiting substantially reduced complexity with a marginal performance loss. An effective legality examination method has been also developed to couple with the subcarrier wise detector. Computer simulation and numerical results have validated the outstanding performance and the low complexity of both proposed detectors.

Ertugrul Basar et al.“Multiple-Input Multiple-Output OFDM with Index Modulation”, A novel scheme called MIMO-OFDM-IM with index modulation has been proposed as an alternative multicarrier transmission technique for 5G networks. It has been shown via extensive computer simulations that the proposed scheme can provide significant BER performance improvements over classical MIMO-OFDM-IM for several different configurations. The following points remain unsolved in this study:

i) performance analysis,

ii) the selection of optimal N and K values,

iii) diversity techniques for MIMO-OFDM-IM, and

iv) Implementation scenarios for high mobility. Ertugrul Basar et al. “On Multiple-Input Multiple-Output OFDM with Index Modulation for Next Generation Wireless Networks”, In this study, the recently proposed MIMO-OFDM-IM scheme has been investigated for next-generation 5G wireless networks. For the MIMO-OFDM-IM scheme, new detector types such as ML, near-ML, simple MMSE, MMSE-LLR-OSIC detectors have been proposed and their ABEP has been theoretically examined. It has been shown via extensive computer simulations that the MIMO-OFDM-IM scheme provides an interesting trade-off between complexity, spectral efficiency and error performance compared to the classical MIMO-OFDM-IM scheme and it can be considered as a possible candidate for 5G wireless networks. The main features of MIMO-OFDM-IM can be summarized as follows:

i) better BER performance,

ii) flexible system design with a variable number of active OFDM subcarriers and

iii) better compatibility to higher MIMO setups. However, interesting topics such as diversity methods, generalized OFDM-IM cases, high mobility implementation and transmit antenna indices selection remain to be investigated for the MIMO-OFDM-IM scheme.

Ertugrul Bas¸ et al. “Performance of Multiple-Input Multiple-Output OFDM with Index Modulation”, In this paper, proposed ML and near-ML detectors for the recently introduced MIMO-OFDM-IM scheme to improve its error performance compared to MMSE based detection.

The ABEP upper bound of the MIMO-OFDM-IM scheme with ML detection has been derived and it has been shown that the derived theoretical upper bound can be used as an efficient tool to predict the BER performance of the MIMO-OFDM-IM scheme. It has been shown via computer simulations that the MIMO- FDM-IM scheme can provide significant improvements in BER performance over classical MIMO-OFDM-IM using a different type of detectors and MIMO configurations.

Beixiong Zheng et al. “Low-Complexity ML Detector and Performance Analysis for OFDM With In-Phase/Quadrature Index Modulation”, In this letter, we've planned a low-complexity detector supported the millilitre criterion, that dispenses with a priori data of the noise variance and also the potential realizations of the active subcarrier indices. supported the framework of OFDM-I/Q-IM using the planned millilitre detector, the straight line ABEP and also the actual coding gain achieved by OFDMI/Q-IM are derived, that matches the simulation results.

Moreover, the exact coding gain including the spectral efficiency price has provided a clear plan of a basic trade-off between the system performance and also the spectral efficiency of OFDM-I/Q-IM by the adjustment of the number of active subcarriers.

Sheng Wu et al.“Low-Complexity Iterative Detection for Large-Scale Multiuser MIMO-OFDM-IM Systems Using Approximate Message Passing”, For the detection of large-scale multiuser MIMO-OFDM-IM systems, we have proposed a range of low-complexity approximate message passing algorithms that can offer a desirable tradeoff between performance and complexity.

It is verified through extensive simulations that our proposed approximate message passing algorithms can achieve near-optimal performance with low complexity. Compared with existing turbo detection algorithms, the proposed schemes can achieve or even outperform the performance of some complex algorithms, such as the iterative decoding based on STS-SD and MMSESIC. In addition, the number of iterations required to achieve near-optimal performance is small and does not increase with the system dimension.

CHAPTER 3

SOFTWARE REQUIREMENTS

3.1 INTRODUCTION TO MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modelling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In the industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available to include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

3.2 THE MATLAB SYSTEM:

The MATLAB system consists of five main parts

- **Development Environment:**

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library:**

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language:**

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both ―programming in the small‖ to rapidly create quick and dirty throw-away programs, and ―programming in the large‖ to create large and complex application programs.

- **Graphics:**

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using the IMAGE PROCESSING toolbox.

3.3 GRAPHICAL USER INTERFACE (GUI):

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG file contains binary data that does not need to be parsed when the associated GUI-based M function is executed.
- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited and callback functions that are executed when a user interacts with GUI objects for example when a button is pushed.

To launch GUIDE from the MATLAB command window, type
guide filename
Where filename is the name of an existing FIG-file on the current path. If the filename is omitted, GUIDE opens a new (i.e., blank) window.

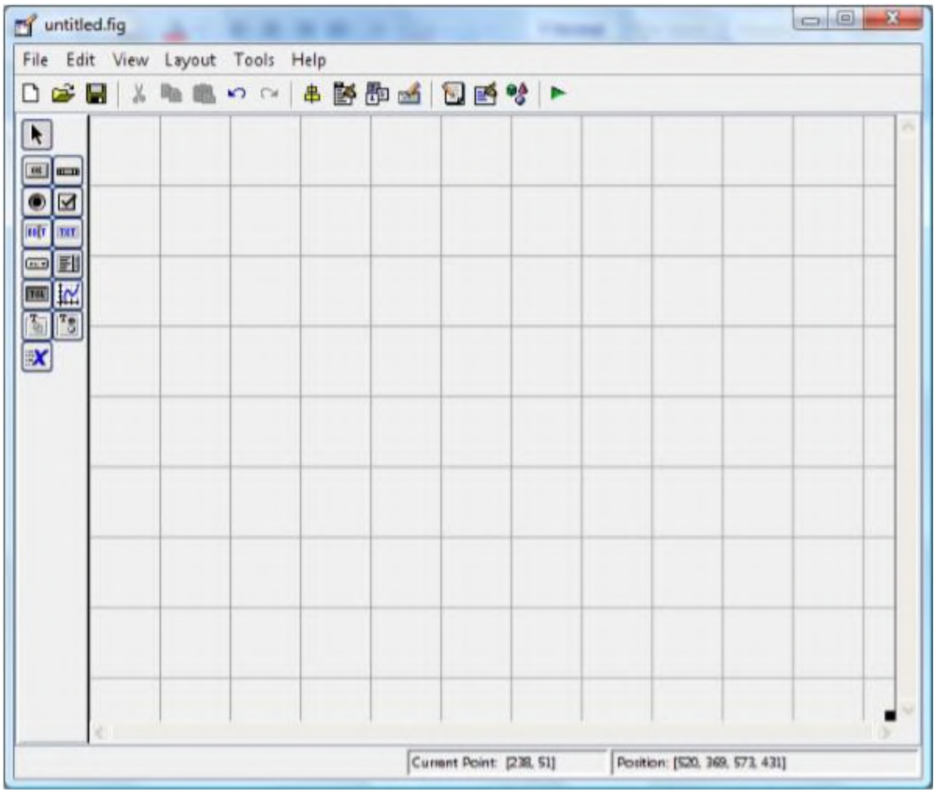


Figure 3.1 : Matlab procedure

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, pushbuttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

3.4 GETTING STARTED

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

3.4.1 DESCRIPTION - describes the components of the MATLAB system.

3.4.2 DEVELOPMENT ENVIRONMENT - describes the MATLAB development environment, including information about tools and the MATLAB desktop.

3.4.3 MANIPULATING MATRICES - describes how to use MATLAB to generate matrices and perform mathematical operations on matrices.

3.4.4 GRAPHICS - describes MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images. 2.4.5 Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

3.5 DEVELOPMENT ENVIRONMENT

3.5.1 INTRODUCTION

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

3.5.2 STARTING MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop. On a UNIX platform, to start MATLAB, type Matlab

At the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup and reduce startup time in some situations.

3.5.3 QUITTING MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu on the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

3.5.4 MATLAB DESKTOP

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB. The first time MATLAB starts, the desktop appears as shown in the following illustration, although your LaunchPad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

3.5.5 DESKTOP TOOLS

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- LaunchPad
- Help Browser Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacsmagik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

LaunchPad

MATLAB's LaunchPad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the Help button in the toolbar, or type help browser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name. Favourites tab - View a list of documents you previously designated as favourites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term on the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find Mfiles and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT file, which has a .mat extension. There are options for saving to different formats. To read in a MAT file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to run MATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop.

CHAPTER 4
PROJECT DESCRIPTION

A typical FDD mode massive MIMO-OFDM-IM system was established. We consider a single-cell massive MIMO system consisting of one Base station equipped with M antennas and K single-antenna users, where M≫K. We propose the case of a single-cell system that means there is no interference co-channel. The downlink or broadcast system is described as follows. The base station with M antennas transmits data simultaneously to K active users chosen from a total of U users, each with one receive antenna. The base station separates the multiuser data streams by beamforming, i.e. assigning a beamforming vector to each of the K active users. The beamforming vectors $wn \;\; n$=1 M are selected from multiple sets of unitary orthogonal vectors following the beam and user selection algorithm. Equal power allocation over-scheduled users are considered. The received signal of u the scheduled user is expressed as:

$$y_u = \sqrt{\frac{P}{M}} h_k^* \sum_{n \in A} w_n x_n + v_u \qquad u \in A$$

Where we use the following notation: M: number of transmit antennas and also the number of scheduled users; hu (M × 1 vector) downlink channel; xu transmitted symbol with E[|xu|2] = 1; yu received symbol; wu (M × 1 vector) beamforming vector, A The index set of scheduled users, P: transmission power;

B. Zero-Forcing Beamforming Zero-forcing (ZF) decomposes the channel into several parallel scalar channels with only additive noise, and the interference is removed completely by transmitting beamforming techniques [8]. Suppose a base station with M transmit antennas transmits information to K users, each user is equipped with one single receive antenna. The received signal by the kth user is

$$y_k = h_k^H v_k x_k + h_k^H \sum_{j=1, j \neq k}^{K} v_j x_j + n_k \qquad 1 \leq k \leq K$$

ZF method transmits the signals towards the intended user with nulls steered in the direction of the other users, i.e., ($hj \;\; Hvk$ = 0 $\forall j \neq k$). The users will receive only the desired signal without any interference because of the perfect nulling. In this case, the received data at the kth user can be written

The corresponding vector equation is

$$y = HhVx + n \ (4)$$

Therefore, if the normalized transmit beamforming vector of the kth user is selected

Therefore, if the normalized transmit beamforming vector of the kth user is selected $vk\ ' = hk\ (\dagger)\ hk\ (\dagger)\ F\ 2$ (5) where $hk\ (\dagger)$ is the kth column of the pseudo inverse of H, denoted as H(y). Then it is shown that the interference can be cancelled completely.

4.1 MULTI INPUT-MULTI OUTPUT

The demand for high bit rate has increased in recent wireless communication networks. Theories by various engineers have proven that the Multiple Input Multiple Output (MIMO) technology can improve the problem of traffic capacity in wireless networks. MIMO systems can be defined as the use of multiple antennas at both the transmitting and receiving ends of a wireless communication network. The systems take advantage of multipath transmission paths. Although various efforts have been made by engineers to improve the data rate, the capacity is never enough for users. Users of mobile wireless devices like to be able to use their devices in streaming live programs, playing more online games and streaming an online movie which involves high data rates.

Telecommunication companies and Internet Service Providers (ISPs) as an example in Africa find it difficult to provide high data rate Internet services to their network users, especially mobile users, due to environmental factors. The only option for most of these companies is to provide the Internet with a high data rate wirelessly. With the limited bandwidth in space, MIMO technology will be of great benefit to these companies in providing high data rate Internet services to their customers. Currently, cellular systems, such as the third-generation (3G) cellular system, satellite communication systems and video broadcasting systems have experienced a great increase in capacity in the implementation of MIMO channel technology. Access point devices such as wireless local area networks (WLAN) routers have also experienced a great change in transmission techniques, with a few using MIMO technologies. The main goal of this project is to explain and illustrate the operation of MIMO channel technology.

4.1.1 Background of Multiple Input Multiple Output:

The multiple input multiple output channel technologies are aimed to increase the capacity in the wireless communication network. With the invention of MIMO, the technology seems to gain popularity as it is being implemented in the current commercial wireless products and networks such as broadband wireless access systems, wireless local area networks (WLAN), 3G networks, etc. Figure 1 shows a line of sight (LOS) antenna setup of a MIMO system.

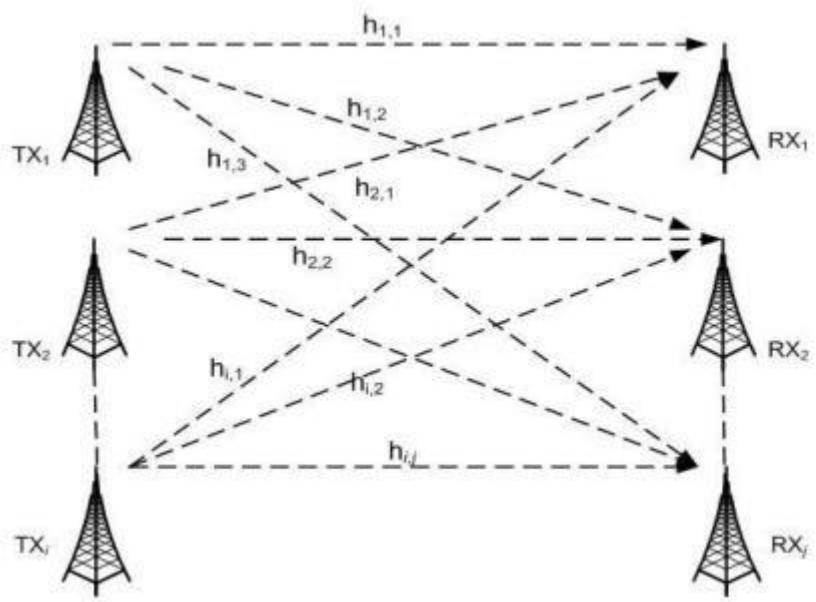


Figure 4.1 : A Generalized MIMO Wireless Communication System

The main idea behind MIMO is that the sampled signals in a spatial domain at both the transmitter and receiver end are combined so that they form effective multiple parallel spatial data streams which increase the data rate. The occurrence of diversity also improves the quality that is the bit-error-rate (BER) of the communication

4.1.2 SINGLE INPUT SINGLE OUTPUT VERSUS MIMO CHANNEL CAPACITY:

In communication systems, input discrete source symbols are mapped into a sequence of channel symbols which are then transmitted through the wireless channel. The transmission of channel symbols through the wireless channel is by nature random and random noise is added to the channel symbols. The measure of how much information can be transmitted and received with a minimum probability of error is called the channel capacity. A Single Input Single Output system involves the use of one antenna both at the transmitter and receiver end. To a telecommunications engineer, there exit a limit at which reliable transmission of information is not possible for a given transmission bandwidth and power. These limits were discovered by Claude Shannon in 1948 when he established the principles of information and communication theory in his various publications.

Shannon also established the conditions that enable the transmission of information over a noisy channel at a given rate, for a given power of the signal and noise. These limiting factors are the finite bandwidth and the S/N of the channel. This is because for a communication channel to accommodate the signal spectrum, enough transmission bandwidth is needed otherwise there will be distortion. “The higher data rate is to be transmitted, the shorter digital pulses must be used and the shorter digital pulses are used for transmission, the wider bandwidth is required”. For a deterministic channel with a bandwidth (B) with additive noise, Shannon proved that information with a rate of r bits per second (bps) can be transmitted with a small error probability provided that the bit rate is less than the capacity of the channel $r < C$. The Shannon formulae that can be applied to determine the maximum capacity C of the channel is of the form C = Blog[1 +T] [(bits s)/H T], where S/N, the signal-to-noise ratio and B the bandwidth of the transmission channel.

The above Equation informs us of how power and bandwidth are related. Assuming we have a channel with additive noise N and that we have some freedom of choosing the average transmission power S, to set up a reliable transmission link to send r bits per second. From the Shannon theorem, the data rate r cannot exceed capacity C, $r < C$ as in equation, but we still have one degree of freedom in the choice of bandwidth B and power S. It can be realized that, for a given signal-to-noise ratio S/N, if we wish to double C, we have to double the bandwidth B. On the other hand, if we double C, for a given B we have to evaluate the S/N.

4.1.3 WHAT IS MIMO?

MIMO (multiple inputs, multiple outputs) is an antenna technology for wireless communications in which multiple antennas are used at both the source (transmitter) and the destination (receiver). The antennas at each end of the communications circuit are combined to minimize errors and optimize data speed. MIMO is one of several forms of smart antenna technology, the others being MISO (multiple inputs, single output) and SIMO (single input, multiple outputs). In conventional wireless communications, a single antenna is used at the source, and another single antenna is used at the destination. In some cases, this gives rise to problems with multipath effects. When an electromagnetic field (EM field) is met with obstructions such as hills, canyons, buildings, and utility wires, the wavefronts are scattered, and thus they take many paths to reach the destination. The late arrival of scattered portions of the signal causes problems such as fading, cut-out (cliff effect), and intermittent reception (picket fencing). In digital communications systems such as wireless Internet, it can cause a reduction in data speed and an increase in the number of errors. The use of two or more antennas, along with the transmission of multiple signals (one for each antenna) at the source and the destination, eliminates the trouble caused by multipath wave propagation, and can even take advantage of this effect.

MIMO technology has aroused interest because of its possible applications in digital television (DTV), wireless local area networks (WLANs), metropolitan area networks (MANs), and mobile communications. In radio, multiple-input and multiple-output, or MIMO is a method for multiplying the capacity of a radio link using multiple transmission and receiving antennas to exploit multipath propagation. MIMO has become an essential element of wireless communication standards including IEEE 802.11n (Wi-Fi), IEEE 802.11ac (Wi-Fi), WiMAX, and Long-Term-Evolution. More recently, MIMO has been applied to power-line communication for 3-wire installations as part of the standard and Home Plug AV2 specification. At one time, in wireless, the term "MIMO" referred to the use of multiple antennas at the transmitter and the receiver. In modern usage, "MIMO" specifically refers to a practical technique for sending and receiving more than one data signal simultaneously over the same radio channel by exploiting multipath propagation. MIMO is fundamentally different from smart antenna techniques developed to enhance the performance of a single data signal, such as beamforming and diversity. MIMO technology has been standardized for wireless LANs, 3G mobile phone networks, and 4G mobile phone networks and is now in widespread commercial use. Greg Raleigh and V. K. Jones founded Airgo Networks in 2001 to develop MIMO-OFDM-IM chipsets for wireless LANs. The Institute of Electrical and Electronics Engineers (IEEE) created a task group in late 2003 to develop a wireless LAN standard delivering at least 100 Mbit/s of user data throughput. There were two major competing proposals: TGn Sync was backed by companies including Intel and Philips, and WWiSE was supported by companies including Airgo Networks, Broadcom, and Texas Instruments. Both groups agreed that the 802.11n standard would be based on MIMO-OFDM-IM with 20 MHz and 40 MHz channel options. TGn

Sync, WWiSE, and a third proposal (MITMOT, backed by Motorola and Mitsubishi) were merged to create what was called the Joint Proposal. In 2004, Airgo became the first company to ship MIMO-OFDM-IM products. Qualcomm acquired Airgo Networks in late 2006. The final 802.11n standard supported speeds up to 600 Mbit/s (using four simultaneous data streams) and was published in late 2009.

Surendra Babu Mandava and Arogyaswami Paulraj founded Beceem Communications in 2004 to produce MIMO-OFDM-IM chipsets for WiMAX. The company was acquired by Broadcom in 2010. Wi-MAX was developed as an alternative to cellular standards, is based on the 802.16e standard, and uses MIMO-OFDM-IM to deliver speeds up to 138 Mbit/s. The more advanced 802.16m standard enables download speeds up to 1 Gbit/s. A nationwide WiMAX network was built in the United States by Clear wire, a subsidiary of Sprint-Nextel, covering 130 million points of presence (PoP) by mid-2012. Sprint subsequently announced plans to deploy LTE (the cellular 4G standard) covering 31 cities by mid-2013 and to shut down its Wi-MAX network by the end of 2015.

The first 4G cellular standard was proposed by NTT DoCoMo in 2004. Long term evolution (LTE) is based on MIMO-OFDM-IM and continues to be developed by the 3rd Generation Partnership Project (3GPP). LTE specifies downlink rates up to 300 Mbit/s, uplink rates up to 75 Mbit/s, and quality of service parameters such as low latency. LTE Advanced adds support for picocells, femtocells, and multi-carrier channels up to 100 MHz wide. LTE has been embraced by both GSM/UMTS and CDMA operators.

The first LTE services were launched in Oslo and Stockholm by TeliaSonera in 2009. There are currently more than 360 LTE networks in 123 countries operational with approximately 373 million connections (devices).

4.1.4 FUNCTIONS OF MIMO:

Precoding is multi-stream beamforming, in the narrowest definition. In more general terms, it is considered to be all spatial processing that occurs at the transmitter. In (single-stream) beamforming, the same signal is emitted from each of the transmit antennas with appropriate phase and gain weighting such that the signal power is maximized at the receiver input. The benefits of beamforming are to increase the received signal gain by making signals emitted from different antennas add up constructively and to reduce the multipath fading effect. In line-of-sight propagation, beamforming results in a well-defined directional pattern. However, conventional beams are not a good analogy in cellular networks, which are mainly characterized by multipath propagation. When the receiver has multiple antennas, the transmit beamforming cannot simultaneously maximize the signal level at all of the receive antennas, and precoding with multiple streams is often beneficial. Note that precoding requires knowledge of channel state information (CSI) at the transmitter and the receiver.

Spatial multiplexing requires MIMO antenna configuration. In spatial multiplexing, a high-rate signal is split into multiple lower-rate streams and each stream is transmitted from a different transmit antenna in the same frequency channel. If these signals arrive at the receiver antenna array with sufficiently different spatial signatures and the receiver has accurate CSI, it can separate these streams into (almost) parallel channels. Spatial multiplexing is a very powerful technique for increasing channel capacity at higher signal-to-noise ratios (SNR). The maximum number of spatial streams is limited by the lesser the number of antennas at the transmitter or receiver. Spatial multiplexing can be used without CSI at the transmitter but can be combined with precoding if CSI is available. Spatial multiplexing can also be used for simultaneous transmission to multiple receivers, known as space-division multiple access or multi-user MIMO, in which case CSI is required at the transmitter. The scheduling of receivers with different spatial signatures allows good separability.

Diversity coding techniques are used when there is no channel knowledge at the transmitter. In diversity methods, a single stream (unlike multiple streams in spatial multiplexing) is transmitted, but the signal is coded using techniques called space-time coding. The signal is emitted from each of the transmit antennas with full or near orthogonal coding. Diversity coding exploits the independent fading in the multiple antenna links to enhance signal diversity. Because there is no channel knowledge, there is no beamforming or array gain from diversity coding. Diversity coding can be combined with spatial multiplexing when some channel knowledge is available at the transmitter. Electronic beam steering with greater efficiency is a vibrant feature of a phased array antenna, but for all the range cells, it is fixed at a specific angle. To mitigate this problem, a frequency diverse array (FDA) antenna was proposed. This study presents a review on the development of FDA technology in radar and navigation applications. FDA is different from a conventional phased array antenna radar in the sense that it uses a small frequency offset across the array, which helps to generate a range, angle and time-dependent beam pattern. This pattern assures the energy transmission towards the desired angle and range cell. In addition, this study also focuses the research on getting range-angle uncoupled beam patterns along with diverse hybrid cognitive FDA design, available in the literature, for improved radar performance. Radar systems provide an important remote sensing capability, and are crucial to the layered sensing vision; a concept of operation that aims to apply the right number of the right types of sensors, in the right places, at the right times for superior battlespace situational awareness. The layered sensing vision poses a range of technical challenges, including radar, that are yet to be addressed. To address the radar-specific design challenges, the research community responded with waveform diversity; a relatively new field of study which aims to reduce the cost of remote sensing while improving performance. Early work suggests that the frequency diverse array radar may be able to perform several remote sensing missions simultaneously without sacrificing performance. With few techniques available for modelling and characterizing the frequency diverse array, this research aims to specify, validate and characterize a waveform diverse signal model that can be used to model a variety of traditional and contemporary radar configurations, including frequency diverse array radars. To meet the aim of the research, a generalized radar array signal model is specified. A representative hardware system is built to generate the arbitrary radar signals, then the measured and simulated signals are compared to validate the model. Using the generalized model, expressions for the average transmit signal power, angular resolution, and the ambiguity function are also derived. The range, velocity and direction-of-arrival measurement accuracies for a set of signal configurations are evaluated to determine whether the configuration improves fundamental measurement accuracy.

The main importance of the MIMO channel technique is to improve the capacity of the channel and therefore it is important to compare the capacity of a SISO system to a MIMO system. In SISO a system, the Shannon formula in the equation can be applied to determine the capacity of the system. However, for a precise comparison, the MIMO system must be transmitting with power the same as that of a SISO system. Therefore if the power radiated by a SISO system is Ps (SNR), then the power radiated from each antenna of a MIMO system with NTX transmit antennas must be Ps/N.

Hence, for a MIMO system with NTX and MRX antennas using diversity at the transmitter and receiver end, the capacity of the system can be determined by the formula means transpose-conjugate of H and H is the M x N channel matrix. IM indicates the identity matrix of dimension N x M, in this case, M = N =2 or more. However, since signals transmitted over the MIMO channel have to be linearly independent and orthogonal, interference averages to zero. Hence from the equation, it can be seen that if the signal power Ps and the noise level N are the same then the more multiple antennas are used at the receiver, the more power is collected increasing the channel capacity and bandwidth.

4.1.5 REPRESENTATION OF MIMO CHANNEL:

The first channel model to be considered in this project will be a 2 x 2 MIMO system that is a system with 2 transmits (TX) and 2 receive (RX) antennas where different independent data streams are transmitted from multiple antennas to multiple receive antennas. This channel model will be extended to a 3 x 3 MIMO system and even more to illustrate the channel characteristics to the increase in the number of antennas. The signals considered in the MIMO systems of this project are baseband signals ignoring modulation processes and concentrating on the up and down frequency conversion. Therefore the signals on the i-th transmit antenna will be denoted x while the received signal on the j-th receive antenna denoted as y. Figure 2 shows the antenna set-up and the various unknown channel coefficients.

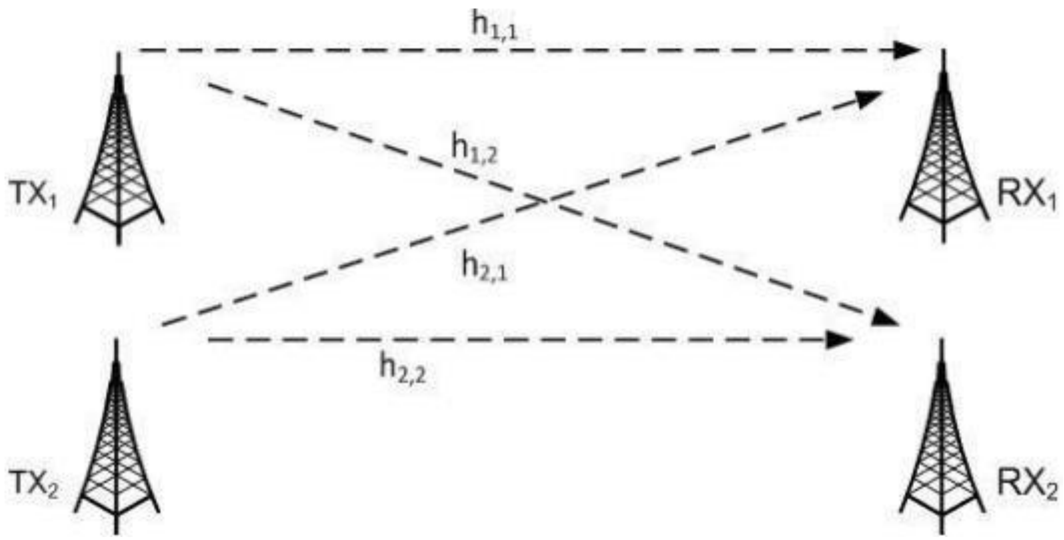


Figure 4.2 : Channel Characteristic of a 2 x 2 MIMO Wireless Communication System

Since the coefficient of the unknown in the channel matrix Wc and the number of transmitted signal X is equal to the number of received signal Y, the equation can be solved if the channel Wc is inversed which in this case a 2 x 2 matrix inversion.

4.1.6 OPERATIONAL PRINCIPLES OF A MIMO SYSTEM:

To derive the channel characteristics, the MIMO system transmits specified and known training signals regularly from all transmitters of the system and these transmitted signals are received at the receiver. Based on the received signals, the receiver calculates the characteristics of all channel paths from each transmitted antenna to each receiving antenna. To prove that MIMO work, the transmitted signal X has to be solved from the group of equations in the equation below. We also assume that the system is noise-free and line of sight (LOS). Reference to figure 3 below, if the transmitted signal is represented to be X and the received signal Y. If the channel characteristics matrix is Wc, we may write Y=X Wc. If the channel matrix has N rows as many as there are transmitting antennas with index i. Then transmitted signal vector is written as X = [x1, x2,…xN] . Also if the channel matrix has M columns, as there are receiving antennas with index j. Then the received signal vector is Y = [y1, y2,…yM]. These vectors are extended later to matrixes by inserting K samples into each column.

4.1.7 CHANNEL ESTIMATION PROCEDURE:

To estimate the channel characteristics, we expand each transmitted and received signals in time and write into signal matrix columns K discrete samples in time. The signals matrixes get K rows and as many columns as we have antennas, N or M. In general the MIMO system involves multiple antennas at the transmitting end and multiple receivers at the receiving end. Figure 4 show a general representation of a 3 x 3 general MIMO system.

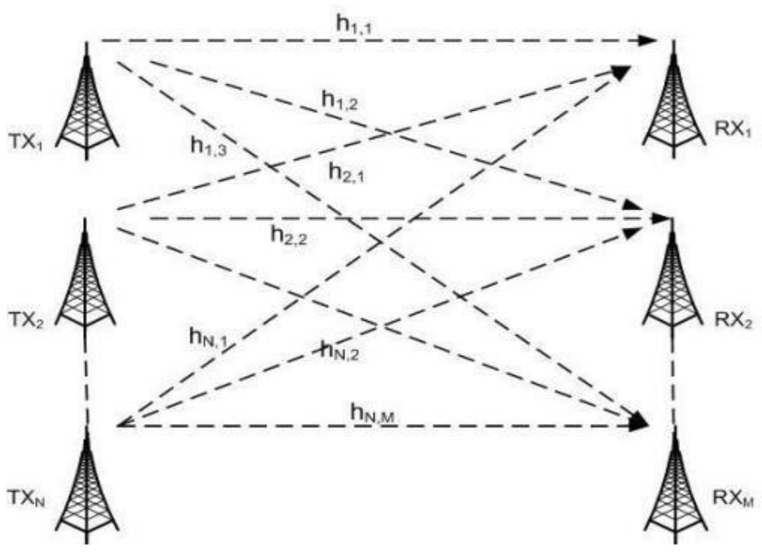


Figure 4.3 : General 3 x 3 MIMO System with Unknown Channel Characteristic

To determine the characteristic of the channel, both the transmitted signal X and the received signal Y have to be known. If the transmitted and received signals are of the form. If the channel transfer matrix Wc can be determined, then it means the transmitted signals can also be determined because the received signal Y is known. We may write the expression between these vector signals as Y = X Wc. To determine the transmitted signals, the channel transfer matrix, Wc -1 have to be inverted and then multiplied with the received signal matrix Y. X = Y Wc -1 The channel transfer matrix that we have to solve with the help of known training signal X and the received signal Y has the form. The channel transfer matrix is calculated periodically with the help of the known training signals and remains constant over information transmission time. It is then recalculated when new information is being transmitted. The channel characteristics in the equation are defined for each signal path at discrete time instants 1,2,...K. However, we need to derive Wc and this can be derived with the help of the transmitted signals X and the corresponding known received signals Y measured at the receiver. The equation expresses the channel matrix Wc which do not need to be a square matrix. Because, for example, if we have a 4x4 MIMO system with 100 samples, we do not need to have a 100x100 channel matrix. However, 5 samples can be transmitted from the four (4) antennas at a time. The next section explains the procedures required to estimate the channel transfer matrix.

4.1.8 CHANNEL IDENTIFICATION ALGORITHM:

To model the transfer channel a common space matrix (orthonormal basis matrix Ux) is first generated and then used to map both the transmitted and received signals that vary in space (multiple antennas) and in time (samples in time). This common orthonormal basis matrix is obtained by decomposing either the transmitted signal X or the received signal Y. Hence, the MIMO channel problem can be solved using a four-step approach under a condition where there is no noise (N = 0) and if the transmitted signals are orthogonal. These steps are summarized as follows:

(1) Finding an orthonormal basis Ux of the transmitted signal matrix X using the Gram-Schmidt procedure.

(2) In the K-dimensional signal vector space spanned by Ux, we express the N column vectors of the transmitted signal X by the projection onto the orthogonal axes of Ux. Wx = Ux X.

(3) In the K-dimensional signal vector space spanned by Ux, we express the M column vectors of the received signal matrix Y by their projection onto the orthogonal axes of Ux. Wy = Ux Y (4.2.2)

(4) Calculate the inverse of the pseudo inverse of the Fourier coefficients of the transmitted signal Wx and find an estimate of channel transfer matrix Oc = Wx -1 Wy or Oc = Wx + Wy.

4.1.9 ORTHONORMAL BASIS UX ESTIMATION:

To be able to estimate the transfer channel matrix Oc, a common space matrix is required to map both the transmitted signal X and the received signal Y together. The orthonormal base matrix Ux serves as the common space matrix needed to map the transmitted signal X to the received signal Y. It is important to know that, the orthonormal basis needed for the mapping can be derived either using the transmitted signal X or the received signal Y. Hence, the orthonormal basis Ux of the transmitted signal X is calculated by taking the matrix Ux obtained from the decomposition of transmitted signal X. In linear algebra, a matrix such as the transmitted signal X can be decomposition into the product X= Ux R where Ux is an orthogonal matrix in this case the orthonormal basis and R an upper triangular matrix. It should be noted that the format of matrix X is K-by-N while the size of the orthonormal base matrix Ux is always K-by-K. The Gram-Schmidt procedure is one way to decompose a column rank matrix and this procedure will be used in this project.

There are also other methods of decomposing a matrix such as QR-decomposition, LU decomposition, Cholesky decomposition, etc. See appendix 1 for the Gram-Schmidt process. Fourier coefficients of transmitted signal Wx After determining the orthonormal basis Ux, we have to map the known transmitted signal X to the orthonormal space matrix Ux and this is done by calculating the generalized Fourier coefficients of the transmitted signal Wx. The mapping of the transmitted signal X for the orthonormal basis Ux is expressed Wx = Ux X. The generalized Fourier coefficients are coefficients of any orthogonal set of functions over which signals are split up. Therefore the generalized Fourier coefficients of the transmitted signal Wx tell us how much each column (signal) of the transmitted signal X contains each orthogonal column component in the orthonormal base matrix Ux. In this case, we are splitting the transmitted signal with the help of the orthonormal basis Ux. It should be noted that the multiplication of Ux Ux T = I where I is a K-by-K identity matrix and the size of Wx is K-by-N.

Fourier coefficients of training (received) signal Wy The received signal Y is also mapped to the orthonormal space matrix Ux by multiplying the received signal Y with the orthonormal base matrix Ux. This means that we have to split the received signal Y with the help of the orthonormal basis Ux. Hence, the calculation of the Fourier coefficients of the received signal as Wy = Ux Y Where the size of Wy is K-by-M. Transfer channel matrix estimation Oc Finally, we derive the most favourable estimate of the channel transfer matrix from the expression Oc = Wx -1 Wy.

If the Fourier coefficient of the transmitted signal Wx is not a square matrix then we use the expression Oc = Wx + Wy where Wx + the pseudo inverse of Wx. This follows from the formula Y = X Oc Oc = X-1 Y =Wx -1 Ux Y =Wx -1 Ux Ux -1Wy = Wx -1 Wy From equation again it can be seen that to estimate the channel transfer matrix Oc, the inverse of the generalized Fourier coefficients of the transmitted signal Wx is needed since it is a square matrix. However, in the case where Wx is not a square matrix, the pseudo inverse of Wx is calculated. The orthonormal basis Ux plays an important role in determining the channel characteristics and therefore it is important to develop an algorithm to generate Ux. Not only to generate the orthonormal basis matrix but to illustrate the entire matrixes required to estimate the transfer channel. Hence, Microsoft Excel is used to develop an algorithm for the Gram-Schmidt procedure to generate Ux, the Fourier coefficients of both transmitted and received signals as well as the transfer channel. This algorithm can be accessed from a compact disk (CD) attached to this project. The next chapter explains with an example how to determine the orthonormal basis Ux using the Gram-Schmidt procedure by decomposing the transmitted signal X and applying this procedure to two different scenarios. First the calculation of the orthonormal basis Ux in the case where the transmitted signal X is a square matrix and in the second case where the transmitted signals X is not a square matrix.

4.1.10 ORTHONORMAL SPACE CONCEPT:

An orthonormal basis is a coordination system where we can present as many dimensions as is the maximum number of antennas of the transmitted and received signals. The main reason why we need the orthonormal base is to have a common coordinated system to combine the transmitted signal X and the received signal Y. The number of dimension of the orthonormal base matrix depends on the maximum number of antennas at both the transmitter and the receiver end of the MIMO system. It does not matter what kind of signal (X or Y) used in generating the orthonormal space but it is most convenient to use the transmitted training signal X which is defined to contain linearly independent column signals. Suppose K = 3-time samples per transmitted signal X and per received signal Y respectively of a 3 x 3 channel transmission system were observed. The figure shows a Line of Sight (LOS) MIMO system made up of three transmitters and three receivers. In figure 6, N = 3 transmitted signals and M = 3 received signals, the format of the unknown channel transfer matrix Oc would be 3 x 3. Assume that all signals are real-valued and the transmitted signals and received signals are observed as signal matrices. Then we need the orthonormal basis matrix Ux to solve for an estimated channel transfer matrix Oc. It is easier to calculate the orthonormal base matrix Ux of a square matrix; therefore in the next section, we estimate the channel model by calculating first the orthonormal basis matrix Ux of a simple 2 x 2 MIMO system and then proceed to a more complex 3 x 3 MIMO system.

This section explains how to calculate the orthonormal basis Ux of a 2 x 2 MIMO system with the help of the transmitted signal X. To demonstrate the steps in subchapter, we will have a numerical example to explain how to generate the orthonormal basis Ux of a full rank square matrix using linearly independent transmitted signals in X. The Gram-Schmidt procedure will be used in the decomposition of the transmitted signal X. See appendix 1 for the Gram-Schmidt procedure. The equation to decomposed in our calculation is of the form X = Ux R; R = Ux -1 X. In the equation, the transmitted signal X is divided into two components, Ux the orthonormal basis and R the upper triangular matrixes. The upper triangular matrix R is calculated by first finding the inverse of the orthonormal basis matrix Ux and multiplying it with the transmitted signal X. However, the upper triangular matrix R will is not needed in our analysis. The next very simple example explains how the Gram Schmidt procedure can be applied to generate orthonormal base Ux of the transmitted signal X.

The transmitted known training signals are used in modelling the transfer channel Oc as shown in examples. Therefore it is important to estimate the channel output if the transmitted signals are transmitted. The next example drives the received signal U in two different scenarios. In both scenarios, the transmitted signals are the same but different transfer channels. This will help us estimate the received signals produced by the different transfer channels.

To further understand the operational principles of the MIMO channel, we extend the same principles listed in subchapter 4.2 to a case of a 3 x 3 MIMO system where there are three antennas at both transmit and receive ends. Assuming three-column transmitted signals and three-column received signals each are observe over K uniformly spaced discrete time instances The channel can be estimated by first calculating the common signal space matrix (orthonormal base matrix). In the next example, the orthonormal base of the transmitted signal X is calculated using the Gram-Schmidt procedure

Based on the Gram Schmidt process, the orthonormal basis Ux matrix is an orthogonal matrix. The column signals of the orthonormal base Ux in the equation are orthogonal and the inner product of any pair of the column vectors result in zero. The column vectors are orthonormal and the norm of every column vector signal result in value 1. To prove that the column signals of the orthonormal base Ux are orthogonal, the inner product of these column signals. The results obtained from the inner product calculations prove that the column signals of the orthonormal base Ux are orthogonal. It should be noted that the results obtained in example (5.3.2) are not exactly zero due to the rounding of values in calculating the orthonormal basis matrix.

4.2 OFDM

OFDM has several significant benefits over single-carrier QAM modulation which has lead to its adoption in many of the modern wireless standards, including ADSL, European Digital Video Broadcast, IEEE 802.11a/g/n (WiFi), WiMax, and 3GPP Long Term Evolution (LTE). First of all, OFDM enables easy equalization, solving many of the equalization complexity issues we encountered for single-carrier QAM by working instead in the frequency domain. Second of all, the ability to control the size of the constellation on each subcarrier in OFDM also allows it to mimic the “water-filling” construction which maximizes the amount of information that can be reliably transmitted over the channel: subcarriers (frequencies) with better channel gains can utilize higher-order modulations and coding rates. A final significant benefit that we will encounter later in the course is that OFDM allows for an especially easy to implement and understand multiple access strategy OFDMA, in which different subcarriers (frequencies) are allocated to different users at different times. To understand how OFDM modems work, we must first revisit several properties of the discrete Fourier Transform and its efficient implementation of the Fast Fourier Transform.

The circular convolution between two length N signals x1[n] and x2[n], n ∈ {0, 1, . . . , N − 1}, is defined as x1[n] * x2[n] = N X−1 =0 x1*x2[((n −1))N] where ((n))N = N − n n < 0 n n ≥ 0 Circularly convolving two length N signals x1[n] and x2[n] is the same thing as multiplying their discrete Fourier transforms x1[n]* x2[n] DF T ←→ X1[k]X2[k] .The underlying principle of OFDM is based on this property of the DFT

The architecture of an OFDM transmitter and receiver pair is depicted in Fig. 1. First, during the th block, N QAM symbols c[k], k ∈ {0, . . . , N − 1} are mapped to "sub-carriers", one per subcarrier, which are the inputs to a size N IFFT, which computes the DFT shown below efficiently q[n] = N X−1 k=0 c[k]e j 2π N kn, n ∈ {0, . . . , N}

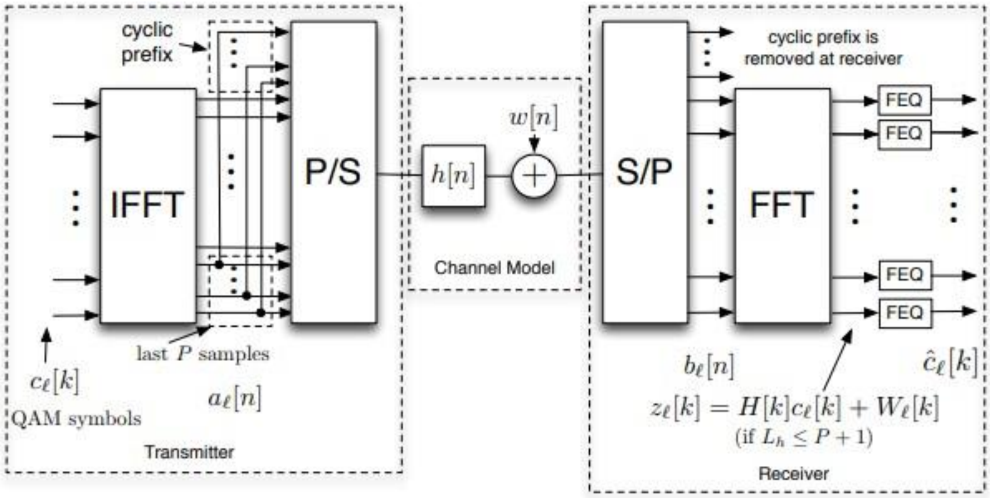


Figure 4.4 : An OFDM Transmitter, Channel model and Receiver

These OFDM blocks are then serialized to form xc[n] = a bn N+P c [n mod (N + P)] which is then put through a digital to analogue converter to form the complex baseband signal to be modulated by an ordinary quadrature modulator. As discussed in the previous notes on equalization, on its way from the transmitter to the receiver, the signal is subject to a multipath channel, as well as potentially some additive interference, and at the receiver, there is additive white Gaussian noise in the electronics. As discussed in the lecture on quadrature modulation, we can model the effects of this multipath channel by modelling the signal at the receiver, after demodulation and sampling as yc[n] = L Xh−1=0 h + w[n] where Δ indicates the delay before the channel is perceived as having any significant energy, and h[n] is the sampled impulse response of the channel, whose delay spread is Lh samples long, and w[n] is the additive white Gaussian noise. The receiver then proceeds to process this signal yc[n] in the exact opposite manner as at the transmitter. First the blocks are collected b[n] = yc[n + Δ + (N + P)], n ∈ {0, 1 . . . , N + P − 1} (8) then the cyclic prefix is removed r[n] = b[n + P], n ∈ {0, 1, . . . , N − 1} . The Fast Fourier transform is then taken, which calculates the DFT shown below efficiently z[k] = X N n=0 r[n]e −j 2π N kn.

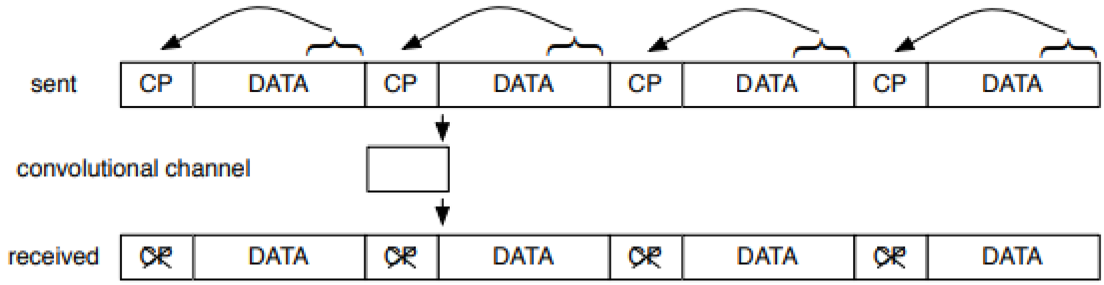


Figure 4.5 : Convolutional Channel

The Purpose of the Cyclic Prefix: Removing Inter-Block & Inter-Carrier Interference. The cyclic prefix allows an OFDM transmitter-receiver pair to accomplish important objectives, provided that the length of the multipath channel is less than or equal to the length of the cyclic prefix plus one:

1) it eliminates interblock interference, and

2) it enables easy equalization, as for the remainder of the block it makes the linear convolution with the ISI channel into a circular convolution. Indeed, as is indicated in Fig. 2, if we consider the output of the channel at times $n \in (\{0, \ldots, P-1\} + k(N+P)$, this will contain a contribution from both the previous OFDM symbol and the next one. However, if the channel length $Lh \leq P + 1$ then these are the only samples containing the inter-block interference, and hence, once we drop these samples at the receiver (when we remove the cyclic prefix), we will have no interblock interference. Additionally, under the same criterion $Lh \leq P + 1$ the remaining samples in each block after dropping the cyclic prefix will be the circular convolution of the channel with the associated transmitted block. Hence, by the circular convolution property of the DFT, when the FFT is taken of these received blocks, they will be the product of the FFT of the channel H[k] and the original QAM symbols, plus the FFT of the additive noise, as depicted in Fig 1. $z[k] = H[k]c[k] + w[k]$. The estimated QAM symbols can then be reconstructed by processing the subcarriers k individually by multiplying them with a frequency domain equalizer F[k] $c[k] = F[k]z[k]$ (12) where the F[k] can select as $H[k] * \ |H[k]|\ 2{+}\sigma2$, which minimizes the average magnitude squared error between c[k] and c[k], where σ 2 is the variance w[k]. These EQs F[k] can be estimated by placing training information, also called pilot signals or reference signals, agreed upon in advance between the transmitter and receiver, on various subcarriers k which are spread out in frequency. Because the channel is shorter than the cyclic prefix, and the cyclic prefix is

typically far shorter than the FFT size, the channel coefficients H[k] vary slowly with frequency k, and hence the channel estimates from the pilot signals can be interpolated to yield improved channel estimates for other carriers Considerations when Selecting a Cyclic Prefix Length and FFT Size There are two important considerations when selecting a cyclic prefix length and FFT size for an OFDM system in practice. We already noted that the cyclic prefix needs to be longer than one less than the length of the channel (its delay spread). As the cyclic prefix does not bear any extra information itself, it is desirable to select a very large FFT size so that the redundancy is introduced is low. However, the multipath channel must remain fixed over the entire duration of the OFDM symbol (i.e. the entire cyclic prefix and received data block), and this places an upper limit on the length of the FFT based on how quickly the channel is varying.

Bit-Loading & OFDMA Because the OFDM construction turns the wideband multipath channel into multiple parallel lower rate channels associated with evenly spaced frequencies within the band, it enables several intuitive other technologies. As mentioned in the introduction, those carriers k with a higher signal to noise ratios |H[k]| 2 σ2 can be given higher-order QAM symbols and higher code rates. This process is called bit-loading and adaptive modulation and coding in different contexts. Additionally, using OFDM modems also enable an especially intuitive form of multiple access. If the transmitter wishes to send individual messages to multiple receivers (as in a cellular system) listening to the same wideband channel, one can place information to be transmitted to different users on different subcarriers. Furthermore, since the multipath profiles H[k] of different receivers listening to the same transmitter will be different, if these multipath profiles are reliably fed back to the transmitter, it can also select those subcarriers for different users which they are receiving with high signal to noise ratios. On the other hand, if the multipath profiles are not fed back, or are varying too quickly, the transmitter can diversify which frequencies are used for different users by spreading the carriers/blocks of carriers given to each particular user out in frequency.

Finally, we note that you are likely to encounter at least two other variants of OFDM: discrete multitone (DMT) and SC-FDMA, both of which differ from pure OFDM solely by what is modulated on each of the subcarrier k (bins of the IFFT at the transmitter). Discrete multitone, which is used in ADSL systems, simply select the blocks of QAM symbols to be conjugate symmetric, so that their IFFT is real-valued, and hence only a single analog to digital converter is required at the receiver. An SC-FDMA transmitter selects a contiguous sub-block of subcarriers and places the result of the FFT of a sequence of QAM symbols of these subcarriers. This “spreading” of the inputs to the subcarriers with an FFT enables the peak to average power ratio of the output of the IFFT to be reduced, which allows the battery-operated power amplifiers in the analog RF front ends of the transmitters in mobiles to operate in more efficient regimes. SC-FDMA is used in the uplink of LTE (from mobile phones to the base stations), while OFDMA is used in the downlink (from base stations to mobile phones). Note that the overall effect of this operation is similar to pulse-shaping a single carrier QAM signal, since the zeros on the other subcarriers effectively determine a pulse shape, and which subcarriers are selected determine an effective overall carrier frequency. This is the reason this type of modulation is called “single-carrier FDMA” in the LTE standard.

Orthogonal frequency-division multiplexing (OFDM) is a method of digital signal modulation in which a single data stream is split across several separate narrowband channels at different frequencies to reduce interference and crosstalk. The original data stream bits -- that in a conventional single-channel modulation scheme would be sent serially (one after the other) -- are transmitted in parallel (several at once on separate channels) but at lower speed in each substream (a stream within another stream) relative to the original signal. This means symbols sent in the substreams are longer and spaced farther apart. In the original stream, each bit might be represented by a 1-nanosecond (ns) segment of the signal, with 0.25 ns spacing between bits. Splitting the signal across four component streams lets each bit be represented by 4 ns of the signal with 1ns spacing between. This reduces interference among symbols and makes it easier to receive each symbol accurately while maintaining the same throughput (in this example, 4 bits every 5 ns).

OFDM technology was first conceived in the 1960s and 1970s during research into minimizing interference among channels near each other in frequency and to achieve clean data transmission in situations prone to interference and signal corruption when more conventional modulation schemes are used.

OFDM is used in Wi-Fi, DSL internet access, 4Gwireless communications, and digital television and radio broadcast services

Orthogonal Frequency Division Multiplexing (OFDM) is a special form of multi-carrier modulation, patented in 1970. It is particularly suited for transmission over a dispersive channel. (See further discussion of MCM over the wireless channel.)

In a multipath channel, most conventional modulation techniques are sensitive to intersymbol interference unless the channel symbol rate is small compared to the delay spread of the channel. OFDM is significantly less sensitive to intersymbol interference because a special set of signals is used to build the composite transmitted signal. The basic idea is that each bit occupies a frequency-time window which ensures little or no distortion of the waveform. In practice, it means that bits are transmitted in parallel over several frequency-nonselective channels. Applications of OFDM are found in

- Digital Audio Broadcasting (DAB) and
- Digital Video Broadcasting over the terrestrial network: Digital Terrestrial Television Broadcasting (DTTB). In the DTTB OFDM transmission standard, about 2,000 to 8,000 subcarriers are used.
- UMTS. The UMTS Forum is selecting an appropriate radio solution for the third generation mobile standard, as a successor to GSM. OFDM is one of the five competing proposals.
- Wireless LANs. OFDM is used in HIPERLAN Phase II, which supports 20 Mbit/s in propagation environments with delay spreads up to 1 second.

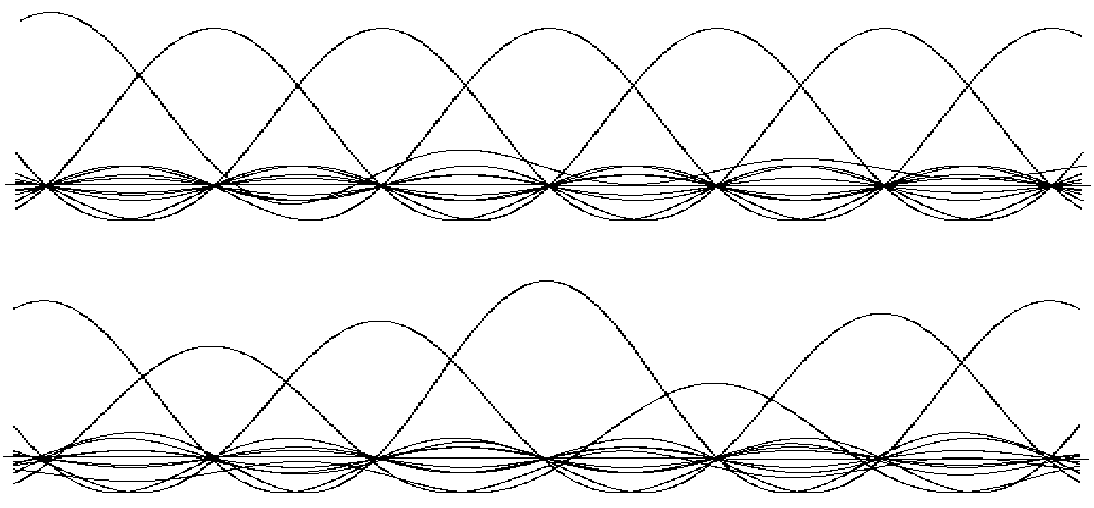


Figure 4.6 : Signal spectrum of an OFDM signal

which consists of the spectra of many bits, in parallel. Rectangular pulses in the time domain produce sinc-functions in the frequency domain.

- above: signal spectrum as transmitted.
- Below: as received in over a dispersive, time-invariant channel.

The effect of multipath scattering on OFDM differs from what happens to other forms of modulation. A qualitative description and mathematical description of OFDM is presented by Dusan Matic. Jean-Paul Linnartz reviews the effects of a Doppler spread and the associated rapid channel variations. Dusan Matic also studied the system design aspects of OFDM at mm-wavelengths.

4.2.1 CODED OFDM

Multi-Carrier Modulation on its own is not the solution to the problems of communication over unreliable multipath channels. The channel time dispersion will excessively attenuate some subcarriers such that the throughput on these sub-channels would be unacceptably small. Only if the joint signal of many subcarriers is processed appropriately, the diversity advantages of MCM can be exploited. The need for coding across subcarriers was addressed by Sari et al. warning against the overly enthusiastic pursuit of MCM. The advantages of frequency-domain implementations of equalizers (using an FFT) should not be mistaken for an "inherent" diversity gain of OFDM, which may not exist.

In an OFDM transmitter, blocks of k incoming bits are encoded into n channel bits. Before transmission, an n-point Inverse-FFT operation is performed. When the signals at the I-FFT output are transmitted sequentially, each of the n channel bits appears at a different (subcarrier) frequency. Such coding across subcarriers is necessary. If one subcarrier experiences deep fading, this leads to the erasure of the bit on this subcarrier.

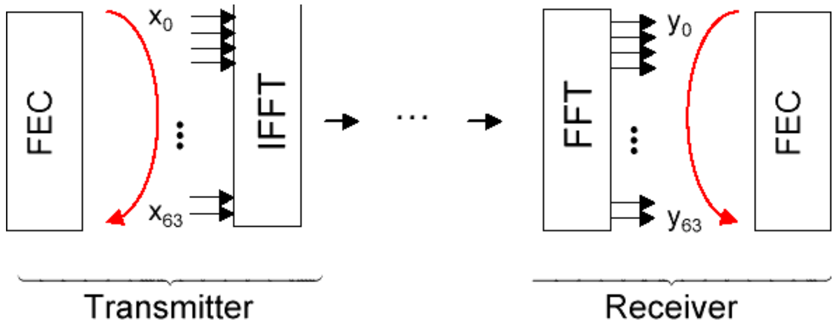


Figure 4.7 : OFDM Transmitter and Receiver

But of course, coding across subcarriers is not the only mechanism that can be invoked to combat dispersion or to exploit diversity. Other possibilities are

- Interleaving in frequency or time domain with coding in the other domain,
- The use of different signal constellations at different frequencies, i.e., adapting the subcarrier bit rate to the channel state,
- Signal to spread over various subcarriers, e.g., according to a linear matrix operation, as is proposed in Orthogonal Multi-Carrier Code Division Multiplexing.

If in a point-to-point MCM link, the receiver and the transmitter can cooperate by adaptively distributing their power budget over the individual subcarriers. For instance, the signal-to-noise ratios selected according to Gallager's water-pouring theorem can (under certain conditions) be proved to be optimum. Efficient loading of the various subcarriers can for instance significantly enhance the performance of MCM over twisted-pair telephone subscriber loops with crosstalk from other nearby copper pairs.

4.2.2 IMPLEMENTATIONAL ASPECTS

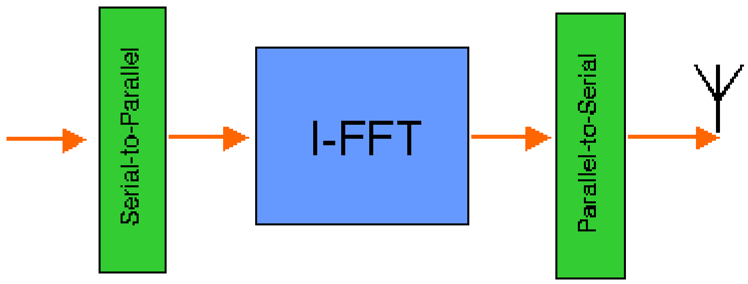


Figure 4.8 : OFDM transmitter using an (inverse) Fast Fourier Transform (FFT)

- OFDM is not a constant-envelope modulation method. Therefore transmit power amplifiers must be **highly linear**.

- OFDM receiver performance is very sensitive to phase noise.
- Synchronization to an OFDM signal also requires frame synchronization, to support an FFT operation at the receiver.

Single Frequency Networks:

OFDM allows very efficient frequency reuse. Transmitters broadcasting the same program can use the same frequency in a Single Frequency Network.

Code Division Multiple Access:

OFDM can be combined with CDMA transmission, e.g. in Multi-Carrier CDMA.
Channel Modeling:

Many channel simulation models follow the narrowband model. Wideband channels are often simulated by extending the model assuming multiple time-delayed resolvable paths. This allows the simulation of the channel impulse response, including its stochastic behaviour. To determine the performance of a multicarrier, OFDM or MC-CDMA system, another approach can be to model a set of fading subchannels. Considering a single subcarrier, the channel may be modelled as a narrowband fading channel, for instance with Rician or Rayleigh amplitude distributions. The collection of multiple subcarriers can be modelled as a set of mutually dependent fading channels. In such a model, it is important to address the correlation of the fading of various subchannels using the models of delay spread and coherence bandwidth. See a discussion of such a model. Also: read about the discrete-frequency model for OFDM with Delay spread and Doppler.

OFDM, Orthogonal Frequency Division Multiplexing is a form of signal waveform or modulation that provides some significant advantages for data links.

Accordingly, OFDM, Orthogonal Frequency Division Multiplexing is used for many of the latest wide bandwidth and high data rate wireless systems including Wi-Fi, cellular telecommunications and many more.

The fact that OFDM uses a large number of carriers, each carrying low bit rate data, means that it is very resilient to selective fading, interference, and multipath effects, as well as providing a high degree of spectral efficiency.

Early systems using OFDM found the processing required for the signal format was relatively high, but with advances in technology, OFDM presents few problems in terms of the processing required.

4.2.3 WHAT IS OFDM?

OFDM is a form of multicarrier modulation. An OFDM signal consists of several closely spaced modulated carriers. When modulation of any form - voice, data, etc. is applied to a carrier, then sidebands spread out either side. A receiver must be able to receive the whole signal to be able to successfully demodulate the data. As a result, when signals are transmitted close to one another they must be spaced so that the receiver can separate them using a filter and there must be a guard band between them. This is not the case with OFDM. Although the sidebands from each carrier overlap, they can still be received without the interference that might be expected because they are orthogonal to each other. This is achieved by having the carrier spacing equal to the reciprocal of the symbol period.

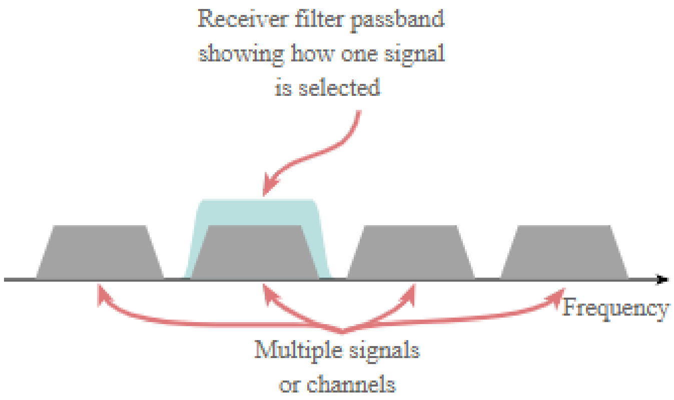


Figure 4.9 : Traditional Selection if signals on different channels

To see how OFDM works, it is necessary to look at the receiver. This acts as a bank of demodulators, translating each carrier down to DC. The resulting signal is integrated over the symbol period to regenerate the data from that carrier. The same demodulator also demodulates the other carriers. As the carrier spacing equal to the reciprocal of the symbol period means that they will have a whole number of cycles in the symbol period and their contribution will sum to zero - in other words, there is no interference contribution.

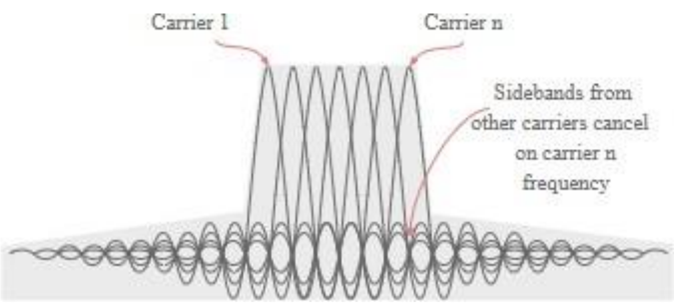


Figure 4.10 : Basic Concept of OFDM

One requirement of the OFDM transmitting and receiving systems is that they must be linear. Any non-linearity will cause interference between the carriers as a result of inter-modulation distortion. This will introduce unwanted signals that would cause interference and impair the orthogonality of the transmission. In terms of the equipment to be used the high peak to average ratio of multi-carrier systems such as OFDM requires the RF final amplifier on the output of the transmitter to be able to handle the peaks whilst the average power is much lower and this leads to inefficiency. In some systems the peaks are limited. Although this introduces distortion that results in a higher level of data errors, the system can rely on the error correction to remove them.

4.2.4 DATA ON OFDM

The data to be transmitted on an OFDM signal is spread across the carriers of the signal, each carrier taking part of the payload. This reduces the data rate taken by each carrier. The lower data rate has the advantage that interference from reflections is much less critical. This is achieved by adding a guard band time or guard interval into the system. This ensures that the data is only sampled when the signal is stable and no new delayed signals arrive that would alter the timing and phase of the signal.

4.2.5 GUARD INTERVAL ON OFDM SIGNALS

The distribution of the data across a large number of carriers in the OFDM signal has some further advantages. Nulls caused by multi-path effects or interference on a given frequency only affect a small number of the carriers, the remaining ones being received correctly. By using error-coding techniques, which does mean adding further data to the transmitted signal, it enables many or all of the corrupted data to be reconstructed within the receiver. This can be done because the error correction code is transmitted in a different part of the signal.

4.2.6 KEY FEATURES OF OFDM

The OFDM scheme differs from traditional FDM in the following interrelated ways:

- Multiple carriers (called subcarriers) carry the information stream
- The subcarriers are orthogonal to each other.
- A guard interval is added to each symbol to minimize the channel delay spread and intersymbol interference.

4.2.7 OFDM ADVANTAGES & DISADVANTAGES

4.2.7.1 OFDM ADVANTAGES

OFDM has been used in many high data rate wireless systems because of the many advantages it provides.

- **Immunity to selective fading**: One of the main advantages of OFDM is that is more resistant to frequency selective fading than single carrier systems because it divides the overall channel into multiple narrowband signals that are affected individually as flat fading sub-channels.
- **Resilience to interference**: Interference appearing on a channel may be bandwidth limited and in this way will not affect all the sub-channels. This means that not all the data is lost.
- **Spectrum efficiency:** Using close-spaced overlapping sub-carriers, a significant OFDM advantage is that it makes efficient use of the available spectrum.
- **Resilient to ISI:** Another advantage of OFDM is that it is very resilient to inter-symbol and inter-frame interference. This results from the low data rate on each of the sub-channels.
- **Resilient to narrow-band effects:** Using adequate channel coding and interleaving it is possible to recover symbols lost due to the frequency selectivity of the channel and narrowband interference. Not all the data is lost.
- **Simpler channel equalisation:** One of the issues with CDMA systems was the complexity of the channel equalisation which had to be applied across the whole channel. An advantage of OFDM is that using multiple sub-channels, the channel equalization becomes much simpler.

4.2.7.2 OFDM DISADVANTAGES

Whilst OFDM has been widely used, there are still a few disadvantages to its use that need to be addressed when considering its use.

- **High peak to average power ratio**: An OFDM signal has a noise like amplitude variation and has a relatively high large dynamic range, or peak to average power ratio. This impacts the RF amplifier efficiency as the amplifiers need to be linear and accommodate the large amplitude variations and these factors mean the amplifier cannot operate with a high-efficiency level.
- **Sensitive to carrier offset and drift**: Another disadvantage of OFDM is that is sensitive to carrier frequency offset and drift. Single carrier systems are less sensitive.

OFDM, orthogonal frequency division multiplexing has gained a significant presence in the wireless marketplace.

The combination of high data capacity, high spectral efficiency, and its resilience to interference as a result of multi-path effects means that it is ideal for the high data applications that have become a major factor in today's communications scene. The use of cyclic prefix is a key element of enabling the OFDM signal to operate reliably.

4.2.8 WHAT IS A CYCLIC PREFIX?

The basic concept behind the OFDM cyclic prefix is quite straightforward. The cyclic prefix performs two main functions.

- The cyclic prefix provides a guard interval to eliminate intersymbol interference from the previous symbol.

It repeats the end of the symbol so the linear convolution of a frequency-selective multipath channel can be modelled as circular convolution, which in turn may transform to the frequency domain via a discrete Fourier transform. This approach accommodates simple frequency-domain processing, such as channel estimation and equalization.

Different OFDM cyclic prefix lengths are available in various systems. For example within LTE a normal length and an extended length are available and after Release 8 a third extended length is also included, although not normally used.

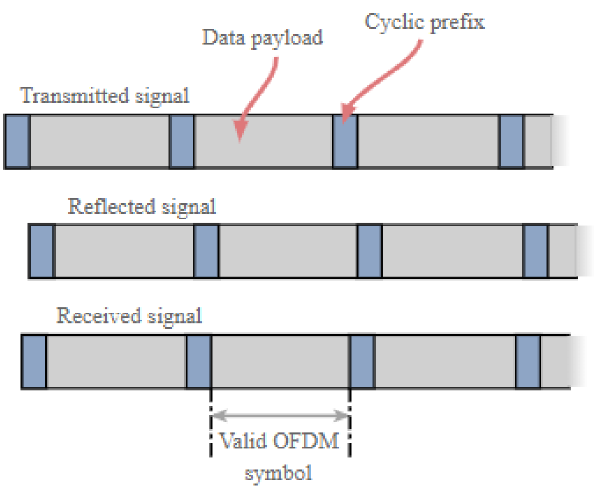


Figure 4.11 : OFDM Cyclic Prefix

4.2.9 CYCLIC PREFIX ADVANTAGES AND DISADVANTAGES

There are several advantages and disadvantages attached to the user for the cyclic prefix within OFDM.

4.2.9.1 ADVANTAGES

- **Provides robustness**: The addition of the cyclic prefix adds robustness to the OFDM signal. The data that is retransmitted can be used if required.
- **Reduces inter-symbol interference**: The guard interval introduced by the cyclic prefix enables the effects of inter-symbol interference to be reduced.

4.2.9.2 DISADVANTAGES

- **Reduces data capacity**: As the cyclic prefix re-transmits data that is already being transmitted, it takes up system capacity and reduces the overall data rate.

The use of a cyclic prefix is standard within OFDM and it enables the performance to be maintained even under conditions when levels of reflections and multipath propagation are high. One of the key requirements for optimum demodulation of OFDM signals is that there is the accurate frequency and timing synchronisation. Fortunately, good OFDM timing and frequency synchronisation are relatively easy to obtain as accurate signals are easy to derive.

Poor timing and frequency synchronisation result in significant levels of degradation to the signal, and as a result, this aspect of the signal chain is key to optimum performance.

4.2.10 OFDM SYNCHRONIZATION BASICS:

OFDM offers many advantages in terms of resilience to fading, reflections and the like. OFDM also offers a high level of spectrum efficiency. However to reap the rewards, it is necessary that the OFDM system operates correctly, and to achieve this, the OFDM synchronization must be effective.

There are many areas in which OFDM synchronisation is critical to the operation of the system:

- OFDM synchronization in terms of frequency offset: The frequencies must be accurately tracked to ensure that orthogonality is maintained.
- OFDM synchronisation in terms of clock accuracy: It is necessary that the sampling occurs at the correct time interval to ensure that the samples are synchronized and data errors are minimised. To ensure that the OFDM system works to its optimum, it is necessary to

ensure that there are schemes in place to ensure the OFDM synchronization is within the required limits.

4.2.11 FREQUENCY OFFSET OFDM SYNCHRONIZATION:

The demodulator in an OFDM receiver must be able to synchronize accurately with the carriers within the OFDM signal. Offsets may arise for many reasons including any frequency errors between the transmitter and the receiver and also as a result of Doppler shifts if there is movement between the transmitter and receiver.

If the frequency synchronisation is impaired, then the orthogonality of the carriers is reduced within the demodulation process and error rates increase. Accordingly, it is essential to maintain orthogonality to reduce errors and maintain the performance of the link.

First look at the way that sampling should occur. With the demodulator in synchronisation, all the contributions from the other carriers sum to zero as shown. In this way all the carriers are orthogonal and the error rate is at its minimum.

OFDM demodulation showing how the interference from other carriers is minimised at the centre of the carrier frequency. If a situation is encountered where the OFDM synchronisation for the frequency aspects are poor, then the demodulator will centre its samples away from the peak of the signal, and also at a point where the contributions from the other signals do not sum to zero. This will lead to a degradation of the signal which could, in turn, lead to an increase in the number of bit errors. OFDM demodulation where interference is high due to poor frequency synchronisation.

4.2.12 CLOCK OFFSET OFDM SYNCHRONIZATION

It is also necessary to maintain OFDM synchronization in terms of the clock. Gain if the clock synchronisation is not accurate, sampling will be offset and again orthogonality will be reduced, and data errors will increase.

When looking at OFDM synchronization concerning the clock offset, the carrier spacing used within the receiver for sampling the received signal will be based upon the internal clock rate. If this differs from that used within the transmitter, it will be found that even if the first carrier within the multiplex is correct, then there will be a growing discrepancy with each carrier away from the first one. Even small levels of discrepancy will cause the error rate to increase.

OFDM demodulation with clock synchronization misalignment: When using OFDM it is necessary to ensure that the synchronisation for both timing and frequency is accurate. By ensuring accurate synchronisation, it is possible to perform the optimum demodulation of the signal. Any misalignment causes the receiver to start to pick up the unwanted interference signals. Fortunately, it is relatively easy to obtain accurate synchronisation signals as these are available from the network, and short term synchronisation can be generated internally.

In telecommunications, orthogonal frequency-division multiplexing (OFDM) is a method of encoding digital data on multiple carrier frequencies. OFDM has developed into a popular scheme for wideband digital communication, used in applications such as digital television and audio broadcasting, DSL internet access, wireless networks, power line networks, and 4G mobile communications.

In coded orthogonal frequency-division multiplexing (COFDM), forward error correction (convolutional coding) and time/frequency interleaving are applied to the signal being transmitted. This is done to overcome errors in mobile communication channels affected by multipath propagation and Doppler effects. COFDM was introduced by Alard in 1986 [1] [2][3] for Digital Audio Broadcasting for Eureka Project 147. In practice, OFDM has become used in combination with such coding and interleaving, so that the terms COFDM and OFDM co-apply to common applications.[4]

OFDM is a frequency-division multiplexing (FDM) scheme used as a digital multi-carrier modulation method. OFDM was introduced by Chang of Bell Labs in 1966.[5][6][7] Numerous closely spaced orthogonal sub-carrier signals with overlapping spectra are emitted to carry data.[8] Demodulation is based on Fast Fourier Transform algorithms. OFDM was improved by Weinstein and Ebert in 1971 with the introduction of a guard interval, providing better orthogonality in transmission channels affected by multipath propagation.[9] Each sub-carrier (signal) is modulated with a conventional modulation scheme (such as quadrature amplitude modulation or phase-shift keying) at a low symbol rate. This maintains total data rates similar to conventional single-carrier modulation schemes in the same bandwidth.

The main advantage of OFDM over single-carrier schemes is its ability to cope with severe channel conditions (for example, attenuation of high frequencies in a long copper wire, narrowband interference and frequency-selective fading due to multipath) without complex equalization filters. Channel equalization is simplified because OFDM may be viewed as using many slowly modulated narrowband signals rather than one rapidly modulated wideband signal. The low symbol rate makes the use of a guard interval between symbols affordable, making it possible to eliminate intersymbol interference (ISI) and use echoes and time-spreading (in analog television visible as ghosting and blurring, respectively) to achieve a diversity gain, i.e. a signal-to-noise ratio improvement. This mechanism also facilitates the design of single frequency networks (SFNs) where several adjacent transmitters send the same signal simultaneously at the same frequency, as the signals from multiple distant transmitters may be re-combined constructively, sparing interference of a traditional single-carrier system.

4.2.13 ORTHOGONALITY

Conceptually, OFDM is specialized frequency-division multiplexing (FDM) method, with the additional constraint that all subcarrier signals within a communication channel are orthogonal to one another.

In OFDM, the sub-carrier frequencies are chosen so that the sub-carriers are orthogonal to each other, meaning that cross-talk between the sub-channels is eliminated and inter-carrier guard bands are not required. This greatly simplifies the design of both the transmitter and the receiver; unlike conventional FDM, a separate filter for each sub-channel is not required.

The orthogonality requires that the sub-carrier spacing is Hertz, where T_U seconds is the useful symbol duration (the receiver-side window size), and k is a positive integer, typically equal to 1. This stipulates that each carrier frequency undergoes k more complete cycles per symbol period than the previous carrier. Therefore, with N sub-carriers, the total passband bandwidth will be $B = N*\Delta f$ (Hz).

The orthogonality also allows high spectral efficiency, with a total symbol rate near the Nyquist rate for the equivalent baseband signal (i.e. near half the Nyquist rate for the double-sideband physical passband signal). Almost the whole available frequency band can be used. OFDM generally has a nearly 'white' spectrum, giving it benign electromagnetic interference properties with respect to other co-channel users.

A simple example: A useful symbol duration T_U = 1 ms would require a sub-carrier spacing of (or an integer multiple of that) for orthogonality. N = 1,000 sub-carriers would result in a total passband bandwidth of $N\Delta f$ = 1 MHz. For this symbol time, the required bandwidth in theory according to Nyquist is (half of the achieved bandwidth required by our scheme), where R is the bit rate and where N = 1,000 samples per symbol by FFT. If a guard interval is applied (see below), the Nyquist bandwidth requirement would be even lower. The FFT would result in N = 1,000 samples per symbol. If no guard interval was applied, this would result in a baseband complex-valued signal with a sample rate of 1 MHz, which would require a baseband bandwidth of 0.5 MHz according to Nyquist. However, the passband RF signal is produced by multiplying the baseband signal with a carrier waveform (i.e., double-sideband quadrature amplitude-modulation) resulting in a passband bandwidth of 1 MHz. A single-sideband (SSB) or vestigial sideband (VSB) modulation scheme would achieve almost half that bandwidth for the same symbol rate (i.e., twice as high spectral efficiency for the same symbol alphabet length). It is however more sensitive to multipath interference.

OFDM requires very accurate frequency synchronization between the receiver and the transmitter; with frequency deviation the sub-carriers will no longer be orthogonal, causing inter-carrier interference (ICI) (i.e., cross-talk between the sub-carriers). Frequency offsets are typically caused by mismatched transmitter and receiver oscillators, or by Doppler shift due to movement. While Doppler shift alone may be compensated for by the receiver, the situation is worsened when combined with multipath, as reflections will appear at various frequency offsets, which is much harder to correct. This effect typically worsens as speed increases and is an important factor limiting the use of OFDM in high-speed vehicles. To mitigate ICI in such scenarios, one can shape each sub-carrier to minimize the interference resulting in a non-orthogonal subcarrier overlapping.[12] For example, a low-complexity scheme referred to as WCP-OFDM (Weighted Cyclic Prefix Orthogonal Frequency-Division Multiplexing) consists of using short filters at the transmitter output to perform a potentially non-rectangular pulse shaping and a near-perfect reconstruction using a single-tap per subcarrier equalization. Other ICI suppression techniques usually increase drastically the receiver complexity.

4.2.14 IMPLEMENTATION USING THE FFT ALGORITHM:

The orthogonality allows for efficient modulator and demodulator implementation using the FFT algorithm on the receiver side, and inverse FFT on the sender side. Although the principles and some of the benefits have been known since the 1960s, OFDM is popular for wideband communications today by way of low-cost digital signal processing components that can efficiently calculate the FFT.

The time to compute the inverse-FFT or FFT transform has to take less than the time for each symbol, which for example for DVB-T (FFT 8k) means the computation has to be done in 896 μs or less.

For an 8192-point FFT, this may be approximated to: MIPS = Million instructions per second

The computational demand approximately scales linearly with FFT size so a double size FFT needs double the amount of time and vice versa.[15]:83 As a comparison an Intel Pentium III CPU at 1.266 GHz can calculate an 8192 point FFT in 576 μs using FFTW. Intel Pentium M at 1.6 GHz does it in 387 μs Intel Core Duo at 3.0 GHz does it in 96.8 μs.

4.2.15 GUARD INTERVAL FOR THE ELIMINATION OF INTERSYMBOL INTERFERENCE

One key principle of OFDM is that since low symbol rate modulation schemes (i.e., where the symbols are relatively long compared to the channel time characteristics) suffer less from intersymbol interference caused by multipath propagation, it is advantageous to transmit several low-rate streams in parallel instead of a single high-rate stream. Since the duration of each symbol is long, it is feasible to insert a guard interval between the OFDM symbols, thus eliminating the intersymbol interference.

The guard interval also eliminates the need for a pulse-shaping filter, and it reduces the sensitivity to time synchronization problems.

The cyclic prefix, which is transmitted during the guard interval, consists of the end of the OFDM symbol copied into the guard interval, and the guard interval is transmitted followed by the OFDM symbol. The reason that the guard interval consists of a copy of the end of the OFDM symbol is so that the receiver will integrate over an integer number of sinusoid cycles for each of the multipath when it performs OFDM demodulation with the FFT.

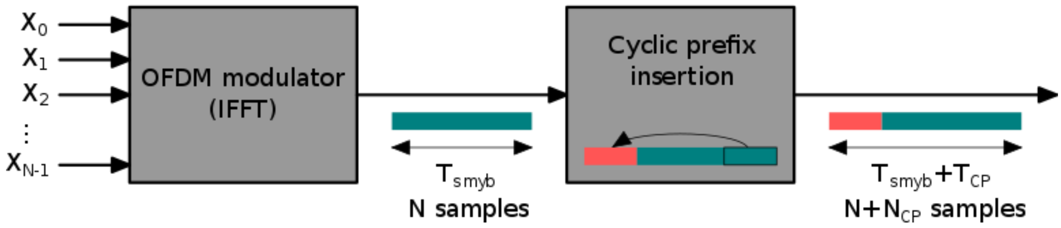


Figure 4.12 : OFDM Modulator

In some standards such as Ultrawideband, in the interest of transmitted power, a cyclic prefix is skipped and nothing is sent during the guard interval. The receiver will then have to mimic the cyclic prefix functionality by copying the end part of the OFDM symbol and adding it to the beginning portion.

4.2.16 SIMPLIFIED EQUALIZATION:

The effects of frequency-selective channel conditions, for example, fading caused by multipath propagation, can be considered as constant (flat) over an OFDM sub-channel if the sub-channel is sufficiently narrow-banded (i.e., if the number of sub-channels is sufficiently large). This makes frequency domain equalization possible at the receiver, which is far simpler than the time-domain equalization used in conventional single-carrier modulation.

In OFDM, the equalizer only has to multiply each detected sub-carrier (each Fourier coefficient) in each OFDM symbol by a constant complex number, or a rarely changed value. On a fundamental level, simpler digital equalizers are better because they require fewer operations, which translate to fewer round-off errors in the equalizer. Those round-off errors can be viewed as numerical noise and are inevitable.

Our example: The OFDM equalization in the above numerical example would require one complex-valued multiplication per subcarrier and symbol (i.e., complex multiplications per OFDM symbol; i.e., one million multiplications per second, at the receiver). The FFT algorithm requires [this is imprecise: over half of these complex multiplications are trivial, i.e. = to 1 and are not implemented in software or HW]. complex-valued multiplications per OFDM symbol (i.e., 10 million multiplications per second), at both the receiver and transmitter side. This should be compared with the corresponding one million symbols/second single-carrier modulation case mentioned in the example, where the equalization of 125 microseconds time-spreading using an FIR filter would require, in a naive implementation, 125 multiplications per symbol (i.e., 125 million multiplications per second). FFT techniques can be used to reduce the number of multiplications for an FIR filter-based time-domain equalizer to a number comparable with OFDM, at the cost of delay between reception and decoding which also becomes comparable with OFDM.

If differential modulation such as DPSK or DQPSK is applied to each sub-carrier, equalization can be completely omitted, since these non-coherent schemes are insensitive to slowly changing amplitude and phase distortion.

In a sense, improvements in FIR equalization using FFTs or partial FFTs leads mathematically closer to OFDM, but the OFDM technique is easier to understand and implement, and the sub-channels can be independently adapted in other ways than varying equalization coefficients, such as switching between different QAM constellation patterns and error-correction schemes to match individual sub-channel noise and interference characteristics.

Some of the sub-carriers in some of the OFDM symbols may carry pilot signals for measurement of the channel condition (i.e., the equalizer gain and phase shift for each sub-carrier). Pilot signals and training symbols (preambles) may also be used for time synchronization (to avoid intersymbol interference, ISI) and frequency synchronization (to avoid inter-carrier interference, ICI, caused by Doppler shift).

OFDM was initially used for wired and stationary wireless communications. However, with an increasing number of applications operating in highly mobile environments, the effect of dispersive fading caused by a combination of multipath propagation and Doppler shift is more significant. Over the last decade, research has been done on how to equalize OFDM transmission over doubly selective channels.

4.2.17 CHANNEL CODING AND INTERLEAVING

OFDM is invariably used in conjunction with channel coding (forward error correction), and almost always uses frequency and/or time interleaving.

Frequency (subcarrier) interleaving increases resistance to frequency-selective channel conditions such as fading. For example, when a part of the channel bandwidth fades, frequency interleaving ensures that the bit errors that would result from those subcarriers in the faded part of the bandwidth are spread out in the bitstream rather than being concentrated. Similarly, time interleaving ensures that bits that are originally close together in the bit-stream are transmitted far apart in time, thus mitigating against severe fading as would happen when travelling at high speed.

However, time interleaving is of little benefit in slowly fading channels, such as for stationary reception, and frequency interleaving offers little to no benefit for narrowband channels that suffer from flat-fading (where the whole channel bandwidth fades at the same time).

The reason why interleaving is used on OFDM is to attempt to spread the errors out in the bit-stream that is presented to the error correction decoder, because when such decoders are presented with a high concentration of errors the decoder is unable to correct all the bit errors, and a burst of uncorrected errors occurs. A similar design of audio data encoding makes compact disc (CD) playback robust.

A classical type of error correction coding used with OFDM-based systems is convolutional coding, often concatenated with Reed-Solomon coding. Usually, additional interleaving (on top of the time and frequency interleaving mentioned above) in between the two layers of coding is implemented. The choice for Reed-Solomon coding as the outer error correction code is based on the observation that the Viterbi decoder used for inner convolutional decoding produces short error bursts when there is a high concentration of errors, and Reed-Solomon codes are inherently well suited to correcting bursts of errors.

Newer systems, however, usually now adopt near-optimal types of error correction codes that use the turbo decoding principle, where the decoder iterates towards the desired solution. Examples of such error correction coding types include turbo codes and LDPC codes, which perform close to the Shannon limit for the Additive White Gaussian Noise (AWGN) channel. Some systems that have implemented these codes have concatenated them with either Reed-Solomon (for example on the MediaFLO system) or BCH codes (on the DVB-S2 system) to improve upon an error floor inherent to these codes at high signal-to-noise ratios.

4.2.18 ADAPTIVE TRANSMISSION

The resilience to severe channel conditions can be further enhanced if information about the channel is sent over a return channel. Based on this feedback information, adaptive modulation, channel coding and power allocation may be applied across all sub-carriers, or individually to each sub-carrier. In the latter case, if a particular range of frequencies suffers from interference or attenuation, the carriers within that range can be disabled or made to run slower by applying more robust modulation or error coding to those sub-carriers.

The term discrete multitone modulation (DMT) denotes OFDM-based communication systems that adapt the transmission to the channel conditions individually for each sub-carrier, utilizing so-called bit-loading. Examples are ADSL and VDSL.

The upstream and downstream speeds can be varied by allocating either more or fewer carriers for each purpose. Some forms of rate-adaptive DSL use this feature in real-time, so that the bitrate is adapted to the co-channel interference and bandwidth is allocated to whichever subscriber needs it most.

4.2.19 OFDM EXTENDED WITH MULTIPLE ACCESS

OFDM in its primary form is considered as a digital modulation technique, and not a multi-user channel access method since it is used for transferring one-bit stream over one communication channel using one sequence of OFDM symbols. However, OFDM can be combined with multiple access using time, frequency or coding separation of the users.

In orthogonal frequency-division multiple access (OFDMA), frequency-division multiple access is achieved by assigning different OFDM sub-channels to different users. OFDMA supports differentiated quality of service by assigning a different number of sub-carriers to different users in a similar fashion as in CDMA, and thus complex packet scheduling or Media Access Control schemes can be avoided. OFDMA is used in: the mobility mode of the IEEE 802.16 Wireless MAN standard, commonly referred to as WiMAX,

- the IEEE 802.20 mobile Wireless MAN standard, commonly referred to as MBWA,
- the 3GPP Long Term Evolution (LTE) fourth-generation mobile broadband standard downlink. The radio interface was formerly named High-Speed OFDM Packet Access (HSOPA), now named Evolved UMTS Terrestrial Radio Access (E-UTRA).
- the now-defunct Qualcomm/3GPP2 Ultra Mobile Broadband (UMB) project, intended as a successor of CDMA2000, but replaced by LTE.
- OFDMA is also a candidate access method for the IEEE 802.22 Wireless Regional Area Networks (WRAN). The project aims at designing the first cognitive radio-based standard operating in the VHF-low UHF spectrum (TV spectrum).

In multi-carrier code division multiple access (MC-CDMA), also known as OFDM-CDMA, OFDM is combined with CDMA spread spectrum communication for coding separation of the users. Co-channel interference can be mitigated, meaning that manually fixed channel allocation (FCA) frequency planning is simplified, or complex dynamic channel allocation (DCA) schemes are avoided.

4.2.20 SPACE DIVERSITY

In OFDM-based wide-area broadcasting, receivers can benefit from receiving signals from several spatially dispersed transmitters simultaneously, since transmitters will only destructively interfere with each other on a limited number of sub-carriers, whereas in general, they will reinforce coverage over a wide area. This is very beneficial in many countries, as it permits the operation of national single-frequency networks (SFN), where many transmitters send the same signal simultaneously over the same channel frequency.

SFNs use the available spectrum more effectively than conventional multi-frequency broadcast networks (MFN), where program content is replicated on different carrier frequencies. SFNs also result in a diversity gain in receivers situated midway between the transmitters. The coverage area is increased and the outage probability decreased in comparison to an MFN, due to increased received signal strength averaged over all sub-carriers.

Although the guard interval only contains redundant data, which means that it reduces the capacity, some OFDM-based systems, such as some of the broadcasting systems, deliberately use a long guard interval to allow the transmitters to be spaced farther apart in an SFN, and longer guard intervals allow larger SFN cell-sizes. A rule of thumb for the maximum distance between transmitters in an SFN is equal to the distance a signal travels during the guard interval — for instance, a guard interval of 200 microseconds would allow transmitters to be spaced 60 km apart.

A single frequency network is a form of transmitter macro-diversity. The concept can be further used in dynamic single-frequency networks (DSFN), where the SFN grouping is changed from timeslot to timeslot.

OFDM may be combined with other forms of space diversity, for example, antenna arrays and MIMO channels. This is done in the IEEE 802.11 Wireless LAN standards.

4.2.21 LINEAR TRANSMITTER POWER AMPLIFIER

An OFDM signal exhibits a high peak-to-average power ratio (PAPR) because the independent phases of the sub-carriers mean that they will often combine constructively. Handling this high PAPR requires:

A high-resolution digital-to-analogue converter (DAC) in the transmitter A high-resolution analogue-to-digital converter (ADC) in the receiver A linear signal chain Any non-linearity in the signal chain will cause intermodulation distortion that Raises the noise floor May cause inter-carrier interference Generates out-of-band spurious radiation

The linearity requirement is demanding, especially for transmitter RF output circuitry where amplifiers are often designed to be non-linear to minimise power consumption. In practical OFDM systems, a small amount of peak clipping is allowed to limit the PAPR in a judicious trade-off against the above consequences. However, the transmitter output filter which is required to reduce out-of-band spurs to legal levels has the effect of restoring peak levels that were clipped, so clipping is not an effective way to reduce PAPR.

Although the spectral efficiency of OFDM is attractive for both terrestrial and space communications, the high PAPR requirements have so far limited OFDM applications to terrestrial systems. The crest factor CF (in dB) for an OFDM system with n uncorrelated sub-carriers is where CF is the crest factor (in dB) for each sub-carrier. (CF is 3.01 dB for the sine waves used for BPSK and QPSK modulation).

For example, the DVB-T signal in 2K mode is composed of 1705 sub-carriers that are each QPSK-modulated, giving a crest factor of 35.32 dB. Many crest factor reduction techniques have been developed.

The dynamic range required for an FM receiver is 120 dB while DAB only requires about 90 dB. As a comparison, each extra bit per sample increases the dynamic range by 6 dB.

The performance of any communication system can be measured in terms of its power efficiency and bandwidth efficiency. Power efficiency describes the ability of a communication system.

Transmission Type	M in M-QAM	No. of Subcarriers	Bit rate	Fiber length	Power at the receiver (at BER of 10^{-9})	Bandwidth efficiency
single carrier	64	1	10 Gbit/s	20 km	-37.3 dBm	6.0000
multicarrier	64	128	10 Gbit/s	20 km	-36.3 dBm	10.6022

Table 4.1: Transmission Types

There is only a 1 dBm increase in receiver power, but we get a 76.7% improvement in bandwidth efficiency by using the multicarrier transmission technique.

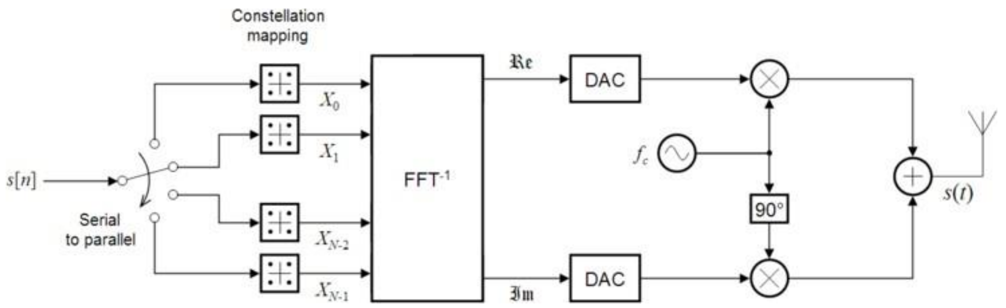


Figure 4.13 : Transmitter

An OFDM carrier signal is the sum of several orthogonal sub-carriers, with baseband data on each sub-carrier being independently modulated commonly using some type of quadrature amplitude modulation (QAM) or phase-shift keying (PSK). This composite baseband signal is typically used to modulate the main RF carrier.

OFDM is used by many powerline devices to extend digital connections through power wiring. Adaptive modulation is particularly important with such a noisy channel as electrical wiring. Some medium speed smart metering modems, "Prime" and "G3" use OFDM at modest frequencies (30–100 kHz) with modest numbers of channels (several hundred) to overcome the intersymbol interference in the power line environment. The IEEE 1901 standards include two incompatible physical layers that both use OFDM. The ITU-T G.hn standard, which provides high-speed local area networking over existing home wiring (power lines, phone lines and coaxial cables) is based on a PHY layer that specifies OFDM with adaptive modulation and a Low-Density Parity-Check (LDPC) FEC code.

An inverse FFT is computed on each set of symbols, giving a set of complex time-domain samples. These samples are then quadrature-mixed to passband in the standard way. The real and imaginary components are first converted to the analogue domain using digital-to-analogue converters (DACs); the analogue signals are then used to modulate cosine and sine waves at the carrier frequency, respectively. These signals are

then summed to give the transmission signal, .

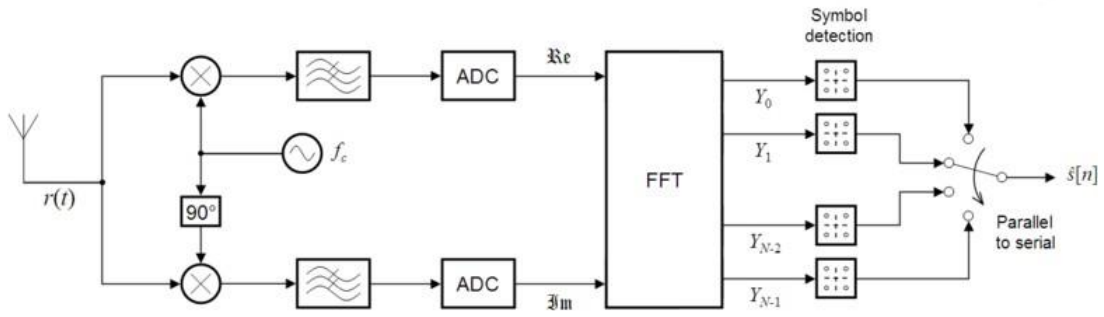


Figure 4.14 : Receiver

The receiver picks up the signal, which is then quadrature-mixed down to baseband using cosine and sine waves at the carrier frequency. This also creates signals centred on, so low-pass filters are used to reject these. The baseband signals are then sampled and digitised using analog-to-digital converters (ADCs), and a forward FFT is used to convert back to the frequency domain.

This returns parallel streams, each of which is converted to a binary stream using an appropriate symbol detector. These streams are then recombined into a serial stream, which is an estimate of the original binary stream at the transmitter.

4.2.22 ADSL

OFDM is used in ADSL connections that follow the ANSI T1.413 and G.dmt (ITU G.992.1) standards, where it is called discrete multitone modulation (DMT). DSL achieves high-speed data connections on existing copper wires. OFDM is also used in the successor standards ADSL2, ADSL2+, VDSL, VDSL2, and G.fast. ADSL2 uses variable sub-carrier modulation, ranging from BPSK to 32768QAM (in ADSL terminology this is referred to as bit-loading, orbit per tone, 1 to 15 bits per sub-carrier).

Long copper wires suffer from attenuation at high frequencies. The fact that OFDM can cope with this frequency-selective attenuation and with narrow-band interference are the main reasons it is frequently used in applications such as ADSL modems.

4.2.23 POWERLINE TECHNOLOGY

OFDM is used by many powerline devices to extend digital connections through power wiring. Adaptive modulation is particularly important with such a noisy channel as electrical wiring. Some medium speed smart metering modems, "Prime" and "G3" use OFDM at modest frequencies (30–100 kHz) with modest numbers of channels (several hundred) to overcome the intersymbol interference in the power line environment. The IEEE 1901 standards include two incompatible physical layers that both use OFDM. The ITU-T G.hn standard, which provides high-speed local area networking over existing home wiring (power lines, phone lines and coaxial cables) is based on a PHY layer that specifies OFDM with adaptive modulation and a Low-Density Parity-Check (LDPC) FEC code.

4.2.24 WIRELESS LOCAL AREA NETWORKS (LAN) AND METROPOLITAN AREA NETWORKS (MAN)

OFDM is extensively used in wireless LAN and MAN applications, including IEEE 802.11a/g/n and WiMAX.

IEEE 802.11a/g/n, operating in the 2.4 and 5 GHz bands, specifies per-stream airside data rates ranging from 6 to 54 Mbit/s. If both devices can use "HT mode" (added with 802.11n), the top 20 MHz per-stream rate is increased to 72.2 Mbit/s, with the option of data rates between 13.5 and 150 Mbit/s using a 40 MHz channel. Four different modulation schemes are used: BPSK, QPSK, 16-QAM, and 64-QAM, along with a set of error-correcting rates (1/2–5/6). The multitude of choices allows the system to adapt the optimum data rate for the current signal conditions.

4.2.25 WIRELESS PERSONAL AREA NETWORKS (PAN)

OFDM is also now being used in the WiMedia/Ecma-368 standard for high-speed wireless personal area networks in the 3.1–10.6 GHz ultrawideband spectrum (see MultiBand-OFDM).

- **Terrestrial digital radio and television broadcasting:**

Much of Europe and Asia have adopted OFDM for terrestrial broadcasting of digital television (DVB-T, DVB-H and T-DMB) and radio (EUREKA 147 DAB, Digital Radio Mondiale, HD Radio and T-DMB).

By Directive of the European Commission, all television services transmitted to viewers in the European Community must use a transmission system that has been standardized by a recognized European standardization body, and such a standard has been developed and codified by the DVB Project, Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television. Customarily referred to as DVB-T, the standard calls for the exclusive use of COFDM for modulation. DVB-T is now widely used in Europe and elsewhere for terrestrial digital TV.

- **SDARS**

The ground segments of the Digital Audio Radio Service (SDARS) systems used by XM Satellite Radio and Sirius Satellite Radio are transmitted using Coded OFDM (COFDM). The word "coded" comes from the use of forwarding error correction (FEC).

- **COFDM vs VSB**

The question of the relative technical merits of COFDM versus **8VSB** for terrestrial **digital television** has been a subject of some controversy, especially between European and North American technologists and regulators. The **United States** has rejected several proposals to adopt the COFDM-based **DVB-T** system for its digital television services and has instead opted for an **8VSB** (**vestigial sideband modulation**) operation.

One of the major benefits provided by COFDM is in rendering radio broadcasts relatively immune to multipath distortion and signal fading due to atmospheric conditions or passing aircraft. Proponents of COFDM argue it resists multipath far better than 8VSB. Early 8VSB DTV (digital television) receivers often had difficulty receiving a signal. Also, COFDM allows single-frequency networks, which is not possible with 8VSB.

However, newer 8VSB receivers are far better at dealing with multipath, hence the difference in performance may diminish with advances in equalizer design.

- **Digital radio**

COFDM is also used for other radio standards, for Digital Audio Broadcasting (DAB), the standard for digital audio broadcasting at VHF frequencies, for Digital Radio

Mondiale(DRM), the standard for digital broadcasting at shortwave and mediumwave frequencies (below 30 MHz) and DRM+ a more recently introduced standard for digital audio broadcasting at VHF frequencies. (30 to 174 MHz)

The USA again uses an alternate standard, a proprietary system developed by iBiquity dubbed HD Radio. However, it uses COFDM as the underlying broadcast technology to add digital audio to AM (medium wave) and FM broadcasts.

Both Digital Radio Mondiale and HD Radio are classified as in-band on-channel systems, unlike Eureka 147 (DAB: Digital Audio Broadcasting) which uses separate VHF or UHF frequency bands instead.

- **BST-OFDM used in ISDB**

The band-segmented transmission orthogonal frequency division multiplexing (BST-OFDM) system proposed for Japan (in the ISDB-T, ISDB-TSB, and ISDB-C broadcasting systems) improves upon COFDM by exploiting the fact that some OFDM carriers may be modulated differently from others within the same multiplex. Some forms of COFDM already offer this kind of hierarchical modulation, though BST-OFDM is intended to make it more flexible. The 6 MHz television channel may therefore be "segmented", with different segments being modulated differently and used for different services.

It is possible, for example, to send an audio service on a segment that includes a segment composed of many carriers, a data service on another segment and television service on yet another segment—all within the same 6 MHz television channel. Furthermore, these may be modulated with different parameters so that, for example, the audio and data services could be optimized for mobile reception, while the television service is optimized for stationary reception in a high-multipath environment.

- **ULTRA-WIDEBAND**

Ultra-wideband (UWB) wireless personal area network technology may also use OFDM, such as in Multiband OFDM (MB-OFDM). This UWB specification is advocated by the WiMedia Alliance (formerly by both the Multiband OFDM Alliance [MBOA] and the WiMedia Alliance, but the two have now merged), and is one of the competing UWB radio interfaces.

- **FLASH-OFDM**

Fast low-latency access with seamless handoff orthogonal frequency division multiplexing (Flash-OFDM), also referred to as F-OFDM, was based on OFDM and also specified higher protocol layers. It was developed by Flarion, and purchased by Qualcomm in January 2006. Flash-OFDM was marketed as a packet-switched cellular bearer, to compete with GSM and 3G networks. As an example, 450 MHz frequency bands previously used by NMT-450 and C-Net C450 (both 1G analogue networks, now mostly decommissioned) in Europe are being licensed to Flash-OFDM operators.

In Finland, the license holder Digita began deployment of a nationwide "@450" wireless network in parts of the country in April 2007. It was purchased by Datame in 2011. In February 2012 Datame announced they would upgrade the 450 MHz networks to competing for CDMA2000 technology

Slovak Telekom in Slovakia offers Flash-OFDM connections with a maximum downstream speed of 5.3 Mbit/s, and a maximum upstream speed of 1.8 Mbit/s, with a coverage of over 70% of the Slovak population The Flash-OFDM network was switched off in the majority of Slovakia on 30 September 2015.

T-Mobile Germany used Flash-OFDM to backhaul Wi-Fi HotSpots on the Deutsche Bahn's ICE high-speed trains between 2005 and 2015, until switching over to UMTS and LTE.

American wireless carrier Nextel Communications field-tested wireless broadband network technologies including Flash-OFDM in 2005. Sprint purchased the carrier in 2006 and decided to deploy the mobile version of WiMAX, which is based on Scalable Orthogonal Frequency Division Multiple Access (SOFDMA) technology.

Citizens Telephone Cooperative launched a mobile broadband service based on Flash-OFDM technology to subscribers in parts of Virginia in March 2006. The maximum speed available was 1.5 Mbit/s. The service was discontinued on April 30, 2009.

OFDM has become an interesting technique for power line communications (PLC). In this area of research, a wavelet transform is introduced to replace the DFT as the method of creating orthogonal frequencies. This is due to the advantages wavelets offer, which are particularly useful on noisy power lines.

Instead of using an IDFT to create the sender signal, the wavelet OFDM uses a synthesis bank consisting of a -band transmultiplexer followed by the transform function.

On the receiver side, an analysis bank is used to demodulate the signal again. This bank contains an inverse transform followed by another -band trans-multiplexer. The relationship between both transform functions is

An example of W-OFDM uses the Perfect Reconstruction Cosine Modulated Filter Bank (PR-CMFB) and Extended Lapped Transform (ELT) is used for the wavelet TF. Thus, and are given as

- **Advantages over standard OFDM**

W-OFDM is an evolution of the standard OFDM, with certain advantages.

Mainly, the sidelobe levels of W-OFDM are lower. This results in less ICI, as well as greater robustness to narrowband interference. These two properties are especially useful in PLC, where most of the lines aren't shielded against EM-noise, which creates noisy channels and noise spikes.

A comparison between the two modulation techniques also reveals that the complexity of both algorithms remains approximately the same.

In a sense, improvements in FIR equalization using FFTs or partial FFTs leads mathematically closer to OFDM, but the OFDM technique is easier to understand and implement, and the sub-channels can be independently adapted in other ways than varying equalization coefficients, such as switching between different QAM constellation patterns and error-correction schemes to match individual sub-channel noise and interference characteristics.

Some of the sub-carriers in some of the OFDM symbols may carry pilot signals for measurement of the channel condition (i.e., the equalizer gain and phase shift for each sub-carrier). Pilot signals and training symbols (preambles) may also be used for time synchronization (to avoid intersymbol interference, ISI) and frequency synchronization (to avoid inter-carrier interference, ICI, caused by Doppler shift).

OFDM was initially used for wired and stationary wireless communications. However, with an increasing number of applications operating in highly mobile environments, the effect of dispersive fading caused by a combination of multipath propagation and Doppler shift is more significant. Over the last decade, research has been done on how to equalize OFDM transmission over doubly selective channels.

4.2.26 CHANNEL CODING AND INTERLEAVING

OFDM is invariably used in conjunction with channel coding (forward error correction), and almost always uses frequency and/or time interleaving.

Frequency (subcarrier) interleaving increases resistance to frequency-selective channel conditions such as fading. For example, when a part of the channel bandwidth fades, frequency interleaving ensures that the bit errors that would result from those subcarriers in the faded part of the bandwidth are spread out in the bitstream rather than being concentrated. Similarly, time interleaving ensures that bits that are originally close together in the bit-stream are transmitted far apart in time, thus mitigating against severe fading as would happen when travelling at high speed.

CHAPTER-5

RESULTS ANALYSIS

5.1 Results

To evaluate its performances, we will simulate the behaviour of the massive MIMO-OFDM-IM system with the classic user selection techniques (ordinary MIMO system with a reduced antenna number at the base station) namely the Greedy algorithm, Random beamforming and SUBF where there is only user transmitting in the cell.

In Figure 5.1, it is the classical case when M=2, the sum rates of ZF (with and without Greedy algorithm as selection method) and RBF (with randomly generated codebooks) are compared for various values of SNR; It is seen that ZF when using a Greedy method maintains a significant advantage over normal ZF, RBF, and SUBF. At a small value of SNR, both schemes of RBF and normal ZF perform similarly, but SUBF gives the poorest performance in comparison with the other schemas.

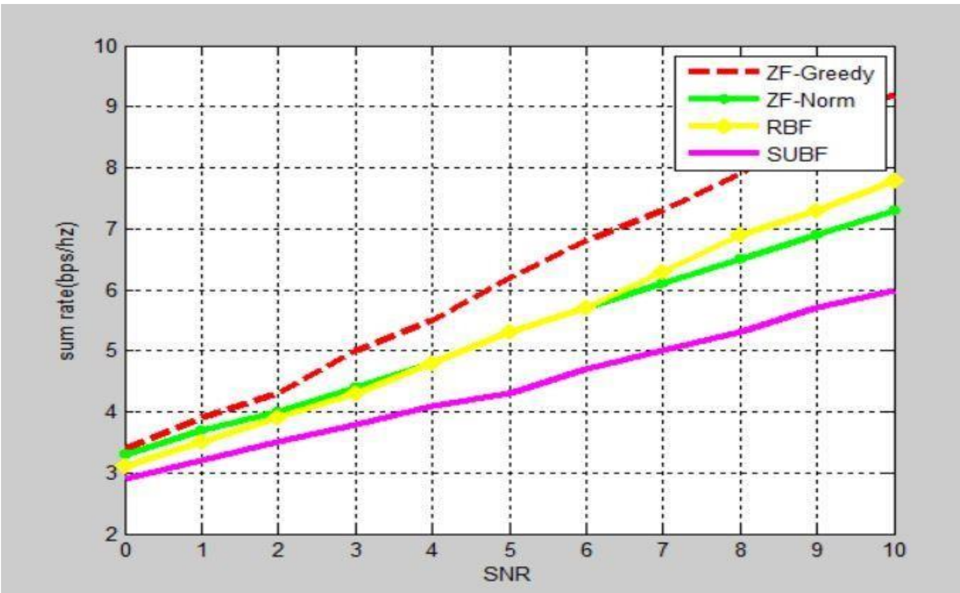


Figure 5.1 : Sum Rate vs SNR for M=2

In Figure 5.2, the same methods are simulated but for a large number of the antennas at the base station namely M = 50. We already notice the big difference between the case of a perfect CSIT and the beamforming random. Intuitively, the difference with perfect CSIT case in terms of sum rates is understandable if we consider the impact of the multiuser interference penalty term and the non-optimality of these algorithms on the other hand.

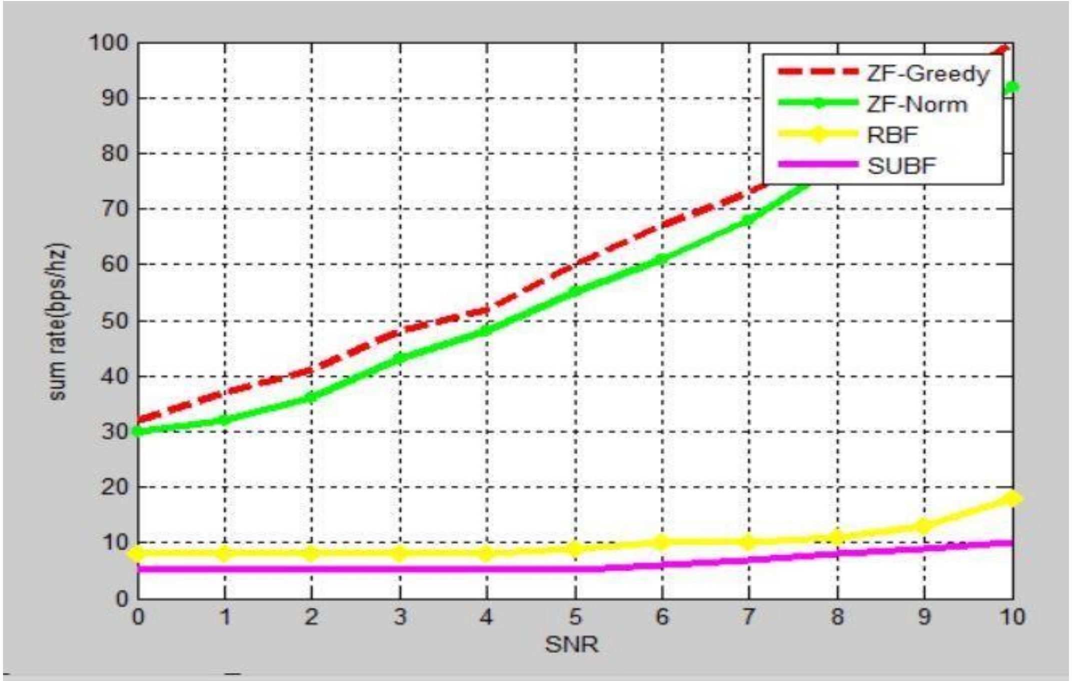


Figure 5.2 : Sum Rate vs SNR for M=50

Figure 5.3 compares the throughput of RBF and ZF for several values of M. The SNR = 10 dBm and feedback size is equal to T=100 bit. As observed from this figure, both RBF and SUBF achieves lower throughput than ZF over the range of M under consideration ($10 \leq M \leq 100$).

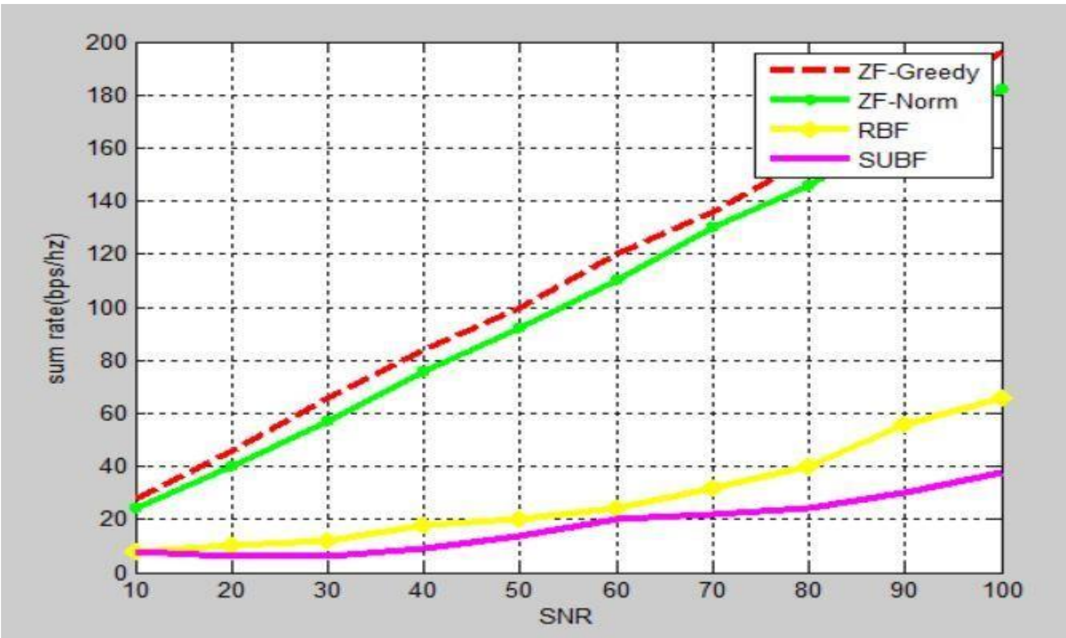


Figure 5.3 : Sum Rate vs Antenna Number; SNR=10, T=100

This should not prevent us from noticing the value of Sum rate in the case of a zero-forcing with the Greedy algorithm which resulted in a value almost equal to 200 bps / Hz for M = 100 which demonstrates the contribution of massive Mimo which intuitively waited.

5.2 Performance Evaluation

S.No	Parameters	Traditional MIMO-OFDM System	Massive MIMO-OFDM-IM
1	Antenna Number	≤8	≥100
2	Throughput	Low	High
3	Signal to Noise Ratio	Low	High
4	Data Rate	Low	High
5	Bit Error Rate	High	Low

Table 5.2: Performance Evaluation

CHAPTER 6

CONCLUSION

6.1 CONCLUSION

This project deals with the case of a downlink massive MIMO-OFDM-IM system in a single cell with a large number of users. Simulation results showed that the system efficiency in the sense of max rate is acceptable with the technique of zero-forcing except that it is very poor with the other classical techniques (like the RBF) especially if we consider the case of imperfect knowledge of the channel at the base station (which is the most realistic case already). These results are interpretable and explainable; indeed this is since that we did not take into account the spatial correlation between transmission antennas, which exists strongly considering a large number of antennas and consequently the minimal distance between them. In another part, pilot contamination has a major influence practically on spectral efficiency. Indeed, a massive MIMO system needs to estimate all the channels between the transmitter and the receivers. The number of these channels is much larger than in the case of conventional MIMO. We are more likely to have more pilots in MIMO systems which cause more interference between pilots as their distance is smaller in the case of MIMO. As perspectives, this aspect can be considered in future projects.

PUBLICATION

ic smec ece
to me

May 30, 2021, 12:11 AM

Dear Author(s),

Warm Greetings from St.Martins Engineering College.

With heartiest congratulations we are pleased to inform you that based on the recommendations of the expert reviewers, your paper has been accepted for journal publication and oral presentation in the online mega International Conference on "Smart Modernistic in Electronics and Communication" (ICSMEC-21).

Paper ID:ICSMEC21-0033
Paper Title:Downlink user selection for massive MIMO-OFDM systems using ZF Preceding

Paper Accepted.

Please Send the final camera ready paper Paper After Registration

Please fill the below "**PAYMENT REGISTRATION FORM**" for completing registration process and follow the checklist to know what to attach to the mail to finish your registration process (**Mandatory to be completed within 5 days of receiving this mail**)

https://forms.gle/8zW1FvyK8XQFBiXQ9

Note: You are now requested to pay the conference registration fees at below mentioned details on or before 5 working days after receiving this email else your paper will not be considered for presentation as well as publication in the Journal. Follow the checklist given below for proceedings.

Account Details for Registration: (Current Account)
A/C Name: PRINCIPAL, St. Martin's Engineering College
Bank Name: SBI, Kukatpally, Hyderabad
Account No: 30667640268
IFSC code: SBIN0011664

Checklist to be followed after payment:

1. Final paper in MS Word and PDF (By mail)
2. Filled and scanned registration form

The project has been approved by international conference that is being conducted by out college i.e., ICSMEC-21 for publication.

REFERENCES

[1] Y. Mehmood, W. Afzal, F. Ahmad, U. Younas, I. Rashid, and I. Mehmood. Large scaled multi-user MIMO system so-called massive MIMO systems for future wireless communication networks. In 2013 19th International Conference on Automation and Computing (ICAC), pages 1–4, 2013.

[2] Joao Vieira, Steffen Malkowsky, Karl Nieman, Zachary Miers, Nikhil Kundargi, Liang Liu, Ian Wong, Viktor Owall, Ove Edfors, and Fredrik Tufvesson. A flexible 100-antenna testbed for Massive MIMO. In IEEE GLOBECOM 2014 Workshop on Massive MIMO : from theory to practice, pages 12–08, 2014.

[3] Hien Quoc Ngo, E.G. Larsson, and T.L. Marzetta. Energy and Spectral Efficiency of Very Large Multiuser MIMO Systems. IEEE Transactions on Communications, 61(4):1436–1449, 2013.

[4] F. Rusek, D. Persson, Buon Kiong Lau, E.G. Larsson, T.L. Marzetta, O. Edfors, and F. Tufvesson. Scaling Up MIMO: Opportunities and Challenges with Very Large Arrays. IEEE Signal Processing Magazine, 30(1):40–60, 2013.

[5] J. Hoydis, S. Ten Brink, and M. Debbah, ―Massive MIMO in the UL/DL of cellular networks: How many antennas do we need?‖ IEEE J. Sel. Areas Commun., vol. 31, no. 2, pp. 160–171, 2013.

[6] T. L. Marzetta, E. G. Larsson, H. Yang, and H. Q. Ngo, Fundamentals of Massive MIMO. Cambridge University Press, 2016.

[7] Erik G. Larsson, Fredrik Tufvesson, Ove Edfors, and Thomas L. Marzetta. Massive MIMO for Next Generation Wireless Systems. arXiv :1304.6690 [cs, math], April 2013.

[8] G. Caire, N. Jindal, M. Kobayashi, and N. Ravindran, Multiuser MIMO Achievable Rates with Downlink Training and Channel State Feedback, IEEE Trans. Information Theory, Vol. 56, No. 6, pp. 2845-2866, June 2010.

[9] Emil Björnson, Luca Sanguinetti, Jakob Hoydis, and Mérouane Debbah. Designing Multi-User MIMO for Energy Efficiency: When is Massive MIMO the Answer? arXiv :1310.3843 [cs, math], October 2013.

[10] E. Björnson, J. Hoydis, and L. Sanguinetti, ―Massive MIMO Networks: Spectral, energy, and hardware efficiency,‖ Foundations and TrendsR in Signal Processing, vol. 11, no. 3-4, pp. 154–655, 2017.

[11] E. Driouch, W. Ajib, ―Efficient Scheduling Algorithms for MultiAntenna CDMA Systems,‖ IEEE Trans. on Vehicular Technology, vol. 61, no. 2, pp.521-532, Feb. 2012.

[12] G. Xu, A. Liu, W. Jiang, H Xiang, W. Luo, ―Joint user scheduling and antenna selection in distributed massive MIMO systems with limited backhaul capacity,‖ China Communications, vol. 11, Issue: 5, May 2014 [13] P.drayi K. Nagaraju―Modelling, Simulation of Multi-User Diversity with Channel State Information in MIMO Systems‖ IRACST – International Journal of Computer Networks and Wireless.

APPENDICES

```
% Downlink user selection for
% massive-MIMO-OFDM systems using ZF precoding

clear  all;
close all;
clc
total_bits = 10^4; % number of bits or symbols
Eb_N0_dB = [0:25]; % multiple Eb/N0 values
transimitters = 2;
receivers = 2;
theta=256;packet_size=6;
channel_length= 64; % channel size
channel_width=64; %channel length
sub_carriers= 32; % number of data subcarriers
nBitPerSym= 48; % number of bits per OFDM symbol
Eb_N0_dB      = [0:25]; % bit to noise ratio
SNRdb=25;
channel_information= Eb_N0_dB + 10*log10(sub_carriers/channel_length) + 10*log10(64/80);
p=1;m=0.2;q=0.02;r=0.4;y1=1;
for ii = 1:length(Eb_N0_dB)

   % Transmitter
   input_random_data = rand(1,total_bits)>0.5;
   modulation_type = 4*input_random_data-1;
   sMod = kron(modulation_type,ones(receivers,1));
   sMod = reshape(sMod,[receivers,transimitters,total_bits/transimitters]); % grouping in
[nRx,nTx,N/NTx ] matrix
   target_signal=transimitters/receivers;
   interference=channel_length/channel_width;
   noise_optimize=target_signal-interference;
   target_int=target_signal+interference+noise_optimize;% ----------------- (1)

   received_info = 1/sqrt(2)*[randn(receivers,transimitters,total_bits/transimitters) +
j*randn(receivers,transimitters,total_bits/transimitters)+target_int]; %
   n = 1/sqrt(2)*[randn(receivers,total_bits/transimitters) + j*randn(receivers,total_bits/transimitters)];
%  variance

   % Channel and noise Noise addition
   squeeze_info = squeeze(sum(received_info.*sMod,2)) + 10^(-Eb_N0_dB(ii)/20)*n;
   complex_valued=SNRdb/theta;
   actual_transmit=complex_valued-transimitters;
   actual_receive=actual_transmit+receivers;
   target_signal=complex_valued*actual_transmit*actual_receive;% -------------------(2)

   % morkov process
   total_data = zeros(2,2,total_bits/transimitters)+target_signal;
   total_data(1,1,:) = sum(received_info(:,2,:).*conj(received_info(:,2,:)),1);
   total_data(2,2,:) = sum(received_info(:,1,:).*conj(received_info(:,1,:)),1);
   computed=y1/packet_size;
   elements=y1-packet_size;
   virtual_steering=computed*elements;% --------- (3)
   total_data(2,1,:) = -sum(received_info(:,2,:).*conj(received_info(:,1,:)),1)+virtual_steering;
   total_data(1,2,:) = -sum(received_info(:,1,:).*conj(received_info(:,2,:)),1);
   hDen = ((total_data(1,1,:).*total_data(2,2,:)) - (total_data(1,2,:).*total_data(2,1,:))); % ad-bc term
```

```
    hDen =
reshape(kron(reshape(hDen,1,total_bits/transimitters),ones(2,2)),2,2,total_bits/transimitters); %
formatting for division
    hInv = total_data./hDen; % inv(H^H*H)

    info_reshape = reshape(conj(received_info),receivers,total_bits); % H^H operation
    convariance_matrix=packet_size/theta;
    variance=channel_length/transimitters;
    computed=convariance_matrix/(variance*convariance_matrix);%------------- (4)
    de_mod_data = kron(squeeze_info,ones(1,2)+computed); % formatting the received symbol for
equalization
    de_mod_data = sum(info_reshape.*de_mod_data,1); % H^H * y
    de_mod_data = kron(reshape(de_mod_data,2,total_bits/transimitters),ones(1,2)); % formatting
    yHat = sum(reshape(hInv,2,total_bits).*de_mod_data,1); % inv(H^H*H)*H^H*y

    % receiver - hard decision decoding
    ipHat = real(yHat)>0;

    % counting the errors
    total_SNR(ii) = size(find([input_random_data- ipHat]),2);

end
[total_SNR proposed_snr]=size(total_SNR);



%figure1
x=[0 1 2 3 4 5 6 7 8 9 10];
exi1
figure,
plot(x,total_SNR,'r--','linewidth',3,'markerface','r','markersize',3);
hold on
plot(x,y2,'g-*','linewidth',3,'markerface','r','markersize',5);
hold on
plot(x,y3,'y-d','linewidth',3,'markerface','r','markersize',5);
hold on
plot(x,y4,'m','linewidth',3,'markerface','r','markersize',5);
hold on
xlabel('SNR');
ylabel('sum rate(bps/hz)');
grid on
legend('ZF-Greedy','ZF-Norm','RBF','SUBF');
%figure
x=[0 1 2 3 4 5 6 7 8 9 10];
exi2
figure,
plot(x,de_mod_data,'r--','linewidth',3,'markerface','r','markersize',3);
hold on
plot(x,y2,'g-*','linewidth',3,'markerface','r','markersize',5);
hold on
plot(x,y3,'y-d','linewidth',3,'markerface','r','markersize',5);
hold on
plot(x,y4,'m','linewidth',3,'markerface','r','markersize',5);
hold on
xlabel('SNR');
ylabel('sum rate(bps/hz)');
grid on
legend('ZF-Greedy','ZF-Norm','RBF','SUBF');
%figure
x=[10 20 30 40 50 60 70 80 90 100];
```

```
exi3
figure,
plot(x,proposed_snr,'r--','linewidth',3,'markerface','r','markersize',3);
hold on
plot(x,y2,'g-*','linewidth',3,'markerface','r','markersize',5);
hold on
plot(x,y3,'y-d','linewidth',3,'markerface','r','markersize',5);
hold on
plot(x,y4,'m','linewidth',3,'markerface','r','markersize',5);
hold on
xlabel('SNR');
ylabel('sum rate(bps/hz)');
grid on
legend('ZF-Greedy','ZF-Norm','RBF','SUBF');
```

A

MAJOR PROJECT REPORT

on

FINGERPRINT VOTING SYSTEM

Submitted by

1) Ms. Ch. Neeraja (17K81A0410)

2) Ms. M. Shravya (17K81A0434)

3) Ms. V. Vaishnavi (17K81A0457)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Mrs.G.Vinatha

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500100

NBA &NAAC A+ Accredited

JUNE 2021

St. MARTIN'S ENGINEERINGCOLLEGE

UGC AUTONOMOUS

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

BONAFIDE CERTIFICATE

This is to certify that the project entitled **"Fingerprint Voting System"** is being submitted by **1.Ms. Ch. Neeraja 17K81A0410, 2.Ms. M. Shravya 17K81A0434 3.Ms. V. Vaishnavi 17K81A0457** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OFTECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Internal Guide	**Head of the Department**
Mrs.G.Vinatha Department of Electronics and Communication iEngineering	Dr. Hari Krishna Department of Electronics and Communication Engineering

Internal Examiner	**External Examiner**

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT CHICOTI NEERAJA WITH ROLL NO.17K81A0410, MULLURI SHRAVYA WITH ROLL NO.17K81A0434, VUPPALA VAISHNAVI WITH ROLL NO.17K81A0457, OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "FINGERPRINT VOTING SYSTEM" AT OUR DEVELOPMENT CENTER, KOMPALLY.

Lasya IT Solutions Pvt. Ltd. ★ Kompally-500014 ★

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

DECLARATION

We, the students of '**Bachelor of Technology in Department of Electronics and Communication Engineering**', session: 2017-2021, **St. Martin's Engineering College,** Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project work entitled '**Fingerprint Voting System**' is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

Ch. Neeraja	M. Shravya	V. Vaishnavi
17K81A0410	17K81A0434	17K81A045

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success**.**

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr. P. Santosh Kumar Patra**, our beloved **Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B. Hari Krishna**, **Head & Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our department **Project Coordinator, Dr. A. Anand Kumar, Professor** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our **Class Project Coordinator, Mrs. A. Soumya, Assistant Professor** Department of ECE and our project **Internal guide, Mrs. G. Vinatha, Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level **Project Coordinators Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. Ch. Neeraja
2. M.Shravya
3. V.Vaishnavi

ABSTRACT

It has always been an arduous task for the election commission to conduct free and fair polls in our country, the largest democracy in the world. Crores of rupees have been spent on this to make sure that the elections are riot free. But, now- a -days it has become common for some forces to indulge in rigging which may eventually lead to a result contrary to the actual verdict given by the People.This project aims to present a new voting system employing biometrics in order to avoid rigging and to enhance the accuracy and speed of the process. The system uses thumb impression for voter identification as we know that the thumb impression of every human being has a unique pattern. Thus, it would have an edge over the present-day voting systems.

As a pre-poll procedure, a database consisting of the thumb impressions of all the eligible voters in a constituency is created. During elections, the thumb impression of a voter is entered as input to the system. This is then compared with the available records in the database. If the particular pattern matches with anyone in the available record, access to cast a vote is granted. But in case the pattern doesn't match with the records of the database or in case of repetition, access to cast a vote is denied or the vote gets rejected. Also, the police station nearby to the election poll booth is informed about the identity of the imposter.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.0 OVERVEIW OF THE PROJECT:

Biometrics is the science and technology of measuring and analyzing biological data. Biometrics refers to technologies that measure and analyze human body characteristics, such as DNA, fingerprints, eye retinas and irises, voice patterns, facial patterns and hand measurements, for authentication purposes. The field of biometrics was formed and has since expanded on to many types of physical identification. Among the several human fingerprints remain a very common identifier and the biometric method of choice among law enforcement. These concepts of human identification have lead to the development of fingerprint scanners that serve to quickly identify individuals and assign access privileges. The basic point of these devices is also to examine the fingerprint data of an individual and compare it to a database of other fingerprints . In our project we have used fingerprint for the purpose of voter identification or authentication. As the thumb impression of every individual is unique, it helps in minimizing the error. A database is created containing the fingerprint images of all the voters as required. Illegal votes and repetition of votes is checked for in this system with accurate coding. Hence with the application of this fingerprint based EVM system elections could be made fair and free from rigging. Further that the elections would are no longer a tedious and expensive job.

1.1 OBJECTIVES OF STUDY:

Tampering with the election process is not uncommon, even in some of the countries with strong rule of law. Ballot stuffing, voter suppression, multiple voting and destruction of legitimately cast ballot still covers the headlines of newspapers. Since every person has the unique fingerprint in this world, their fingerprint can be used to cast their vote with electronic fingerprint voting system. The proposed system is offline version. The finger-print voting system requires to register their fingerprint at the polling booth. The person can now vote on the Election Day just by verifying their identity using their fingerprint. The system uses Arduino and fingerprint technology. Voting does not 2 only include political but can be social, business and educational as well.

The development steps or method that outlines the way to achieve certain goal or in this case design and build a prototype Fingerprint voting system. In other words, it is a constructive framework. Below in figure , methodology flow block diagram of this project is shown.

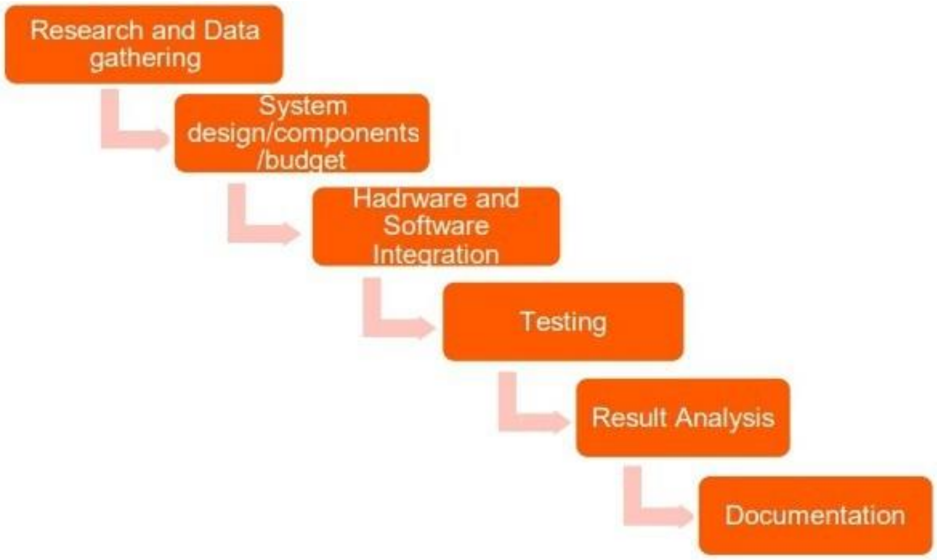


Fig 1.1: Flow diagram methodology

1.2 SCOPE OF THE STUDY:

Election is a feature of democratic government in which people govern themselves and are able to express their choices regarding various issues, such as constitutional amendments, piece of legislation or choosing the right person as their leader. An electoral system is present to layout the rules of the election. Political election is the most common form of election but there are many different fields where election is vital part of their organizational functions. Election is vital for business, informal organizations and nonprofit organizations. Election is the way of democratic world but conducting an election in fair manner has been the challenge of every electoral body especially in the country with high corruption, weak rule or law and less transparency. On the top of that conducting election cost millions and billions of dollars. This study report mainly focuses on developing a prototype of embedded system that deals with fingerprint voting system which can help in progression of election in robust and secure manner. The system integrates different hardware components like microcontroller, fingerprint module, LEDs, switches that facilitates a flawless voting system. For the implementation of this system, DY50 fingerprint sensor is used to take user finger print image and store in internal memory, these images are further processed and analyzed using Arduino. The user interface is implemented using LCD screen, which is mainly used to print user instructions during the execution of the voting process and the result.

1.3 MATERIALS REQUIREMENT:

1.3.1 HARDWARE COMPONENTS:

- ARDUINO UNO (ATMEGA328) MICROCONTROLLER BOARD

Fig:1.2 Arduino Uno board

- FINGER PRINT SCANNER ALONG WITH 3 PUSHBUTTONS

Fig:1.3 Fingerprint scanner

- LCD DISPLAY (16*2)

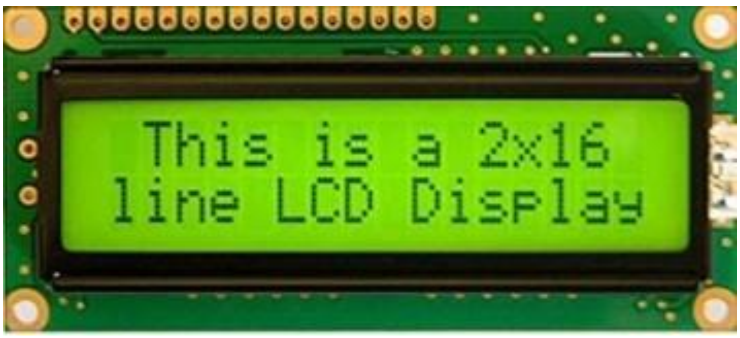


Fig:1.4 LCD Display

- POTENTIOMETER

Fig:1.5 POTENTIOMETER

- MB27L-12V PIEZO BUZZER

Fig:1.6 Piezo buzzer

1.3.2 SOFTWARE REQUIREMENTS:

1.3.2.1 ARDUINO IDE SOFTWARE:

This software is basically used for giving commands to ARDUNO Microcontroller**.**

STEPS FOR INSTALLING ARDUNO IDE SOFTWARE:

1. Download Arduino IDE Software in your PC.

2. When the download finishes, proceed with the installation.

3. Allow the driver installation process when you get a warning from the operating system.

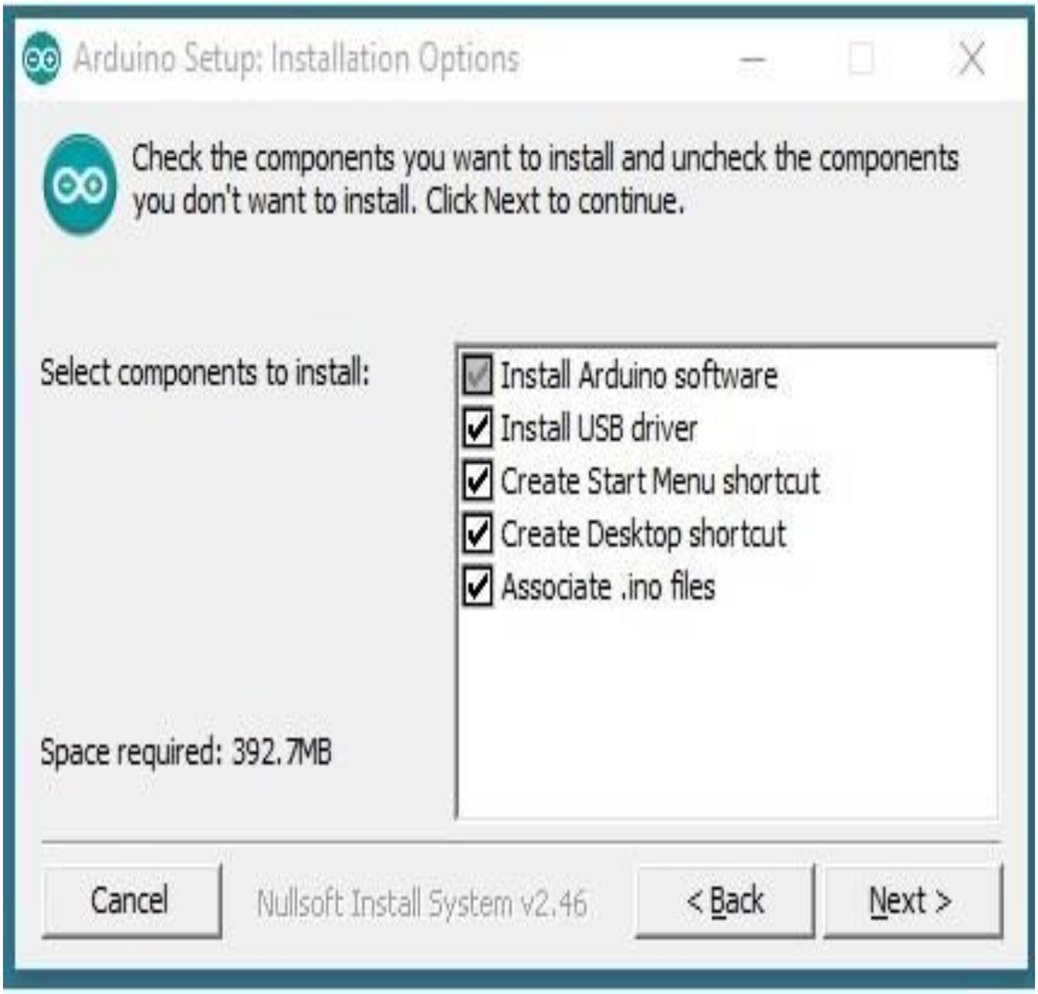


Fig:1.7 Installation process of driver

4. Choose the components to install

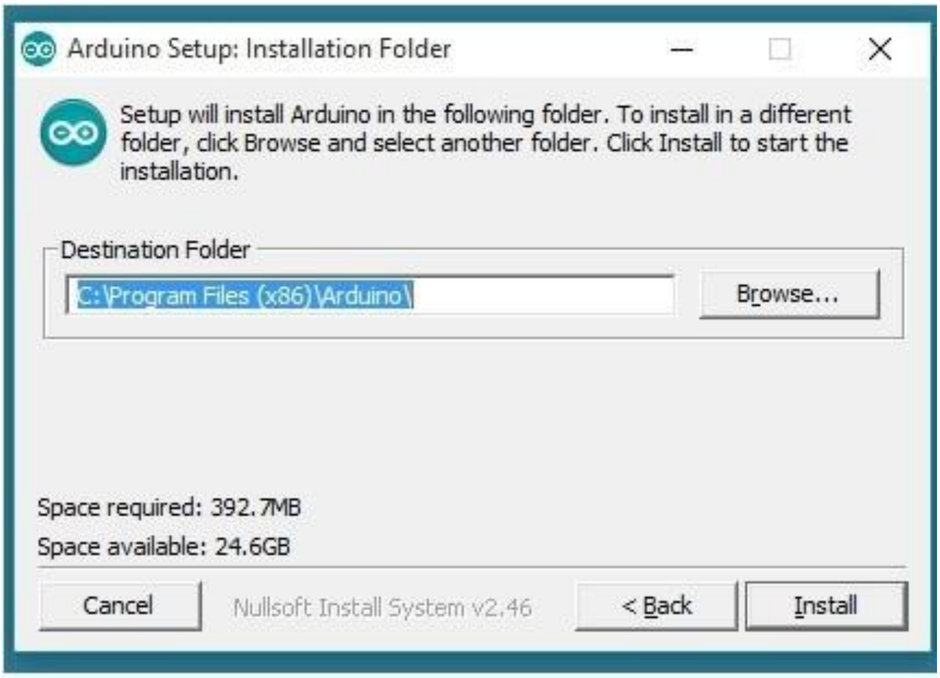


Fig 1.8 Setting destination folder

5. Choose the installation directory (we suggest to keep the default one)

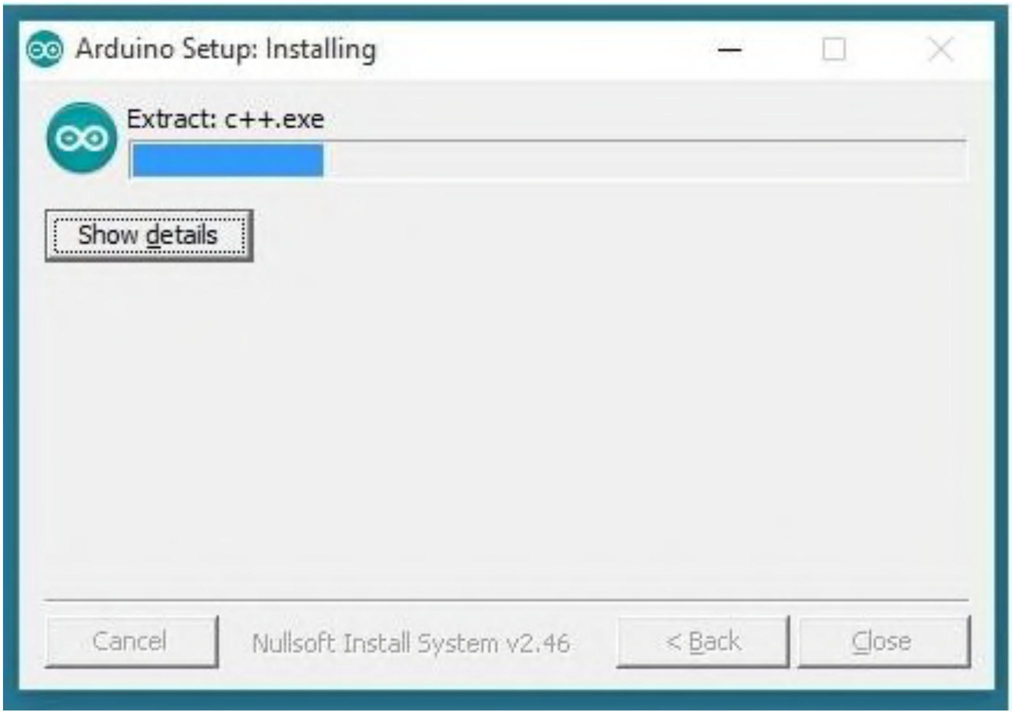


Fig:1.9 Installing directory

6. Successfully completed Installation of Software

7. Writing code on IDE platform

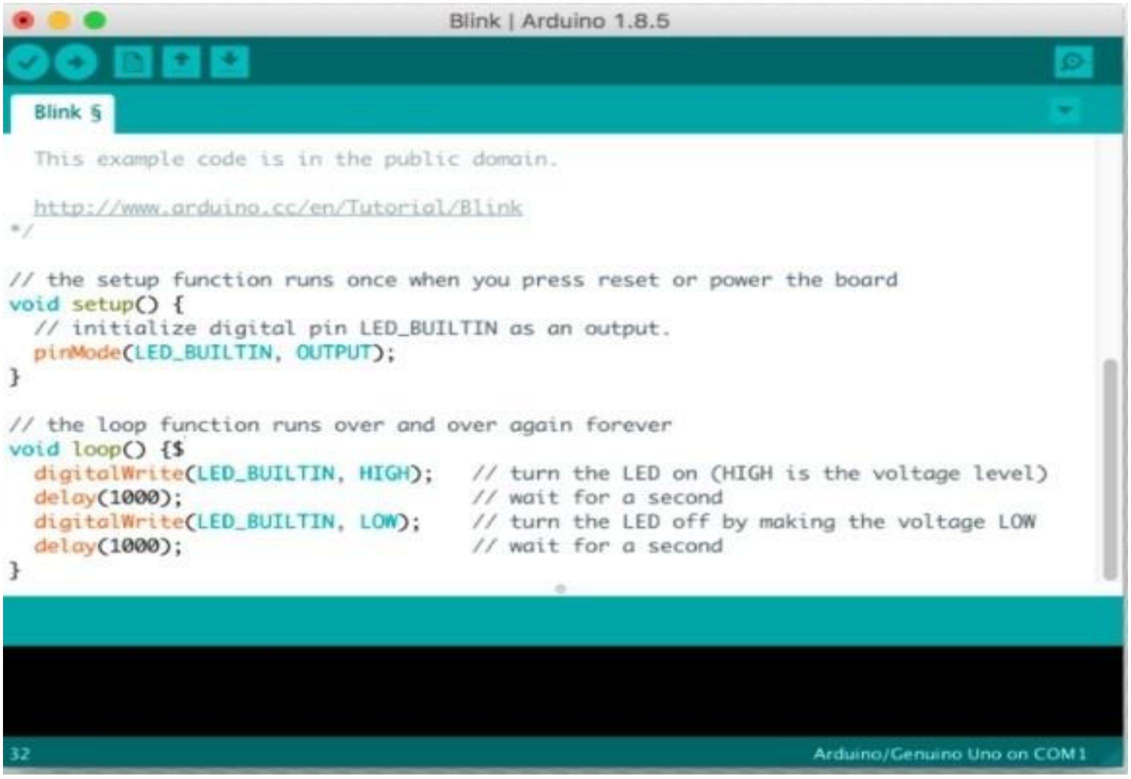


Fig:1.10 Source code written on IDE platform

1.3.2.2 PROTEUS SOFTWARE:

Through this software Schematic implementation of the embedded system will be done.

STEPS FOR DESIGNING SYSTEM ON PROTEUS DESIGN SUITE:

Step 1**:** Open ISIS software and select new design in File menu

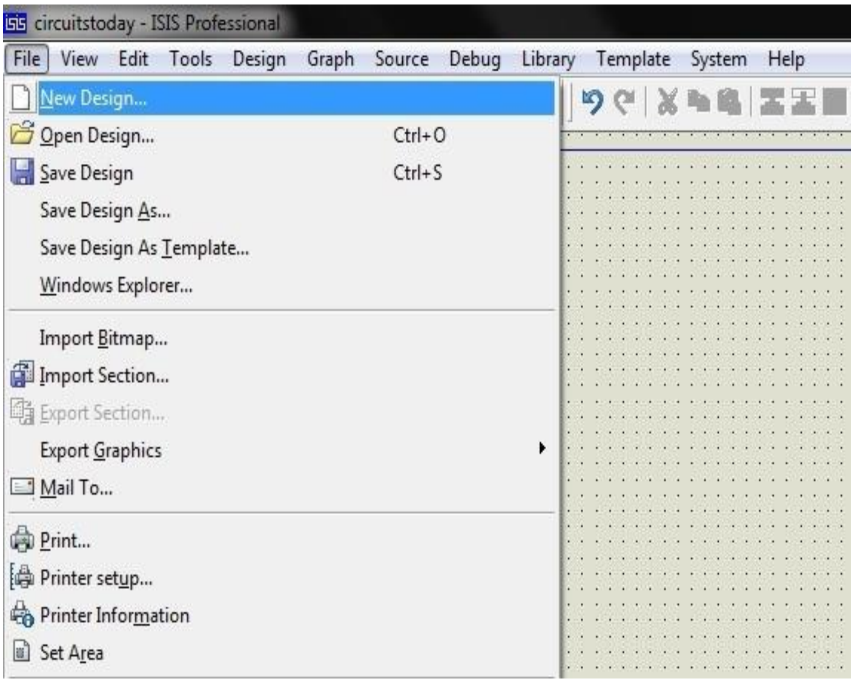


Fig:1.11 Simulation Procedure 1

Step 2**:** A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

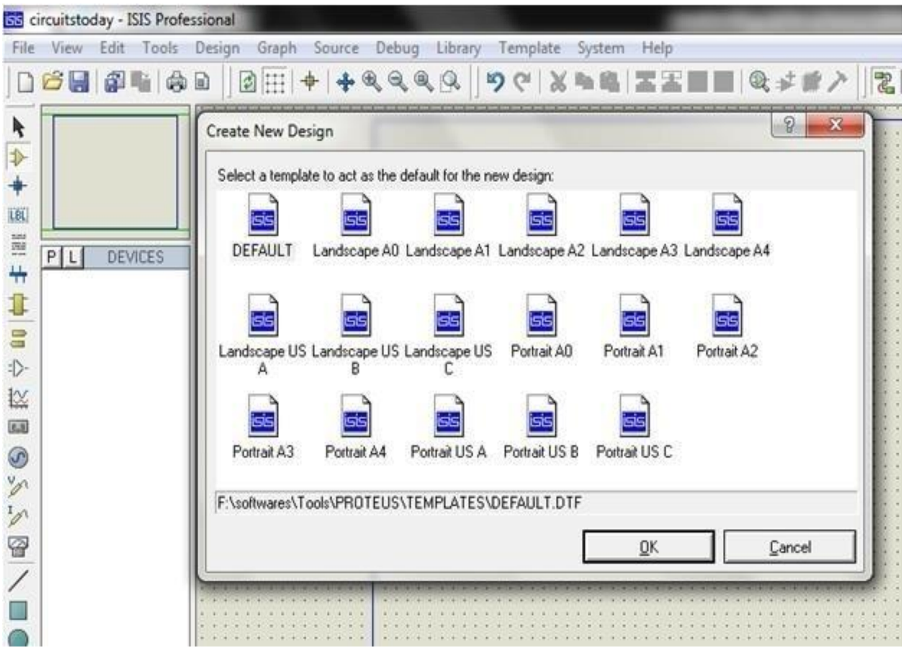


Fig:1.12 Simulation Procedure 2

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design.

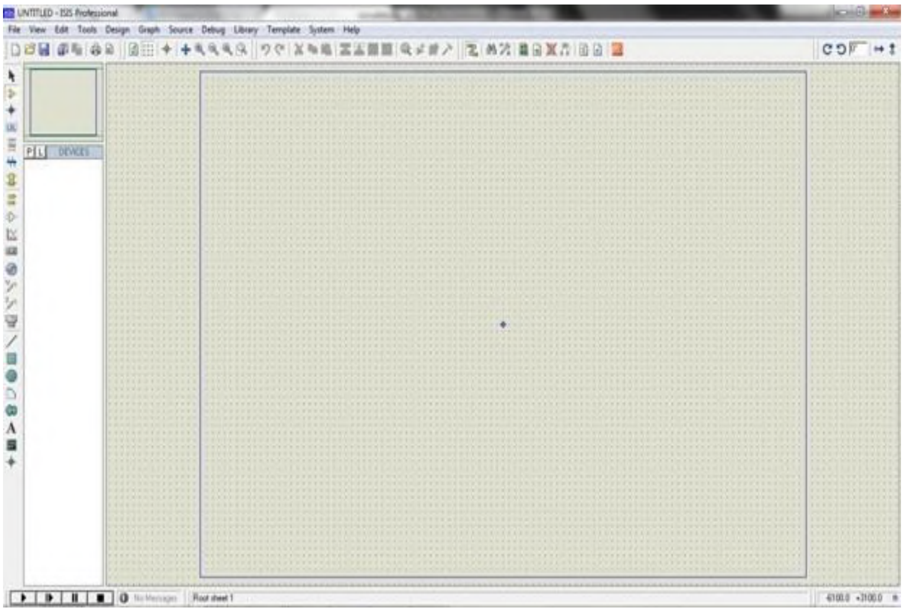


Fig:1.13 Simulation Procedure 3

Step 4: To Select components, Click on the component mode button.

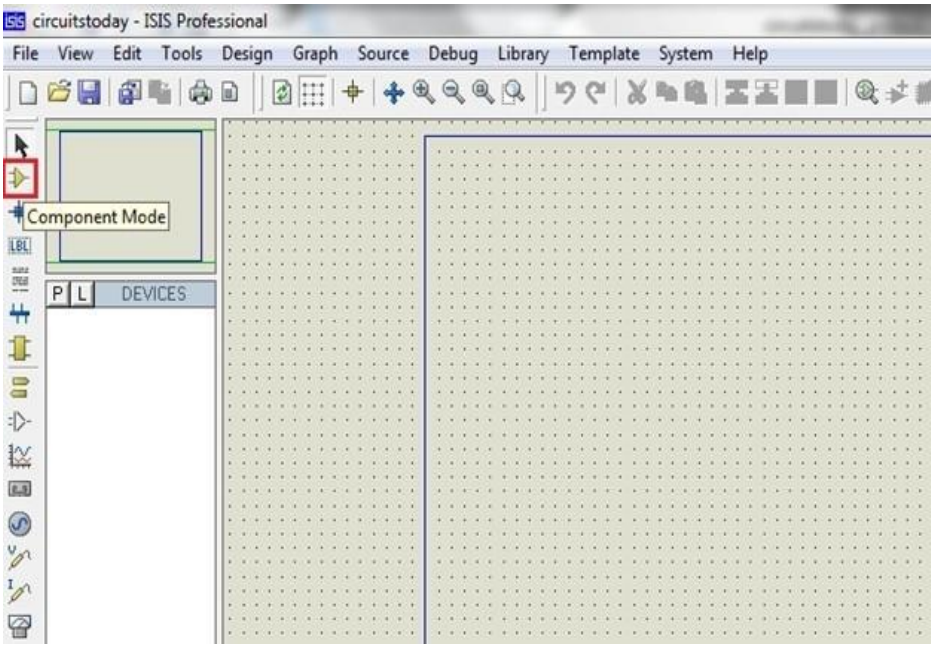


Fig:1.14 Simulation Procedure 4

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

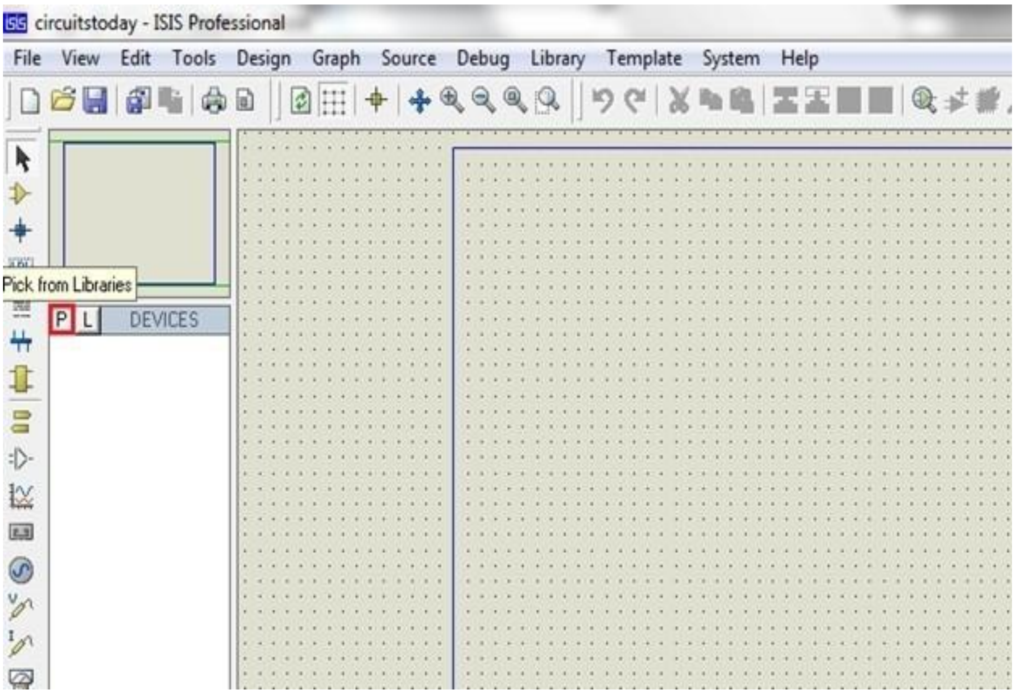
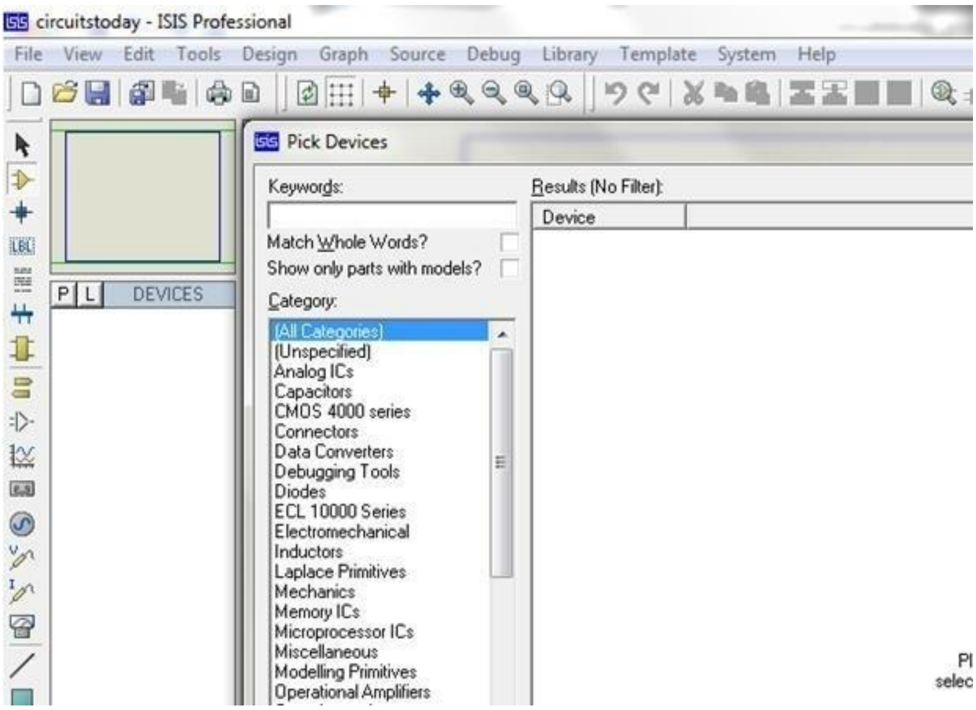


Fig:1.15 Simulation Procedure 5

Fig:1.16 Simulation Procedure 6

Step 6: Select the components from categories or type the part name in Keyword's text box.



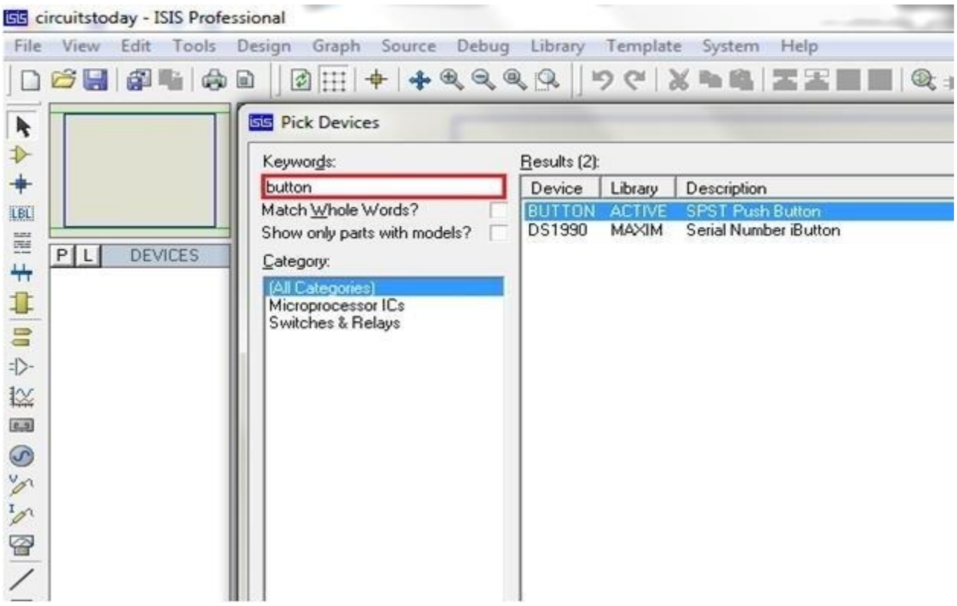


Fig:1.17 Simulation Procedure 7

Step 7**:** The selected components will appear in the devices list. Select the component and place it in the design sheet by left-cl.

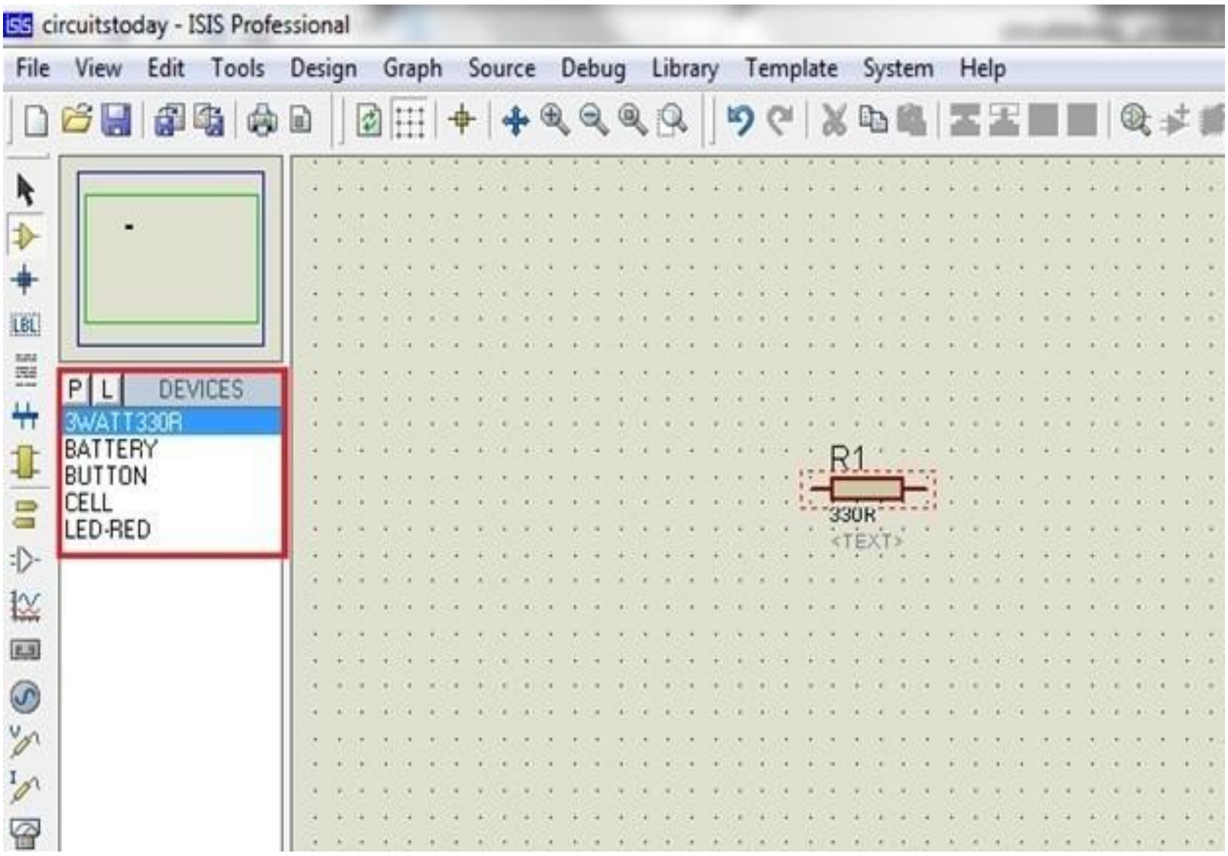


Fig:1.18 Simulation Procedure 8

Step 8: Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

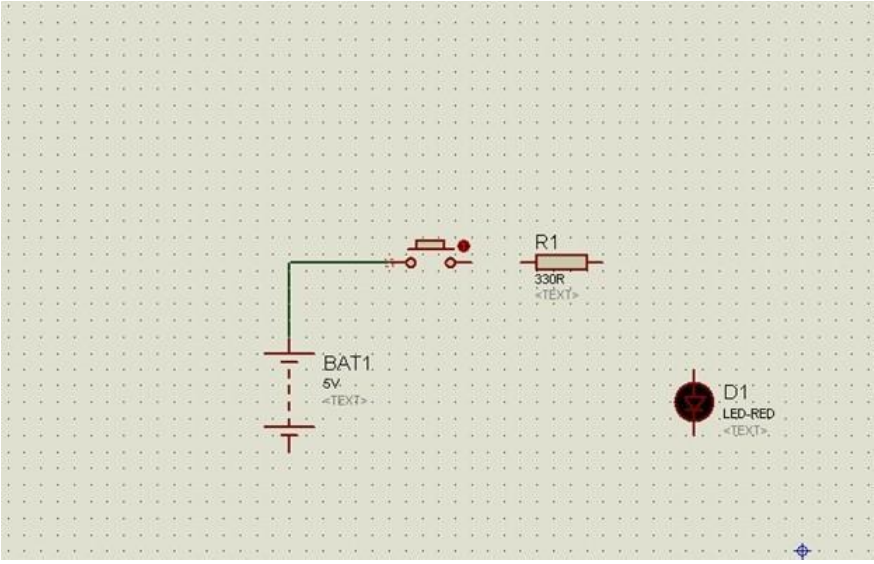


Fig:1.19 Simulation Procedure 9

Step 9: Double click on the component to edit the properties of the components and click on Ok.

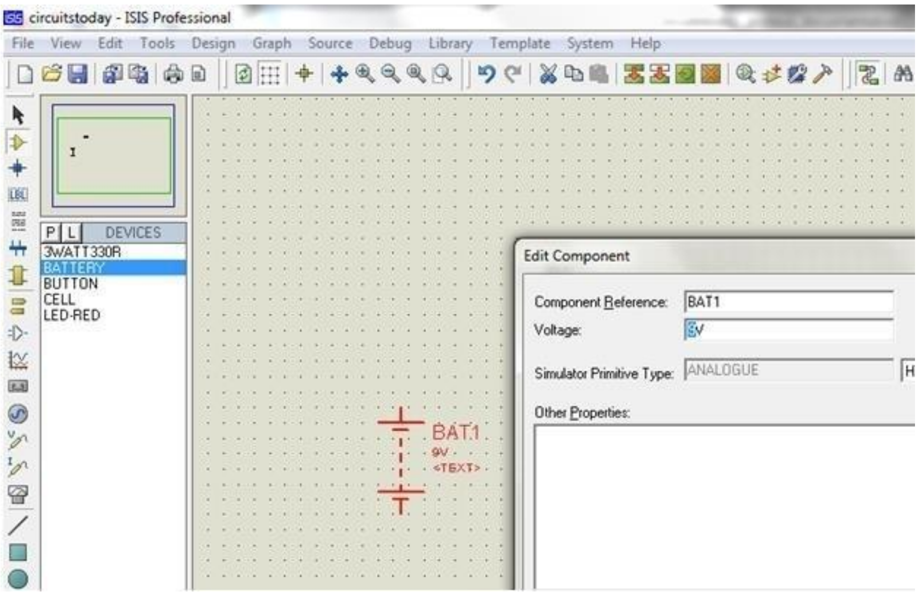


Fig.1.20 Simulation Procedure 10

Step 10: In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

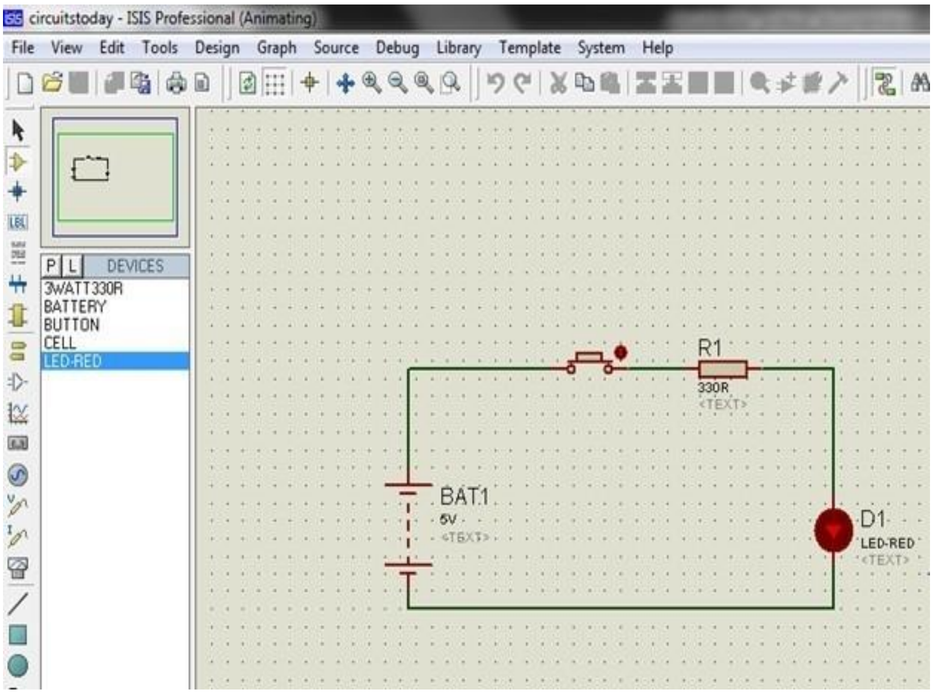


Fig:1.21 Simulation Procedure 11

Step 11: Simulation can be stepped, paused or stopped at any time.

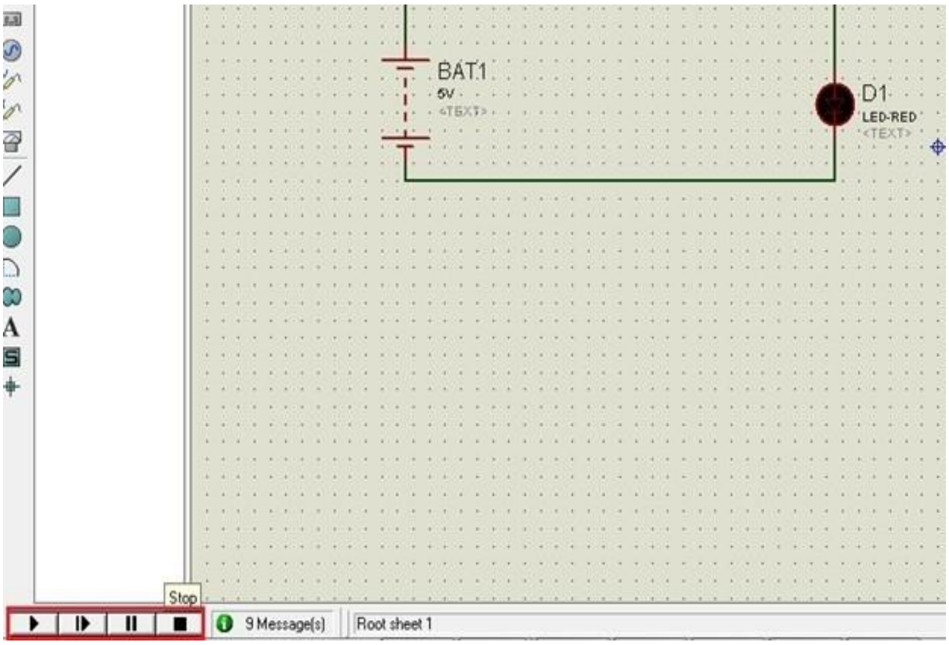


Fig:1.22 Simulation Procedure 12

1.4 PROCURMENT OF EQUIPMENT:

Fingerprints—the small ridges of skin on the fingertips which may enhance texture perception and grip friction—are unique to each person and have been used in identification for more than 100 years. Identification, the process of comparing a set of prints or a single print to others in a database, is used in many aspects of law enforcement and counterterrorism operations and investigations. Fingerprints may be compared to local and national databases for criminal history or to watch lists for border security. Latent prints from crime scenes are used in forensic investigations. Fingerprints can also be used to identify victims and for access control. Each application involves different processes for fingerprint capture and database searching and matching.

1.4.1 Components of Fingerprint Identification Systems

Fig:1.23 Finger print capture device

Fingerprint Capture the first step in fingerprint identification is obtaining an image of prints
interest. Fingerprints deliberately collected from an individual are called exemplar prints, and may include simultaneous four finger images or rolled prints, which capture an image of each side the fingernail to the other. Exemplar prints have historically been obtained using paper and ink, which must then be optically scanned for database input. Live scanners electronically capture and store a digital image, eliminating paper and ink. Digitized prints are transmitted electronically for AFIS searches. Most current live scanners are based on optical sensors. Emerging technologies include thermal, capacitive, and ultrasound sensors, and some have to use a striped pattern projected onto the fingertip to obtain a three-dimensional image.

Minutiae Encoding

The second step in fingerprint identification is to extract the fingerprints which are identifible characteristics to be used in a database search. Fingerprints can be sorted into general classification by

pattern type, such as arch, loop, or whorl, but minute details such as the ridge endings or burifications are unique to each individual. Software identifies
these minutiae and converts them into a system of numerical values. These minutiae templates are stored and used in searches instead of the whole image to reduce data and time requirements.

Recognition Software

An AFIS search involves the interaction of various databases where the minutiae are stored. Depending on the application, proprietary algorithms search segments of the database. For example, software may search the minutiae of two fingerprints of each subject, such as two index fingers, two thumbs, or a combination of finger and thumb. In other cases it may be necessary to search a database of all ten fingers. The simplest type of search is an identity verification used for access control or credentialing applications. Another identifier, such as a username, is submitted with a single fingerprint to specify who is attempting the authentication.

Certifications and Evaluations

A wide variety of fingerprint capture equipment and proprietary matching algorithms are commercially available. Interoperability is critical to the identification process and is achieved through standards and certification. The FBI has established image quality specifications for fingerprint images as well as data compression requirements for storage and transmission. The FBI certifies equipment compatible with the IAFIS, including live scanners, card scanners, and printers. Certified products are listed at https://www.fbibiospecs.org/IAFIS/Default.aspx. The National Institute of Standards and Technology (NIST) leads developments in automating fingerprint identification and conducts evaluations of hardware and software. Some examples are

Fast Capture, an initiative to improve technology to capture 10 rolledequivalent prints in less than 15 seconds in a portable, rugged system; the Minutiae Interoperability Exchange (MINEX), which tests the use of a standardized minutiae template, rather than entire fingerprint images, for exchange of data between 1:1 matching systems; and the Fingerprint Vendor Technology
Evaluation 2012, which assesses the performance of 1:N matching algorithms based on proprietary fingerprint templates using a database of millions of fingerprint sets, including live scan and scanned inked cards.

Standards

Two standards are currently used in FBI certification. The image quality specifications for interoperability with IAFIS for fingerprint scanners and printers are contained in Appendix F of IAFIS-DOC-01078-9.3 Criminal Justice Information Services Electronic Biometric Transmission Specification. Fingerprint verification using 1:1 matching is the focus of the standard PIV-071006.

Development of national and international standards for fingerprint technology is ongoing, and many other standards are available. For example, ANSI/INCITS 381-2004 specifies data compression for image storage and transmission, and ANSI/INCITS 378-2009 defines a fingerprint standard minutiae template.

1.5 ORGANISATION OF THE CHAPTERS

The Overall Overview of Documentation is:

1.5.1 INTRODUCTION

1.5.2 LITERATURE SURVEY

1.5.3 PROJECT DESIGN

1.5.4 PROJECT IMPLEMENTATION

1.5.5 CONCLUSION AND FUTURE ENHANCEMENT

CHAPTER 2

LITERATURE SURVEY

This chapter gives a brief overview of previous works done related to Fingerprint system.

2.0 INTRODUCTION:

The main idea of our project is to secure the places by using fingerprint sensor used to scan the finger of n the persons, the authorized people have to register their fingerprints to the system by using fingerprint sensor and each person is provided a particular ID which is stored in the data, Authorized people are the only eligible for vote, while the remaining people can't. For more security, the project is done along with buzzer indicator .

2.1 RELATED WORK:

Biometrics is a way used to recognize a person based on his physical nature. The fingerprint, iris, face, voice, etc. are the mainly used biometrics to recognize a person. There are two key functions for biometrics, first is one to one matching and other is one too many matchings. In one to many matching the biometric sample is compared with the already stored samples. In one to one matching, it compares with the previously stored sample. Biometric method results in a faster security, and more convenient method for user verification. Biometric method is better than password security. Fingerprint is unique for each individual so it can be used as a mark of signature, verification and authentication. Fingerprint is the biometric which is used in this project. Finger-print will be different for each individual. In this project, fingerprint is used for the authentication of the user and allows him to cast vote based on his fingerprint image. Finger-print matching can be divided into three types: correlationbased matching, minutiae-based matching, pattern-based (or image-based) matching. In correlationbased matching, two fingerprint images are superimposed and therefore the correlation between corresponding pixels is computed for various alignments. In minutiae-based matching, minutiae from the two fingerprints are extracted and stored in a two-dimensional plane as a set. This matching method consists of finding the alignment between the template and the input minutiae sets that result in the maximum number of minutiae pairings. In pattern-based (or image-based) matching method it compares with stored template and the candidate's fingerprint. This requires that the images to be aligned within the same orientation. To do this, the algorithm finds a central point within the fingerprint image and centers there on. In a pattern-based algorithm, the template contains the sort, size, and orientation of patterns within the aligned fingerprint image. Almost all the sectors are storing data digitally. To create digital India, most of the tasks are made through on-line. When the voting is made on-line, it helps the voters to vote from anywhere in the world. Thingspeak is one of the ways which helps in making voting on-line. Obtaining online result makes the system faster. Traditionally voting was done by marking with stamp casting vote for the corresponding candidate and then dropping the paper to a ballot box. To calculate the number of vote each vote must be calculated in each ballot box and then sum all the votes for each candidate and candidate who secured largest vote will be selected as the winner.

2.2 EXISTING METHODOLOGY:

Paper Ballots were used in India before 1997 for conducting the public elections. Even though this method gives uninterrupted voting, Votes captured in ballots can be stored for a very short period as the ink used in voting my discharge or ballot paper may lose its quality. Proper care has to be taken in maintaining these ballots to protect them from humidity, sunlight and other factors, which affect the ballot papers. Once the ballot is corrupted, we cannot recover the original data. After the election process, it takes more time and effort for counting the votes manually by checking each ballot paper. Paper is an Inflammable material accidentally I may catch fire in case all record will be lost and cannot be recovered and hence government has to spend extra money for conducting reelections. Bogus ballots can be made and in-numerous fake votes can be casted. Physically disabled peoples were facing difficulties in casting their votes, in those cases they need others help, but privacy while casting votes is vomited.

2.3 PROPOSED METHODOLOGY:

Fingerprint Based Voting Project is a application where the user is recognized by his finger pattern. Since the finger pattern of each human being is different, the voter can be easily authenticated. The system allow the voter to vote through his fingerprint. Finger print is used to uniquely identify the user. The finger print minutiae features are different for each human being. Finger print is used as a authentication of the voters. Voter can vote the candidate only once, the system will not allow the candidate to vote for the second time. The system will allow admin to add the candidate name and candidate photo who are nominated for the election. Admin only has the right to add candidate name and photo who are nominated. Admin will register the voters name by verifying voter. Admin will authenticate the user by verifying the user's identity proof and then admin will register the voter. The number of candidate added to the system by the admin will be automatically deleted after the completion of the election. Admin has to add the date when the election going to end. Once the user has got the user id and password from the admin the user can login and vote for the candidate who are nominated. The system will allow the user to vote for only one candidate. The system will allow the user to vote for one time for a particular election. Admin can add any number of candidates when the new election will be announced. Admin can view the election result by using the election id. Even user can view the election result.

CHAPTER 3

PROJECT DESIGN

3.0 OVERVEIW OF THE SYSTEM DESIGN

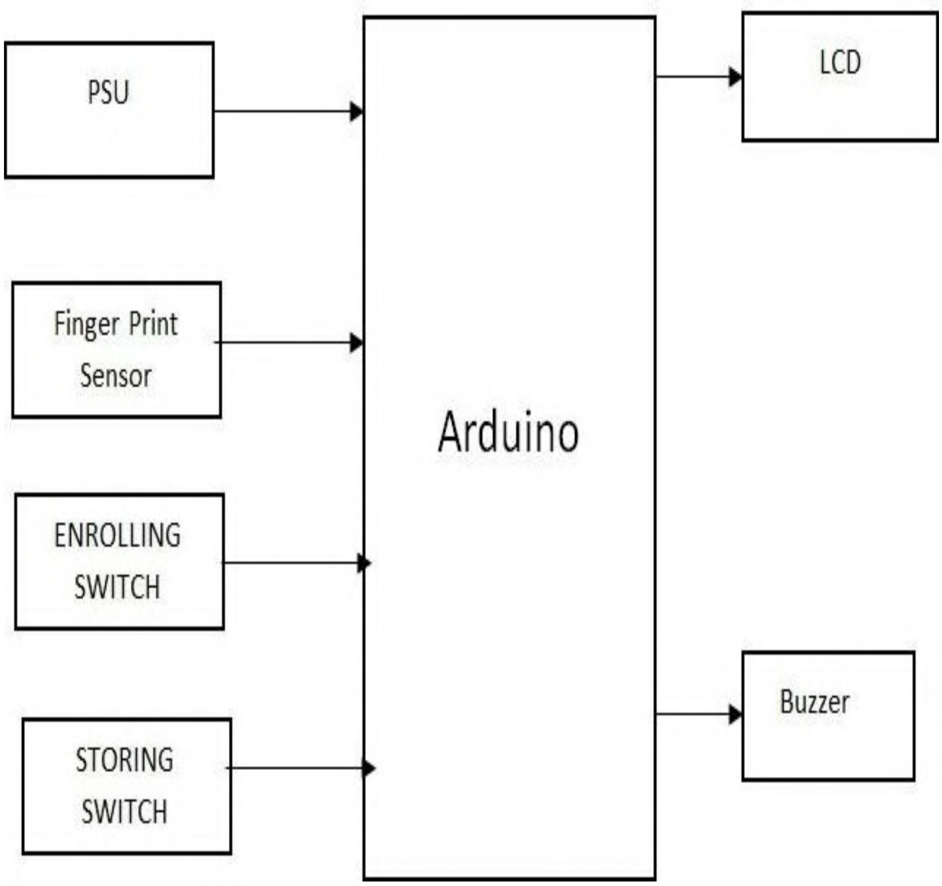


Fig:3.1 Block Diagram of finger print voting system

3.1 SCHEMATIC DIAGRAM:

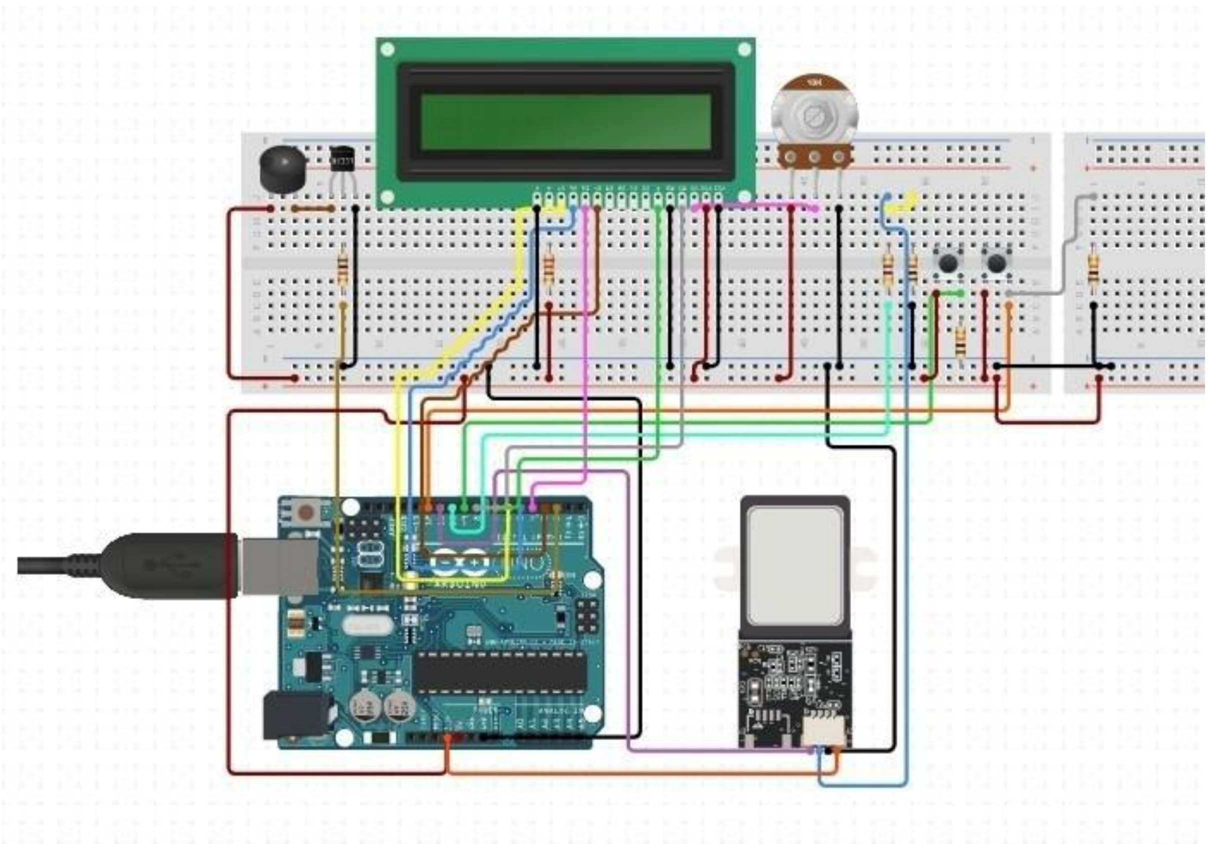


Fig:3.2 Simulation layout

The above schematic diagram of Finger Print Based Security System explains the interfacing section of each component with Arduino uno and input output modules

3.2 DEFINE THE MODULES:

3.2.1 FINGER PRINT MODULE:

Fingerprint Module consists of optical fingerprint sensor, high-speed DSP processor, highperformance fingerprint alignment algorithm, high-capacity FLASH chips and other hardware and software composition, stable performance, simple structure, with fingerprint entry, image processing, fingerprint matching, search and template storage and other functions. Fingerprint module has two interface TTL UART and USB2.0, USB2.0 interface are often connected to the computer; RS232 interface may be a TTL level, the default baud is 57600, can be changed, ask a communication protocol, microcontroller, like ARM, DSP and other serial devices with a connection, 3.3V- 5V microcontroller are often connected.

Fig 3.3 : Fingerprint module

Specifications:

Here's the specifications of the fingerprint sensor module we're using (you should check your sensor datasheet or the specifications provided by your supplier – they shouldn't be much different than these):

- Voltage supply: DC 3.6 to 6.0V
- Current supply: <120mA
- Backlight color: green
- Interface: UART
- Bad rate: 9600
- Safety level: five (from low to high: 1,2,3,4,5)
- False Accept Rate (FAR): <0.001% (security level 3)
- False Reject Rate (FRR): <1.0% (security level 3)
- Able to store 127 different fingerprints

3.2.2 LCD MODULE:

An LCD (Liquid Crystal Display) screen is an electronic display module and has a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. This LCD has two registers, namely, Command and Data.

Command register stores various commands given to the display. Data register stores data to be displayed. The process of controlling the display involves putting the data that form the image of what you want to display into the data registers, then putting instructions in the instruction register. In your arduino project Liquid Crystal Library simplifies this for you so you don't need to know the low-level instructions. Contrast of the display can be adjusted by adjusting the potentiometer to be connected across VEE pin.

3.3 MODULES FUNCTIONALITIES:

3.3.1 Arduino

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are :

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

3.3.1.1 Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header

Table:3.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16Mz	5V	16MHz	14	6	6	1	Native USB
Pro micro-3.3V/8Mz	5V	16MHz	14	6	6	1	Native USB
Lilypad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table:3.2 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table:3.3 Arduino boards based on ATMEGA2560 microcontroller

Operating Volt	Board Name	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
3.3V	Arduino Mega 2560 R3	84MHz	54	12	12	4	USB native

Table:3.4 Arduino boards based on AT91SAM3X8E microcontroller

3.3.1.2 Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common

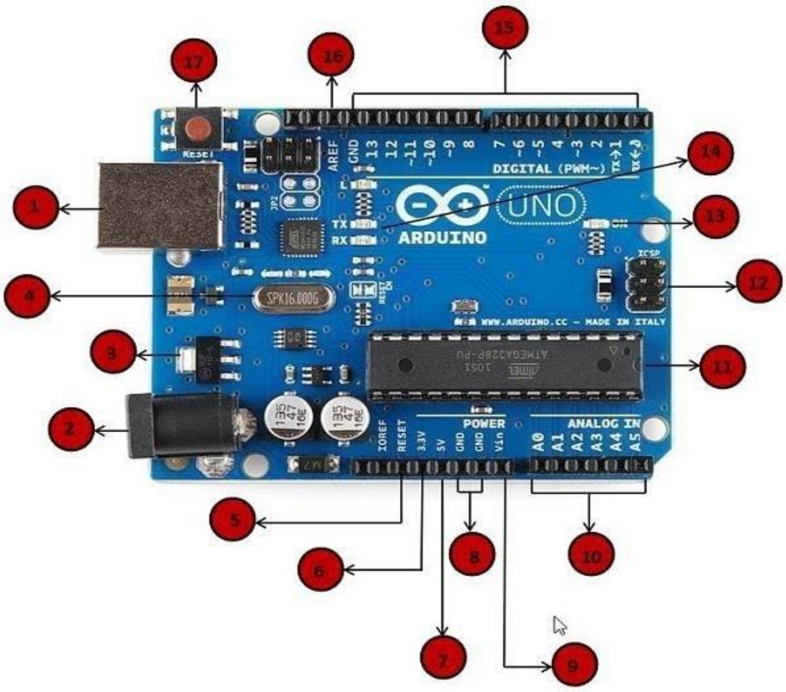


Fig:3.4 Board description of Arduino uno

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt Most of the components used with Arduino board works fine with 3.3 volt and 5 volts. GND (8) (Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.

10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

3.4 Liquid Crystal Display

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to a controller is an LCD display. Some of the most common LCDs connected to the controllers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines (RS RW D7 D6 D5 D6).

When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller

3.4.1 Description of 16x2:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all-Parallel Ports. It however doesn't show the use of the Status Port as an input. So, what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

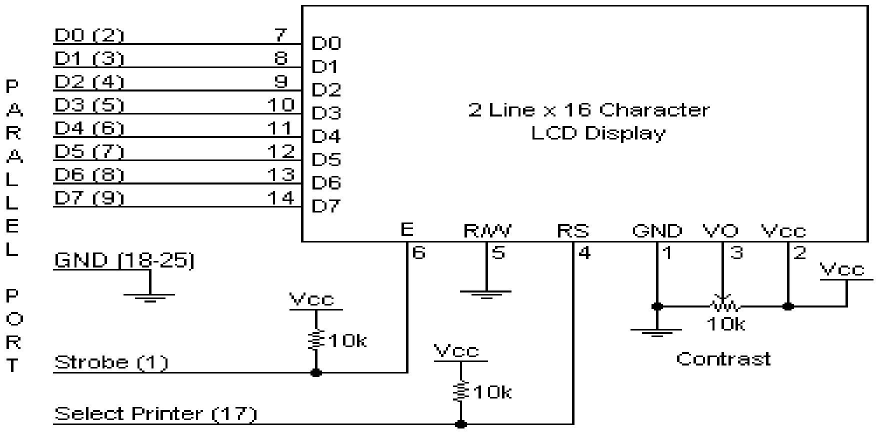


Fig 3.5: Schematic diagram

- Above is the quite simple schematic. The LCD panel's *Enable* and *Register Select* is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore, by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

- We make no effort to place the Data bus into reverse direction. Therefore, we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result, we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

- The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. You can use a bench power supply set to 5v or use an onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.

3.4.2 16 x 2 Alphanumeric LCD Module Features:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area.
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line.
- Can display 224 different symbols.
- Low power consumption (1 mA typical).
- Powerful command set and user-produced characters.
- TTL and CMOS compatible.
- Connector for standard 0.1-pitch pin headers.

Pin	Symbol	Level	Function
1	V_{SS}	-	Power, GND
2	V_{DD}	-	Power, 5V
3	Vo	-	Power, for LCD Drive
4	RS	H/L	Register Select Signal H: Data Input L: Instruction Input
5	R/W	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)
6	E	H, H->L	Enable

Table:3.5 16x2 LCD Features

3.4.3 FEATURES:

- 5 x 8 dots with cursor
- Built-in controller (KS 0066 or Equivalent)
- + 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)
- N.V. optional for + 3V power supply

Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are:

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

Table:3.6 Data Location in LCD

3.4.4 Figure Address locations for a 1x16 line LCD

Even limited to character-based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8,16,20,24,32 and 40 characters are all standard, in one-, two- and four-line versions.

Several different LC technologies exists. “supertwist” types, for example, offer Improved contrast and viewing angle over the older “twisted nematic” types. Some modules are available with back lighting, so that they can be viewed in dimly-lit conditions. The back lighting may be either “electro-luminescent”, requiring a high voltage inverter circuit, or simple LED illumination.

3.4.5 Pin Description In LCD:

Most LCDs with 1 controller have 14 Pins and LCDs with 2 controllers has 16 Pins (two pins are extra in both for back-light LED connections).

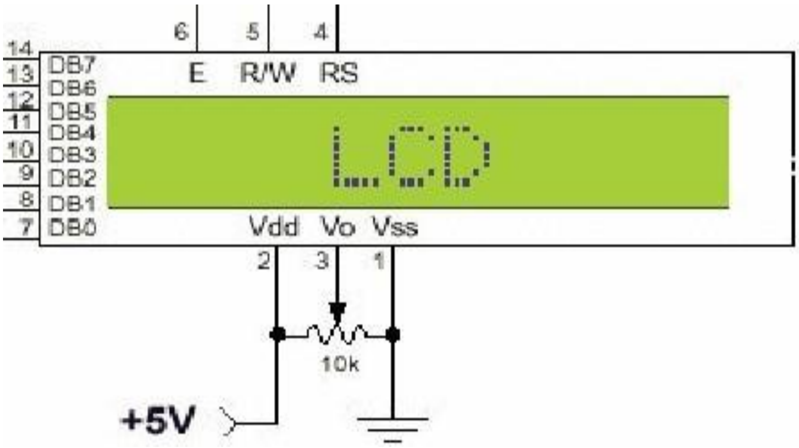


Fig:3.6 Pin Description of 1x16 lines

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Table 3.7 Pin specifications

3.4.6 CONTROL LINES:

EN: Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS: Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW: Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands, so RW will almost always be low. Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Logic status on control lines

- E - 0 Access to LCD disabled.
- 1 Access to LCD enabled.
- R/W - 0 Writing data to LCD.
- 1 Reading data from LCD.
- RS - 0 Instructions.
- 1 Character.

Writing data to the LCD

- Set R/W bit to low.
- Set RS bit to logic 0 or 1 (instruction or character).
- Set data to data lines (if it is writing).
- Set E line to high.
- Set E line to low.

Read data from data lines (if it is reading) on LCD

- Set R/W bit to high.
- Set RS bit to logic 0 or 1 (instruction or character).
- Set data to data lines (if it is writing).
- Set E line to high.
- Set E line to low.

Entering Text

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the d.i.l. switch pack with a of re-wiring is necessary.

LCD Commands

There are some present commands instructions in LCD, which we need to send to LCD through some microcontroller. Some important command instructions are given below

Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left
0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

Table:3.8 LCD Commands

3.5 Fingerprint Sensor

The fingerprint sensor can read different fingerprints and store in its own flash memory. The sensor can perform three functions namely Add (Enroll), Empty Database or search database and return the ID of stored fingerprint. Any of three functions can be called simply by making the pin low of the sensor or pressing onboard three switches. The response is either error or ok which is indicated by onboard LED. The response is also returned as single serial data byte.

The return byte is a valid ID or error code. The response byte is a single byte at 9600 bps thus making whole sensor very easy to use. We have provided indicating LEDs and function switch already so it’s ready to use when you receive it. Just give power and start using the sensor using onboard switches. Then you can move on making external application using these functions.

3.5.1 Inputs and Outputs of Sensor

Input: Two ways to trigger the function of fingerprint sensor

- Onboard switch: Add, Empty or Search.
- Make pin low from external microcontroller for 5ms as per function required to be executed.

Output: Two ways to monitor output response after a function is executed

- Onboard LEDs: ERROR or OK.
- Read byte after executing function.

3.5.2 Types of Sensor Function

There are namely three functions you can call for the fingerprint sensor. We will see each in brief.

Add (Enroll) Function: Adds a fingerprint to database and return a byte of newly added ID. Return values are from 0x00 to 0xFE. In case of error like no finger placed, return code is 0xFF. Here 0xFF means error executing function

Search Function: When a finger is put and search function is called, it returns a matching ID if found in its existing memory. Return values are from 0x00 to 0xFE. In case of error like no finger placed, return code is 0xFF. Here 0xFF means error executing function.

Empty Function: When you wish to empty all fingerprint, data stored on sensor you can use this function. After executing this function, you will get 0xCC as OK or 0xFF in case of error.

3.5.3 Fingerprint Recognition Sensor Module

The sensor is a solid-state fingerprint sensor that reliably captures fingerprint information. It is designed to integrate into devices for improved security and convenience. The sensor provides a reliable, quick and user-friendly alternative to passwords, PIN's and other forms of user authentication.

This fingerprint scanner is capable of gathering and storing unique finger prints. Simply hold your finger on the optical scanner, query the device over serial, and you will be issued a unique ID. Use that ID within your embedded system to determine access levels, time clocks, door locks, etc. Unit includes 4 pin connector cable to connect and read to controller. The outputs are TTL level serial data.

A biometric sensor, fingerprint sensor to be specific, also known as the fingerprint reader, is a fingerprint image capture device, the very front end of the biometric fingerprint identification/verification module. The fingerprint sensor captures the fingerprint images, matches the uniqueness of each print read by the sensor and compares it to the one stored in its module or local system database.

It consists of optical fingerprint sensor, high performance DSP processor and Flash. It boasts of functions such as fingerprint enrollment, fingerprint deletion, fingerprint verification, fingerprint upload, fingerprint download, etc.

3.5.4 Applications

- Computer peripherals – improves security and convenience
- Transportation systems – validation of operators, drivers and inspectors
- Medical equipment – authorization of operator or technician
- Physical access systems – approval for entry
- Kiosks and vending machines – confirmation of person receiving the selection
- Point of Sale terminals – authentication of tellers and cashiers

3.5.5 Features

- Rugged, solid-state optical fingerprint sensor
- High resolution 500 DPI imager
- Adapts to wet/dry fingers
- Simple Interfacing protocol

Fig:3.7 Fingerprint sensor

3.5.6 Specifications of Sensor

- Fingerprint enrollment time <250ms
- Fingerprint search time <1s（100 fingerprint, average value in test） Resolutions 500 DPI
- Security level 5, Adjustable Capacity 768 Templates
- FAR <0.0001%
- FRR <0.01%

- Power Supply 4.3V to 6V Working Current <80mA Peak Current <90mA Communication Interface TTL
- Communication Baud Rate 57600 bps Working Temperature -10 deg C to +40 deg C
- Working Humidity 40% RH to 85% TH (no dew) Module Dimensions 60x21x25 mm (LxWxH)

3.6 BUZZER:

3.6.1 Types of Buzzer

Magnetic Transducer

Magnetic transducers contain a magnetic circuit consisting of a iron core with a wound coil and a yoke plate, a permanent magnet and a vibrating diaphragm with a movable iron piece. The diaphragm is slightly pulled towards the top of the core by the magnet's magnetic field. When a positive AC signal is applied, the current flowing through the excitation coil produces a fluctuating magnetic field, which causes the diaphragm to vibrate up and down, thus vibrating air. Resonance amplifies vibration through resonator consisting of sound hole(s) and cavity and produces a loud sound.

Magnetic Buzzer (Sounder)

Buzzers like the TMB-series are magnetic audible signal devices with built-in oscillating circuits. The construction combines an oscillation circuit unit with a detection coil, a drive coil and a magnetic transducer. Transistors, resistors, diodes and other small devices act as circuit devices for driving sound generators. With the application of voltage, current flows to the drive coil on primary side and to the detection coil on the secondary side. The amplification circuit, including the transistor and the feedback circuit, causes vibration. The oscillation current excites the coil and the unit generates an AC magnetic field corresponding to an oscillation frequency. This AC magnetic field magnetizes the yoke comprising the magnetic circuit. The oscillation from the intermittent magnetization prompts the vibration diaphragm to vibrate up and down, generating buzzer sounds through the resonator.

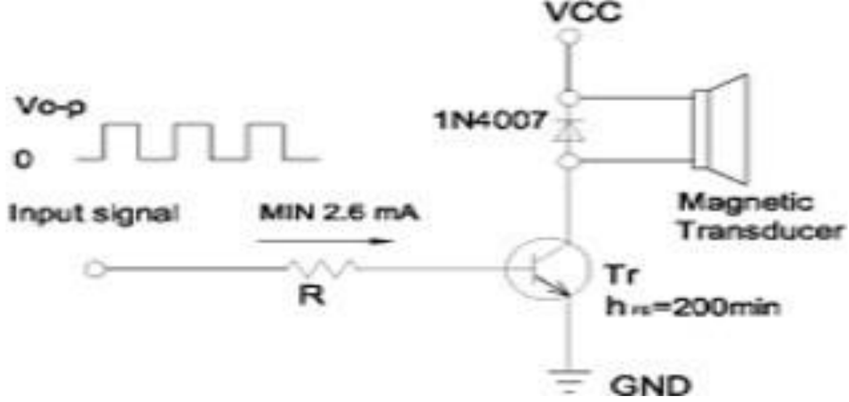


Fig:3.8 Driving Circuit for Magnetic Transducer

3.6.2 Specifications of buzzer:

- **Rated Voltage**: A magnetic buzzer is driven by 1/2 square waves (V o-p). **Operating Voltage**: For normal operating. But it is not guaranteed to make the minimum Sound Pressure Level (SPL) under the rated voltage.
- **Consumption Current**: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.
- **Direct Current Resistance**: The direct current resistance is measured by ammeter directly. **Sound Output**: The sound output is measured by decibel meter. Applying rated voltage and 1/2 square waves, and the distance of 10 cm.
- **Rated Frequency**: A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.
- **Operating Temp**: Keep working well between -30℃ and +70℃.
- **Driving methods**: AX series with built drive circuit will be the best choice when we cannot provide frequency signal to a buzzer, it only needs direct current.
- **Dimension**: Dimension affects frequency, small sizeresult in high frequency.
- **Voltage**: Depend on V o-p (1/2 square waves)
- **Fixed methods**: From the highest cost to the lowest- DIP, wires/ connector, SMD.

3.6.3 How to choose a buzzer

There are many different kinds of buzzer to choose, first we need to know a few parameters, such as voltage, current, drive method, dimension, mounting type, and the most important thing is how much SPL and frequency we want.

Operating voltage: Normally, the operating voltage for a magnetic buzzer is from 1.5V to 24V, for a piezo buzzer is from 3V to 220V. However, in order to get enough SPL, we suggest giving at least 9V to drive a piezo buzzer.

Consumption current: According to the different voltage, the consumption current of a magnetic buzzer is from dozens to hundreds of mill amperes; oppositely, the piezo type much more electricity, only needs a few mill amperes, and consumes three times current when the buzzer starts to work.

Driving method: Both magnetic and piezo buzzer have self-drive type to choose. Because of the internal set drive circuit, the self-drive buzzer can emit sound as long as connecting with the direct current. Due to the different work principle, the magnetic buzzer needs to be driven by 1/2 square waves, and the piezo buzzer need square waves to get better sound output.

Dimension: The dimension of the buzzer affects its SPL and the frequency, the dimension of the magnetic buzzer is from 7 mm to 25 mm; the piezo buzzer is from 12 mm to 50 mm, or even bigger.

Connecting way: Dip type, Wire type, SMD type, and screwed type for big piezo buzzer are usually seen.

Sound Pressure Level (SPL): Buzzer is usually tested the SPL at the distance of 10 cm, if distance double, the SPL will decay about 6 dB; oppositely, the SPL will increase 6 dB when the distance is shortened by one time. The SPL of the magnetic buzzer can reach to around 85 dB/ 10 cm; the piezo buzzer can be designed to emit very loud sound, for example, the common siren, are mostly made of silicon.

Fig:3.9 Buzzer

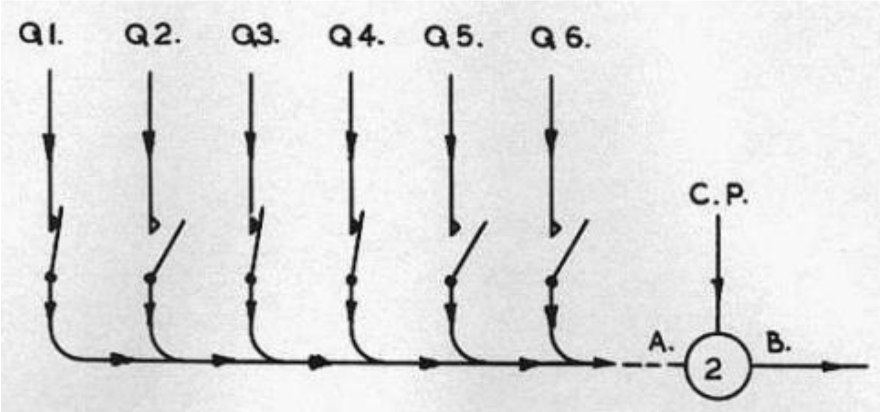


Fig:3.9 Pin Diagram of buzzer

3.7 Switches

A switch is an electrical component that can break an electrical circuit, interrupting the current or diverting it from one conductor to another. The most familiar form of switch is a manually operated electromechanical device with one or more sets of electrical contacts. Each set of contacts can be in one of two states: either 'closed' meaning the contacts are touching and electricity can flow between them, or 'open', meaning the contacts are separated and non-conducting.

Fig:3.10 Switches in a row

Switches with larger numbers of poles or throws can be described by replacing the "S" or "D" with a number or in some cases the letter "T" (for "triple"). In the rest of this article the terms *SPST*, SPDT and intermediate will be used to avoid the ambiguity in the use of the word "way".

3.7.1 Working of Switch

A pair of contacts is said to be "closed" when current can flow from one to the other. When the contacts are separated by an insulating air gap, an air space, they are said to be "open", and no current can flow at typical voltages.

Switches are classified according to the arrangement of their contacts in electronics. Electricians installing building wiring use different nomenclature, such as "one-way", "two- way", "three-way" and "four-way" switches

In a push-button type switch, in which the contacts remain in one state unless actuated, the contacts can either be normally open (abbreviated "no." or "no") until closed by operation of the switch, or normally closed ("n.c. or "nc") and opened by the switch action. A switch with both types of contact is called a changeover switch. These may be "make-before- break" which momentarily connect both circuits, or may be "break-before-make" which interrupts one circuit before closing the other

3.7.2 APPLICATIONS

- They are used for the given instructions to the required device like robot, driving the motor, etc.
- Switches can arrange in the matrix from or in the required from according to the application.
- **Reset circuit of the controller, motor also has the switches for its particular functioning.**

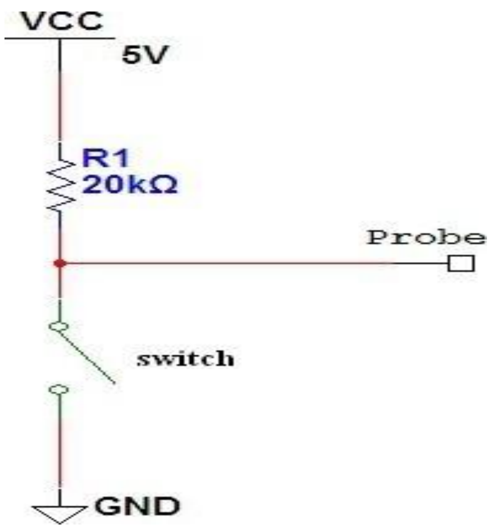


Fig:3.11 Circuit of Switch

From the above circuit it explains the working of the switch when supply is given to the circuit the resistor takes the current and oppose to the certain extend until the switch is pressed. These switches can be connected parallel or serial in required manner.

We can have resistor, capacitor, light resistor diode etc. as the combination of the switch. For example, w the reset circuit in which we the resistor and capacitor as the combination.

CHAPTER 4
PROJECT IMPLEMENTATION

4.0 IMPLEMENTATION STAGES:

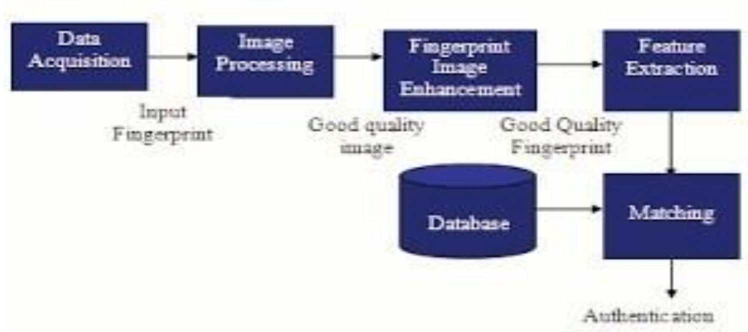


Fig:4.1 Implementation of Finger print voting system

1. Algorithm of Fingerprint Based Voting System:

Step 1: Start

Step 2: Users can register by giving their fingerprints. We use a microcontroller to store them.

Step 3: Scan your finger

Step 4: If match found goto step 5 else return to step3 Step 5: user checks their details

Step 6: cast your vote

Step 7: Press push-button for a party from the party list

Step 8: Vote counted

Step 9: Stop

2. Fingerprint Identification and Matching:

Fundamentally, we utilize advanced imaging innovation in securing, putting away, and dissecting the unique mark information.

1. Acquiring Images:

In the image identification process, the first step is to obtain an image as, withot it, we cannot perform further steps. To acquire an image of a fingerprint, the person needs to place their finger on the sensor. Once the voter places their finger,on one end of the prism, the total internal reflection occurs through which we can capture the image using the image sensor and lens from another end of the prism. But the image extracted in this step is unprocessed.

The position and placement of the finger play a prominent role in the process of capturing an impression. For intensifying the total internal reflection and capture a fingerprint of good quality with the image sensor, we need to make sure that the finger is placed correctly on the module.

2. Storing the images:

The unprocessed image acquired in the previous step is now processed using image segmentation.

Image Segmentation: The captured images may contain some redundant data and noise along with required data, so we use image segmentation in which we divide the image into many segments called pixels to remove the irrelevant data. To ease the process of image analysis, we use image segmentation.

We use the normalization of an image to get even pixels. Once the pixels are uniform, it results in the formation of an image, and then to reduce or remove the noise present in it, we use the Gabor filter. The thresholding technique is implemented on the filtered image to change it into a binary image, then we compare threshold values and pixel values, if the value is higher than the threshold value then we set the pixel value to 1 else 0. Next, to remove some pixels from the foreground, we implement the thinning process. Finally, all the segments form a single image.

3. Analyzing the Image:

Finally, it shows to whom the user cast the vote.

Using image analysis, we can retrieve all related data from the image for further use. Mostly some electrical machines are used to collect the required data. According to this project, the electrical machine is a fingerprint module through which we capture the fingerprint. The retrieved data from the image can be compared with data stored in different storage devices for identification, authentication, etc.

4.1 WORKING FLOW OF THE PROJECT:

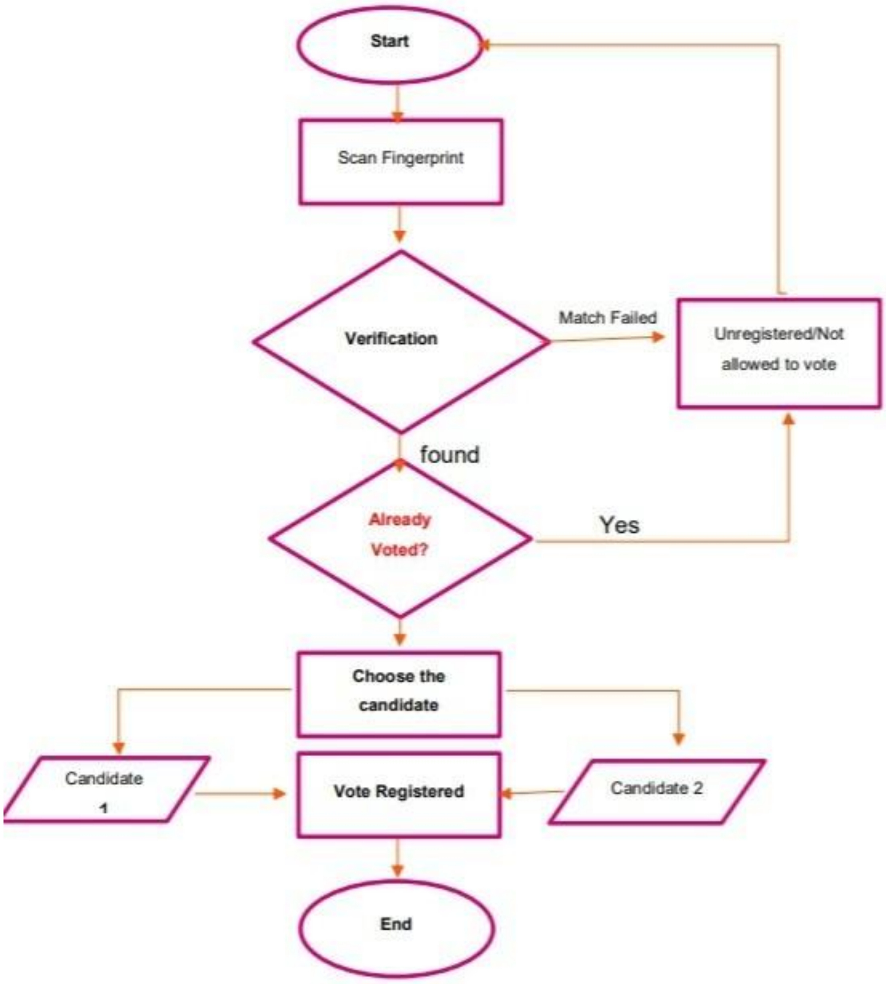


Fig:4.2 Working flow of Finger Print Based Voting System

4.2 EXPERIMENTAL RESULTS:

The Hardware setup gives us a brief idea to how the project ”Fingerprint voting system” works. The proposed system is implemented. There are mainly two units in this system, one for verification and other for voting. Arduino UNO was programmed using Arduino IDE. In the verification unit there are three scenarios- voting for first time, voting more than once, mismatch in fingerprint. If user attempts to vote for the first time, his fingerprint is compared with data in the database, if a match occurs, he can cast vote and message “Authenticated. Proceed” displays on the serial monitor of the arduino. If an authenticated user tries to cast vote more than once then a buzzer sound will be produced and “User not authorized” message displays on serial monitor. If the fingerprint of a person is not available in the database, he can’t cast his vote.Finally the authorized officer gets the summary of voting.

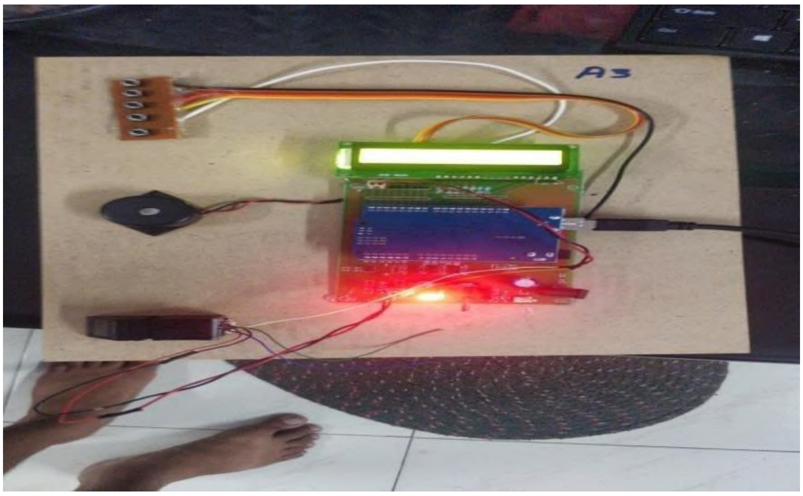


Fig 4.3:Overall design

4.2.1 INTERPRETING EXPERIMENTAL RESULTS:

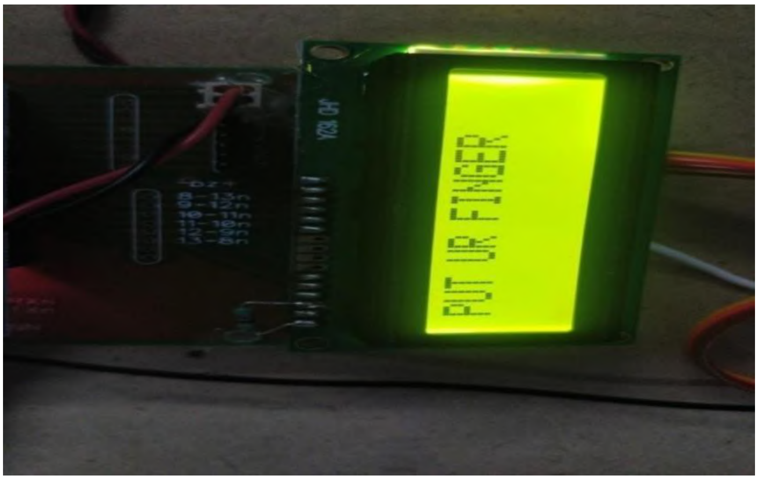


Fig:4.4 Displays Commands in LCD

Fig 4.5 Finger print Scanner

Fig:4.6 Displayed on LCD “User not authorized”

Fig:4.7 It shows the result obtained after voting

4.3 ADVANTAGES AND LIMITATIONS

ADVANTAGES

Some advantages of using this system are discussed below:

- Safe: It is an utmost important that the voting process be secure and no one should be able to tamper with the result before, during and after voting process. Since the system is not connected to internet, no online external influence can occur. Also, it is very easy for security personnel to secure the device if needed, since it is compact, and light compared to many ballot boxes.
- Cost: The conventional paper voting system used in Nepal is very expensive because of papers and printings, transportation, staff expenses and it takes several days to count the votes. On the other hand, the biometric fingerprint system is cheap, compact and can store any amount of data with proper upgrade.
- Accuracy and Reliability: The ink used to mark people who have voted already can be erased with modern technologies but the fingerprint voting system erases the chance of multiple voting and it is very precise. Also, there is a very slim chance of errors happening from electronic system so, people can rely on results to be accurate.
- Time Saving: It takes weeks to count the votes by election personnel sometimes. This time frame is not peaceful for that area so it is ideal to get result as soon as possible. The manual vote counting technique can be seen . With electronic device election results can be produced is minutes rather than days or weeks

LIMITATIONS

- The collection of fingerprints of all voters is little difficult.
- The personal details of the voter does not verifying in this system other than fingerprint.
- Only fingerprints of the voters is identified instead of identifying the person.

CHAPTER 5

APPLICATIONS

5.0 APPLICATIONS

This project can be used as a voting machine to prevent rigging, during the elections in the polling booths.

- Fast track voting which could be used in small scale elections, like resident welfare association, “panchayat” level election and other society level elections, where results can be instantaneous.

- It could also be used to conduct opinion polls during annual shareholders meeting.

- It could also be used to conduct general assembly elections where number of candidates are less than or equal to eight in the current situation, on a small scale basis

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.0 CONCLUSION

The project "Fingerprint Based Voting Machine" was mainly intended to develop a fingerprint based advanced Electronic Voting Machine (EVM) which helps in free and fair way of conducting elections which are basis for democratic country like India. Fingerprint Based Voting Machine is designed to make the procedure of voting easier and more convenient as it is a modified system. It has proved to be very advantageous in providing security EVM is capable of saving considerable printing stationery and transport of large volumes of electoral material. It is easy to transport, store, and maintain. It completely rules out the chance of invalid votes. In total, the complete system (including all the hardware components and software routines) is working as per the initial specifications and requirements of our project. So certain aspects of the system can be modified as operational experience is gained with it. As the users work with the system, they develop various new ideas for the development and enhancement of the project. The proposed system has been designed and implemented successfully using a PIC microcontroller, which was shown to be superior over the existing Electronic Voting Machine. The proposed system has the benefit of using a biometric authentication and controls the process of voting avoiding unnecessary things like rigging, ballot papers, casings etc.

6.1 FUTURE SCOPE

- Number of active candidates may increase.
- It could be interfaced with the computer and results can be displayed quickly and easily or the result can be stored in the central server whose data can be stored in the backend servers.
- It could be interfaced with the printer so that the hardcopy of the data can be taken.
- It could make the results available at any corner of the world in minimum time.

REFERENCES

[1] A. K. Sharma, A. Raghuwanshi and V. K. Sharma, "Biometric System- A Review," International Journal of Computer Science and Information Technology, vol. 6, p. 1, 2015.

[2]B.Aleksandra, "handle/10024/44684," 2012. [Online]. Available: https://www.theseus.fi/handle/10024/44684.

[3] "technology," 2013. [Online]. Available: http://www.nitgen.com/eng/technology/technology_02.html. [Accessed 18 10 2019].

[4] "Adafruit," Adafruit, [Online]. Available: https://www.adafruit.com/product/751?gclid=CjwKEAj. [Accessed 06 11 2019].

[5] "arduino," 11 11 2019. [Online]. Available: https://store.arduino.cc/.

[6] B. Sigdel, "Thesis," Metropolia UAS, Helsinki, 2017.

[7] "software," 11 11 2019. [Online]. Available: https://www.arduino.cc/.

[8] "components101," [Online]. Available: https://components101.com/microcontrollers/arduino-uno. [Accessed 14 11 2019].

[9] Dejan, "lcd tutorials," [Online]. Available: https://howtomechatronics.com/tutorials/arduino/lcd-tutorial/. [Accessed 08 10 2019].

[10] U. Hussaini, "technobyte," [Online]. Available: https://www.technobyte.org. [Accessed 17 11 2019].

[11] "wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/USB. [Accessed 17 11 2019].

[12] 19 11 2019. [Online]. Available: https://en.wikipedia.org/wiki/Electoral_fraud. [Accessed 20 11 2019].

[13] P. Pokharel, "Study on the Election Campaign Finance of Local, Provincial, and Federal Elections in Nepal," Pradip Pokharel, Kathmandu, 2017.

[14] "Kathmandu Post," 14 05 2017. [Online]. Available: https://kathmandupost.com/national/2017/05/15/ballot-papers-confusing

APPENDICES

```
/*
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2,3);*/
#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);
char buff[200],k=0;
char res[130];
unsigned char enroll[12]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X03,0X01,0X00,0X05}; // ok
unsigned char generate_ch[13]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X04,0x02,0X01,0X00,0X08}; //ok
unsigned char generate_ch1[13]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X04,0x02,0X02,0X00,0X09}; //ok
unsigned char un_cmd[12]={0xef,0x01,0xff,0xff,0xff,0xff, 0x01,0x00,0x03,0x05,0x00,0x09 };
unsigned char store[12]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X06,0X06,0X02,0x00};        //ok
unsigned char identify[17]={0xef,0x01,0xff,0xff,0xff,0xff,0x01,0x00,0x08,0x1b,0x01,0x00,0x00,0x01,0x01,0x00,0x27}; void
serialFlush(){
while(Serial.available() > 0)
{ char t = Serial.read();
}
}
int fpenroll(char); int
fpsearch();
int s1=2,s2=3,s3=4,s4=5,s5=6,buz=7;
void setup() {
char ret;
pinMode(s1, INPUT_PULLUP);
pinMode(s2, INPUT_PULLUP);
pinMode(s3, INPUT_PULLUP);
pinMode(s4, INPUT_PULLUP);
pinMode(s5, INPUT_PULLUP);
pinMode(buz, OUTPUT
digitalWrite(buz,HIGH);
Serial.begin(9600); lcd.begin(16,2);
lcd.clear();lcd.setCursor(0, 0);lcd.print("FINGERPRINT BASED VOTING MACHINE");
delay(2000);
}
int err =0;
int idk = 0,eid=0;
int trs=0,bjp=0,result=0;
 int count1=0,count2=0,count3=0;
void loop()
{
  lcd.clear();lcd.setCursor(0, 0);lcd.print("PUT UR FINGER"); delay(1000);
if(digitalRead(s1) == 0)
```

```
{
//Serial.println("Enrolling");
lcd.clear();lcd.setCursor(0, 0);lcd.print("ENROLLING..");
if(fpenroll(eid) == -1)
{
// Serial.print("Enroll failed:");Serial.print(err);Serial.println("");
err=0;
lcd.clear();lcd.setCursor(0, 0);lcd.print("ENROLL FAILED");
digitalWrite(buz,LOW);delay(1000);digitalWrite(buz,HIGH);
}
else
{
lcd.clear();lcd.setCursor(0, 0);lcd.print("ENROLLED:");lcd.print((int)eid);
//Serial.print("Enroll Success to id:");Serial.print((int)eid);Serial.println("");
//Serial.print("*E");Serial.print((int)eid);Serial.println("#");
eid++;
}
delay(2000);
// lcd.clear();lcd.setCursor(0, 0);lcd.print("SELECT OPTION");
}
if(digitalRead(s2) == 0)//identify
{
lcd.clear();lcd.setCursor(0, 0);lcd.print("SEARCHING.."); idk =
fpsearch();
if(idk == -1)
{
err=0;lcd.clear();lcd.setCursor(0, 0);lcd.print("UNAUTHOISED ");
lcd.setCursor(0, 1);lcd.print("PERSON");

digitalWrite(buz,LOW);
delay(2000);
digitalWrite(buz,HIGH);

}
else
{
lcd.clear();lcd.setCursor(0, 0);lcd.print("IDENTIFIED:");lcd.print((int)idk);
delay(2000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("ENTER YOUR VOTE ...");
delay(2000);
if( count1 >= 1)
{
lcd.clear();lcd.setCursor(0, 0);lcd.print("DOUBLE VOTE TRYING !!! ");
digitalWrite(buz,LOW);
delay(400);
```

```
digitalWrite(buz,HIGH);

}
if(idk == 0 && count1== 0)
{

if(digitalRead(s3) == 0)//trs
{
trs++;
lcd.clear();lcd.setCursor(0, 0);lcd.print("YOUR VOTE FOR TRS !!! ");delay(2000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("THANK YOU FOR YOUR VOTE!!! ");delay(2000);
}
if(digitalRead(s4) == 0)//BJP
{
bjp++;
lcd.clear();lcd.setCursor(0, 0);lcd.print("YOUR VOTE FOR BJP !!! ");delay(2000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("THANK YOU FOR YOUR VOTE!!! ");delay(2000);
}
count1++;
}
if( count2 >= 1)
{
lcd.clear();lcd.setCursor(0, 0);lcd.print("DOUBLE VOTE TRYING !!! ");
digitalWrite(buz,LOW);
delay(400);
digitalWrite(buz,HIGH);

}
if(idk == 1 && count2== 0)
{

if(digitalRead(s3) == 0)//trs
{
trs++;
lcd.clear();lcd.setCursor(0, 0);lcd.print("YOUR VOTE FOR TRS !!! ");delay(2000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("THANK YOU FOR YOUR VOTE!!! ");delay(2000);
}
if(digitalRead(s4) == 0)//BJP
{
bjp++;
lcd.clear();lcd.setCursor(0, 0);lcd.print("YOUR VOTE FOR BJP !!! ");delay(2000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("THANK YOU FOR YOUR VOTE!!! ");delay(2000);
}
count2++;
}
```

```
if( count3 >= 1)
{
lcd.clear();lcd.setCursor(0, 0);lcd.print("DOUBLE VOTE TRYING !!! ");
digitalWrite(buz,LOW);
delay(400);
digitalWrite(buz,HIGH);

}
if(idk == 2 && count3== 0)
{

if(digitalRead(s3) == 0)//trs
{
trs++;
lcd.clear();lcd.setCursor(0, 0);lcd.print("YOUR VOTE FOR TRS !!! ");delay(2000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("THANK YOU FOR YOUR VOTE!!! ");delay(2000);
}
if(digitalRead(s4) == 0)//BJP
{
bjp++;
lcd.clear();lcd.setCursor(0, 0);lcd.print("YOUR VOTE FOR BJP !!! ");delay(2000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("THANK YOU FOR YOUR VOTE!!! ");delay(2000);
}
count3++;
}
delay(2000);
}
if(digitalRead(s5) == 0)//result
{
lcd.clear();lcd.setCursor(0, 0);lcd.print("RESULTS !!! ");
lcd.setCursor(0, 1);lcd.print("TRS:");lcd.print(trs);
lcd.setCursor(8, 1);lcd.print("BJP:");lcd.print(bjp);
delay(4000);

}
}

int ct=0;
char dummy=0x0f; int
fpenroll(char id)
{

serialFlush();
        //enroll buffer send 12 bytes
```

```
Serial.write(enroll[i]); res[9]
= 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=1;return -1;}

//generate ch buffer
for(int i =0;i<13;i++)
Serial.write(generate_ch[i]); res[9]
= 1;
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=2;return -1;}

//enroll buffer send 12 bytes
for(int i =0;i<12;i++)
Serial.write(enroll[i]); res[9]
= 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=3;return -1;}

//generate ch1 buffer
for(int i =0;i<13;i++)
Serial.write(generate_ch1[i]); res[9]
= 1;
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=4;return -1;}

//uncmd buffer send 12 bytes
for(int i =0;i<12;i++)
Serial.write(un_cmd[i]); res[9]
= 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=5;return -1;}

//store buffer send 12 bytes
```

```
Serial.write(store[i]); dummy =
0x0f+id; Serial.write((uint8_t)id);
Serial.write((uint8_t)0x00);
Serial.write((uint8_t)dummy);
res[9] = 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){return id;}
else{err=6;return -1;}
}
int fpsearch()
{
ct=0; serialFlush();
//enroll buffer send 12 bytes
for(int i =0;i<12;i++)
Serial.write(enroll[i]); res[9]
= 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=1;return
//generate ch buffer
for(int i =0;i<13;i++)
Serial.write(generate_ch[i]); res[9]
= 1;
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=2;return -1;}
//enroll buffer send 12 bytes
for(int i =0;i<17;i++)
Serial.write(identify[i]); res[9]
= 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){return (int)res[11];}
else{err=1;return -1;}

}
```

A

PROJECT REPORT

ON

SATELLITE IMAGE ENHANCEMENT FOR SMALL PARTICLE OBSERVATION USING DECORRELATION STRETCHER

Submitted by

A.SURYATEJA REDDY	**(17K81A0404)**
CH.SIIDDARDHA REDDI	**(17K81A0412)**
S.VAMSHI	**(17K81A0445)**

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

In

ELETRONICS AND COMMUNICATION ENGINEERING

Under the Guidance of

Mr.VENKANNA MOOD

M.Tech(Ph.D)

Associate professor

DEPARTMENT OF ELETRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN’S ENGINEERING COLLEGE
(An Autonomous Institute)
Dhulapally,secunderbad-500 100

JUNE 2021

BONAFIDE CERTIFICATE

This is to certify that the project entitled "**Satellite Enhancement For Small Particle Observation Using Decorrelation Stretcher"**, is being submitted by **1.SuryaTeja reddy(17K81A0404), 2.Siiddardha reddi (17K81A0412), 3.Vamshi(17K81A0445)** in partial fulfillment of the requirement for the award of the degree of BACHELOR OF TECHNOLOGY "Electronics and Communication Engineering " is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

< Signature>	Head of the Department
Mr .VENKANNA MOOD	Dr.B.HARIKRISHNA
Department of E.C.E.	Department of E.C.E.

Internal Examiner External Examiner

Place:
Date:

LASYA INFOTECH

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **BSS. PRABHATH** WITH ROLL NO.**17K81A0407, CH. ROHITH KIRAN** WITH ROLL NO.**17K81A0408**, **C.S SAI TEJA** WITH ROLL NO.**17K81A0411**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**DOWNLINK USER SELECTION FOR MASSIVE-MIMO-OFDM-IM SYSTEMS USING ZF PRECODING**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We , the students of **“Bachelor of Technology in Department of Electronics and Communication Engineering”,** session: 2017-2021, **St. Martin’s Engineering College**, hereby declare that the project work Entitled **“Satellite Enhancement For Small Particle Observation Using Decorrelation Stretcher”** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. The result embodied in this project report has not been submitted in any university for award of any degree.

A.SURYATEJA REDDY 17K81A0404
CH.SIIDDARDHA REDDI 17K81A0412
S.VAMSHI 17K81A0445

ACKNOWLEDGEMENT

We sincerely thank the management of the college **ST . MARTIN'S ENGINEERING COLLEGE** for providing required facilities during our project work.

We derive our great pleasure in expressing our sincere gratitude and respect to our Principal **Dr.SANTHOSH KUMAR PATRA** for his timely suggestions, which helped us to complete the project successfully.

We would like to express our thanks and immense gratitude to Dr.B.HARIKRISHNA, Professor, Head of the Department, ECE for his consistent encouragement during the Progress of this project.

Our deep sense of gratitude to **Mr .VENKANNA MOOD**, Associate Professor ,our Project Mentor and Internal Guide, for the stimulating guidance and encouragement that we have received from him throughout the project.

We take it as a privilege to thank our project coordinator **Dr. A.ANAND** department of ECE for the ideas that lead to complete the project Work and we also thank him for the continuous guidance , support and unfailing patience, throughout the course of this work.

We would also like to thank all the staff of Department of ECE who has been helpful to complete the project successfully.

Abstract

In this research work, we have improved the image visual appearance using image decorrelation stretcher techniques. Landsat 8 images are used in this research work to show the improved visual presentation for the images. Landsat 8 images contain 11 different spectrum bands. The visual appearance of the images produces some specific details in terms of appearance, which were not present in the original image. Through the image decorrelation stretcher technique, the image quality is improved and enhanced.

LIST OF CONTENT

LIST OF FIGURES

NAME OF THE FIGURE **PAGE NO.**

LIST OF ABBREVATIONS

1. CDF	Cumulative Distributive Function
2. GLCM	Grey Level Cooccurrence Matrix
3. JPEG	Joint Photographic Experts Group
4. JFIF	JPEG File Interchange Format
5. EXIF	Exchangeable Image File Format
6. TIFF	Tagged Image File Format
7. PNG	Portable Network Graphics
8. GIF	Graphics Interchange Format
9. CGM	Computer Graphics Metafile
10. SVG	Scalable Vector Graphics
11. GCM	Grayscale Co-occurrence Matrix

Chapter 1

INTRODUCTION

The earth is associate aquatic planet and the maximum amount as eightieth of its surface is roofed by water.. Moreover, there is a strong interest in knowing what lies in underwater. Present days, an image of deep waters has a scope to large investigation to explore the underwater for sea floor expedition and navigation. Enthusiasm of underwater imaging includes the inspection of plants, seabed exploration, the search for wrecks up and to the exploration of natural resources. There were several issues faced by the human in the underwater, if he dives deep into the ocean and stay there for a long time to perform experimentation. [1]. Due to the above reasons, unmanned remote vehicles are used to sea floor exploration.

A. Historical Development

Underwater image quality improvement approaches present a path to magnify the object recognition in underwater surrounding. A heap of research started for the upgradation of image visual quality, but a little amount of work has been carried out in this area. In the deep waters, image quality is degrades due to poor illumination conditions and the light properties differ in water compared to air.[2]. There were several parameters which decreases the quality of an image in underground waters. So inorder to remove all these effects

there are several techniques has been implemented and practiced.

B. Need for Pre process

Initially processing is necessary for deep water images due to their poor quality during acquisition. Necessity for pre-processing of deep water images[1] are discussed below:

(i) Quality of images taken from deep water is deteriorated due to light ray attributes like scattering and absorption of light.

(ii) Specificity of surroundings such as lighting inequalities, water torridness, and blue complexion is more or less influential when vehicles move.

(iii) Video or image captured from deep waters like unknown rigid scene, and the depth of the scene and low light sensitivity due to Marine snow etc.

TRADITIONAL TECHNIQUES FOR IMAGE ENHANCEMENT

There are several techniques which are used very frequently for processing the image to improve the visual quality. Some of them are as follows:

(i) Contrast Stretching

(ii) Adaptive Histogram Equalization

A. Contrast stretching

The contrast stretching is a method to transform high intense region of image into more brighter and less intense region into more darker by using a predefined transformation function T(r) [2]. Generally, the underwater images will have less grey values. There are 256 grey values. '0' indicates black and '255' indicates white. In this method the current grey value of the image is stretched towards 255 i.e., from black to white, pixel by pixel. That means the contrast of the image is stretched, so that the quality of the image is improved for better vision.

For example:

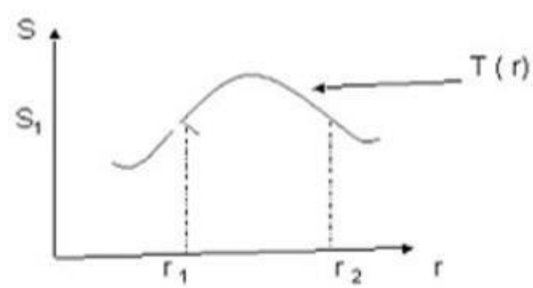


Fig.1. Two different gray levels look same

Here two different thresholds are considered for the entire image and the values between them are stretched to the maximum extent, so that the contrast increases. And more over by this method the entire global image contrast is enhanced.

Fig.2. (a) Raw image (b) Enhanced image

But the disadvantage here is that the transformation function is not unique. Depending on the application the suitable transformation function is chosen.

B. Adaptive histogram equalization

Adaptive histogram equalization is a PC based image processing technique which is used to improve the quality of image properties like contrast. It is similar to contrast stretching method but with a slight difference. It computes several intensities of specific gray value, each corresponding to a distinct portion of an image, and with the help of them intensities are rearranged by applying a suitable transformation function. For example, a simple transformation function such as each pixel transformed based on the histogram of a square surrounding the pixel [3]. Existing values will be mapped to new values keeping actual number of intensities in the resulting image equal or less than the original number of intensities. The transformation function applied on the histogram is proportional to the cumulative distributive function (CDF) of pixel values in the neighbourhood. Therefore it suits for enhancing the local details and enhancing the edge information of each region of an image.

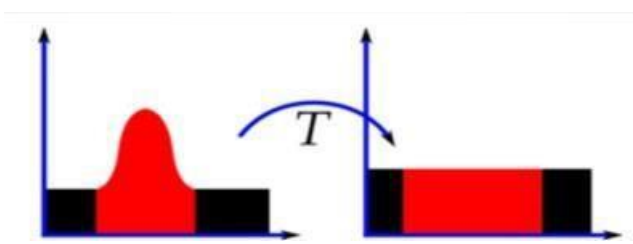


Fig.3. Histograms of an image before and after histogram equalization

Histogram equalization is a technique for changing the overall pixel intensities based on transformation function and contrast of an image. Histogram equalization is an effective technique which will benefit for the images with extreme contrast values. The limitation of this technique highlights the unwanted noise present in the background of an image and lead to loss in the information signal. It results in undesired effects in the resultant images [4].

Fig.4. (a) Raw image **(b) Enhanced image**

Here the noise in relatively homogeneous regions of the image are amplified which results in poor SNR. And also only the local objects of the image are enhanced and the background is left unenhanced.

Decorrelation can be understood as the process through which autocorrealtion within the signal and cross-correlation within the set of signals or images can be removed. During this process, all the other details of the image are conserved. There are many image improvement algorithms like Karhunen–Loève transform and discrete cosine transform (D.C.T.) . Decorrelation stretcher technique is applied in many fields like image processing, neuroscience, artificial neural network, cryptography and hardware number random generator. In simple words, decorrelation stretching can be understood as an artificial image contrast enhancement technique through which image contrast can be improved. The images on which these improvements can be implemented are digital images in communication and medicines (D.I.C.O.M.) images , satellite images , ultraviolet (U.V.) images, multispectral images etc.,.The purpose of performing decorrelation stretcher in an 8-bit image are as follows.

- We are changing the image pixel value distribution over the entire image with pixel values in the range of 0-255.
- These pixel values are utilized on the full range in a line fashion.
- Decorrelation Stretching is performed only on principal components of the image, not on the entire image.

Decorrelation Stretcher is used to improve the image quality to a great extend. The detailed review of the application of the Decorrelation Stretcher is as follows. Uji et al. used the Decorrelation Stretcher technique to improve the image quality and color of the images obtained from the "TRCNW8F Plus Mydriatic/Non-Mydriatic Retinal Camera" for the eye defect purpose. Their research concluded that the Decorrelation Stretcher has the ability to improve the image quality in color images. P. Sinha et al. used the Decorrelation Stretcher technique to enhance the visible image quality for sedimentary rock available in the Earth and the Mars surface. Thus the Decorrelation Stretcher technique can also be used for the image enhancement for the satellites and surface outside the Earth surface. G. Ch. Miliaresis used the Decorrelation Stretcher technique to enhance the visual appearance of the images having annual precipitation range on a 1 km grid for the state of "California, Nevada, Arizona, and Utah." A. Jellouli used the Decorrelation Stretcher technique to enhance the visual appearance of the Landsat 8 O.L.I. Images to enhance spatial, spectral, and radiometric characteristics of these images. E. Gürbüz used the Decorrelation Stretcher technique to improve the image quality for the multispectral image using ASTER data for the Turkey region by studying the vapourization of the minerals. An R Gillespie et al. used the image Decorrelation Stretcher technique on three different images, which are highly correlated with each other. They performed two different types of stretching on the models discussed as Decorrelation Stretch and saturation intensity stretching on the images to obtain superior results.

Image enhancement is performed to provide sufficient information that is hidden from human visual perception. There are several image enhancement techniques; one of the most popular image enhancement techniques is the grey level cooccurrence matrix (G.L.C. M) technique . In this technique, the changes developed in the images are quantified, and later for the visual image interpretation, a grey level image is created. The purpose of this image quantification and change detection represents the number of changes developed in the image. This techniques is used in medical image compression, satellite image analysis texture classification purposes etc

Chapter 2

LITERATURE SURVEY

There are many image improvement algorithms like Karhunen–Loève transform and discrete cosine transform (D.C.T.). Decorrelation stretcher technique is applied in many fields like image processing, neuroscience, artificial neural network, cryptography and hardware number random generator. In simple words, decorrelation stretching can be understood as an artificial image contrast enhancement technique through which image contrast can be improved. The images on which these improvements can be implemented are digital images in communication and medicines (D.I.C.O.M.) images , satellite images , ultraviolet (U.V.) images, multispectral images Uji et al. [7] used the Decorrelation Stretcher technique to improve the image quality and color of the images obtained from the "TRCNW8F Plus Mydriatic/Non-Mydriatic Retinal Camera" for the eye defect purpose.

Their research concluded that the Decorrelation Stretcher has the ability to improve the image quality in color images. P. Sinha et al. used the Decorrelation Stretcher technique to enhance the visible image quality for sedimentary rock available in the Earth and the Mars surface. Thus the Decorrelation Stretcher technique can also be used for the image enhancement for the satellites and surface outside the Earth surface. G. Ch. Miliaresis used the Decorrelation Stretcher technique to enhance the visual appearance of the images having annual precipitation range on a 1 km grid for the state of "California, Nevada, Arizona, and Utah." A. Jellouli used the Decorrelation Stretcher technique to enhance the visual appearance of the Landsat 8 O.L.I. Images to enhance spatial, spectral, and radiometric characteristics of these images. E. Gürbüz used the Decorrelation Stretcher technique to improve the image quality for the multispectral image using ASTER data for the Turkey region by studying the vapourization of the minerals. An R Gillespie et al. used the image Decorrelation Stretcher technique on three different images, which are highly correlated with each other

CHAPTER 3

INTRODUCTION TO IMAGE PROCESSING

INTRODUCTION

3.1.IMAGE:

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Fig 1 General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

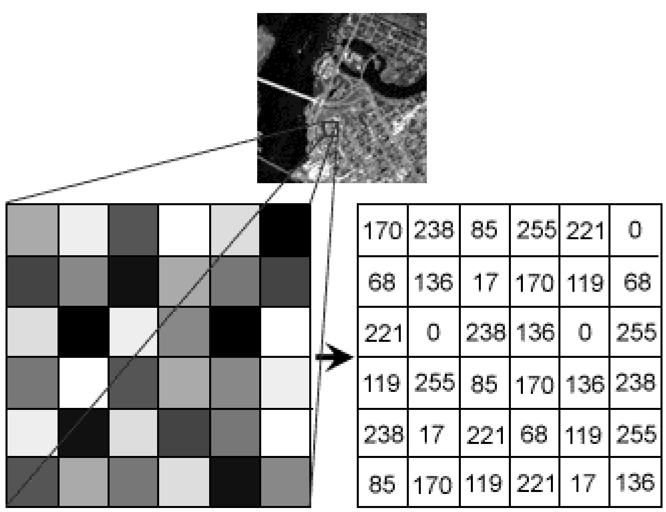


Fig 1.1 Image pixel

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
Transparency	Red	Green	Blue

Fig1.2 Transparency image

3.2 IMAGE FILE SIZES:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

3.3 IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

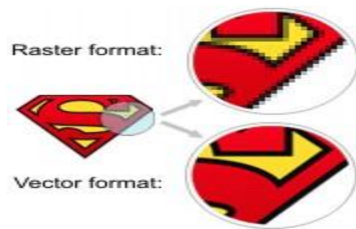


Fig1.3 Resolution image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

3.3.1 RASTER FORMATS:

These formats store images as bitmaps (also known as pixmaps)

- **JPEG/JFIF:**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

- **EXIF:**

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

- **TIFF:**

  The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

- **PNG:**

  The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

- **GIF:**

  GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

- **BMP:**

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

3.3.2 VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM:**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

- **SVG:**

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

3.4 IMAGE PROCESSING:

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

3.5 FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:

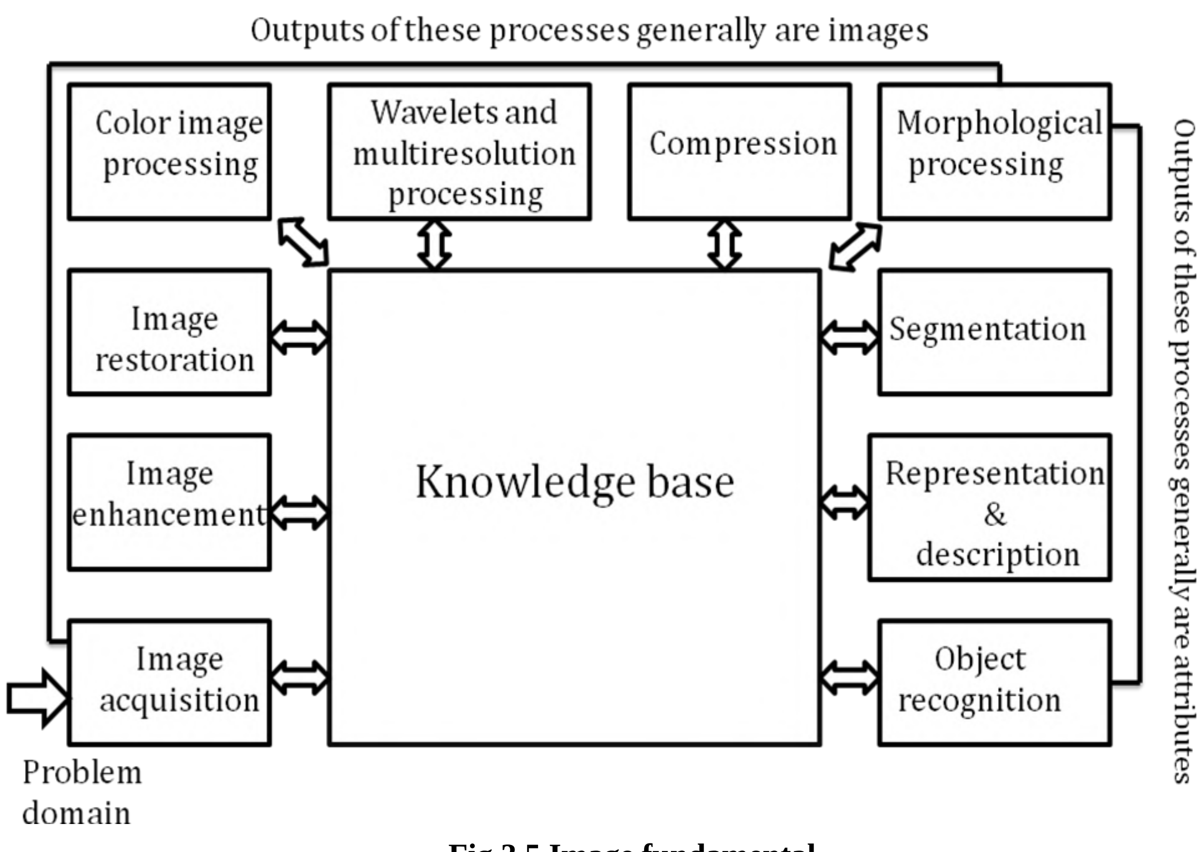


Fig 3.5 Image fundamental

3.5.1 Image Acquisition:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Fig 3.5.1 Digital camera image

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Fig 1.5.2 digital camera cell

3.5.2 Image Enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because "it looks better." It is important to keep in mind that enhancement is a very subjective area of image processing.

Fig 3.5.3 Image enhancement

3.5.3 Image restoration:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Fig 3.5.4 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, where as removal of image blur by applying a deblurring function is considered a restoration technique.

3.5.4 Color image processing:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Fig 3.5.5 Color & Gray scale image

3.5.5 Wavelets and multiresolution processing:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform based image processing since the late1950's, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

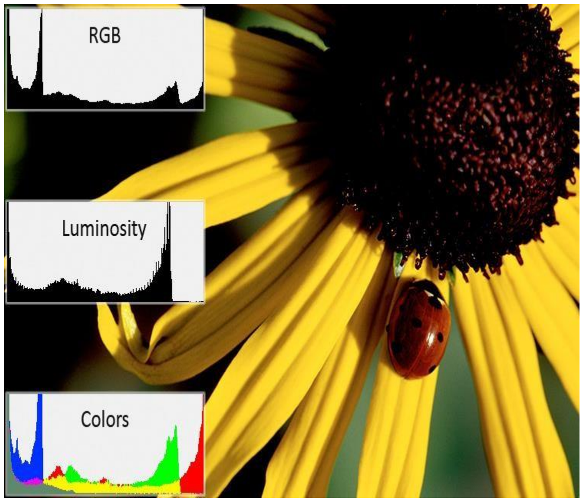


Fig 3.5.6 rgb histogram image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called **Multiresolution** theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

3.5.6 Compression:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

3.5.7 Morphological processing:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Fig 3.5.7 blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

3.5.8 Segmentation:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

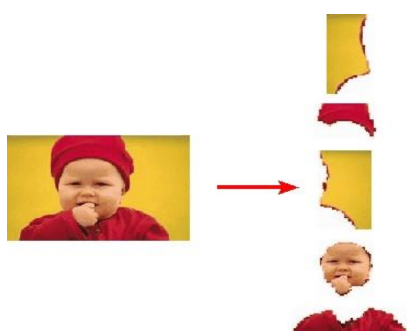


Fig 3.5.8 Image segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

3.5.9 Representation and description:

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

3.5.10 Object recognition:

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

3.5.11 Knowledgebase:

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an inter related to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

3.6 COMPONENTS OF AN IMAGE PROCESSING SYSTEM:

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

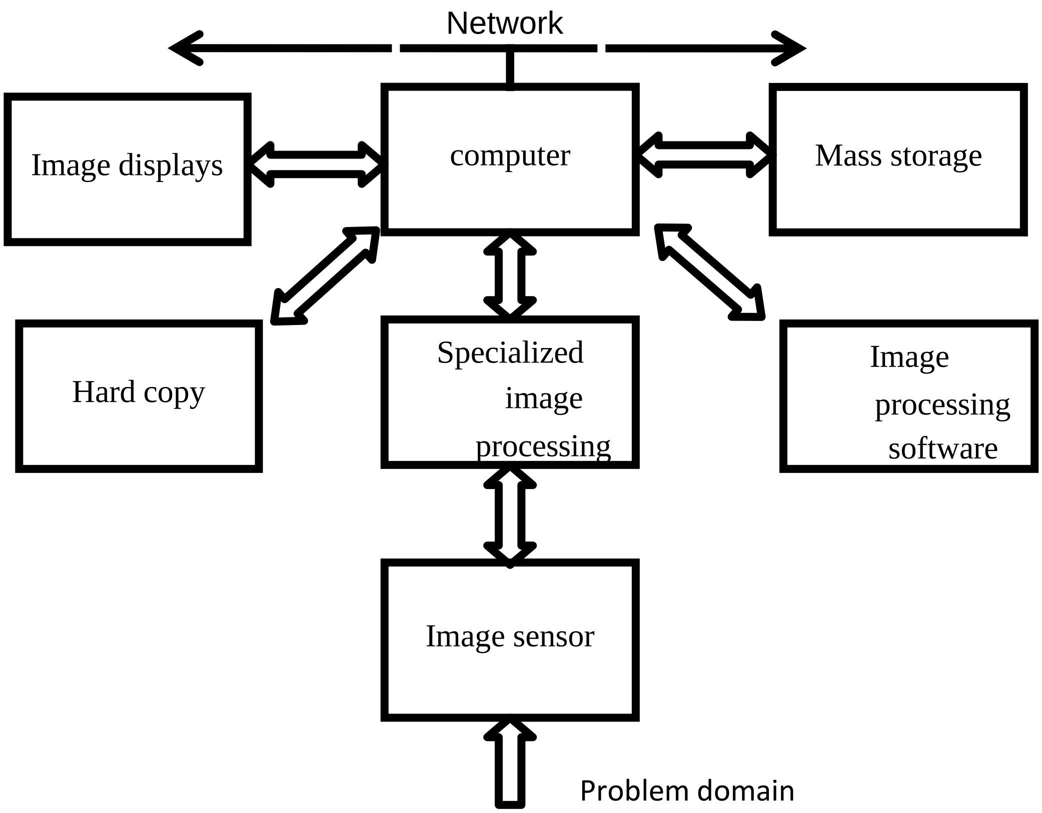


Fig 3.6 Component of image processing

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 1.24 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

- **Image sensors:**

With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

- **Specialized image processing hardware:**

  Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

- **Computer:**

  The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

- **Image processing software:**

  Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

- **Mass storage:**

  Mass storage capability is a must in image processing applications. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage forimage processing applications fall into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), Gbytes (meaning giga, or one billion, bytes), and Tbytes (meaning tera, or one trillion, bytes)

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig. 1.24. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in "jukeboxes" are the usual media for archival applications.

- **Image displays:**

  Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are there requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

- **Hardcopy:**

  Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

- **Network:**

  Networking is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of optical fiber and other broadband technologies.

Color and texture are two low-level features widely used for image classification, indexing and retrieval. Color is usually represented as a histogram, which is a first order statistical measure that captures global distribution of color in an image One of the main drawbacks of the histogram- based approaches is that the spatial distribution and local variations in color are ignored. Local spatial variation of pixel intensity is commonly used to capture texture information in an image. Grayscale Co-occurrence Matrix (GCM) is a well-known method for texture extraction in the spatial domain. A GCM stores the number of pixel neighborhoods in an image that have a particular grayscale combination. Let I be an image and let p and Np respectively denote any arbitrary pixel and its neighbor in a given direction. If GL denotes the total number of quantized gray levels and gl denotes the individual gray levels, where, gl $\in$ {0, . . .,GL _ 1}, then each component of GCM can be written as follows:

$$\mathrm{gcm}(i,j) = \mathrm{Pr}((\mathrm{gl}\ \ \mathrm{gl}$$

gcm(i, j) is the number of times the gray level of a pixel p denoted by glp equals i, and the gray level of its neighbor Np denoted by glNp equals j, as a fraction of the total number of pixels in the image. Thus, it estimates the probability that the gray level of an arbitrary pixel in an image is i, and that of its neighbor is j. One GCM matrix is generated for each possible neighborhood direction, namely, 0, 45, 90 and 135.Average and range of 14 features like Angular Second Moment, Contrast, Correlation, etc., are generated by combining all the four matrices to get a total of 28 features. In the GCM approach for texture extraction, color information is completely lost since only pixel gray levels are considered.

To incorporate spatial information along with the color of image pixels, a feature called color correlogram has recently been proposed. It is a three dimensional matrix that represents the probability of finding pixels of any two given colors at a distance ‘d’ apart Auto correlogram is a variation of correlogram, which represents the probability of finding two pixels with the same color at a distance ‘d’ apart. This atproach can effectively represent color distribution in an image. However, correlogram features do not capture intensity variation Many image databases often contain both color as well as gray scale images. The color correlogram method does not constitute a good descriptor in such databases.

Another method called Color Co-occurrence Matrix (CCM) has been proposed to capture color variation in an image. CCM is represented as a three-dimensional matrix, where color pair of the pixels p and Np are captured in the first two dimensions of the matrix and the spatial distance ‘d’ between these two pixels is captured in the third dimension. This approach is a generalization of the color correlogram and reduces to the pure color correlogram for d = 1. CCM is generated using only the Hue plane of the HSV (Hue, Saturation and Intensity Value) color space. The Hue axis is quantized into HL number of levels. If individual hue values are denoted by hl, where $\mathrm{hl} \in \{0, \ldots, \mathrm{HL}-1\}$, then each component of CCM can be written as follows:

$$\mathrm{ccm}(i,j) = \Pr((\mathrm{hl}_{\mathrm{p}}, \mathrm{hl}_{N_{\mathrm{p}}}) = (i,j))$$

Four matrices representing neighbors at angles 0, 90, 180 and 270 are considered. This approach was further extended by separating the diagonal and the non-diagonal components of CCM to generate a Modified Color Co-occurrence Matrix (MCCM). MCCM, thus, may be written as follows: MCCM = (CCMD;CCMND)

Here, CCMD and CCMND correspond to the diagonal and off-diagonal components of CCM. The main drawback of this approach is that, like correlogram, it also captures only color information and intensity information is completely ignored.

An alternative approach is to capture intensity variation as a texture feature from an image and combine it with color features like histograms using suitable weights . One of the challenges of this approach is to determine suitable weights since these are highly application-dependent. In certain applications like Content-based Image Retrieval (CBIR), weights are often estimated from relevance feedback given by users.

While relevance feedback is sometimes effective, it makes the process of image retrieval user-dependent and iterative. There is also no guarantee on the convergence of the weight-learning algorithms. In order to overcome these problems, researchers have tried to combine color and texture features together during extraction.

proposed two approaches for capturing color and intensity variations from an image using the LUV color space. In the Single-channel Co-occurrence Matrix (SCM), variations for each color channel, namely, L, U and V are considered independently. In the Multi channel Co-occurrence Matrix (MCM), variations are captured taking two channels at a time – UV, LU and LV. Since the LUV color space separates out chrominance (L and U) from luminance (V), SCM in effect, generates one GCM and two CCMs from each image independently. As a result, correlation between the color channels is lost

However, in MCM, the count of pair wise occurrences of the values of different channels of the color space is captured. Thus, each component of MCM can be written as follows:

mcmUV(i; j) = Pr((up; vNp) = (i; j))

mcmLU(i; j) = Pr((lp; uNp) = (i; j))

mcmLV(i; j) = Pr((lp; vNp) = (i; j))

Here, mcmUV(i, j) is the number of times the U chromaticity value of a pixel p denoted by up equals i, and the V chromaticity value of its neighbor Np denoted by vNp equals j, as a fraction of the total number of pixels in the image. Similarly, mcmLU(i, j) and mcmLV(i, j) are defined. One MCM matrix is generated for each of the four neighborhood directions, namely, 0, 45, 90 and 135.

Deng and Manjunath (2001) proposed a two-stage method called JSEG, which combines color and texture after image segmentation. In the first stage, colors are quantized to the required levels for differentiating between various regions of an image.

Pixel values of the regions are then replaced by their quantized color levels to form a color map. Spatial variation of color levels between different regions in the map is viewed as a type of texture composition of the image.

Yu et al. (2002) suggested the use of color texture moments to represent both color and texture of an image. This approach is based on the calculation of Local Fourier Transformation (LFT) coefficients. Eight templates equivalent to LFT are operated over an image to generate a characteristic map of the image. Each template is a $3 \cdot 3$ filter that considers eight neighbors of the current pixel for LFT calculation. First and second order moments of the characteristic map are then used to generate a set of features.

In this paper, we propose an integrated approach for capturing spatial variation of both color and intensity levels in the neighborhood of each pixel using the HSV color space. In contrast to the other methods, for each pixel and its neighbor, the amount of color and intensity variation between them is estimated using a weight function. Suitable constraints are satisfied while choosing the weight function for effectively relating visual perception of color and the HSV color space properties. The color and intensity variations are represented in a single composite feature known as Integrated Color and Intensity Co-occurrence Matrix (ICICM). While the existing schemes generally treat color and intensity separately, the proposed method provides a composite view to both color and intensity variations in the same feature. The main advantage of using ICICM is that it avoids the use of weights to combine individual color and texture features. We use ICICM feature in an image retrieval application from large image databases.

Early result on this work was reported in (Vadivel et al., 2004a). In the next section, we describe the proposed feature extraction technique after introducing some of the

properties of the HSV color space. Choice of quantization levels for color and intensity axes, selection of parameter values and a brief overview of the image retrieval application

Integrated color and intensity co-occurrence matrix:

We propose to capture color and intensity variation around each pixel in a two-dimensional matrix called Integrated Color and Intensity Co-occurrence Matrix (ICICM). This is a generalization of the Grayscale Co-occurrence Matrix and the Color Co-occurrence Matrix techniques. For each pair of neighboring pixels, we consider

their contribution to both color perception as well as gray level perception to the human eye. Some of the useful properties of the HSV color space and their relationship to human color perception are utilized for extracting this feature. In the next sub-section, we briefly explain relevant properties of the HSV color space. In the subsequent subsection, we describe how the properties can be effectively used for generating ICICM.

HSV color space:

HSV Color space: Basically there are three properties or three dimensions of color that being hue, saturation and value HSV means Hue, Saturation and Value. It is important to look at because it describes the color based on three properties. It can create the full spectrum of colors by editing the HSV values. The first dimension is the Hue. Hue is the other name for the color or the complicated variation in the color. The quality of color as determined by its dominant wavelength. This Hue is broadly classified into three categories. They are primary Hue, Secondary Hue and Teritiary Hue. The first and the foremost is the primary Hue it consists of three colors they are red, yellow and blue. The secondary Hue is formed by the combination of the equal amount of colors of the primary Hue and the colors of the secondary Hue which was formed by the primary Hue are Orange, Green and violet. The remaining one is the teritiary Hue is formed by the combination of the primary Hue and the secondary Hue. The limitless number of colors are produced by mixing the colors of the primary Hue in different amounts. Saturation is the degree or the purity of color. Then the second dimension is the saturation. Saturation just gives the intensity to the colors. The saturation and intensity drops just by mixing the colors or by adding black to the color. By adding the white to the color in spite of more intense the color becomes lighter. Then finally the third dimension is the Value. The value is the brightness of the color. When the value is zero the color space is totally black with the increase in the color there is also increase in the brightness and shows the various colors. The value describes the contrast of the color. That means it describes the lightness and darkness of the color. As similar to the saturation this value consists of the tints and shades. Tints are the colors with the added white and shades are the colors with the added black.

Properties of the HSV color space:

Sensing of light from an image in the layers of human retina is a complex process with rod cells contributing to scotopic or dim-light vision and cone cells to photopic or bright-light vision (Gonzalez and Woods, 2002). At low levels of illumination, only the rod cells are excited so that only gray shades are perceived. As the illumination level increases, more and more cone cells are excited, resulting in increased color perception. Various color spaces have been introduced to represent and specify colors in a way suitable for storage, processing or transmission of color information in images. Out of these, HSV is one of the models that separate out the luminance component (Intensity) of a pixel color from its chrominance components (Hue and Saturation). Hue represents pure color, which is perceived when incident light is of sufficient illumination and contains a single wavelength. Saturation gives a measure of the degree by which a pure color is diluted by white light. For light with low illumination, corresponding intensity value in the HSV color space is also low.

The HSV color space can be represented as a Hexa cone, with the central vertical axis denoting the luminance component, I (often denoted by V for Intensity Value). Hue, is a chrominance component defined as an angle in the range [0,2p] relative to the red axis with red at angle 0, green at 2p/3, blue at 4p/3 and red again at 2p. Saturation, S, is the other chrominance component, measured as a radial distance from the central axis of the hexacone with value between 0 at the center to 1 at the outer surface. For zero saturation, as the intensity is increased, we move from black to white through various shades of gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the color represented by its hue. When saturation is near 0, all the pixels in an image look alike even though their hue values are different.

As we increase saturation towards 1, the colors get separated out and are visually perceived as the true colors represented by their hues. Low saturation implies presence of a large number of spectral components in the incident light, causing loss of color information even though the illumination level is sufficiently high. Thus, for low values of saturation or intensity, we can approximate a pixel color by a gray level while for higher saturation and intensity, the pixel color can be approximated by its hue. For low intensities, even for a high saturation, a pixel color is close to its gray value. Similarly, for low saturation even for a high value of intensity, a pixel is perceived as gray. We use these properties to estimate the degree by which a pixel contributes to color perception and gray level perception.

One possible way of capturing color perception of a pixel is to choose suitable thresholds on the intensity and saturation. If the saturation and the intensity are above their respective thresholds, we may consider the pixel to have color dominance; else, it has gray level dominance. However, such a hard thresholding does not properly capture color perception near the threshold values. This is due to the fact that there is no fixed level of illumination above which the cone cells get excited. Instead, there is a gradual transition from scotopic to photopic vision. Similarly, there is no fixed threshold for the saturation of cone cells that leads to loss of chromatic information at higher levels of illumination caused by color dilution. We, therefore, use suitable weights that vary smoothly with saturation and intensity to represent both color and gray scale perception for each pixel.

NON INTERVAL QUANTIZATION:

Due to the large range for each component by directly calculating the characteristics for the retrieval then the computation will be very difficult to ensure rapid retrieval. It is essential to quantify HSV space component to reduce computation and improve efficiency. At the same time, because the human eye to distinguish colors is limited, do not need to calculate all segments. Unequal interval quantization according the human color perception has been applied on H , S ,V components.

Based on the color model of substantial analysis, we divide color into eight parts. Saturation and intensity is divided into three parts separately in accordance with the human eyes to distinguish. In accordance with the different colors and subjective color perception quantification, quantified hue(H), saturation(S) and value(V)

In accordance with the quantization level above, the H, S, V three-dimensional feature vector for different values of with different weights to form one dimensional feature vector and is given by the following equation:

G = Qs*Qv*H+Qv*s+V

Where Qs is the quantized series of S and Qv is the quantized series of V. And now by setting Qs = Qv = 3, Then G = 9H+3S+V

$$H = \begin{cases} 0 \ \ if \ \ h \in [316,20] \\ 1 \ \ if \ \ h \in [21,40] \\ 2 \ \ if \ \ h \in [41,75] \\ 3 \ \ if \ \ h \in [76,155] \\ 4 \ \ if \ \ h \in [156,190] \\ 5 \ \ if \ \ h \in [191,270] \\ 6 \ \ if \ \ h \in [271,295] \\ 7 \ \ if \ \ h \in [296,315] \end{cases} \qquad S = \begin{cases} 0 \ \ if \ \ s \in [0,0.2) \\ 1 \ \ if \ \ s \in [0.2,0.7) \\ 2 \ \ if \ \ s \in [0.7,1) \end{cases} \qquad V = \begin{cases} 0 \ \ if \ \ v \in [0,0.2) \\ 1 \ \ if \ \ v \in [0.2,0.7) \\ 2 \ \ if \ \ v \in [0.7,1) \end{cases}$$

In this way three component vector of the HSV from one dimensional vector, Which quantize the whole color space for the 72 kinds of the main colors. So we can handle 72 bins of one dimensional histogram. This qualification is effective in reducing the images by the effect of the light intensity, but also reducing the computational time and complexity.

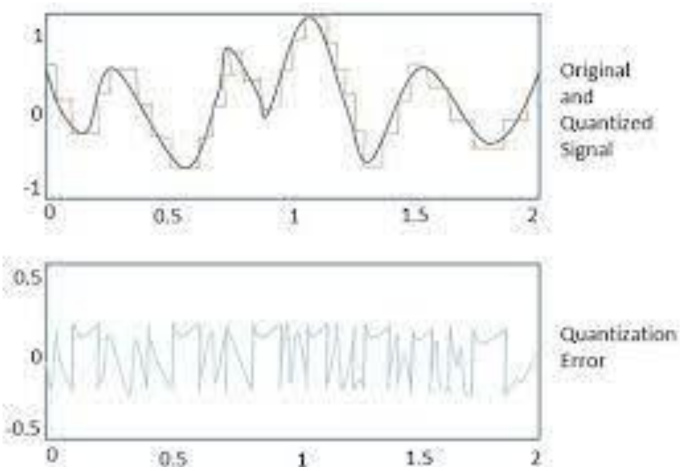


IMAGE RETRIEVAL:

Image retrieval is nothing but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords or the descriptions to the images so that the retrieval can be performed. Manual image annotation is time consuming, expensive and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc..,

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others. one can today generate volumes of images with content as diverse as family get-togethers and national park visits. Low-cost storage and easy Web hosting has fueled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchable image data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real-world image search system designers.

As far as technological advances are concerned, growth in Content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real world implications, applications, and constraints of the technology. Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future.

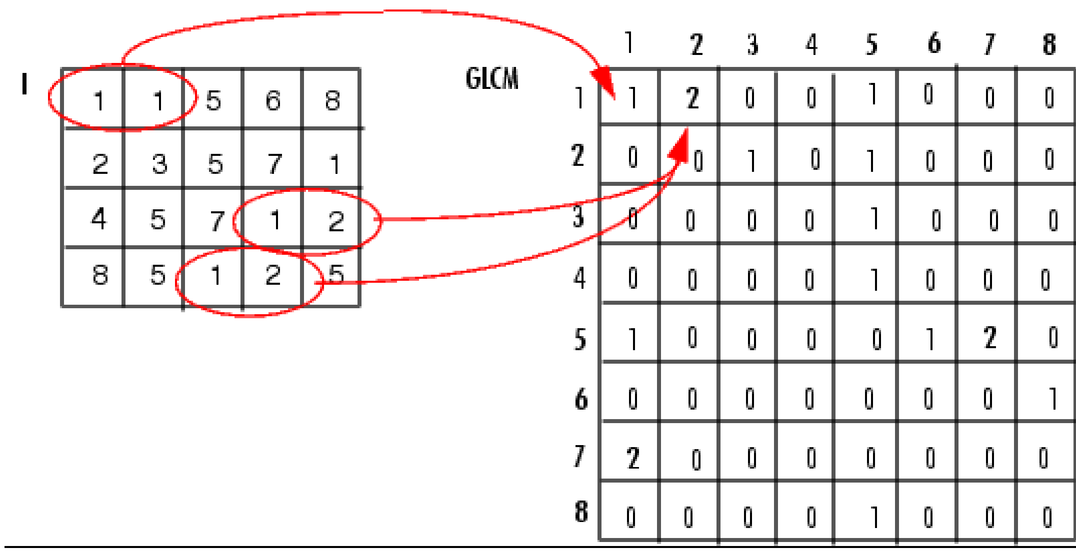
An image retrieval system designed to serve a personal collection should focus on features such as personalization, flexibility of browsing, and display methodology. For example, Google's Picasa system [Picasa 2004] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods. The notable few applications include an FPGA implementation of a color-histogram-based image retrieval system [Kotoulas and Andreadis 2003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 2003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 2002].

Discussion. Regardless of the nature of the collection, as the expected user-base grows, factors such as concurrent query support, efficient caching, and parallel and distributed processing of requests become critical. For future real-world image retrieval systems, both software and hardware approaches to address these issues are essential. More realistically, dedicated specialized servers, optimized memory and storage support, and highly parallelizable image search algorithms to exploit cluster computing powers are where the future of large-scale image search hardware support lies.

OVERVIEW OF TEXTURE:

We all know about the term Texture but for defining it is a hard time. One can differentiate the two different Textures by recognizing the similarities and differences. Commonly there are three ways for the usage of the Textures:

Based on the Textures the images can be segmented To differentiate between already segmented regions or to classify them.We can reproduce Textures by producing the descriptions. The texture can be analyzed in three different ways. They are Spectral, Structural and Statistical:



CHAPTER-4

DIGITAL IMAGE PROCESSING

Digital image processing

Background:

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An important characteristic underlying the design of image processing systems is the significant level of testing & experimentation that normally is required before arriving at an acceptable solution. This characteristic implies that the ability to formulate approaches &quickly prototype candidate solutions generally plays a major role in reducing the cost & time required to arrive at a viable system implementation.

What is DIP

An image may be defined as a two-dimensional function f(x, y), where x & y are spatial coordinates, & the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital image by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensor, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis& computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting

& somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low-level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low- level process is characterized by the fact that both its inputs & outputs are images.

Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images. Finally higher- level processing involves “Making sense” of an ensemble of recognized objects, as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

What is an image?

An image is represented as a two dimensional function f(x, y) where x and y are spatial co-ordinates and the amplitude of ‘f’ at any pair of coordinates (x, y) is called the intensity of the image at that point.

Gray scale image:

A grayscale image is a function I (xylem) of the two spatial coordinates of the image plane.

I(x, y) is the intensity of the image at the point (x, y) on the image plane.

I (xylem) takes non-negative values assume the image is bounded by a rectangle [0, a] ×[0, b]I: [0, a] × [0, b] → [0, info)

Color image:

It can be represented by three functions, R (xylem) for red, G (xylem) for green *and* B (xylem) for blue.

An image may be continuous with respect to the x and y coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate’s values is called sampling. Digitizing the amplitude values is called quantization.

Coordinate convention:

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image f(x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size M X N. The values of the coordinates (xylem) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates.

In many image processing books, the image origin is defined to be at (xylem)=(0,0).The next coordinate values along the first row of the image are (xylem)=(0,1).It is important to keep in mind that the notation (0,1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Following figure shows the coordinate convention. Note that x ranges from 0 to M-1 and y from 0 to N-1 in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (xylem) the toolbox uses the notation (race) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate topples, (alb), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentation refers to the coordinates. Less frequently the toolbox also employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refers to rows. This is the opposite of our use of variables x and y.

Image as Matrices:

The preceding discussion leads to the following representation for a digitized image function:

$$f\ (\text{xylem}) = \begin{matrix} f(0,0) & f(0,1) & \ldots\ldots\ldots & f(0,N-1) \\ f(1,0) & f(1,1) & \ldots\ldots\ldots & f(1,N-1) \\ . & . & & . \\ . & . & & . \\ f(M-1,0) & f(M-1,1) & \ldots\ldots\ldots & f(M-1,N-1) \end{matrix}$$

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements.

A digital image can be represented naturally as a MATLAB matrix:

$$\begin{matrix} f(1,1) & f(1,2) \ldots\ldots\ldots & f(1,N) \\ f(2,1) & f(2,2) \ldots\ldots\ldots & f(2,N) \end{matrix}$$

.   .   .

f =   .   .   .

f (M,1) f(M,2) …….f(M,N)

Where f (1,1) = f(0,0) (note the use of a monoscope font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation f(p ,q) denotes the element located in row p and the column q. For example f(6,2) is the element in the sixth row and second column of the matrix f. Typically we use the letters M and N respectively to denote the number of rows and columns in a matrix. A 1xN matrix is called a row vector whereas an Mx1 matrix is called a column vector. A 1x1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using mono-scope characters. We use conventional Roman, italic notation such as f(x ,y), for mathematical expressions

Reading Images:

Images are read into the MATLAB environment using function imread whose syntax is

Imread (‘filename’)

Format name	**Description**	**recognized extension**
TIFF	Tagged Image File Format	.tif, .tiff
JPEG	Joint Photograph Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png

XWD	X Window Dump	.xwd

Here filename is a spring containing the complete of the image file(including any applicable extension).For example the command line

>> f = imread ('8. jpg');

Reads the JPEG (above table) image chestxray into image array f. Note the use of single quotes (') to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output If a semicolon is not included. MATLAB displays the results of the operation(s) specified in that line. The prompt symbol (>>) designates the beginning of a command line, as it appears in the MATLAB command window.

Data Classes:

Although we work with integers coordinates the values of pixels themselves are not restricted to be integers in MATLAB. Table above list various data classes supported by MATLAB and IPT are representing pixels values. The first eight entries in the table are refers to as numeric data classes. The ninth entry is the char class and, as shown, the last entry is referred to as logical data class.

All numeric computations in MATLAB are done in double quantities, so this is also a frequent data class encounter in image processing applications. Class unit 8 also is encountered frequently, especially when reading data from storages devices, as 8 bit images are most common representations found in practice. These two data classes, classes logical, and, to a lesser degree, class unit 16 constitute the primary data classes on which we focus. Many ipt functions however support all the data classes listed in table. Data class double requires 8 bytes to represent a number uint8 and int 8 require one byte each, uint16 and int16 requires 2bytes and unit 32.

Name	**Description**
Double	Double _ precision, floating_ point numbers the Approximate.
Uint8	unsigned 8_bit integers in the range [0,255] (1byte per Element).

Uint16 unsigned 16_bit integers in the range [0, 65535] (2byte per element).

Uint 32 unsigned 32_bit integers in the range [0, 4294967295](4 bytes per element). Int8 signed 8_bit integers in the range [-128,127] 1 byte per element)

Int 16 signed 16_byte integers in the range [32768, 32767] (2 bytes per element).

Int 32 Signed 32_byte integers in the range [-2147483648, 21474833647] (4 byte per element).

Single single _precision floating _point numbers with values

In the approximate range (4 bytes per elements)

Char characters (2 bytes per elements).

Logical values are 0 to 1 (1byte per element).

Int 32 and single required 4 bytes each. The char data class holds characters in Unicode representation. A character string is merely a 1*n array of characters logical array contains only the values 0 to 1,with each element being stored in memory using function logical or by using relational operators.

Image Types:

The toolbox supports four types of images:

1 .Intensity images;

2. Binary images;

3. Indexed images;

4. R G B images.

Most monochrome image processing operations are carried out using binary or intensity images, so our initial focus is on these two image types. Indexed and RGB colour images.

Intensity Images:

An intensity image is a data matrix whose values have been scaled to represent intentions. When the elements of an intensity image are of class unit8, or class unit 16, they have integer values in the range [0,255] and [0, 65535], respectively. If the image is of class double, the values are floating point numbers. Values of scaled, double intensity images are in the range [0, 1] by convention.

Binary Images:

Binary images have a very specific meaning in MATLAB.A binary image is a logical array 0s and1s.Thus, an array of 0s and 1s whose values are of data class, say unit8, is not considered as a binary image in MATLAB .A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, we create an array B using the statement.

B=logical (A)

If A contains elements other than 0s and 1s.Use of the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.

Using relational and logical operators also creates logical arrays.

To test if an array is logical we use the I logical function: islogical(c).

If c is a logical array, this function returns a 1.Otherwise returns a 0. Logical array can be converted to numeric arrays using the data class conversion functions.

Indexed Images:

An indexed image has two components:

A data matrix integer, x

A color map matrix, map

Matrix map is an m*3 arrays of class double containing floating point values in the range [0, 1].The length m of the map are equal to the number of colors it defines. Each row of map specifies the red, green and blue components of a single color. An indexed images uses "direct mapping" of pixel intensity values color map values. The color of each pixel is determined by using the corresponding value the integer matrix x as a pointer in to map. If x is of class double ,then all of its components with values less than or equal to 1 point to the first row in map, all components with value 2 point to the second row and so on. If x is of class units or unit 16, then all components value 0 point to the first row in map, all components with value 1 point to the second and so on.

RGB Image:

An RGB color image is an M*N*3 array of color pixels where each color pixel is triplet corresponding to the red, green and blue components of an RGB image, at a specific spatial location. An RGB image may be viewed as “stack” of three gray scale images that when fed in to the red, green and blue inputs of a color monitor

Produce a color image on the screen. Convention the three images forming an RGB color image are referred to as the red, green and blue components images. The data class of the components images determines their range of values. If an RGB image is of class double the range of values is [0, 1].

Similarly the range of values is [0,255] or [0, 65535].For RGB images of class units or unit 16 respectively. The number of bits use to represents the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8bit image, the corresponding RGB image is said to be 24 bits deep.

Generally, the number of bits in all component images is the same. In this case the number of possible color in an RGB image is (2^b) ^3, where b is a number of bits in each component image. For the 8bit case the number is 16,777,216 colors

CHAPTER 5

Software Introduction:

5.1. Introduction to MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

5.2 The MATLAB system:

The MATLAB system consists of five main parts

- **Development Environment**:

  This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library**:

  This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language**:

  This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

- **Graphics:**

  MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

5.3 GRAPHICAL USER INTERFACE (GUI):

MATLAB’s Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function’s GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.

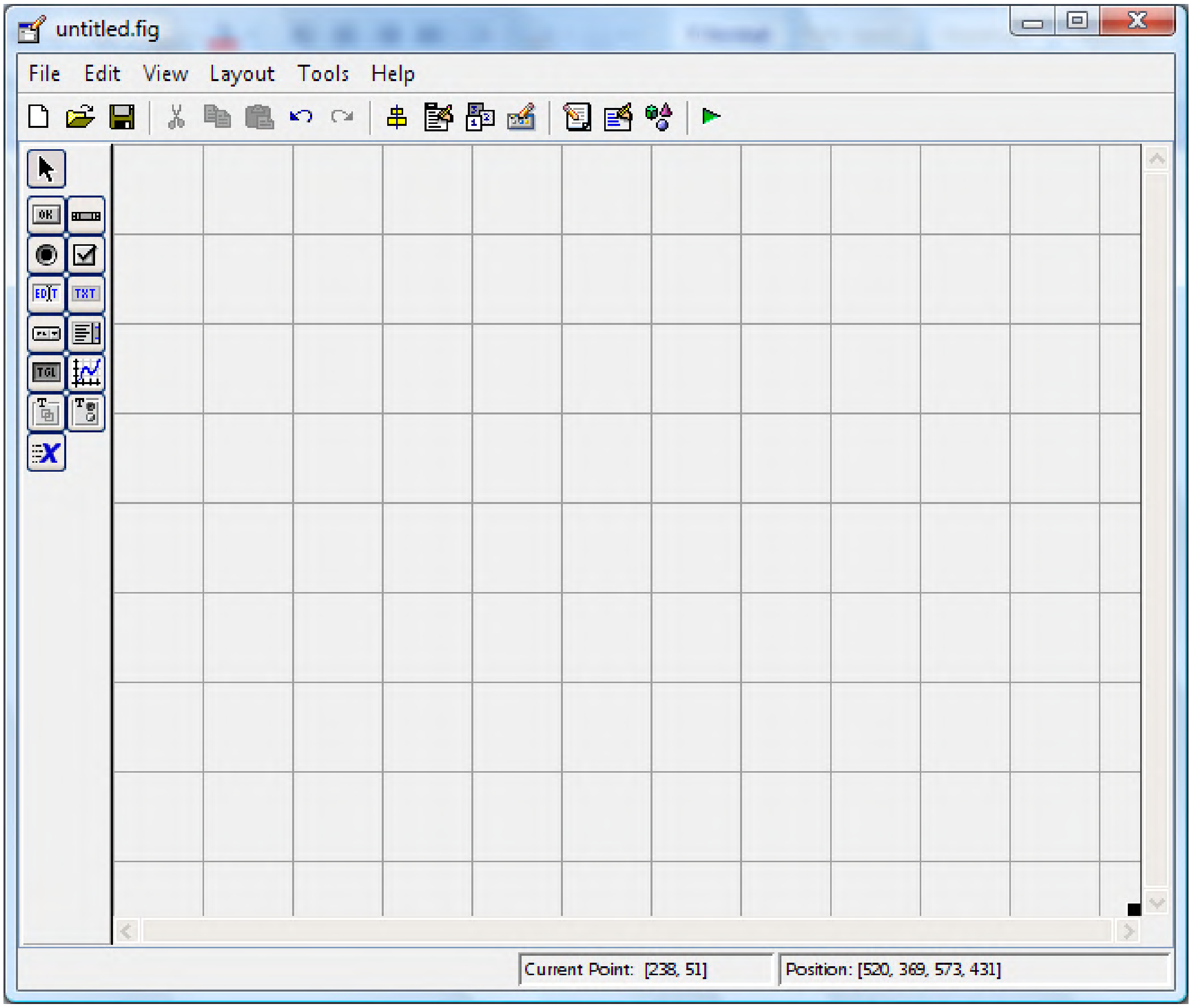
- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type

guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted,

GUIDE opens a new (i.e., blank) window.



A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

5.4 Getting Started

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator,

and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

5.4.1 Introduction - describes the components of the MATLAB system.

5.4.2 Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

5.4.3 Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

5.4.4 Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

5.4.5 Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

5.5 DEVELOPMENT ENVIRONMENT

5.5.1 Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

5.5.2 Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

5.5.3 Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

5.5.4 MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

5.5.5 Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type helpbrowser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to runMATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

5.6 MANIPULATING MATRICES

5.6.1 Entering Matrices

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Dürer's matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

MATLAB displays the matrix you just entered.

```
A =
   16    3    2   13
    5   10   11    8
    9    6    7   12
    4   15   14    1
```

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

5.6.2 Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example, num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

```
3               -99             0.0001
9.6397238       1.60210e-20     6.02252e23
1i              -3.14159j       3e5i
```

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

5.6.3 Operators

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

5.6.4 Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.14159265...
I	Imaginary unit, $\sqrt{-1}$
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

5.7 GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

5.7.1 Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

5.7.2 GUI Development Environment

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of its children (uicontrols and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the callbacks, which are defined as subfunctions. This M-file is referred to as the application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

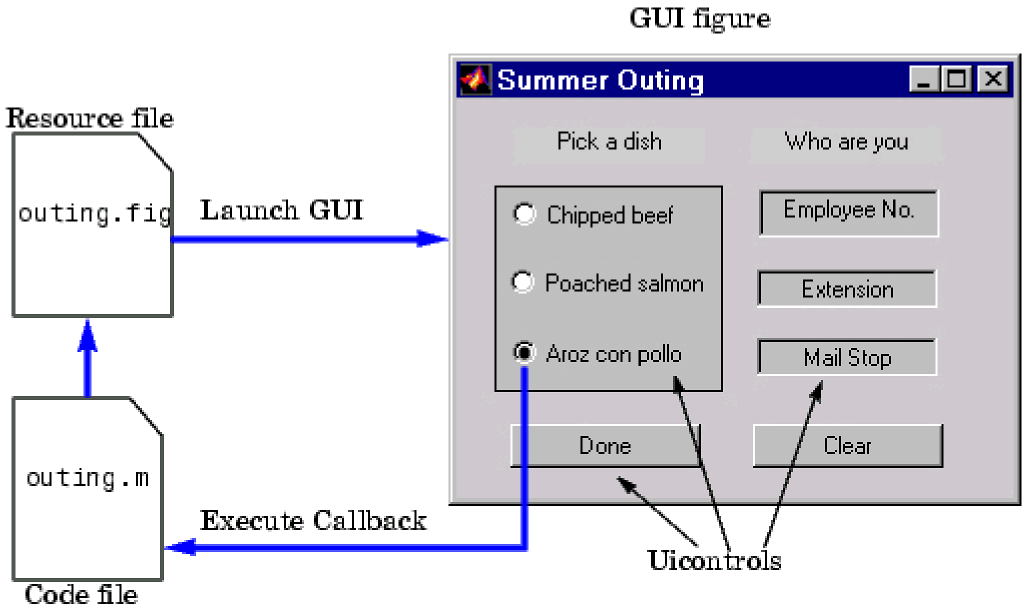


FIG 5.7 graphical user blocks

5.7.3 Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

5.7.4 Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques used in the application M-file.

Application Examples - a collection of examples that illustrate techniques which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

5.7.5 User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- Listboxes
- Checkboxes
- Popup Menus

- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)
button_state = get(h,'Value');
if button_state == get(h,'Max')
```

```
    % toggle button is pressed
elseif button_state == get(h,'Min')
    % toggle button is not pressed
end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);
a(:,:,2) = rand(16,128);
a(:,:,3) = rand(16,128);
set(h,'CData',a)
```

Radio Buttons

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)
set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)
off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];
mutual_exclude(off)
% Continue with callback
   .
   .
   .
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)
if (get(h,'Value') == get(h,'Max'))
   % then checkbox is checked-take approriate action
else
   % checkbox is not checked-take approriate action
end
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)
user_string = get(h,'string');
% proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)
user_entry = str2double(get(h,'string'));
if isnan(user_entry)
   errordlg('You must enter a numeric value','Bad Input','modal')
end
% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menubar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menubar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

Listboxes differentiate between single and double clicks on an item and set the figure SelectionType property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure SelectionType property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)
val = get(h,'Value');
switch val
case 1
% The user selected the first item
case 2
% The user selected the second item
% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)
val = get(h,'Value');
string_list = get(h,'String');
selected_string = string_list{val}; % convert from cell array to string
% etc.
```

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes ButtonDownFcn property to define the callback.

3.7.6 Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

```
axes(handles.axes1)
```

makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axes. See GUI with Multiple Axes for and example that uses two axes.

Figure

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control.

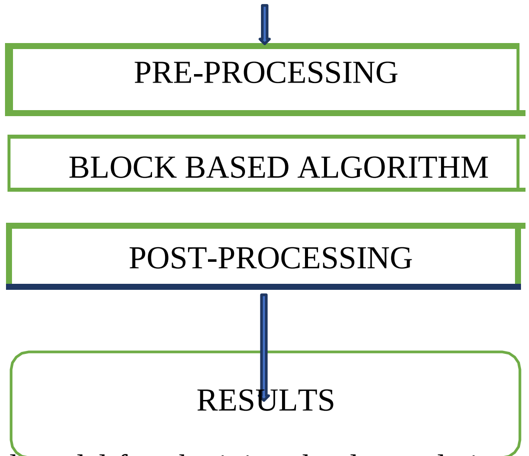
Chapter 6

Proposed method

Landsat 8 is one of the essential satellite programs of the National Aeronautics and Space Administration (N.A.S.A). This satellite is currently operational. This satellite consists of two different sensors "Thermal InfraRed Sensor (T.I.R.S.) and Operational Land Imager (O.L.I.)."b) At any point of time, for each IMF average value of the envelope computed from local maxima or local minima must be zero.

By considering the above two conditions, it came to know that IMFs are correlated to oscillatory modes in time series. Addition of all IMFs and the residual signal recovers the entire signal. The IMFs contain different frequency ranges, where the highest frequencies are usually found in the first IMF which contains the edge information with reference to image and lower frequencies in subsequent IMFs consists of spatial information of the image.

BLOCK DIAGRAM:



Now the mathematical model for obtaining the decorrelation stretcher can be understood from the following expressions. Let us assume that for the original image having (k, l) number of pixels. The decorrelation stretcher image has (k, m) number of pixels.

$$SUMX_{l,m} = \sum_{k=1}^{n} P_{k,l} * P_{k,m}$$

$$SUM_l = \sum_{k=1}^{n} P_{k,l}$$

Where Pk,l is the value of the kth pixel for the channel l.

The element of the covariance matrix

$$COV_{l,m=} \frac{1}{(n-1)}\left[SUMX_{l,m} - \frac{1}{n} * SUM_l * SUM_m\right]$$

For the correlation matrix element

$$COV_{l,m} = \frac{COV_{l,m}}{(COV_{l,l} * COV_{m,n})^{1/2}}$$

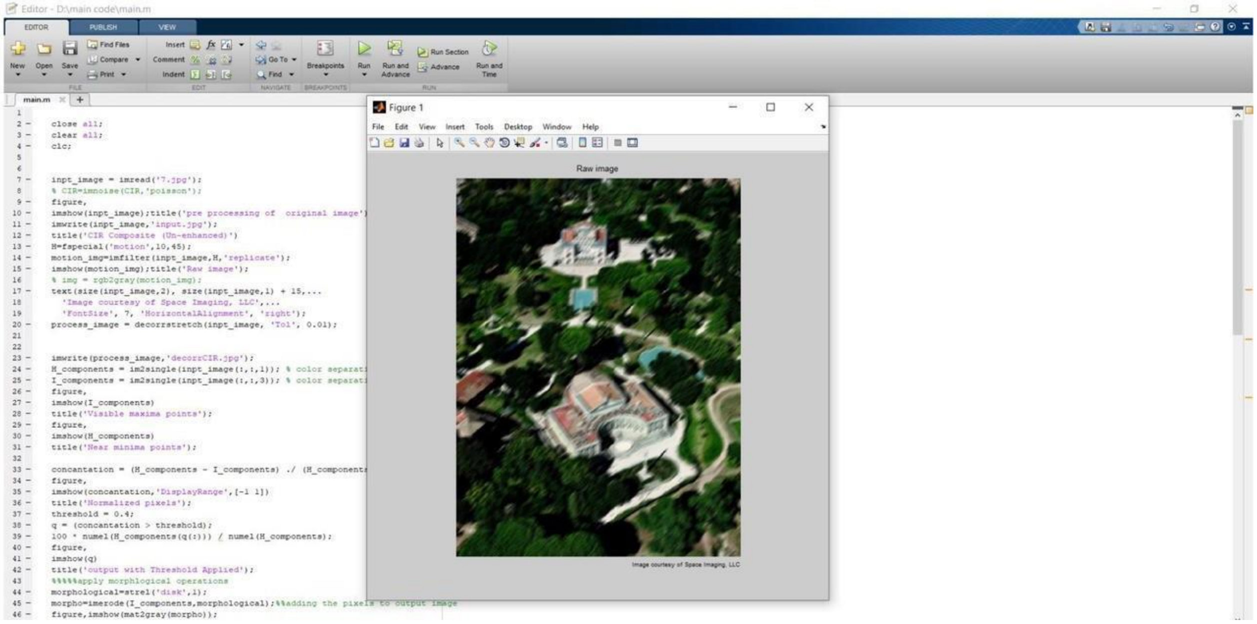
Table. 1 Landsat 8 Band Information [22]

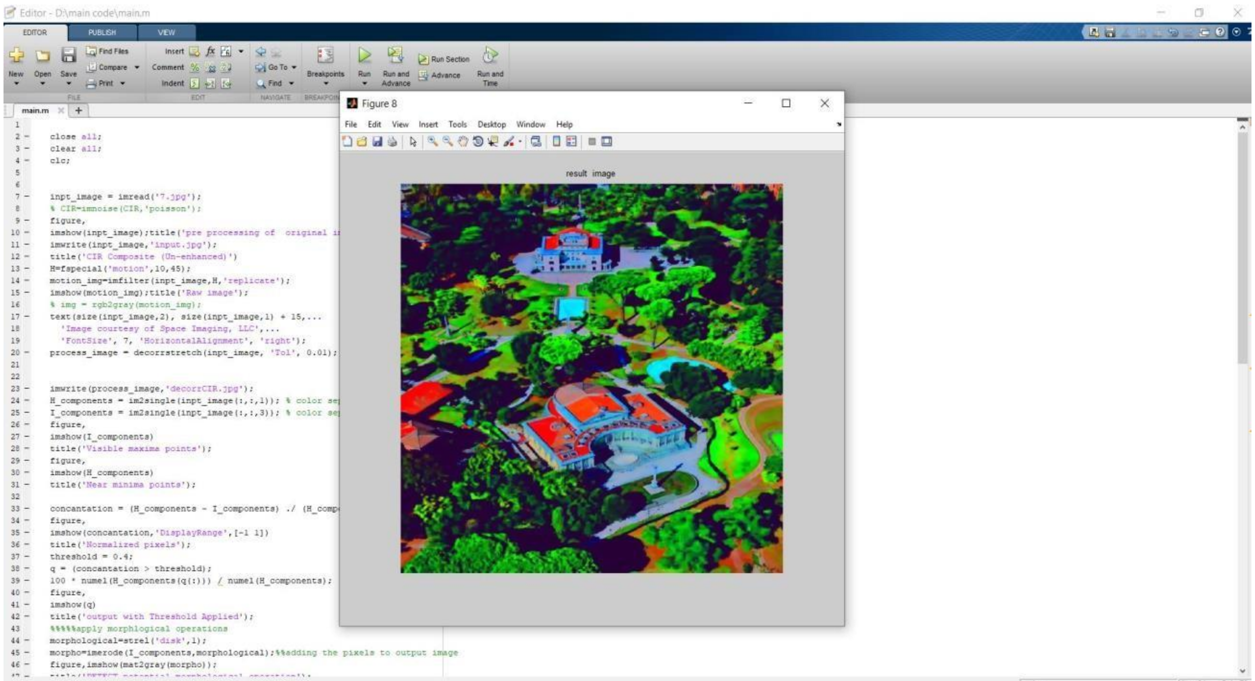
S.No	Spectral Band	Wavelength	Resolution	Solar Irradiance
1	Band 1	0.433 – 0.453	30 *m*	2031 W/(m2μm)
2	Band 2	0.450 – 0.515	30 *m*	1925 W/(m2μm)
3	Band 3	0.525 – 0.600	30 *m*	1826 W/(m2μm)
4	Band 4	0.630 – 0.680	30 *m*	1574 W/(m2μm)
5	Band 5	0.845 – 0.885	30 *m*	955 W/(m2μm)
6	Band 6	1.560 – 1.660	30 *m*	242 W/(m2μm)
7	Band 7	2.100 – 2.300	30 *m*	82.5 W/(m2μm)
8	Band 8	0.500 – 0.680	15 *m*	1739 W/(m2μm)
9	Band 9	1.360 – 1.390	30 *m*	361 W/(m2μm)

Chapter 7

Experiment results

Input Image:





Output Image (fig:7.1 & 7.2)

Chapter 8

CONCLUSION

In the proposed research work, we have presented a new approach for image enhancement. This image enhancement is based on an image decorrelation stretcher. This approach is suitable for mainly satellite images. Besides, this approach can be used for medical diagnostic purposes. In the World of today, image enhancement can also be used to provide information regarding Novel Coronavirus.

REFERENCES

[1] V. Aristov, "Distance localization of the moving object by applying the Karhunen–Loeve transform to ultra-wideband impulse signals," Automatic Control and Computer Sciences , vol. 51, no. https://doi.org/10.3103/S0146411617050029, pp. 294-300, 2017.

[2] M. H. Alkawaz, G. Sulong, T. Saba and A. Rehman, "Detection of copymove image forgery based on discrete cosine transform," Neural Computing and Applications , vol. 30, no. 1, pp. 183-192, 2018.

[3] A. K. Shakya, A. Ramola, H. S. Pokhariya and A. Kandwal, "Fusion of IoT and Machine Learning Approach to Prevent Confidential Data from Digital Crimes and Cyber Mugging for Covert Transmission," Lecture Notes in Electrical Engineering , vol. 553, no. 1, pp. 563-579, 2019.

[4] A. K. Shakya, A. Ramola and D. C. Pandey, "Polygonal region of interest based compression of DICOM images," in International Conference on Computing, Communication and Automation (ICCCA), Noida, 2017.

[5] A. K. Shakya, A. Ramola, Y. Tandon and A. Vidhyarthi, "Suomi NPP Satellite Data Processing to Estimate Impact of Cyclone Fani on LU/LC," in 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, 2019.

[6] A. K. Shakya, A. Ramola, K. Sawant and A. Kandwal, "Comparison of Supervised Classification Techniques for High-Resolution Optical Aerial Image," in International Conference on Automation and Computational Engineering (ICACE), Greater Noida, India,, 2018.

[7] A. M. Uji, Y. M. Muraoka and N. M. Yoshimura, "Hidden information in color fundus photographs is revealed by the decorrelation stretching method," Retinal Cases & Brief Reports, vol. 13, no. 2, pp. 176-180, 2019.

[8] P. Sinha, B. Horgan, E. Ewing and E. Rampe, "Decorrelation Stretches (DCS) of Visible Images as a Tool for Sedimentary Provenance Investigations on Earth and Mars," SPACE SCIENCES (GENERAL), no. JSC-E-DAA-TN78923, pp. 1-2, 2020.

[9] G. C. Miliaresis, "Spatial decorrelation stretch of annual (2003–2014) Daymet precipitation summaries on a 1-km grid for California, Nevada, Arizona, and Utah," Environmental Monitoring and Assessment volume, vol. 188, no. Article number: 361 , pp. 1-21, 2016.

[10] A. Jellouli, A. E. Harti, Z. Adiri, E. M. Bachaoui and A. E. Ghmari, "Application of Remote Sensing Data in Lithological Discrimination of Kerdous Inlier in the Anti Atlas Belt of Morocco," International Journal on Informatics Visualization , vol. 3, no. 2-2, pp. 227-232, 2019.

[11] E. Gürbüz, "Multispectral mapping of evaporite minerals using ASTER data: A methodological comparison from central Turkey," Remote Sensing Applications: Society and Environment, vol. 15, no. 100240, pp. 1-13, 2019.

[12] A. R. Gillespie, A. B, K. Richard and E. Walker, "Color enhancement of highly correlated images. I. Decorrelation and HSI contrast stretches," Remote Sensing of Environment, vol. 20, no. 3, pp. 209-235, 1986.

[13] A. K. Shakya, A. Ramola, A. Kandwal and R. Prakash, "Change Over Time in Grey Levels of Multispectral Landsat 5TM/8OLI Satellite Images," in Lecture Notes in Electrical Engineering , Ranchi, Springer, Singapore, 2019, pp. 309-356.

[14] A. K. Shakya, A. Ramola, A. Kandwal, P. Mittal and R. Prakash, "Morphological Change Detection in Terror Camps of Area 1 and 2 by Pre- and Post-strike Through MOAB: A," in Lecture Notes in Electrical Engineering , Sikkim, Springer, Singapore, 2019, pp. 253-263.

[15] A. K. Shakya, A. Ramola and D. C. Pandey, "Polygonal region of interest based compression of DICOM images," in International Conference on Computing, Communication and Automation (ICCCA), Noida, 2017.

[16] K. M. Ratheesh, L. K. Seah and V. M. Murukeshan, "Spectral phasebased automatic calibration scheme for swept source-based optical coherence tomography systems," Physics in Medicine & Biology, , vol. 61, no. 21, 2016.

[17] R. K. Meleppat, M. V. Matham and L. K. Seah, "An efficient phase analysis-based wavenumber linearization scheme for swept source optical coherence tomography systems," Laser Physics Letters, vol. 12, no. 5, 2015.

[18] R. K. Meleppat, E. B. Miller, S. K. Manna, P. Zhang, E. N. P. Jr and R. J. Zawadzki, "Multiscale Hessian filtering for enhancement of OCT angiography images," in Proc. SPIE 10858 Ophthalmic Technologies XXIX, 108581K , 2019.

[19] Z. Dong, D. Yang, T. Reindl and W. M.Walsh, "Satellite image analysis and a hybrid ESSS/ANN model to forecast solar irradiance in the tropics," Energy Conversion and Management, vol. 79, no. 1, pp. 66-73, 2014.

[20] A. K. Shakya, A. Ramola, A. Kashyap, D. V. Pham and A. Vidyarthi, "Susceptibility Assesment of Changes Developed in the Landcover Caused Due to the Landslide Disaster of Nepal from Multispectral LANDSAT Data," in Communications in Computer and Information Science, Mohali, Springer Nature Singapore, 2020, pp. 406-418.

A

PROJECT REPORT

On

IOT BASED COAL MINE SAFETY MONITORING AND ALERTING SYSTEM

Submitted by

1) **Ms.D.Rishika** **(17K81A0414)**

2) **Mr.E.Joseph David** **(18K81A5410)**

3) **Mr.Piyush Roy** **(15K81A04A6)**

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

G.Upender

M.Tech,(Ph.D)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

ENGINEERINNG

ST.MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500100

JUNE 2021

BONAFIDE CERTIFICATE

This is to certify that the project entitled IOT BASED COAL MINE SAFETY MONITORING AND ALERTINNG SYSTEM, is being submitted by **1.Ms.D.Rishika(17K81A0414), 2.Mr.E.Joseph David(18K85A0410) 3.Mr.Piyush Roy(15K81A04A6)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

G.UPENDER
Department of ECE

Head of the Department
B.HARI KRISHNA
Department of ECE

Internal Examiner

External Examiner

Place:

Date:

DECLARATION

We, the student of **Bachelor of Technology** in Department of Electronics And Communication Engineering’, session: <2017 – 2021>, St. Martin’s Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled IOT BASED COAL MINE SAFETY MONITORING AND ALERTING SYSTEM is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

D.Rishika (17K81A0414)
E.Joseph David (18K85A0410)
Piyush Roy (15K81A04A6)

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his in valuable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerableways.

We would like to express our gratitude to **B.HARI KRISHNA, Head&Professor**, the Dept. of Information Technology for his/her advice, supervision and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept. project coordinator A.ANAND** for his/her involvement with originality has triggered and nourished my intellectual maturity that will help me for along time to come.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

1. D.Rishika
2. E.Joseph David
3. Piyush Roy

ABSTRACT

Coal mining is the process of extracting coal from the ground. Steel and cement industries use coal as a fuel for extraction of iron from iron ore and for cement production. Underground mining industry comes to the category, where each and every parameter such as methane gas, high temperature, fire accidents and so on has to be monitored regularly. Safe production level of coal mine is still low, disasters in coal mine occur frequently, which lead to great loss of possession and life. The disasters happening in coal mine are due to the complexity of mine environment and the variety of work carried out in coal mine, so it is very necessary to monitor the working environment of coal mine. To get over this problem, projects has proposed a wireless sensor network's application in coal mining safety system. In this wireless sensor network's application system there will, be master controllers and slave controllers. Slave controllers will detect the danger and give alert through RF to master controller and it will raise the alarm in all tunnels and also raise message on IOT, which will help to take action as soon as possible. This monitoring and alerting system is powered by Atmega328 microcontroller for master controller and rollers for slave controller and consists of temperature sensor, water sensor, methane and carbon dioxide sensor and RF transmitter which collects temperature, humidity and methane values underground of coal mine.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVEIW OF THE PROJECT:

Coal is one of the most important commodities and raw materials for a number of industries. It is used for power generation as well as the extraction of many by-product chemicals and materials. The extraction of coal from the coal mine is, however, a complex and dangerous process. Many accidents take place in the coal mines world over which causes fatalities and economic losses. The dangers and hazards can be reduced significantly by making use of the latest smart technologies.

1.2 OBJECTIVES OF THE PROJECT

The objective of developing this systems is to monitor and alerting the workers if there is any change in temperature and water level. This project is developed to provide a better safety to the coal mine workers**.**

1.3 SCOPE OF THE PROJECT

With the growing innovations future work of this experimentation may include, more development of the system by using other advanced sensors for monitoring the underground. Threats. Also, all the underground operations can be carried out from the ground surface. New developing communication technologies can be used for highspeed data transfer in integration with smart sensors for sensing the mine conditions. Also, more IOT enabled systems can be developed for more advanced functionality.

1.4 MATERIAL REQUIREMENT:

1.4.1 HARDWARE REQUIREMENT:

1. Arduino UNO
2. LCD
3. Power Supply Unit
4. LM35
5. Water level indicator
6. RF Module
7. Wifi Module

8. Buzzer

ARDUINO

Arduino is open source software. Arduino Uno is a microcontroller, ATmega328Pcontain in it. It has 14 digital i/o pins and 6 analog input pin, a 16 MHz quartz crystal, a USB connection, a power jack and a reset button. Simply connect to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. Embedded C program is used for coding.

FIG.1.1 ARDUINO UNO

LCD

The LCD is used to display whenever there is a gas leakage or fire detected in the area. Whenever there is a gas leakage the Arduino that is connected to the sensor and LCD sends a message to LCD to display ‘Gas leakage’. When there is fire outbreak the Arduino again sends a message to LCD to display ‘Fire Detected’ notifying us if there is a gas leakage or fire detected in the vicinity.

FIG 1.2 LCD DISPLAY

POWER SUPPLY

Every electrical and electronic device that we use in our day-to-day life will require a power supply. In general, we use an AC supply of 230V 50Hz, but this power has to be changed into the required form with required values or voltage range for providing power supply to different types of devices. There are various types of power electronic converters such as step-down converter, step-up converter, voltage 31 stabilizer, AC to DC converter, DC 25 to DC converter, DC to AC converter, and so on. For example, consider the microcontrollers that are used frequently for developing many embedded systems based projects and kits used in real-time applications. These microcontrollers require a 5V DC supply.

LM35

LM35 is a precession Integrated circuit Temperature sensor, whose output voltage varies, based on the temperature around it. It is a small and cheap IC which can be used to measure temperature anywhere between -55°C to 150°C. There will be rise of 0.01V (10mV) for every degree Celsius rise in temperature.

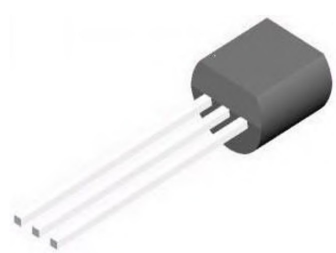


Fig 1.3 LM35

WATER LEVEL INDICATOR

A water level indicator is a system that relays information back to a control panel to indicate whether a body of water has a high or low water level. Some water level indicators use a combination of probe sensors or float switches to sense water levels. “The Water Level Indicator employs a simple mechanism to detect and indicate the water level in an overhead tank or any other water container.”

Fig.1.4 Water level indicator

RF MODULE

An RF module (short for radio-frequency module) is a (usually) small electronic device used to transmit and/or receive radio signals between two devices. In an embedded system it is often desirable to communicate with another device wirelessly. This wireless communication may be accomplished through optical communication or through radio frequency(RF) communication. For many applications, the medium of choice is RF since it does not require line of sight. RF communications incorporate a transmitter and a receiver. They are of various types and ranges. Some can transmit up to 500 feet. RF modules are typically fabricated using RF CMOS technology.

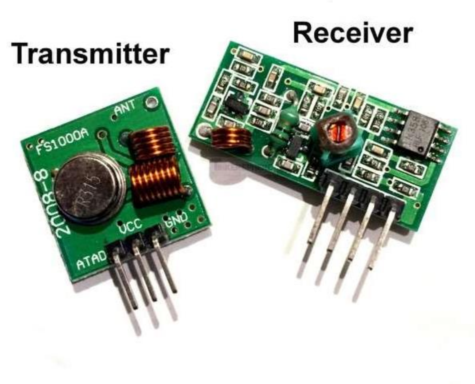


FIG 1.5 RF MODULE

WIFI MODULE(ESP-01)

The ESP8266 Wi-Fi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

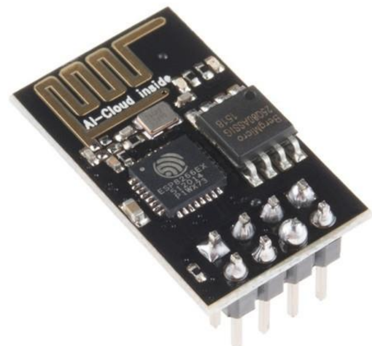


FIG 1.6 WIFI MODULE

BUZZER

FIG 1.7 BUZZER

1.4.2 SOFTWARE REQUIREMENTS:

1.4.2.1 ARDUNO IDE SOFTWARE:

This software is basically used for giving commands to ARDUNO Microcontroller**.**

STEPS FOR INSTALLING ARDUNO IDE SOFTWARE:

1. Download Arduino IDE Software in your PC.

2. When the download finishes, proceed with the installation.

3. Allow the driver installation process when you get a warning from the operating system.

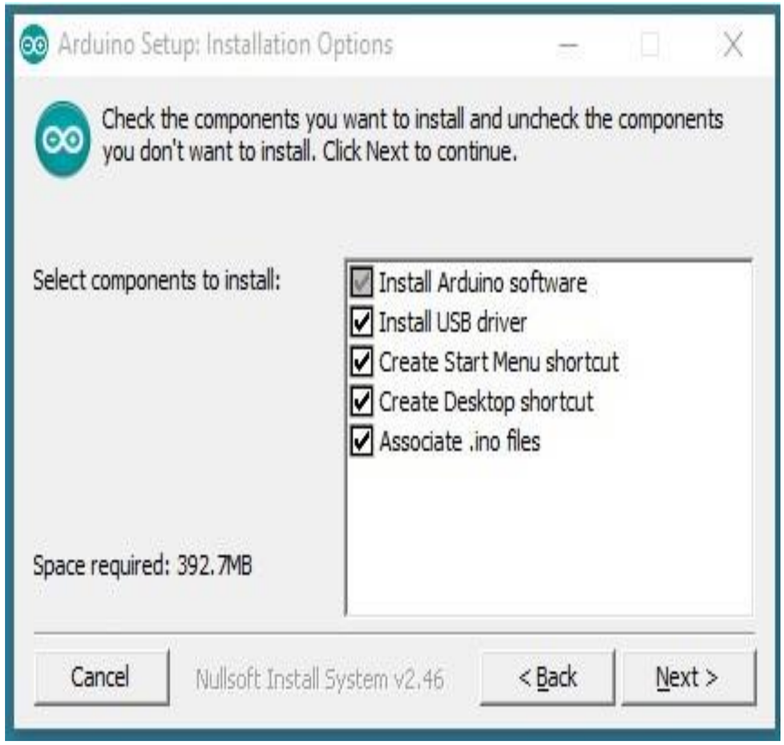


Fig:1.8 Installation process of driver

4. Choose the components to install

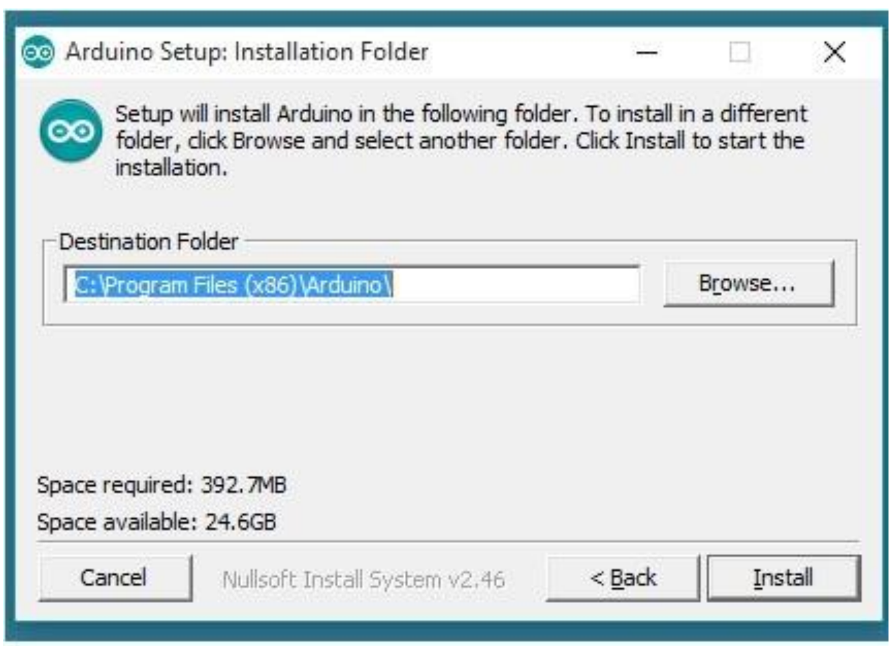


Fig 1.9 Setting destination folder

5. Choose the installation directory (we suggest to keep the default one)

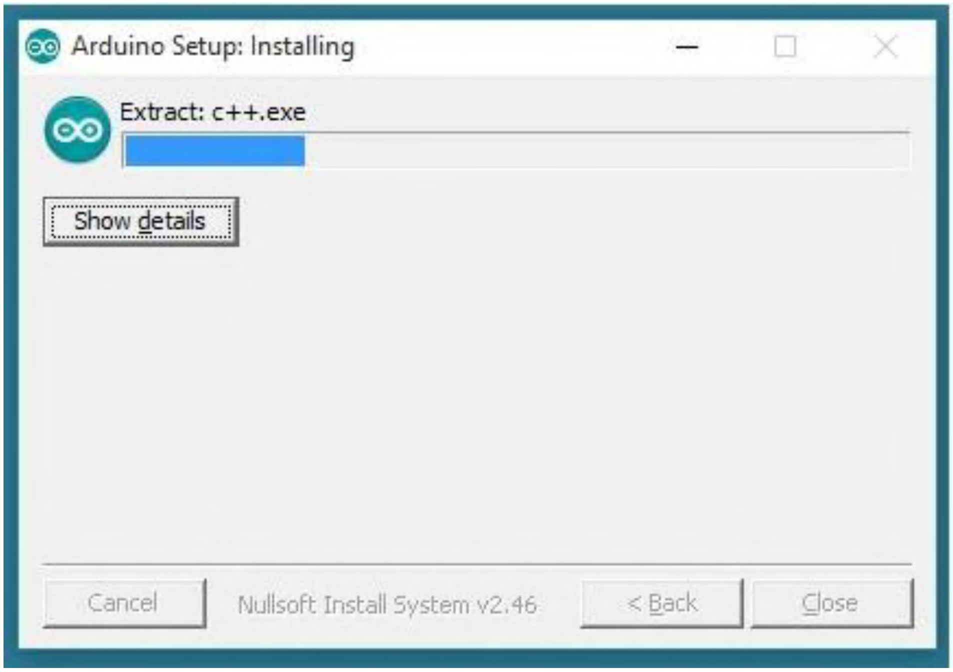


Fig:1.10 Installing directory

6. Successfully completed Installation of Software

7. Writing code on IDE platform

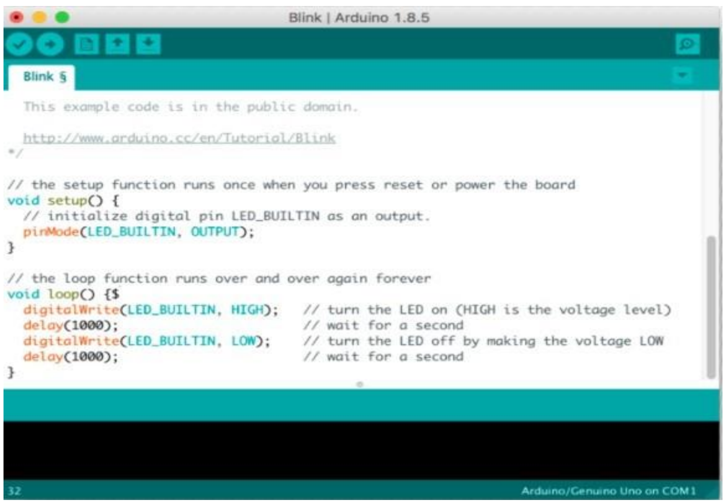


Fig:1.11 Source code written on IDE platform

1.4.2.2 PROTEUS SOFTWARE:

Through this software Schematic implementation of the embedded system will be done**.**

STEPS FOR DESIGNING SYSTEM ON PROTEUS DESIGN SUITE:

Step 1: Open ISIS software and select new design in File menu

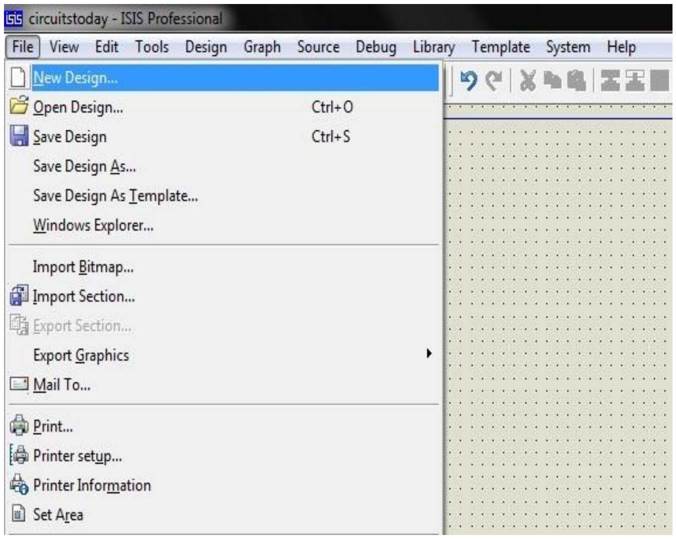


Fig:1.12 Simulation Procedure 1

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

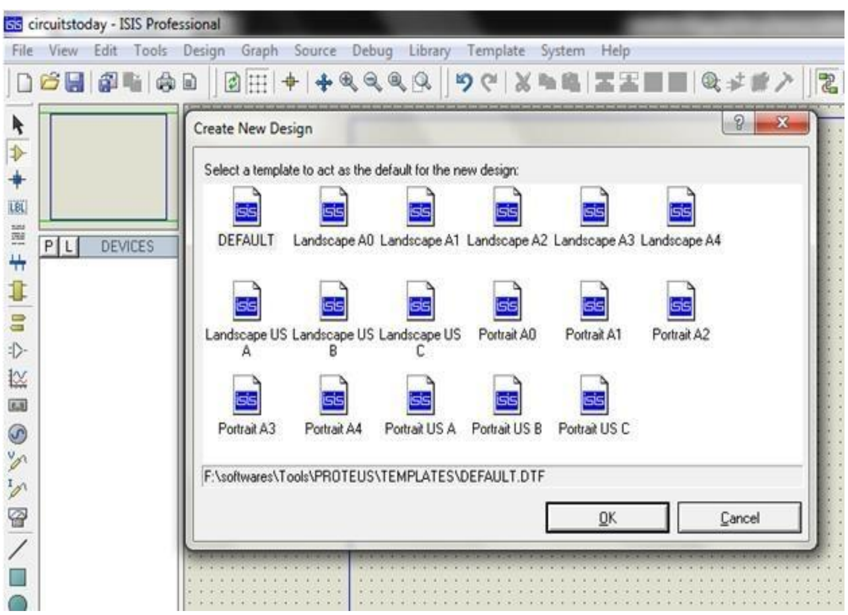


Fig:1.13 Simulation Procedure 2

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a

new folder for every layout as it generates other files supporting your design.

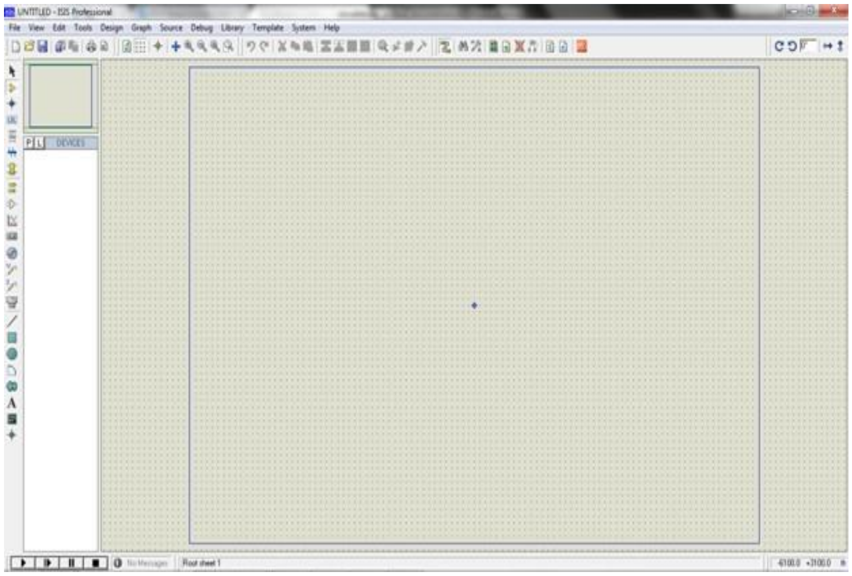


Fig:1.14 Simulation Procedure 3

Step 4: To Select components, Click on the component mode button.

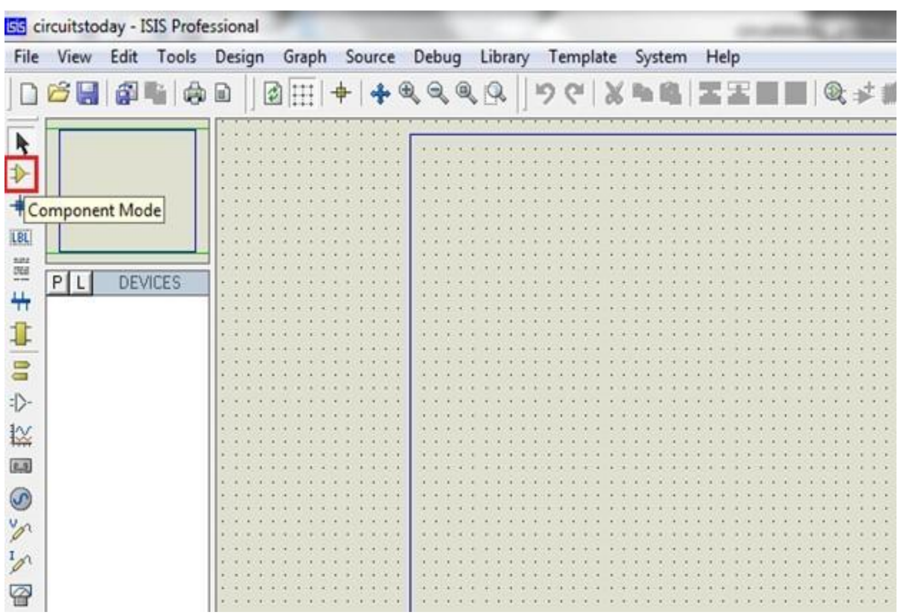


Fig:1.15 Simulation Procedure 4

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

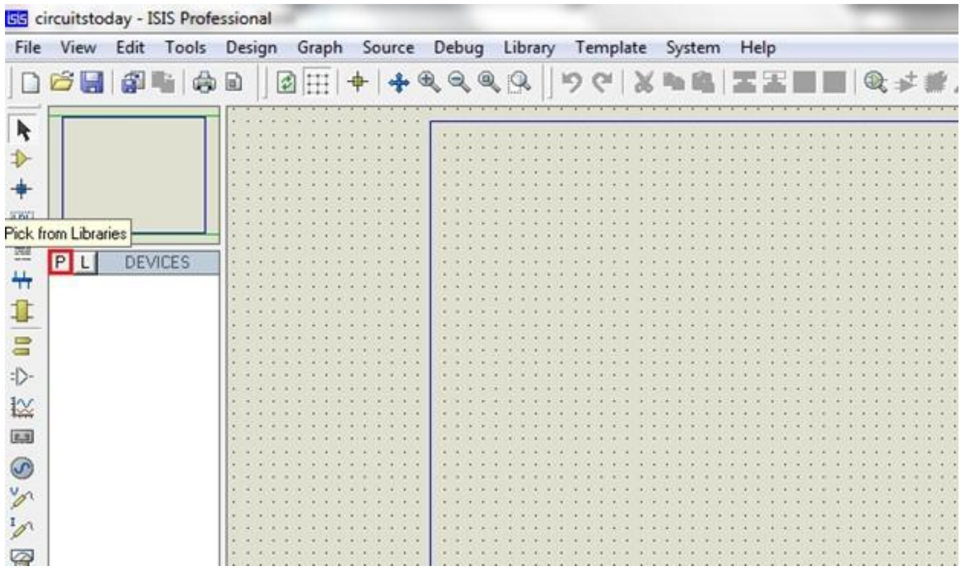


Fig:1.16 Simulation Procedure 5

Step 6: Select the components from categories or type the part name in Keyword's text box.

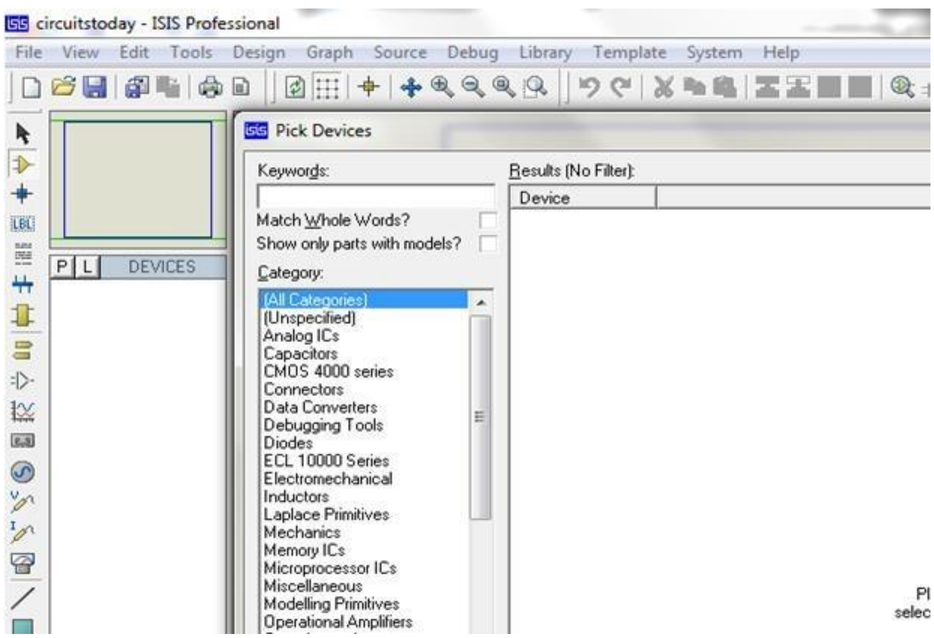


Fig:1.17 Simulation Procedure 6

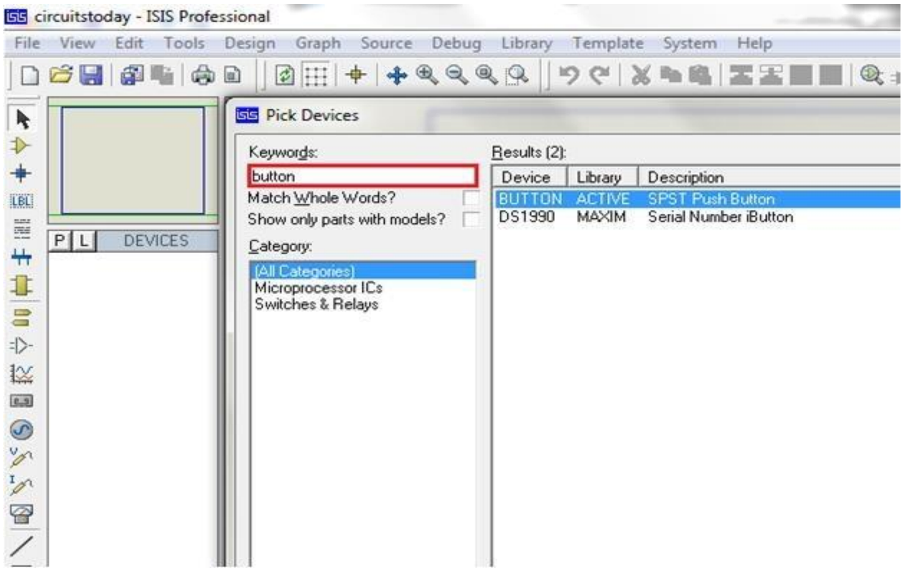


Fig:1.18 Simulation Procedure 7

Step 7: The selected components will appear in the devices list. Select the component and place itin the design sheet by left-cl.

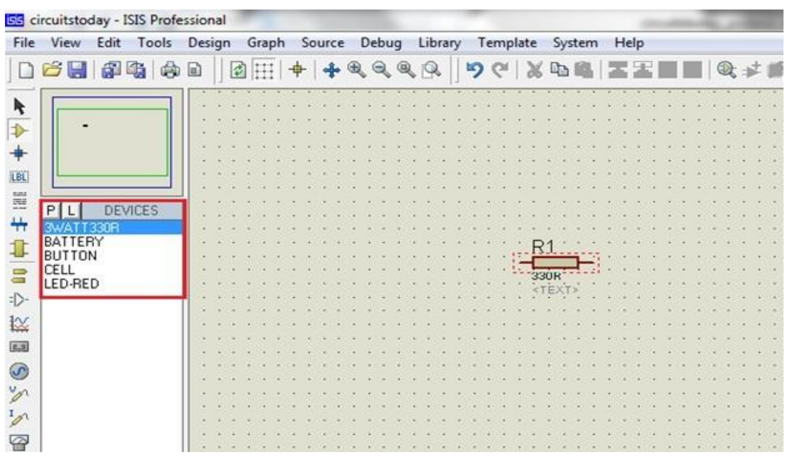


Fig.1.19 Simulation Procedure 8

Step 8: Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

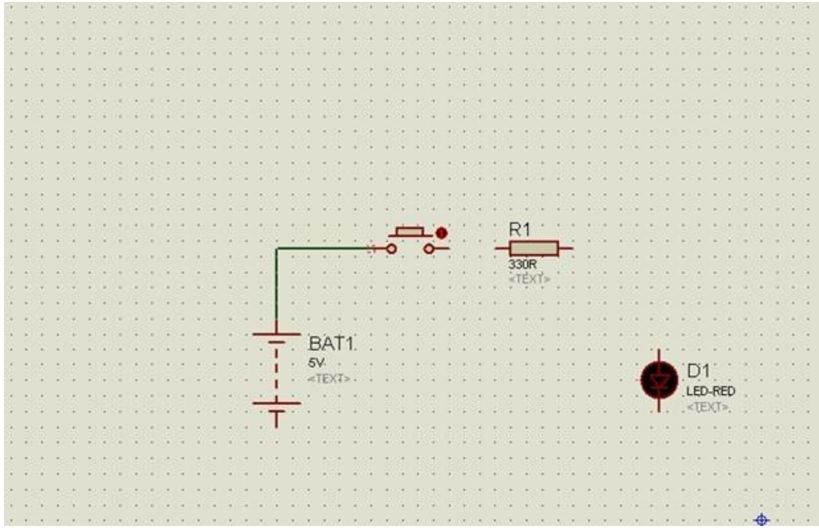


Fig:1.20 Simulation Procedure 9

Step 9: Double click on the component to edit the properties of the components and click on Ok.

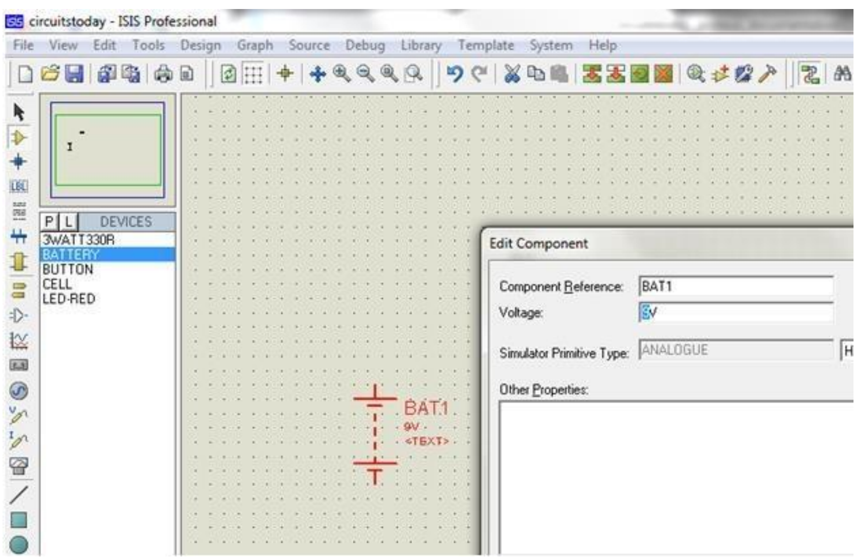


Fig.1.21 Simulation Procedure 10

Step 10: In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

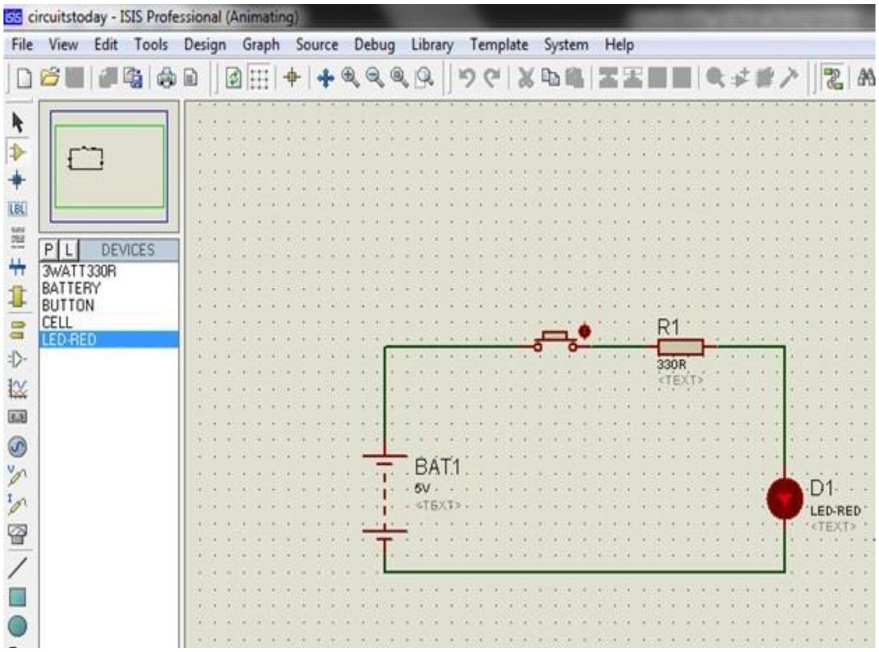


Fig:1.22 Simulation Procedure 11

Step 11: Simulation can be stepped, paused or stopped at any time.

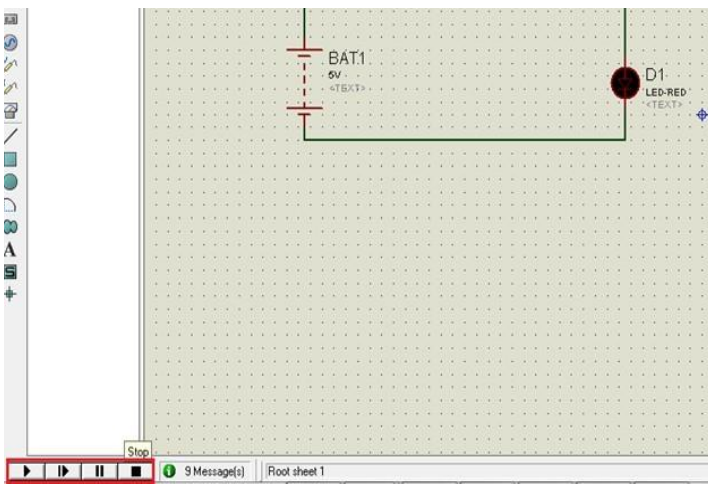


Fig:1.23 Simulation Procedure 12

1.5 PROCUMENT OF EQUIPMENT:

We brought all the required hardware components in an electronic manufacturing shop and before assembling and interfacing hardware components to Arduino UNO micro-controller, we connected individual sensors to Arduino in Proteus Software and understood working of every individual sensor in Proteus Software. Now, we assembled all sensors and interfaced to Arduino

UNO micro-controller using connecting wires through bread board and we wrote program to work all the hardware sensors in Arduino IDE, we understood the working of all sensors in a practical way. We observed temperature values and water level values inside the mine and all the information is sent to the online server.

1.6 ORGANISATION OF THE CHAPTERS

The Overall Overview of Documentation is:

1.6.1 INTRODUCTION

1.6.2 LITERATURE SURVEY

1.6.3 PROJECT DESIGN

1.6.4 PROJECT IMPLEMENTATION

1.6.5 PROJECT TESTING

1.6.6 CONCLUSION AND FUTURE ENHANCEMENT

CHAPTER 2

LITERATURE SURVEY AND EXISTING MODELS

2.1 RELATED WORK

Coal mining is the process of extracting coal from the ground. Steel and cement industries use coal as a fuel for extraction of iron from iron ore and for cement production. Underground mining industry comes to the category, where each and every parameter such as methane gas, high temperature, fire accidents and so on has to be monitored regularly. Safe production level of coal mine is still low, disasters in coal mine occur frequently, which lead to great loss of possession and life. The disasters happening in coal mine are due to the complexity of mine environment and the variety of work carried out in coal mine, so it is very necessary to monitor the working environment of coal mine. To get over this problem, projects has proposed a wireless sensor network's application in coal mining safety system. In this wireless sensor network's application system there will, be master controllers and slave controllers. Slave controllers will detect the danger and give alert through RF to master controller and it will raise the alarm in all tunnels and also raise message on IOT, which will help to take action as soon as possible. This monitoring and alerting system is powered by Atmega328 microcontroller for master controller and rollers for slave controller and consists of temperature sensor, water sensor, methane and carbon dioxide sensor and RF transmitter which collects temperature, humidity and methane values underground of coal mine.

2.2 EXISTING SYSTEM

In existing method, there is no data transmission from mine territory to watching station for checking the status of excavators and the environment. Difficult to screen each and every person to the barometrical status. There is no fast wellbeing endeavours available at the hour of emergency.

2.3 PROPOSED SYSTEM

IOT Based Coal Mine Safety Monitoring project consists of two hardware modules, one transmitter and one receiver. The main controller in both the modules is an Arduino board. The transmitter module is installed inside the coal mine. The transmitter module contains the smoke sensor, temperature sensor and methane sensor. The transmitter module also has an LCD, all the sensor data is displayed on the LCD screen by the Arduino controller. The RF transmitter present on the transmitter module sends the sensor data to the receiver module. The receiver module also has an Arduino controller, LCD display, and an RF receiver. The RF receiver receives the sensor values from the RF transmitter on the transmitter module. The received sensor values are displayed on the LCD screen. The Arduino also sends the sensor data to the remote IOT server using the WiFi module every two minutes. If any of the sensor values exceeds a particular threshold level, the buzzer is turned on to notify the concerned personnel. The remote server has an IOT platform installed on it which displays the relevant data using the GUI which helps the users in monitoring and system control.

CHAPTER 3

PROJECT DESIGN

3.1 OVERVEIW OF THE SYSTEM DESIGN

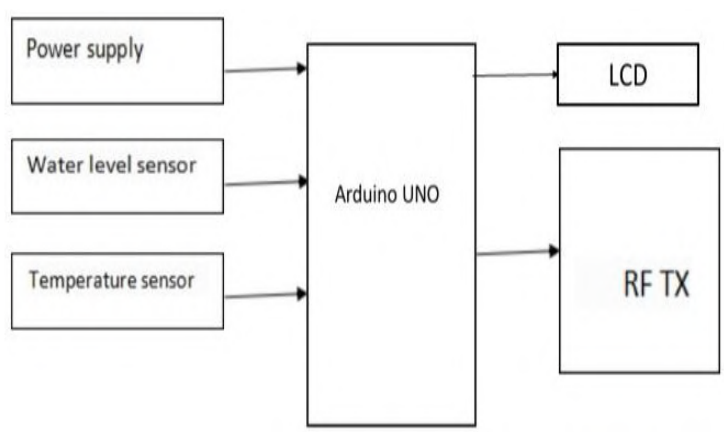


Fig 3.1 Block diagram transmitter

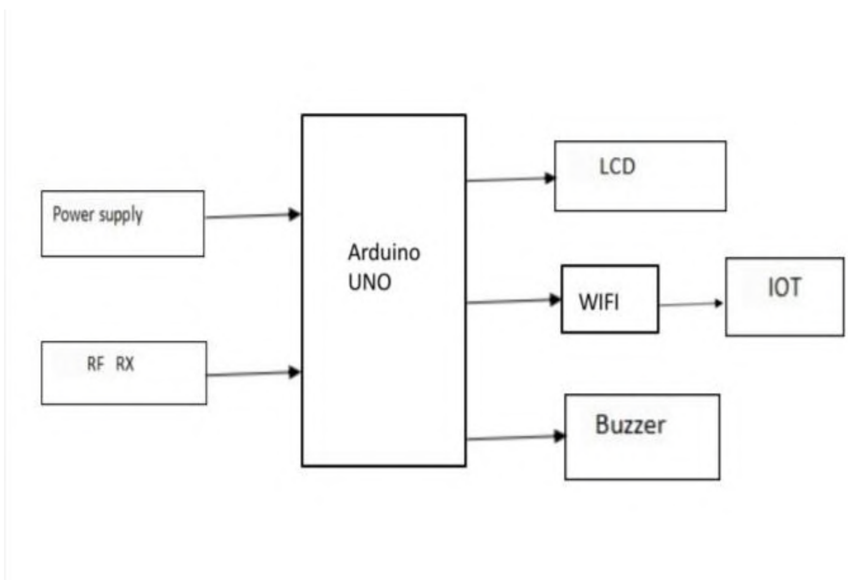


Fig 3.2 Block diagram of receiver

3.2 SCHEMATIC DESIGN

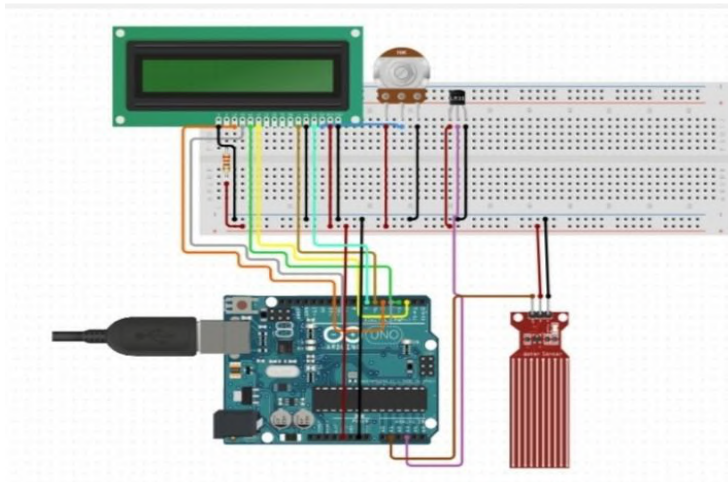


Fig 3.3 Schematic diagram of transmitter

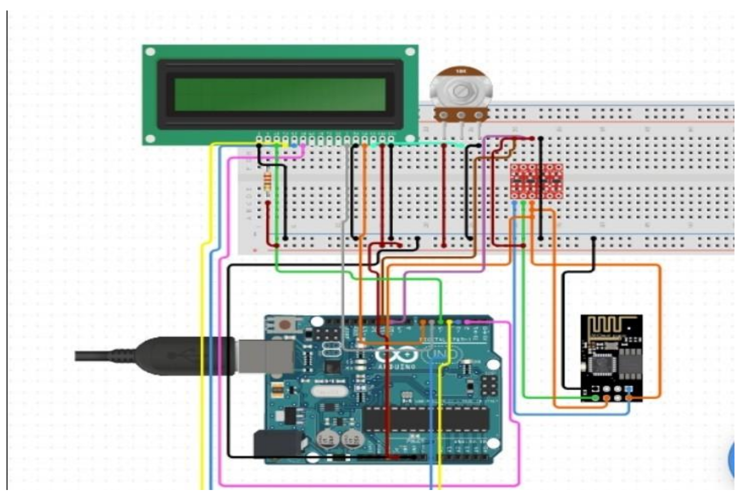


Fig 3.4 Schematic Diagram Of Receiver

3.2.1 SCHEMATIC EXPLANATION

The above schematic diagram of **IOT Based Coal Mine Safety Monitoring And Alerting System** explains the interfacing section of each component with Arduino Uno and input output modules

3.3 MODULES FUNCTIONALITIES:

3.3.1 Arduino

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

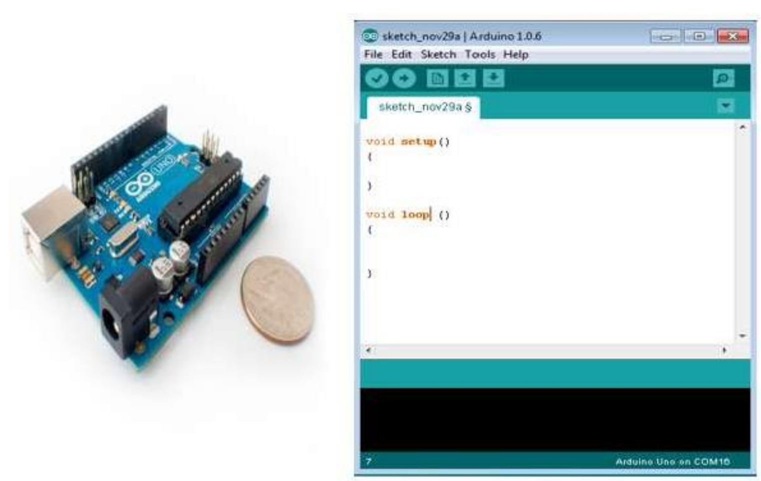


Figure 3.5 : Arduino UNO board

Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V. Here is a list of different Arduino boards available.

Table 3.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible

							Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.2Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 3.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro	3.3V	8MHz	54	16	14	4	FTDI-Compatible

Mini 3.3V							Header

3.4 Arduino boards based on AT91SAM3X8E microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

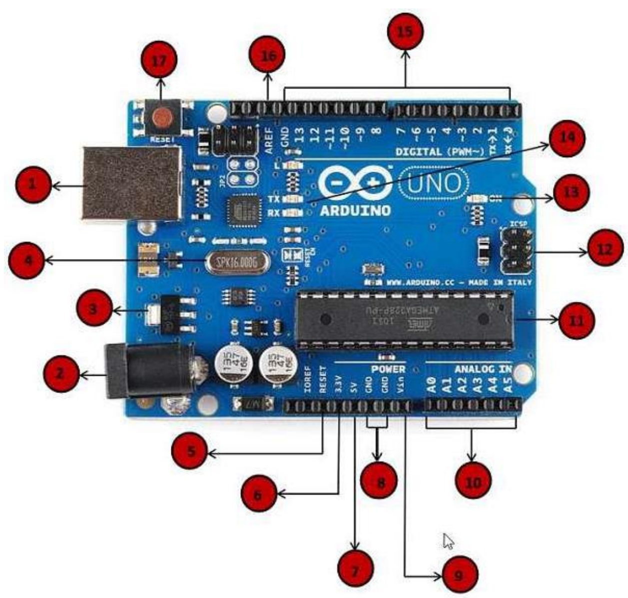


Figure 3.6 : Board description of Arduino Uno

	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).

3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.

13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

3.3.2 Liquid Cristal Display

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCDs connected to the contollers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines(RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

Description Of 16x2:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all-Parallel Ports. It however doesn't show the use of the Status Port as an input. So what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

Schematic Diagram

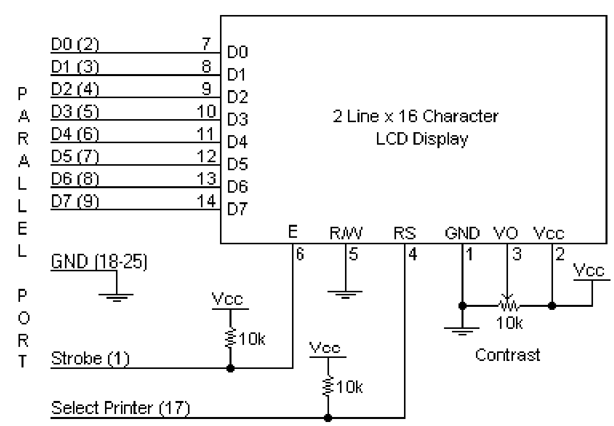


Figure 3.7: Schematic diagram of LCD

- Above is the quite simple schematic. The LCD panel's *Enable* and *Register Select* is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

- We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.
- The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.

16 x 2 Alphanumeric LCD Module Features:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

16 x 2 Alphanumeric LCD Module Specifications:

Pin	Symbol	Level	Function
1	V_{SS}	-	Power, GND
2	V_{DD}	-	Power, 5V
3	Vo	-	Power, for LCD Drive
4	RS	H/L	Register Select SignalH: Data Input L: Instruction Input
5	R/W	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)
6	E	H,H->L	Enable
7-14	DB0-DB7	H/L	Data Bus; Software selectable 4- or 8-bit mode

15	NC	-	**NOT CONNECTED**
16	NC	-	**NOT CONNECTED**

FEATURES:

• 5 x 8 dots with cursor

• Built-in controller (KS 0066 or Equivalent)

• + 5V power supply (Also available for + 3V)

• 1/16 duty cycle

• B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)

• N.V. optional for + 3V power supply

Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are:

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

Figure Address locations for a 1x16 line LCD

Even limited to character based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8, 16,20,24,32 and 40 characters are all standard, in one, two and four line versions.

Several different LC technologies exist. “Supertwist” types, for example, offer improved contrast and viewing angle over the older “twisted nematic” types. Some modules are available with back lighting, so that they can be viewed in dimly-lit conditions. The back lighting may be either “electro-luminescent”, requiring a high voltage inverter circuit, or simple LED illumination.

PIN DESCRIPTION:

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections).

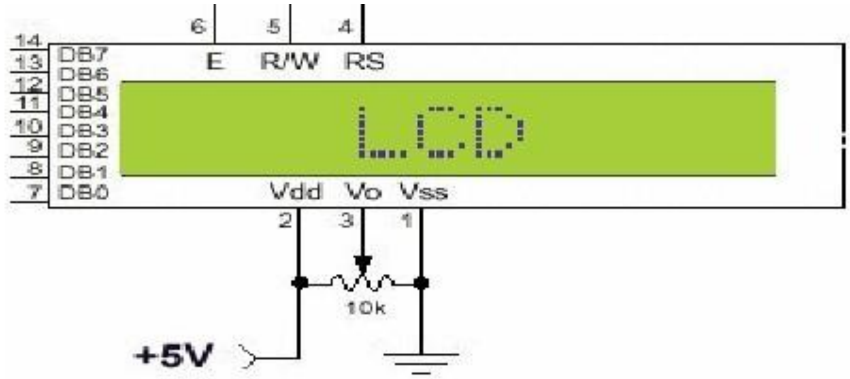


Figure3.8 Pin diagram of 1x16 lines LCD

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Figure 3.9 Pin specifications

CONTROL LINES:

EN: Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS: Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW: Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands, so RW will almost always be low.Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Logic status on control lines:

• E - 0 Access to LCD disabled

- 1 Access to LCD enabled

• R/W - 0 Writing data to LCD

- 1 Reading data from LCD

• RS - 0 Instructions

- 1 Character

Writing data to the LCD:

1) Set R/W bit to low

2) Set RS bit to logic 0 or 1 (instruction or character)

3) Set data to data lines (if it is writing)

4) Set E line to high

5) Set E line to low

Read data from data lines (if it is reading) on LCD:

1) Set R/W bit to high

2) Set RS bit to logic 0 or 1 (instruction or character)

3) Set data to data lines (if it is writing)

4) Set E line to high

5) Set E line to low

Entering Text:

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the d.i.l. switch pack with a of re-wiring is necessary.

LCD Commands:

There are some present commands instructions in LCD, which we need to send to LCD through some microcontroller. Some important command instructions are given below:

Table 3.5 Commands

Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left
0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

3.3.3 TEMPERATURE SENSOR (LM35)

General Description:

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in °Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1/4°C at room temperature and ±3/4°Cover a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package (Fig 3.2.1).

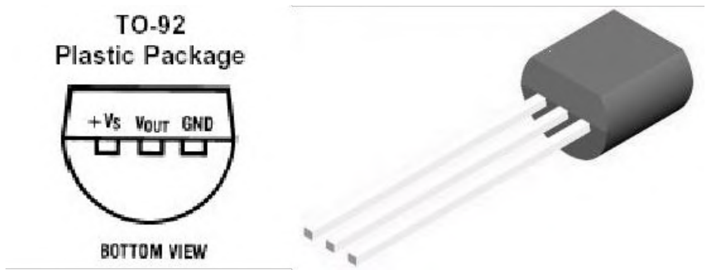


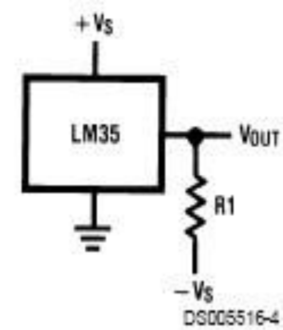
Fig 3.10 LM35

Applications:

The LM35 can be applied easily in the same way as other integrated-circuit temperature sensors. It can be glued or cemented to a surface and its temperature will be within about 0.01°C of the surface temperature. This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an intermediate temperature between the surface temperature and the air temperature. This is especially true for the TO-92 plastic package, where the copper leads are the principal thermal path to carry heat into the device, so its temperature might be closer to the air temperature than to the surface temperature. To minimize this problem, be sure that the wiring to the LM35, as it leaves the device, is held at the same temperature as the surface of interest. The easiest way to do this is to cover up these wires with a bead of epoxy which will insure that the leads and wires are all at the same temperature as the surface, and that the LM35 die's temperature will not be affected by the air temperature. The TO-46 metal package can also be soldered to a metal surface or pipe without damage. Of course, in that case the V− terminal of the circuit will be grounded to that metal. Alternatively, the LM35 can be mounted inside a sealed-end metal tube, and can then be dipped into a bath or screwed into a threaded hole in a tank. As with any IC, the LM35 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion. This is especially true if the circuit may operate at cold temperatures where condensation can occur. Printed-circuit coatings and varnishes such as Humiseal and epoxy paints or dips are often used to insure that moisture cannot corrode the

LM35 or its connections. These devices are sometimes soldered to a small light-weight heat fin, to decrease the thermal time constant and speed up the response in slowly-moving air. On the other

hand, a small thermal mass may be added to the sensor, to give the steadiest reading despite small deviations in the air temperature.



Choose $R_1 = -V_S/50\ \mu A$

$V_{OUT} = +1{,}500$ mV at +150°C

$= +250$ mV at +25°C

$= -550$ mV at −55°C

Features:

Calibrated directly in ° Celsius (Centigrade)

Linear + 10.0 mV/°C scale factor

0.5°C accuracy guaranteeable (at +25°C)

Rated for full −55° to +150°C range

Suitable for remote applications

Low cost due to wafer-level trimming

Operates from 4 to 30 volts

Less than 60 µA current drain

Low self-heating, 0.08°C in still air

Nonlinearity only ±1/4°C typical

Low impedance output, 0.1 W for 1 mA load

Testing of Temperature sensor LM35 using OP-AMP Circuit:

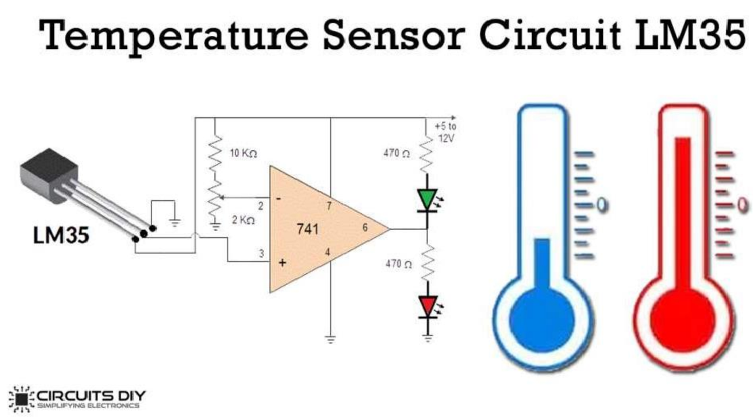


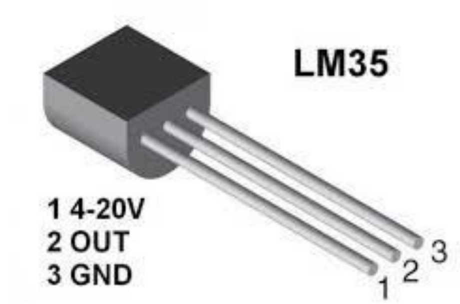
Fig3.11: LM35 testing circuit

Gain calculation:

Gain A = 1+ R_f / R_1
If R_f =4k & R_1=1k Then Gain A=5

Output of LM35:

		Before Amplification,	After amplification With gain of 5
0^0 C	$\longrightarrow$	0V,	0V
1^0C	$\longrightarrow$	10mV,	50mV
25^0C	$\longrightarrow$	250mV,	1.25V
100^0C	$\longrightarrow$	1000mV,	5V



3.3.4 WIFI Module:

The ESP8266 is a low-cost Wi-Fi module that can be integrated easily into IoT devices. We’ve featured several projects using this module, such as **How To Make Smart Home Electronics: A Smart Mailbox** and **How To Read Your Arduino’s Mind: Building A Childproof Lock**. This tutorial will walk you through setting up ESP8266 Wi-Fi module which can be used with Arduino. The ESP8266 comes in many models with different functionalities. We’ll be focusing on the ESP8266 ESP-01 module, the most common and basic one available.

What is ESP8266?

The ESP8266 is a small Wi-Fi module built around the ESP8266 chip that can connect your microcontroller to the internet wirelessly for a very small cost. It can be a great option for Internet of Things (IoT) projects, but can be difficult to work with for beginner hobbyists who do not have prior experience with the module. In this tutorial, we hope to show you how to interface the ESP8266 with an Arduino and perform some basic functions like connecting it to a Wi-Fi network.

- & 0.4ms guard interval
  Wake up and transmit packets in < 2msStandby power consumption of < 1.0mW (DTIM3)

The first feature to notice about the ESP8266 is its awkwardly spaced header pins. The module has 8 pins that serve different functions, but they are packed in a 4×2 arrangement that makes plugging the module into a breadboard impossible. This means that to prototype projects on a breadboard, you’ll need male-female jumper wires to connect the pins on the ESP8266 to rows on the breadboard. If you’d like to make your prototyping more compact, you can also purchase breadboard breakouts forthe ESP8266 such as this one. For prototyping, I chose to just use jumper wires.

The pinout for the ESP8266’s pins are according to the foll

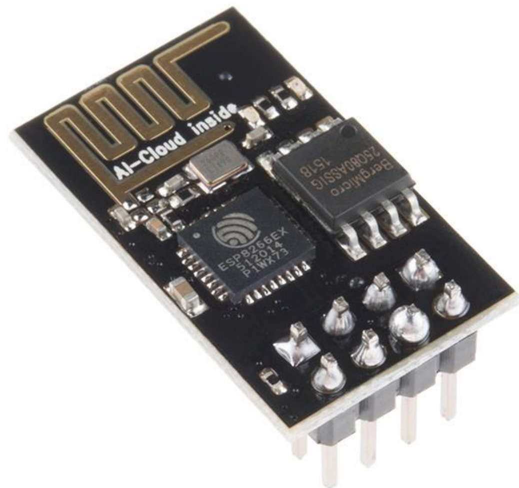


Fig 3.12 ESP8266 ESP-01 module / ©Sparkfun

ESP-01 Features – Spark fun**:**

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 1MB Flash Memory
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation

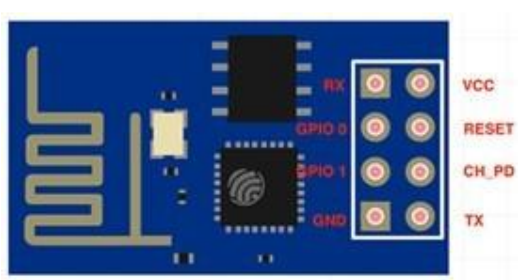


Fig 3.13. ESP8266 Pinout / ©Github.com/ydonnelly

RX: UART serial communication receive pin

GPIO 0: GPIO pin (unused in this project)

GPIO 1: GPIO pin (unused in this project)

GND: Connection to Ground

VCC: Connection to 3.3V Vcc (Vcc cannot exceed 3.3V!)

RESET: Reset pin (pull down to reset)

CH_PD: Chip enable and power down pin

TX: UART serial communication transmit pin

Note that the maximum voltage input for the ESP8266 is 3.3V. Any input voltage greater than 3.3V will damage the module! To program settings on the ESP8266, we'll first need to connect it to a serial terminal on a computer through which we can send it special commands. Settings that we'll have to program include, for example, the SSID and password for the wifi network the module will be connected to. To connect the ESP8266 to a computer and configure its settings, we'll need a USB to serial adapter with 3.3V logic, along with a serial terminal program. Fortunately for us, we have the Arduino and the Arduino IDE's serial monitor! This means that we'll just have to connect the ESP8266 module to the Arduino and upload a custom sketch to the Arduino.

Use the following diagram to connect the ESP8266 module to the Arduino:

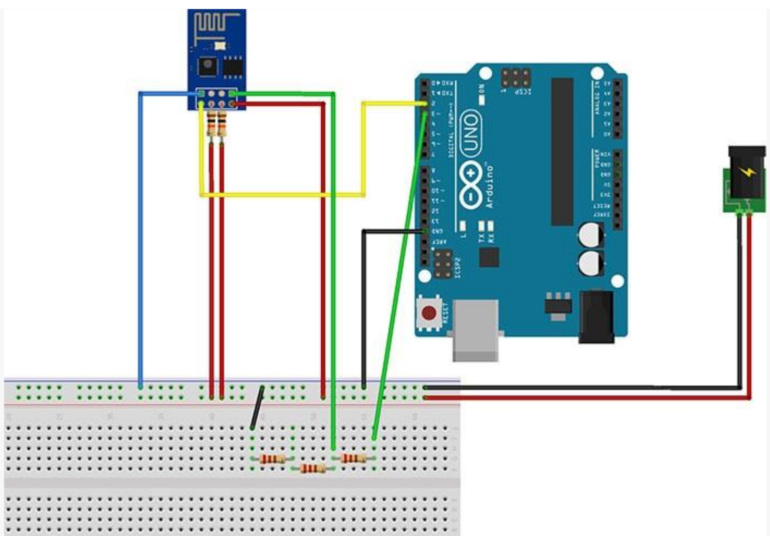


Figure 3.14. Connection between ESP8266 and Arduino Uno

You will need the following parts:

- Arduino Uno or similar module
- ESP8266 ESP-01 module
- 2 x 1kΩ resistors for CH_PD and RESET pull-up
- 3 x 220Ω resistors for serial line voltage divider

3.3 V regulated power supply for ESP8266 (you cannot plug the power line into the Arduino's 3.3V out pin because the ESP8266 draws a lot of current!)

- You can use something like this: https://www.sparkfun.com/products/114

- Breadboard and jumper wires
  - A couple of features of this circuit stand out immediately.

  - First, the ESP8266 module is powered by a 3.3V regulated power supply. As described earlier, the ESP8266 has a maximum voltage input rating of 3.3 volts. Using a power supply rated higher than this recommended 3.3 voltscould fry the module! When connecting the ESP8266 to this power supply, you should not only connect the Vcc and Ground pins to the power supply lines, but also connect the power supply's ground line to the Arduino's ground pin. Since the Arduino will be powered by the USB connection to the laptop, creating a common ground essentially creates a common reference (you can think of it as a baseline) to compare voltages and thereby interpret digital high and low signals.

Second, the receive (RX) line for the ESP8266 module is connected to the output of a resistor voltage-divider circuit. We do this to shift the serial communication logic level (the highs and lows of the digital signals that make up the serial communications) from a logic high of 5 volts on the Arduino to a logic high of 3.3 volts on the ESP8266 module. Again, the ESP8266 is specified with 3.3V logic, so connecting the module's receive line directly to the Arduino's transmit line could damage the device. We do not need to shift levels on the ESP8266's transmit line because the module's 3.3V logic high is a high enough voltage to also register as a logic high on the Arduino.

Finally, there are two pull-up resistors on the ESP8266's CH_PD and RESET pins. These are specified by the module's creators to ensure that the device functions properly while it is being used. To reset the ESP8266 or disable it, these pins must be pulled down, but since we do not want those processes to occur, we will instead pull these pins high through some 10kΩ pull-up resistors

3.3.5 Water Level Indicator

Description

This is the circuit diagram of a simple **corrosion free water level indicator** for home and industries.In fact the the level of any conductive non corrosive liquids can be measured using this

circuit.The circuit is based on 5 transistor switches.Each transistor is switched on to drive the corresponding LED , when its base is supplied with current through the water through the electrode probes.

One electrode probe is (F) with 6V AC is placed at the bottom of tank. Next probes are placed step by step above the bottom probe. When water is rising the base of each transistor gets electrical connection to 6V AC through water and the corresponding probe. Which in turn makes the transistors conduct to glow LED and indicate the level of water. The ends of probes are connected to corresponding points in the circuit as shown in circuit diagram. Insulated Aluminium wires with end insulation removed will do for the probe. Arrange the probes in order on a PVC pipe according to the depth and immerse it in the tank.AC voltage is use to prevent electrolysis at the probes. So this setup will last really long. I guarantee at least a 2 years of maintenance free operation .That's what I got and is still going.

Components

T1 – T5 BC 548 or 2N2222 Transistors

R1-R5 2.2K 1/4 W Resistors

R6-R10 22K 1/4 W Resistors

D1 – D5 LED's (color your choice)

Notes:

Use a transformer with 6V 500 mA output for power supply. Do not use a rectifier! we need pure AC. Use good quality insulated Aluminium wire for probes. If Aluminium wires are not available try Steel or Tin. Copper is the worst. Try the circuit first on a bread board and if not working properly, make adjustments with the resistance values .This is often needed because conductivity of water changes slightly from place to place .The type number of the transistors used here are not critical and any small signal NPN transistor will do the job. Few other suitable type numbers are BC546, BC107, PN2222, BC337, BF494, ZTX300, BEL187 etc. The circuit can be enclosed in a plastic box with holes for revealing the LEDs .

Water Level Indicator Circuit Diagram and Sensor Arrangement

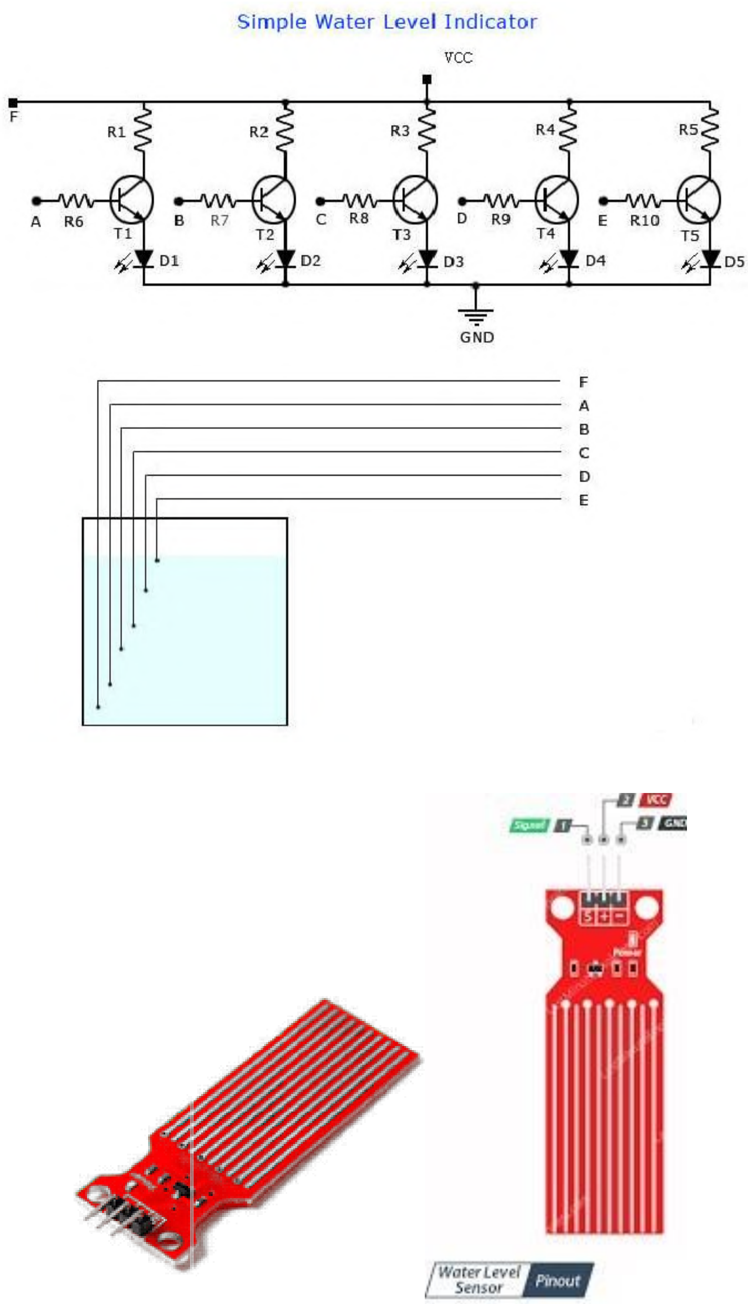


Figure 3.15 Water level indicator

3.3.6 Radio Frequency

The nRF24L01+ transceiver module is designed to operate in 2.4 GHz worldwide ISM frequency band and uses GFSK modulation for data transmission. The data transfer rate can be one of 250kbps, 1Mbps and 2Mbps.

WHAT IS 2.4GHZ ISM BAND?

2.4 GHz band is one of the Industrial, Scientific, and Medical (ISM) bands reserved internationally for the use of unlicensed low-powered devices. Examples are Cordless phones, Bluetooth devices, near field communication (NFC) devices, and wireless computer networks (WiFi) all use the ISM frequencies.

Power consumption

The operating voltage of the module is from **1.9 to 3.6V**, but the good news is that the **logic pins are 5-volt tolerant**, so we can easily connect it to an Arduino or any 5V logic microcontroller without using any logic level converter.

The module supports programmable output power viz. 0 dBm, -6 dBm, -12 dBm or -18 dBm and consumes unbelievably around **12 mA during transmission** at 0 dBm, which is even lower than a single LED. And best of all, it consumes 26 µA in standby mode and 900 nA at power down mode. That’s why they’re the go-to wireless device for low-power applications.

SPI Interface

The nRF24L01+ transceiver module communicates over a 4-pin Serial Peripheral Interface (**SPI**) with a maximum data rate of **10Mbps**. All the parameters such as frequency channel (125 selectable channels), output power (0 dBm, -6 dBm, -12 dBm or -18 dBm), and data rate (250kbps, 1Mbps, or 2Mbps) can be configured through SPI interface.

The SPI bus uses a concept of a Master and Slave, in most common applications our Arduino is the Master and the nRF24L01+ transceiver module is the Slave. Unlike the I2C bus the number of

slaves on the SPI bus is limited, on the Arduino Uno you can use a **maximum of two SPI slaves** i.e. two nRF24L01+ transceiver modules.

Here are complete specifications:

Frequency Range	2.4 GHz ISM Band
Maximum Air Data Rate	2 Mb/s
Modulation Format	GFSK

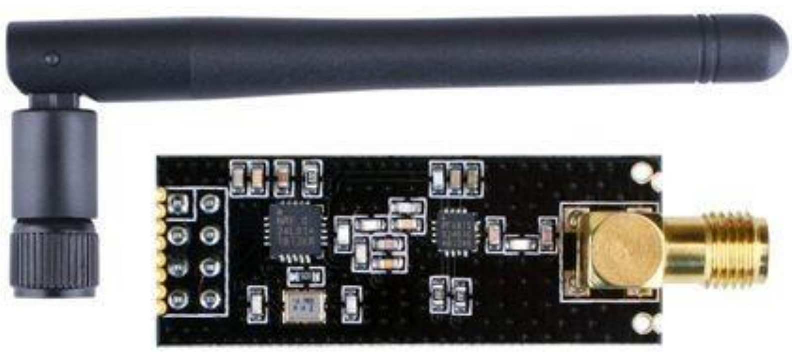
Max. Output Power	0 dBm
Operating Supply Voltage	1.9 V to 3.6 V
Max. Operating Current	13.5mA
Min. Current(Standby Mode)	26µA
Logic Inputs	5V Tolerant
Communication Range	800+ m (line of sight)

nRF24L01+ module Vs nRF24L01+ PA/LNA module

There are a variety of modules available based upon the nRF24L01+ chip. Below are the most popular versions.

nRF24L01+ Wireless Module

The first version uses on-board antenna. This allows for a more compact version of the breakout. However, the smaller antenna also means a lower transmission range. With this version, you'll be able to communicate over a distance of **100 meters**. Of course that is outdoors in an open space. Your range indoors, especially through walls, will be slightly weakened.

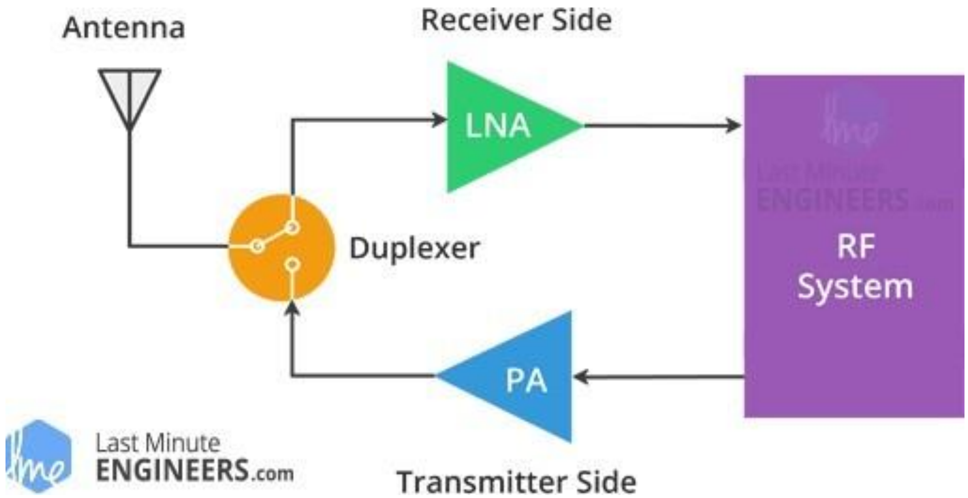


nRF24L01+ PA LNA Wireless Transceiver Module with External Antenna

The second version comes with a SMA connector and a duck-antenna but that's not the real difference. The real difference is that it comes with a special RFX2401C chip which integrates the PA, LNA, and transmit-receive switching circuitry. This range extender chip along with a duck-antenna helps the module achieve a significantly larger transmission range about **1000m**.

What is PA LNA?

The PA stands for **Power Amplifier**. It merely boosts the power of the signal being transmitted from the nRF24L01+ chip. Whereas, LNA stands for **Low-Noise Amplifier**. The function of the LNA is to take the



nRF24L01+ PA/LNA Block Diagram

extremely weak and uncertain signal from the antenna (usually on the order of microvolts or under -100 dBm) and amplify it to a more useful level (usually about 0.5 to 1V)

The low-noise amplifier (LNA) of the receive path and the power amplifier (PA) of the transmit path connect to the antenna via a duplexer, which separates the two signals and prevents the relatively powerful PA output from overloading the sensitive LNA input. For more information check out this article on digikey.com

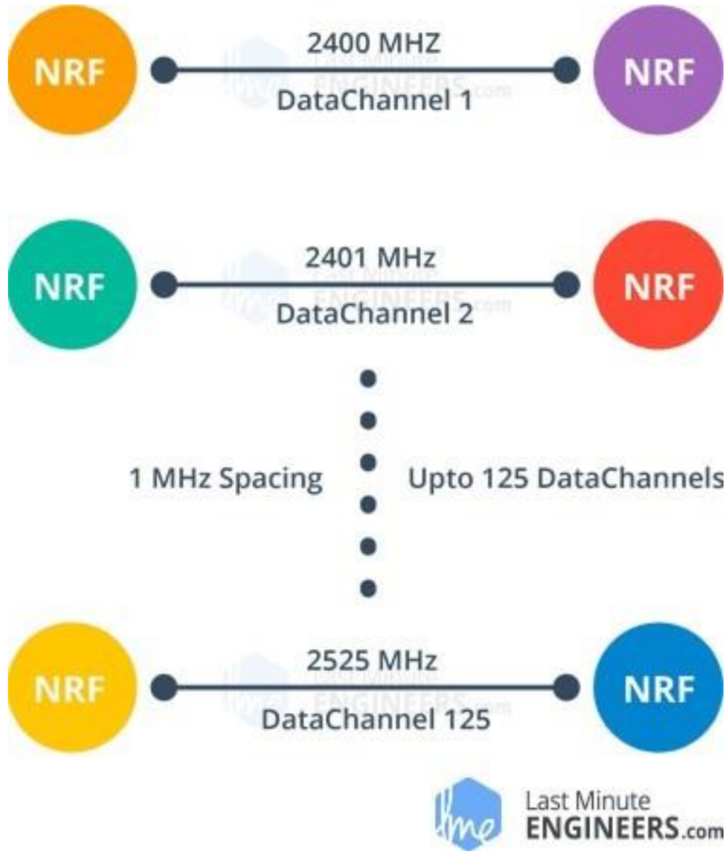
Except this difference, both modules are drop-in compatible. Meaning, if you build your project with one you can just unplug it and use another without need to make any changes to the system.

How nRF24L01+ transceiver module works?

RF Channel Frequency

The nRF24L01+ transceiver module transmits and receives data on a certain frequency called **Channel**. Also in order for two or more transceiver modules to communicate with each other, they need to be on the same channel. This channel could be any frequency in the 2.4 GHz ISM band or to be more precise, it could be between 2.400 to 2.525 GHz (2400 to 2525 MHz).

Each channel occupies a bandwidth of less than 1MHz. This gives us 125 possible channels with 1MHz spacing. So, the module can use 125 different channels which give a possibility to have a network of 125 independently working modems in one place.



The channel occupies a bandwidth of less than 1MHz at 250kbps and 1Mbps air data rate. However at 2Mbps air data rate, 2MHz bandwidth is occupied (wider than the resolution of RF channel frequency setting). So, to ensure non-overlapping channels and reduce cross-talk in 2Mbps mode, you need to keep 2MHz spacing between two channels.

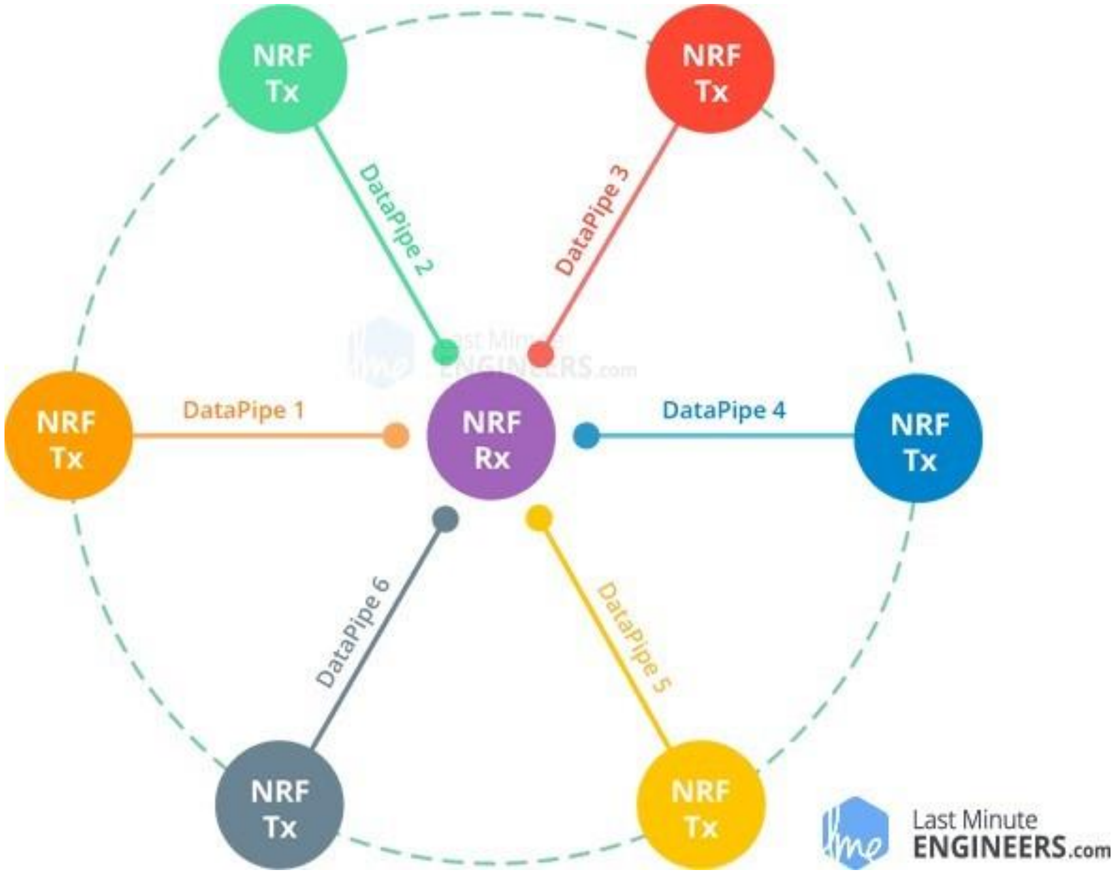
RF channel frequency of your selected channel is set according to the following formula:

$$Freq_{(Selected)} = 2400 + CH_{(Selected)}$$

For example, if you select 108 as your channel for data transmission, the RF channel frequency of your channel would be 2508MHz (2400 + 108)

nRF24L01+ Multiceiver Network

The nRF24L01+ provides a feature called **Multiceiver**. It’s an abbreviation for **Multi**ple Transmitters Single Re**ceiver**. In which each RF channel is logically divided into 6 parallel data channels called **Data Pipes**. In other words, a data pipe is a logical channel in the physical RF Channel. Each data pipe has its own physical address (Data Pipe Address) and can be configured. This can be illustrated as shown below.

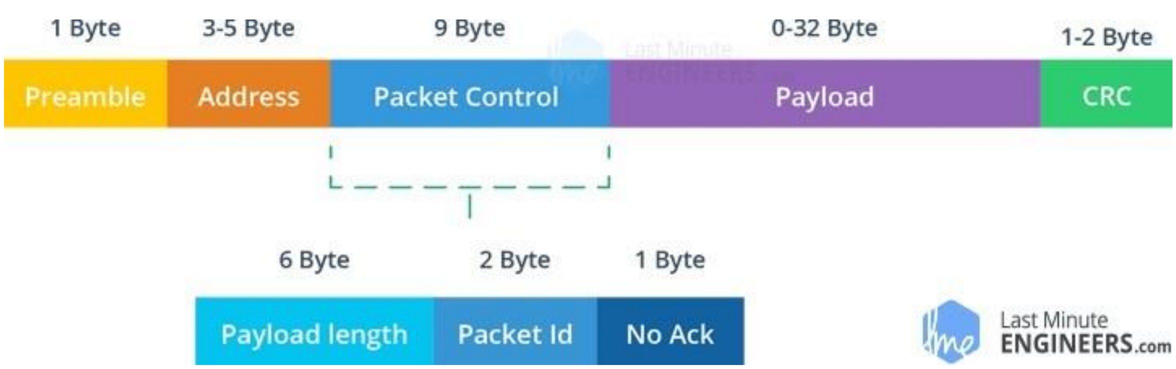


nRF24L01+ Multiceiver Network – Multiple Transmitters Single Receiver

To simplify the above diagram, imagine the primary receiver acting as a hub receiver collecting information from 6 different transmitter nodes simultaneously. The hub receiver can stop listening any time and acts as a transmitter. But this can only be done one pipe/node at a time.

Enhanced Shock Burst Protocol

The nRF24L01+ transceiver module uses a packet structure known as Enhanced ShockBurst. This simple packet structure is broken down into 5 different fields, which is illustrated below.



nRF24L01+ Enhanced Shock Burst Packet Structure

The original ShockBurst structure consisted only of Preamble, Address, Payload and the Cyclic Redundancy Check (CRC) fields. Enhanced Shock Burst brought about greater functionality for more enhanced communications using a newly introduced **Packet Control Field** (PCF).

This new structure is great for a number of reasons. Firstly, it allows for variable length payloads with a payload length specifier, meaning payloads can vary from 1 to 32 bytes.

Secondly, it provides each sent packet with a packet ID, which allows the receiving device to determine whether a message is new or whether it has been retransmitted (and thus can be ignored).

Finally, and most importantly, each message can request an acknowledgement to be sent when it is received by another device.

nRF24L01+ Automatic Packet Handling

Now, let's discuss three scenarios to get a better understanding of how two nRF24L01+ modules transact with each other.

Transaction with acknowledgement and interrupt This is an example of positive scenario. Here the transmitter starts a communication by sending a data packet to the receiver. Once the whole packet is transmitted, it waits (around 130 µs) for the acknowledgement packet (ACK packet) to receive. When the receiver receives the packet, it sends ACK packet to the transmitter. On receiving the ACK packet the transmitter asserts interrupt (IRQ) signal to indicate the new data is available.

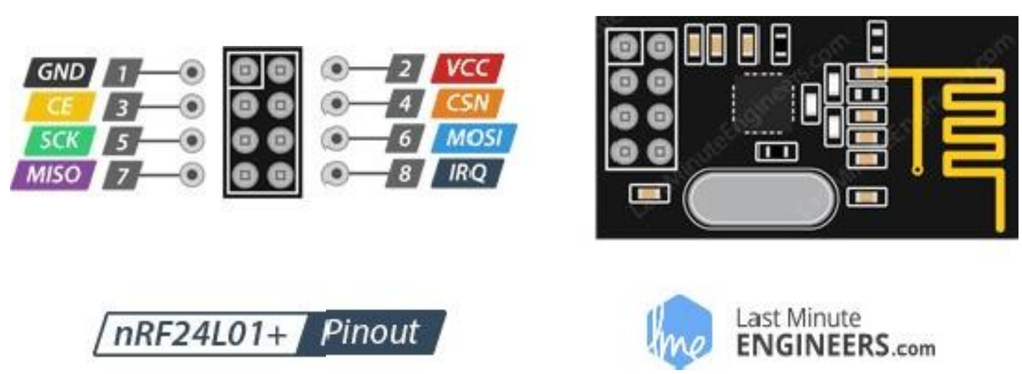
Transaction with data packet lost This is a negative scenario where a retransmission is needed due to loss of the packet transmitted. After the packet is transmitted, the transmitter waits for the ACK packet to receive. If the transmitter doesn't get it within Auto-Retransmit-Delay (ARD) time, the packet is retransmitted. When the retransmitted packet is receiv d by the receiver, the ACK packet is transmitted which in turn generates interrupt at the transmitter.

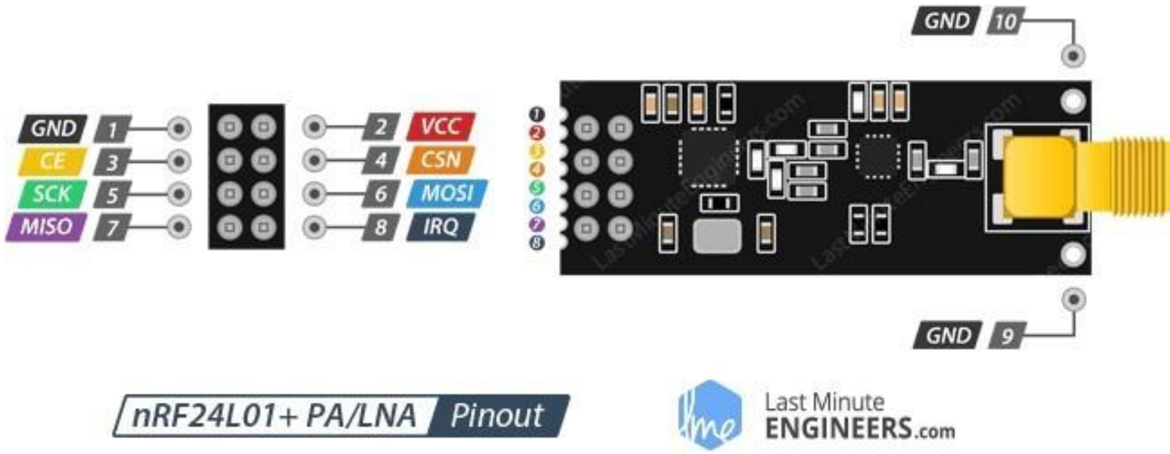
Transaction with acknowledgement lost This is again a negative scenario where a retransmission is needed due to loss of the ACK packet. Here even if the receiver receives the packet in the first attempt, due to the loss of ACK packet, transmitter thinks the receiver has not got the packet at all. So, after the Auto-Retransmit-Delay time is over, it retransmits the packet. Now when receiver receives the packet containing same packet ID as previous, it discards it and sends ACK packet again.

This whole packet handling is done automatically by the nRF24L 1+ chip without involvement of the microcontroller.

nRF24L01+ Transceiver Module Pinout

Let's have a look at the pinout of both the versions of nRF24L01+ transceiver Module.





GND is the Ground Pin. It is usually marked by encasing the pin in a square so it can be used as a reference for identifying the other pins.

VCC supplies power for the module. This can be anywhere from 1.9 to 3.9 volts. You can connect it to 3.3V output from your Arduino. Remember connecting it to 5V pin will likely destroy your nRF24L01+ module!

CE (Chip Enable) is an active-HIGH pin. When selected the nRF24L01 will either transmit or receive, depending upon which mode it is currently in.

CSN (Chip Select Not) is an active-LOW pin and is normally kept HIGH. When this pin goes low, the nRF24L01 begins listening on its SPI port for data and processes it accordingly.

SCK (Serial Clock) accepts clock pulses provided by the SPI bus Master.

MOSI (Master Out Slave In) is SPI input to the nRF24L01.

MISO (Master In Slave Out) is SPI output from the nRF24L01.

IRQ is an interrupt pin that can alert the master when new data is available to process

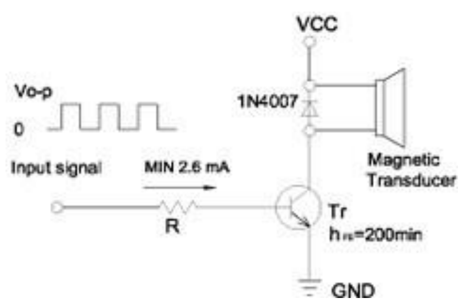
3.3.7 BUZZER:

Magnetic Transducer

Magnetic transducers contain a magnetic circuit consisting of a iron core with a wound coil and a yoke plate, a permanent magnet and a vibrating diaphragm with a movable iron piece. The diaphragm is slightly pulled towards the top of the core by the magnet's magnetic field. When a positive AC signal is applied, the current flowing through the excitation coil produces a fluctuating magnetic field, which causes the diaphragm to vibrate up and down, thus vibrating air. Resonance amplifies vibration through resonator consisting of sound hole(s) and cavity and produces a loud sound.

Magnetic Buzzer (Sounder)

Buzzers like the TMB-series are magnetic audible signal devices with built-in oscillating circuits. The construction combines an oscillation circuit unit with a detection coil, a drive coil and a magnetic transducer. Transistors, resistors, diodes and other small devices act as circuit devices for driving sound generators. With the application of voltage, current flows to the drive coil on primary side and to the detection coil on the secondary side. The amplification circuit, including the transistor and the feedback circuit, causes vibration. The oscillation current excites the coil and the unit generates an AC magnetic field corresponding to an oscillation frequency. This AC magnetic field magnetizes the yoke comprising the magnetic circuit. The oscillation from the intermittent magnetization prompts the vibration diaphragm to vibrate up and down, generating buzzer sounds through the resonator.



Recommended Driving Circuit for Magnetic Transducer

Specifications:

Rated Voltage: A magnetic buzzer is driven by 1/2 square waves (V o-p).

Operating Voltage: For normal operating. But it is not guaranteed to make the minimum Sound Pressure Level (SPL) under the rated voltage.

Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.

Direct Current Resistance: The direct current resistance is measured by ammeter directly.

Sound Output: The sound output is measured by decibel meter. Applying rated voltage and 1/2 square waves, and the distance of 10 cm.

Rated Frequency: A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.

Operating Temp: Keep working well between -30℃ and +70℃.

Driving methods: AX series with built drive circuit will be the best choice when we cannot provide frequency signal to a buzzer, it only needs direct current.

Dimension: Dimension affects frequency, small size result in high frequency.

Voltage: Depend on V o-p (1/2 square waves)

Fixed methods: From the highest cost to the lowest- DIP, wires/ connector, SMD.

Soldering methods: AS series is soldered by hand, the frequency is lower because of the holes on the bottom. On the other hand, we suggest AC series for the reflow soldering, the reliability is better.

How to choose a buzzer

There are many different kinds of buzzer to choose, first we need to know a few parameters, such as voltage, current, drive method, dimension, mounting type, and the most important thing is how much SPL and frequency we want.

Operating voltage: Normally, the operating voltage for a magnetic buzzer is from 1.5V to 24V, for a piezo buzzer is from 3V to 220V. However, in order to get enough SPL, we suggest giving at least 9V to drive a piezo buzzer.

Consumption current: According to the different voltage, the consumption current of a magnetic buzzer is from dozens to hundreds of mill amperes; oppositely, the piezo type saves much more electricity, only needs a few mill amperes, and consumes three times current when the buzzer start to work.

Driving method: Both magnetic and piezo buzzer have self drive type to choose. Because of the internal set drive circuit, the self drive buzzer can emit sound as long as connecting with the direct current. Due to the different work principle, the magnetic buzzer need to be driven by 1/2 square waves, and the piezo buzzer need square waves to get better sound output.

Dimension: The dimension of the buzzer affects its SPL and the frequency, the dimension of the magnetic buzzer is from 7 mm to 25 mm; the piezo buzzer is from 12 mm to 50 mm, or even bigger.

Connecting way: Dip type, Wire type, SMD type, and screwed type for big piezo buzzer are usually seen.

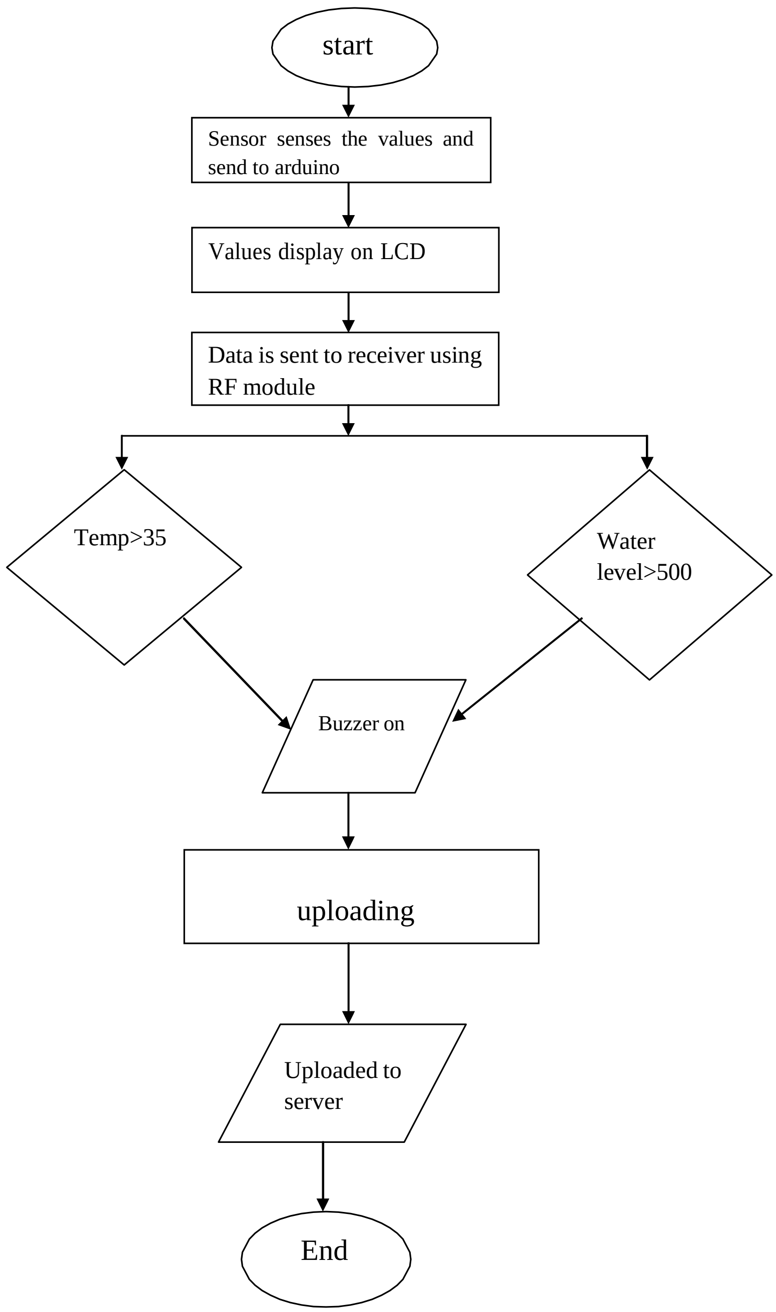
Sound Pressure Level (SPL): Buzzer is usually tested the SPL at the distance of 10 cm, if distance double, the SPL will decay about 6 dB; oppositely, the SPL will increase 6 dB when the distance is shortened by one time. The SPL of the magnetic buzzer can reach to around 85 dB/ 10 cm; the piezo buzzer can be designed to emit very loud sound, for example, the common siren, are mostly made of piezo buzzer.

Fig 3.16 BUZZER

CHAPTER 4

PROJECT IMPLEMENTATION

4.1 IMPLEMENTATION STAGES



4.2 EXPERIMENTAL RESULTS:

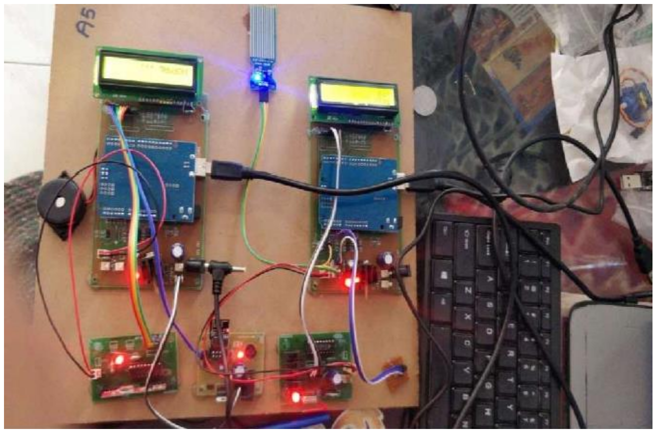


Figure 4.1: All the components connected on board

Figure 4.2 : Temperature and water level values displayed on LCD

Figure 4.3: "HIGH TEMPERATURE" displayed on LCD

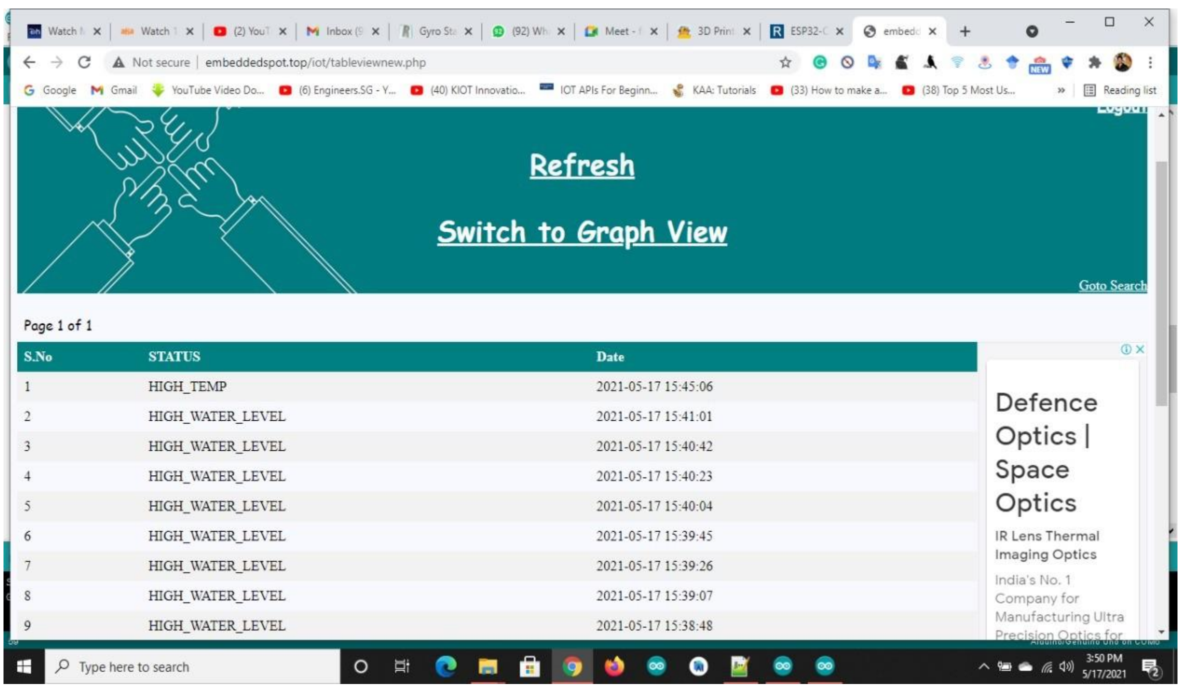


Figure 4.4: Message sent to Server

CHAPTER 5

ADVANTAGES AND APPLICATIONS

5.1 Advantages:

- The project is built from easily available and reasonably priced components. Therefore, the cost is reasonable and maintenance is easy.

5.2 Application

- This project is crucial in coal mining industry and is critical for the health and safety of the mine workers.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

This system is displaying the parameters on the monitoring unit; it will be helpful to all miners present inside the mine to save their life before any casualty occurs. Alarm triggers when sensor values crosses the threshold level. This system also stores all the data in the computer for future inspection.

6.2 FUTURE SCOPE

This is a safe critical project, therefore this system can be improved by making it fail-safe. We can implement the fail-safe operation using redundancy in the system. If the one module fails then the parallel module will take over the operation.

REFERENCES

[1] B. Wang et al., "Quantitative examination of break association by assessing crack shape during incidentally stopping organized cracking," J. Oil Sci. Eng., vol. 172, pp. 559–571, Jan. 2019.

[2] D. Shang et al., "Local topsy-turvy breaking to build complex crack system in tight permeable repositories during subsurface coal mining: A trial study," J. Gaseous petrol Sci. Eng., vol. 59, pp. 343–353, Jan. 2018.

[3] M. He, W. J. Chen, L. Tian, B. Shao, and Y. Lin, "Plantmicrobial synergism: A viable methodology for the remediation of shale-gas breaking flowback and delivered water," J. Unsafe Mater., vol. 363, no. 5, pp. 170–178, Feb. 2019.

[4] G. Liu, T. Zhang, M. He, M. Li, and J. Li, "Thermal execution examination of penetrating even wells in high temperature arrangements," Appl. Therm. Eng., vol. 78, pp. 217–227, Mar. 2015.

[5] K. H. S. M. Sampath et al., "Effect of coal development on CO2-based water driven breaking process in coal crease gas repositories," Fuel, vol. 236, pp. 179–189, Jan. 2019.

[6] B. Figueiredo, C. F. Tsang, J. Rutqvist, and A. Niemi, "Study of water driven breaking forms in shale arrangements with complex geographical settings," J. Oil Sci. Eng., vol. 152, pp. 361–374, Apr. 2017.

[7] Z. Zhang, J. Mao, X. Yang, J. Zhao, and G. S. Smith, "Advances in waterless cracking advances for whimsical repositories," Energy Sources A, Recovery, Utilization, Environ. Impacts, vol. 41, no. 2, pp. 237–251, 2019.

[8] M. Marjani et al., "Big IoT information examination: Architecture, openings, and open research difficulties," IEEE Access, vol. 5, pp. 5247–5261, 2017.

[9] F. Ghofrani, Q. He, R. M. P. Goverde, and X. Liu, "Recent utilizations of enormous information investigation in railroad transportation frameworks: A review," Transp. Res. C, Emerg. Technol., vol. 90, pp. 226–246, May 2018.

APPENDIX CODE

Transmitter code:

```
/*#include <SoftwareSerial.h>
SoftwareSerialZIGBEE(2, 3); //TX, RX respetively
*/
#include <LiquidCrystal.h>
LiquidCrystallcd(13,12,11,10,9,8);
int  sw1=2,sw2=3,sw3=4,sw4=5;
int water = A4,temp = A5;


void setup() {
Serial.begin(9600);
pinMode(sw1,OUTPUT);
pinMode(sw2,OUTPUT);
pinMode(sw3,OUTPUT);
pinMode(sw4,OUTPUT);
  //d1 = 0,d2= 0 ,d3 = 0,d4 =0 ;
lcd.begin(16,2);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("***WELCOME***");
delay(3000);

}


void loop() {

        /*    digitalRead(sw1)==HIGH;
digitalRead(sw2)==HIGH;
digitalRead(sw3)==HIGH;
digitalRead(sw4)==HIGH;*/

int td = analogRead(temp)/10;
td = td + 20;
intwd = analogRead(water);
lcd.clear();lcd.setCursor(0,0);lcd.print("T:");lcd.print(td);
lcd.setCursor(0,8);lcd.print("w:");lcd.print(wd);
delay(1000);

if( td > 35 )
        {
 lcd.clear();lcd.setCursor(0,0);lcd.print("HIGH  TEMP");

digitalWrite(sw1,LOW);
digitalWrite(sw2,HIGH);
digitalWrite(sw3,HIGH);
```

```
digitalWrite(sw4,HIGH);


delay(2000);

        }

else if( wd> 500)
        {
lcd.clear();lcd.setCursor(0,0);lcd.print("HIGH WATER LEVEL");

          // delay(2000);
digitalWrite(sw2,LOW);
digitalWrite(sw1,HIGH);
digitalWrite(sw3,HIGH);
digitalWrite(sw4,HIGH);
        }
else
        {
digitalWrite(sw2,HIGH);
digitalWrite(sw1,HIGH);
digitalWrite(sw3,HIGH);
digitalWrite(sw4,HIGH);
        }

          /*     if(wd< 20)
        {
lcd.clear();lcd.setCursor(0,0);lcd.print("LOW WATER LEVEL");

digitalWrite(sw3,LOW);
digitalWrite(sw1,HIGH);
digitalWrite(sw2,HIGH);
digitalWrite(sw4,HIGH);

delay(2000);

        }*/

     /*
if(d1 == 1 && d2 == 0 && d3 ==0 && d4 == 0)
    {
lcd.clear();lcd.setCursor(0,1);lcd.print("readvoice");delay(1000);
    }
if(d1 == 0 && d2 == 1 && d3 ==0 && d4 == 0)
    {
lcd.clear();
lcd.setCursor(0,1);
lcd.print("readvoice");
delay(1000);
    }
if(d1 == 0 && d2 == 0 && d3 ==1 && d4 == 0)
    {
lcd.clear();
```

```
lcd.setCursor(0,1);
lcd.print("readvoice");
delay(1000);
    }
     */

}
```

Receiver code:

```
#include <LiquidCrystal.h>
LiquidCrystallcd(13,12,11,10,9,8);

intbuz = 7;
int sw1=2,sw2=3,sw3=4,sw4=5;
char buff[200],k=0;


void upload1(unsigned char *chr);
char res[130];
char check(char* ex,int timeout)
{
inti=0;
int j = 0,k=0;
while (1)
  {
sl:
if(Serial.available() > 0)
    {
res[i] = Serial.read();
if(res[i] == 0x0a || res[i]=='>' || i == 100)
      {
i++;
res[i] = 0;break;
      }
i++;
    }
j++;
if(j == 30000)
    {
k++;
      //Serial.Sprintln("kk");
      j = 0;
    }
if(k > timeout)
    {
     // Serial.println("timeout");
return 1;
     }
  }//while 1
if(!strncmp(ex,res,strlen(ex)))
```

```
  {
    //Serial.println("ok..");
return 0;
   }
else
  {
   // Serial.print("Wrong ");
   // Serial.println(res);
i=0;
gotosl;
   }
}
voidserialFlush(){
while(Serial.available() > 0) {
char t = Serial.read();
  }
}
const char* ssid = "project";
const char* password = "project8215";

void setup() {
char ret;
Serial.begin(9600);
pinMode(sw1,INPUT);
pinMode(sw2,INPUT);
pinMode(sw3,INPUT);
pinMode(sw4,INPUT);
pinMode(buz,OUTPUT);

digitalWrite(buz,HIGH);


// d1 = 0,d2= 0 ,d3 = 0,d4 =0 ;
lcd.begin(16,2);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("***WELCOME***");
delay(3000);
Serial.begin(115200);
st:
Serial.println("ATE0");
ret = check((char*)"OK",50);
Serial.println("AT");
ret = check((char*)"OK",50);
if(ret != 0)
  {
delay(1000);
gotost;
  }

lcd.clear();lcd.setCursor(0, 0);lcd.print("CONNECTING");
Serial.println("AT+CWMODE=1");
ret = check((char*)"OK",50);
```

```
cagain:

serialFlush();
Serial.print("AT+CWJAP=\"");
Serial.print(ssid);
Serial.print("\",\"");
Serial.print(password);
Serial.println("\"");
if(check((char*)"OK",300))gotocagain;
Serial.println("AT+CIPMUX=1");
delay(1000);
lcd.clear();lcd.setCursor(0,  0);lcd.print("CONNECTED");

}


void loop() {


if ((digitalRead(sw1)==LOW) && (digitalRead(sw2)==HIGH) && (digitalRead(sw3)==HIGH)
&& (digitalRead(sw4)==HIGH))

   {
lcd.clear();lcd.setCursor(0,1);lcd.print("HIGH TEMP ...");
upload1("HIGH_TEMP");
digitalWrite(buz,LOW);delay(2000);digitalWrite(buz,HIGH);

   }
if ((digitalRead(sw1)==HIGH) && (digitalRead(sw2)==LOW) && (digitalRead(sw3)==HIGH)
&& (digitalRead(sw4)==HIGH))
   {
lcd.clear();lcd.setCursor(0,1);lcd.print("HIGH  WATER  LEVEL  ...");
upload1("HIGH_WATER_LEVEL");
digitalWrite(buz,LOW);delay(2000);digitalWrite(buz,HIGH);

   }
if ((digitalRead(sw1)==HIGH) && (digitalRead(sw2)==HIGH) && (digitalRead(sw3)==HIGH)
&& (digitalRead(sw4)==HIGH))
   {
lcd.clear();lcd.setCursor(0,1);lcd.print("NORMAL  ...");delay(1000);

   }
  /* if ((digitalRead(sw1)==HIGH) && (digitalRead(sw2)==HIGH) &&
(digitalRead(sw3)==HIGH) && (digitalRead(sw4)==HIGH))
   {
lcd.clear();lcd.setCursor(0,1);lcd.print("LOW  WATER  LEVEL  ...");
  // upload1("WRONG_USER");
digitalWrite(buz,LOW);delay(2000);digitalWrite(buz,HIGH);
   }*/


}
```

```
char bf2[50];
void upload1(unsigned char *chr)
{

lcd.clear();lcd.setCursor(0, 1);lcd.print("UPLOADING");
delay(2000);
serialFlush();
Serial.println("AT+CIPSTART=4,\"TCP\",\"embeddedspot.top\",80");

delay(8000);
sprintf(buff,"GET http://embeddedspot.top/iot/storedata.php?name=iot303&s1=%s\r\n\r\n",chr);
serialFlush();
sprintf(bf2,"AT+CIPSEND=4,%u",strlen(buff));
Serial.println(bf2);

delay(5000);


serialFlush();
Serial.print(buff);

delay(2000);

Serial.println("AT+CIPCLOSE");
lcd.setCursor(0, 1);lcd.print("UPLOADDED");

}
```

A

Project report

On

APPLICATION OF MIMO-OFDM SYSTEM IN UAV COMMUNICATION NETWORK

Submitted by

1) Ms.G.Samyuktha (17K81A0416)

2) Ms.K.Kusuma (17K81A0428)

3) Mr. K.Vijay (18K85A0405)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

DR.A. CHAITHANYA KRISHNA

Professor

Department of Electronics and Communication Engineering

ST.MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute, NBA &NAAC A+ Accredited

Dhulapally, Secunderabad-500 100.

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the project entitled APPLICATION OF MIMO-OFDM SYSTEM IN UAV COMMUNICATION NETWORK, is being submitted by **1) Ms. G.Samyuktha (17K81A0416) 2) Ms. K.Kusuma (17K81A0428) 3) Mr. K.Vijay (18k85A0405)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

DR.A.Chaitanya Krishna
Department of electronics and communication engineering

DR.B.Hari Krishna
Head of the Department
Department of electronics and communication engineering

Internal Examiner

External Examiner

Place: Hyderabad

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **G.SAMYUKTHA** WITH ROLL NO.**17k81A0416, KORABOHINA VIJAY** WITH ROLL NO.**18K85A0405**, **KOTTETI KUSUMA** WITH ROLL NO.**17k81A0428**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**APPLICATION OF MIMO-OFDM TECHNOLOGY IN UAV COMMUNICATION NETWORK**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014 Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of **Bachelor of Technology** in Department of ‘ELECTRONICS AND COMMUNICATION ENGINEERING’, session: 2017 – 2021, St. Martin’s Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled Application of MIMO-OFDM system in UAV communication network is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university foraward of any degree.

G. SAMYUKTHA (17K81A0416)
K. KUSUMA (17K81A0428)
K. VIJAY (18K85A0405)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna**, **Head &Professor**, the Dept. of electronics and communication engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept. project coordinator Dr.A.Anand** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide, **DR.A.Chaitanya Krishna, professor**, department of ECE and our class project coordinator **MS,A.Aleti Soumya, Assistant professor**, department of ECE for guidance and encouragement in carrying out this project work

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. G.Samyuktha (17K81A0416)
2. K.Kusuma (17K81A0428)
3. K.Vijay (18K85A0405)

ABSTRACT

With the application of various new mission loads in unmanned aerial vehicle (UAV), how to improve the communication quality of UAV data link has become a widespread concern. With the rapid development of UAV, multi-UAVs cooperative self-organizing network can accomplish tasks more reliably, efficiently and economically than a single large UAV. This paper analyzes the characteristics of mobile self-organizing network composed of multiple UAVs, and focuses on how to improve the communication quality of UAV network. After analyzing the technical principles and system models of multiple input multiple output (MIMO) and orthogonal frequency division multiplexing (OFDM) technologies, a MIMO-OFDM communication system is constructed. The simulation results show that the application of MIMO-OFDM technology in UAV communication network can effectively improve the data transmission ability.

LIST OF FIGURES

LIST OF TABLES

LIST OF OUTPUT SCREENS

LIST OF ACRONYMS AND DEFINITIONS

UAV	Unmanned Aerial Vehicle
MIMO	Multiple Input Multiple Output
OFDM	Orthogonal Frequency Division Multiplexing
UAS	Unmanned Aerial System
RPA	Remotely Piloted Aircraft
GPS	Global Positioning System
FPV	First Person View
IEEE	Institute of Electrical and Electronics Engineer
WIFI	Wireless Fidelity
HSPA	High Speed Packet Access
WIMAX	Worldwide Interoperability for Microwave Access
ITU	International Telecommunication Union
SM	Spatial multiplexing
BER	Bit Error Rate
STBC	Space Time Block Coding
WLAN	Wireless Lan
LTE	Long Term Evolution
CSI	Channel State Information
SNR	Signal to noise ratio
FDA	Frequency diverse array
SISO	Single Input Single Output
LOS	Line of Sight
FFT	Fast Fourier transform
ISI	Inter symbol Interference
CDMA	Code Division Multiple Access
RF	Radio Frequency
DAB	Digital Audio Broadcasting
DTTB	Digital Terrestrial Television Broadcasting
UMTS	Universal Mobile Telecommunication System
GSM	Global System for Mobile Communication
HIPERLAN	High performance Radio LAN
CM	Complex Multiplication
LLR	Log likelihood ratio
MMSE	Minimum mean squared error
STS-SD	Single tree search sphere decoding
MMSESIC	Minimum mean squared error with successive interference cancellation
QPSK	Quadrature Phase Shift Keying
IFFT	Inverse fast Fourier transform
OSIC	Ordered Successive Interference Cancellation
ABEP	Average bit error probability
ADC	Analog to digital converter
DAC	Digital to analog convertor

TABLE OF CONTENTS

CHAPTER-1

1.1 OVERVIEW OF PROJECT

Choosing transmission technology is one of the key issues to improve the performance of UAV communication network. Multiple input multiple output (MIMO) technology can reduce the influence of multipath fading without increasing power and sacrificing bandwidth and make full use of the channel capacity of wireless communication system.

1.2 OBJECTIVES OF THE STUDY

At present, UAV and ground control center mainly use a single antenna for communication, whose channel capacity seriously restricts the maximum transmission rate, resulting in long transmission time. So, the main objective of this project is to overcome the drawbacks of existing system.

1.3 SCOPE OF PROJECT

This project is going to cover the technical properties of MIMO and OFDM systems and mostly focuses on combination of these two systems for the improvement of transmission ability.

1.4 ORGANISATION OF CHAPTERS

1.4.1 Introduction.

1.4.2 Literature survey.

1.4.3 Proposed system.

1.4.4 Main program.

1.4.5 Simulation results and analysis.

1.4.6 Conclusion and future enhancement.

1.4.1 INTRODUCTION

With the application of new mission loads in UAV, such as high-resolution photoelectric camera, infrared thermal imager and synthetic aperture radar, a large number of real time color images, thermal images, videos and other information will be generated, resulting in a geometric increase in the amount of returned information. How to improve the communication capacity of data link has become a hot issue in the development of UAV system. At present, UAV and ground control center mainly use a single antenna for communication, whose channel capacity seriously restricts the maximum transmission rate, resulting in long transmission time. Multiple input multiple output (MIMO) technology can reduce the influence of multipath fading without increasing power and sacrificing bandwidth and make full use of the channel capacity of wireless communication system, which can effectively overcome the problem of insufficient communication capacity and low transmission rate of UAV. UAV communication network requires multiple UAVs for data exchange and transmission, and wireless transmission technology is used as the underlying communication means. But the physical characteristics of wireless channel determine the network bandwidth it can provide. Due to the coupling of collision, signal attenuation, multipath interference and other factors generated by wireless channels, UAV terminals can only obtain a bandwidth much smaller than the theoretical maximum bandwidth value. Therefore, choosing the appropriate physical layer transmission technology is one of the key issues to improve the performance of UAV communication network. MIMO technology can greatly improve the quality and capacity of a communication system by implementing full diversity, and it can resist multipath fading to some extent, but it is powerless for frequency selective fading. Orthogonal frequency division multiplexing (OFDM) technology can effectively utilize spectrum resources and resist frequency selective fading. As a combination of MIMO and OFDM technology, MIMO-OFDM can not only improve the communication capacity, but also effectively combat frequency selective fading and improve the data transmission ability.

What is an UAV?

An unmanned aerial vehicle (UAV) or uncrewed aerial vehicle, commonly known as a drone, is an aircraft without a human pilot on board. UAVs are a component of an unmanned aircraft system (UAS), which include additionally a ground-based controller and a system of communications with the UAV

Fig: 1.1 UAV

The flight of UAVs may operate under remote control by a human operator remotely piloted aircraft (RPA) or with various degrees of autonomy, such as autopilot assistance, up to fully autonomous aircraft that does not allow human intervention. Compared to crewed aircraft, UAVs were originally used for missions too "dull, dirty or dangerous" for humans. While drones originated mostly in military applications, their use is rapidly finding many more applications including aerial photography, product deliveries, agriculture, policing and surveillance, infrastructure inspections, science, smuggling, and drone racing. Extremely small UAVs, in some cases hand-launched, are used to extend the vision of ground combat units beyond their front lines.

Applications of Unmanned Aerial Vehicle (UAV)

Some of the most common commercial applications and uses for UAV Drones are:

1. Aerial Photography & Videography
2. Real estate photography
3. Mapping & Surveying
4. Asset Inspection

5. Payload carrying
6. Agriculture
7. Bird Control
8. Crop spraying
9. Crop monitoring
10. Multispectral/thermal/NIR cameras
11. Live streaming events
12. Roof inspections
13. Emergency Response
14. Search and Rescue
15. Marine Rescue
16. Disaster zone mapping
17. Disaster Relief
18. Forensics
19. Mining
20. Firefighting
21. Monitoring Poachers
22. Insurance
23. Aviation
24. Meteorology
25. Product Deliver

ADVANTAGES OF UNMANNED AERIAL VEHICLE(UAV)

MAINTAINING SAFE ENVIRONMENT - UAVs are utilized in numerous occurrences due to their advancement in safety. With their remote control abilities, Drones monitor locations, communicate possible hazards, and notify threatening conditions .such as oil and gas refineries, pipelines and flare stacks. Not only this, Drone Technology is employed in the military during high-risk periods as well. Their features allow them to obtain real-time data to create and preserve a safe environment.

COST SAVING TECHNOLOGY - As drone's applicability becomes more extensive, their prices also drive towards being more pocket-friendly. People now acquire Drones not just for their industrial practices but also to fulfill their tech-savvy gadget's passion. UAVs are no longer equipped only for the military, law authorities, or the elite. Since UAVs take over several workforces, vehicles, and operation activities in commercial uses, many costs are preserved. For example, a Drone is more economical to buy, sustain, and fuel than airplanes for inspections. In addition you don’t need to hire a ladder, aerial lifts, and other heavy equipment.

QUALITY OF AERIAL IMAGING-AERIAL PHOTOGRAPHY With their high-resolution cameras furnished with top-notch sensors, UAVs can take excellent Aerial Photographs, aerial videos and accumulate large volumes of accurate data. The data obtained is transformed into detailed 3D Maps and 3D Models for a complete analysis. 3D Mapping is particularly relevant to disclose cracks, damages, or other hazardous elements in disaster areas. Drones, when paired along with high-resolution images or 4K video abilities, is well-known for live streaming significant events such as entertainment, personal, political, and global affairs.

PRECISION - UAVs appropriate GPS (the Global Positioning System) in their software, which is why they can be programmed and guided precisely to specific locations. For example, in Precision Agriculture, a Drone Aircraft is employed to perform many farming obligations like pesticide spraying, identification of weeds, monitoring crop health, crop damage, crop assessment, field soil analysis, Irrigation Monitoring etc. This feature of precision through the GPS conserves time and expenses for farmers.

EASY CONTROLLABLE OR DEPLOYABLE - The regular advancement in drone-control technology allows operators to quickly deploy and operate drones even with a relatively minimal technical background. With an extensive range of low-cost drones available for several purposes, drones are open to a broad spectrum of operators. Unmanned aerial vehicles (UAVs) have a more comprehensive range of movement, fly lower in all Directions, and can navigate effortlessly when contrasted to a crewed aircraft.

SECURITY - Another advantage that weighs out the pros and cons of a drone is the security centered on them. With relevant permissions and licenses, drone operators can utilize an Unmanned Aircraft System (UAS) to render safety and surveillance to private organizations, potential venues, and other expenses. Drones can also accumulate reliable information from natural catastrophes to support safety and recovery efforts.

MINIMIZES OBVIOUS DANGER AND HEALTH RISKS - With the support of a Drone, numerous dangers like elevation, wind, weather, and radiation that were earlier suffered by crew members have been replaced with more viable and safer alternatives. Drones facilitate straightforward and secure inspections of towering and complicated constructions like oil and gas refineries, flare stacks, and pipelines.

IN-DEPTH AND DETAIL DATA INPLACE - Many drone models are launched into the market with obstacle avoidance capacities. They can operate quite close to constructions, and this encourages them to seize precise data. They capture high-resolution images or 4K videos that explicitly reveal cracks, damages, displaced wires, and additional defects that we cannot detect through our naked eye. UAVs allow obtaining complete data without endangering inspection crew members of the company.

FLEXIBILITY FOR QUICK INSPECTIONS - Since Drones come with varied specifications, several can provide high or low altitude Inspections. The versatility of these characteristics empowers clients to customize the tools with ease for their projects. Drones are suitable for both regular and emergency scenarios, the Construction Industry abides by these advantages, especially building developers for Rooftop Inspections. Drones can carry out multiple roles, such as capturing high-quality photos, videos, thermal images, etc. This data is then transmitted and processed immediately, as opposed to the time- consuming conventional method.

REACH HAZARDOUS AREA - UAVs make obtaining efficient data from hard-to-reach locations a cakewalk for industry professionals. It is the most suitable alternative to overcome limitations of traditional methods regarding worker's safety, especially in hazardous situations like radiation monitoring, inspecting high-voltage lines. Drones also allow a more cost-effective approach toward inspections of these locations

DISADVANTAGES OF UNMANNED AERIAL VEHICLE (UAV)

PRIVACY - While drone's benefits are endless, drone technology has several downsides to it. UAVs can quickly fall prey to manipulation and trespass a group or individual's privacy. Though many desire to utilize drones for retaining safety, it could violate numerous individual liberties in the name of public security.

LEGISLATIVE UNCERTAINTY - The use of Unmanned Aircraft Systems (UAS) has become widespread; however, the law is still developing, considering it is a novel technology in the industry. Specific practices installed for tiny drones also apply to commercial and recreational applications but are still vague in several dimensions. Rules for the regulation of drone movement and property protection from aerial trespassing are still in the making; thus, UAV technology functions in a judicial gray zone. There are numerous frictions between governmental regulations and any state or city laws to manage airspace property rights, because of which drone operators may violate rules they didn't know about.

SAFETY - Safety is a fundamental element to prioritize when operating drone technology. UAVs outfitted with high-quality sensors recognize possible collisions and safely engineer their way around them, making them a significant trait. These drone capacities must resemble those of the manned aircraft navigators. It is commendable to hire professional drone service providers who can operate an aerial drone without crashing it. Drones operated in heavily-populated regions have an amplified risk of ground impact or damage, mainly due to system malfunction or hacking.

SOFTWARE ISSUES OR MALFUNCTION - There have previously been many drones that have fired weapons to commoners, generating a significant amount of casualties, injuries, and damages due to malfunctions or software blunders. Drone mishaps strike other military personnel's safety as well. Drones are still in the process of improvement to limit accidents or hazards that can affect the health and safety of human lives.

VULNERABLE TO WILD ANIMALS - Drones are susceptible to wild animal attacks and are sometimes also dangerous to nature. It is possible that when a drone operator is flying in a domain with a considerable number of wild animals, they crash against a tree or possibly conflict with a vulnerable animal. Large flying birds like eagles are regularly attacking and even capturing drones operating in their space to obtain crucial data.

SPYING – spying many offenders employ drones as a strategy to target their victims and to maintain a track on them. The blatant propeller noises are no longer a concern and are unnoticeable, enabling criminals to invade someone's privacy. Many drones furnished with thermal and night sensors identify life signs and efficiently target those currently of interest by the spy. Since UAVs can seize accurate data, they can register regular habits and recognize suspicious activities without permission.

EASY TO HACK - One substantial downside to drone technology's growth is its vulnerability. Hackers can quickly attack a drone's central control system and become the drone's original controller. The primary control system includes significant knowledge crucial for hackers to evade without the initial operator's awareness. Hackers can acquire private information, corrupt or damage the files, and leak data to unauthorized third parties.

WEATHER DEPENDENT - Drones are more vulnerable to weather conditions when contrasted to traditional aircraft. For example, if the climatic conditions are unfavorable, the UAV will not maneuver appropriately or gather reliable data or imagery. However, there are drones available that are more stable and can withstand gusts of wind successfully.

KNOWLEDGE AND SKILL - As discussed earlier, if one necessitates seizing accurate, high-quality data, they need to possess the demanded skillset. This specification would indicate that an average farmer would require comprehensive training or a third party drone service provider to capture, process, and analyze farming data. With expanding operators in the industry, drone costs and its accompanying resource expenses will gradually reduce.

DATA TRANSFER SPEED IS SLOW - One of the cons in expanding drone technology in precision agriculture is its data transmission speed, which some suppose could be a week. If the time necessitated for data delivery results in a farmers' unproductivity and damage to fertilizers, crops, or pesticides, the operation of the drone would be a waste in the end. Thus, if data transfer speed is slow, suffering and damage can occur in that period, following all efforts going to waste.

UAV communication network

UAV communications systems work by using one frequency to control the aerial vehicle from the ground via a remote pilot while the other frequency is used to beam data or relay First-Person View (FPV) video. By utilizing high-quality, reliable communications links, civilian UAVs are able to relay aerial visuals and data to those on the ground with ease, while still remaining in flight.

Since UAV can move quickly and flexibly, it is absolutely impossible to do the communication by using wire when they are flying. Even using wireless techniques, the flying attitude and the speed of UAV should be considered. However, there is also a number of advantages when UAV using wireless networking.

- UAVs can provide on-demand, high-quality communication due to line-of-sight signal propagation.
- UAVs can be sensing and data fusion nodes dynamically deployable in the region of interest.

- UAVs can tailor their flight paths to enhance the quality of wireless networking and communication.
- UAVs can themselves carry and forward huge amounts of data.

At present, the structure of UAV communication network usually adopts mobile self-organizing network, which is a kind of wireless communication network with special structure. Its communication is in response to the mutual cooperation between nodes, and is completed in the way of wireless multi hop. Each node in the network has a transceiver, which adopts distributed control. In addition, the node has the functions of a host and a router and can be quickly deployed without relying on pre-existing network infrastructure.

The mobile self-organizing network of UAV has strong anti-destroy ability. It can quickly and automatically form a network at any time and any place without the need of any other preset network facilities. It can dynamically change the network structure, and even if the UAV of one node is attacked, it can automatically reconstruct the network topology without affecting other nodes. The mobile self-organizing network of UAV has an efficient routing protocol algorithm, which can sense the network changes in time, automatically configure or reconstruct the network, ensure the real-time connectivity of the data link, and has a high degree of autonomy and adaptability. In addition, the mobile self-organizing network of UAV can realize information sharing, process the received information, make decision independently, and realize intelligent task.

MIMO (multiple input and multiple output)

Multiple-input and multiple-output, or MIMO is a method for multiplying the capacity of a radio link using multiple transmission and receiving antennas to exploit multipath propagation. MIMO has become an essential element of wireless communication standards including IEEE802.11n (Wi-Fi), IEEE802.11ac (Wi- Fi), HSPA+ (3G), WiMAX, and Long Term Evolution (LTE). More recently, MIMO has been applied to power-line communication for three-wire installations as part of the ITU G.hn standard and of the Home Plug AV2 specification.

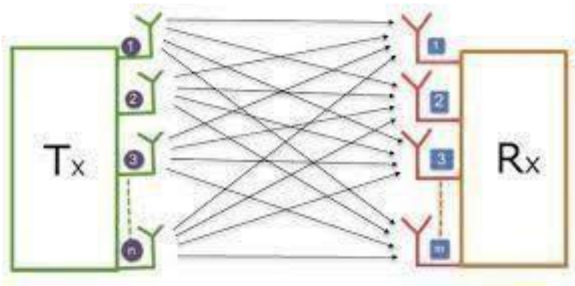


Fig: 1.2 Basic structure of MIMO

As a result of the use multiple antennas, MIMO wireless technology is able to considerably increase the capacity of a given channel while still obeying Shannon's law. By increasing the number of receive and transmit antennas it is possible to linearly increase the throughput of the channel with every pair of antennas added to the system. This makes MIMO wireless technology one of the most important wireless techniques to be employed in recent years. As spectral bandwidth is becoming an ever more valuable commodity for radio communications systems, techniques are needed to use the available bandwidth more effectively. MIMO wireless technology is one of these techniques.

Advantages of MIMO

- The higher data rate can be achieved with the help of multiple antennas and SM (Spatial Multiplexing) technique. This helps in achieving higher downlink and uplink throughput.
- It helps in achieving reduction in BER (Bit Error Rate) due to application of advanced signal processing algorithms on the received data symbols by multiple antennas.
- The techniques such as STBC (Space Time Block Coding) and BF (Beamforming) when employed in MIMO system helps in achieving extension of cell coverage.
- MIMO based system minimize fading effects seen by the information traveling from transmit to receive end. This is due to various diversity techniques such as time, frequency and space.
- There is lower susceptibility of tapping by unauthorized persons due to multiple antennas and algorithms.

- The systems with MIMO offers high QoS (Quality of Service) with increased spectral efficiency and data rates.
- The wide coverage supported by MIMO system helps in supporting large number of subscribers per cell.
- The MIMO based system is widely adopted in latest wireless standards viz. WLAN (802.11n, 802.11ac etc.), WiMAX (IEEE 802.16e), LTE, LTE-Advanced etc.

Disadvantages of MIMO

- The resource requirements and hardware complexity is higher compare to single antenna based system. Each antenna requires individual RF units for radio signal processing. Moreover advanced DSP chip is needed to run advanced mathematical signal processing algorithms.
- The hardware resources increase power requirements. Battery gets drain faster due to processing of complex and computationally intensive signal processing algorithms. This reduces battery lifetime of MIMO based devices.
- MIMO based systems cost higher compare to single antenna based system due to increased hardware and advanced software requirements

FUNCTIONS OF MIMO

Precoding is multi-stream beamforming, in the narrowest definition. In more general terms, it is considered to be all spatial processing that occurs at the transmitter. In (single-stream) beamforming, the same signal is emitted from each of the transmit antennas with appropriate phase and gain weighting such that the signal power is maximized at the receiver input. The benefits of beamforming are to increase the received signal gain by making signals emitted from different antennas add up constructively and to reduce the multipath fading effect. In line-of-sight propagation, beamforming results in a well-defined directional pattern. However, conventional beams are not a good analogy in cellular networks, which are mainly characterized by multipath propagation. When the receiver has multiple antennas, the transmit beamforming cannot simultaneously maximize the signal level at all of the receive antennas, and precoding with multiple streams is often beneficial. Note that precoding requires knowledge of channel state information (CSI) at the transmitter and the receiver.

Spatial multiplexing requires MIMO antenna configuration. In spatial multiplexing, a high-rate signal is split into multiple lower-rate streams and each stream is transmitted from a different transmit antenna in the same frequency channel. If these signals arrive at the receiver antenna array with sufficiently different spatial signatures and the receiver has accurate CSI, it can separate these streams into (almost) parallel channels. Spatial multiplexing is a very powerful technique for increasing channel capacity at higher signal-to-noise ratios (SNR). The maximum number of spatial streams is limited by the lesser of the number of antennas at the transmitter or receiver. Spatial multiplexing can be used without CSI at the transmitter, but can be combined with precoding if CSI is available. Spatial multiplexing can also be used for simultaneous transmission to multiple receivers, known as space-division multiple access or multi-user MIMO, in which case CSI is required at the transmitter. The scheduling of receivers with different spatial signatures allows good separability.

Diversity coding techniques are used when there is no channel knowledge at the transmitter. In diversity methods, a single stream (unlike multiple streams in spatial multiplexing) is transmitted, but the signal is coded using techniques called space-time coding. The signal is emitted from each of the transmit antennas with full or near orthogonal coding. Diversity coding exploits the independent fading in the multiple antenna links to enhance signal diversity. Because there is no channel knowledge, there is no beamforming or array gain from diversity coding. Diversity coding can be combined with spatial multiplexing when some channel knowledge is available at the transmitter. Electronic beam steering with greater efficiency is a vibrant feature of a phased array antenna, but for all the range cells, it is fixed at a specific angle. To mitigate this problem, frequency diverse array (FDA) antenna was proposed. This study presents a review on the development of FDA technology in radar and navigation applications. FDA is different from a conventional phased array antenna radar in a sense that it uses a small frequency offset across the array, which helps to generate a range, angle and time-dependent beam pattern. This pattern assures the energy transmission towards the desired angle and range cell. In addition, this study also focuses the research on getting range-angle uncoupled beam patterns along with diverse hybrid cognitive FDA design, available in the literature, for improved radar performance. Radar systems provide an important remote sensing capability and are crucial to the layered sensing vision; a concept of operation that aims to apply the right number of the right types of sensors, in the right places, at the right times for superior battle space situational awareness. The layered sensing vision poses a range of technical challenges, including radar, that are yet to be addressed. To address the radar-specific design challenges, the research community responded with waveform diversity; a relatively new field of study which aims reduce the cost of remote sensing while improving performance. Early work suggests that the frequency diverse array radar may be able to perform several remote sensing missions simultaneously without sacrificing performance. With few techniques available for modeling and characterizing the frequency diverse array, this research aims to specify, validate and characterize a waveform diverse signal model that can be used to model a variety of traditional and contemporary radar configurations, including frequency diverse array radars. To meet the aim of the research, a generalized radar array signal model is specified. A representative hardware system is built to generate the arbitrary radar signals, then the measured and simulated signals are compared to validate the model. Using the generalized model, expressions for the average transmit signal power, angular resolution, and the ambiguity function are also derived. The range, velocity and direction-of-arrival measurement accuracies for a set of signal configurations are evaluated to determine whether the configuration improves fundamental measurement accuracy.

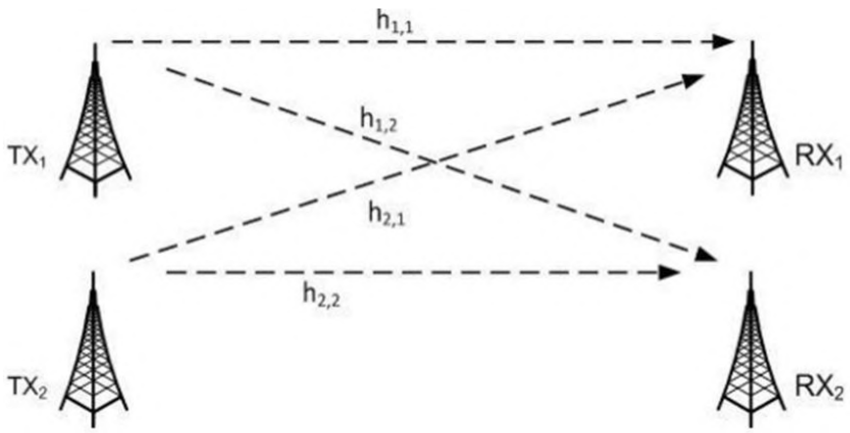
The main importance of MIMO channel technique is to improve the capacity of the channel and therefore it is important to compare the capacity of a SISO system to MIMO system. In SISO a system, the Shannon formula in equation can be applied to determine the capacity of the system. However, for a precise comparison, it is important that the MIMO system is transmitting with a power the same as that of a SISO system. Therefore, if the power radiated by a SISO system is Ps (SNR), then the power radiated from each antenna of a MIMO system with NTX transmit antennas must be Ps/N.

Hence, for a MIMO system with NTX and MRX antennas using diversity at transmitter and receiver end, the capacity of the system can be determined by the formula means transpose-conjugate of H and H is the M x N channel matrix. IM

Indicates the identity matrix of dimension N x M, in this case M = N =2 or more. However, since signals transmitted over MIMO channel have to be linearly independent and orthogonal, interference averages to zero. Hence from equation it can be seen that if the signal power Ps and the noise level N are the same then the more multiple antennas are used at the receiver, the more power is collected increasing the channel capacity and bandwidth.

Representation of MIMO channel:

The first channel model to be considered in this project will be a 2 x 2 MIMO system that is a system with 2 transmits (TX) and 2 receive (RX) antennas where different independent data streams are transmitted from multiple antennas to multiple receive antennas. This channel model will be extended to a 3 x 3 MIMO system and even more to illustrate the channel characteristics in relation to the increase in the number of antennas. The signals considered in the MIMO systems of this project are baseband signals ignoring modulation processes and concentrating on the up and down frequency conversion. Therefore, the signals on the i-th transmit antenna will be denoted x while the received signal on the j-th receive antenna denoted as y. Figure 1.3 shows the antenna set-up and the various unknown channel coefficients.



. Figure 1.3 antenna set-up

Since the coefficient of the unknown in the channel matrix Wc and the number of transmitted signal X is equal to the number of received signal Y, the equation can be solved if the channel Wc is inversed which in this case a 2 x 2 matrix inversion.

Operational principles of a MIMO system:

To derive the channel characteristics, MIMO system transmits specified and known training signals regularly from all transmitters of the system and these transmitted signals are received at the receiver. Based on the received signals, the receiver calculates the characteristics of all channel paths from each transmitted antenna to each receiving antenna. In order to prove that MIMO work, the transmitted signal X has to be solved from the group of equations in equation below. We also assume that the system is noise free and line of sight (LOS). Reference to figure 3 below, if the transmitted signal is represented to be X and the received signal Y. If the channel characteristics matrix is Wc, we may write Y=X Wc. If the channel matrix has N rows as many as there are transmitting antennas with index i. Then transmitted signal vector is written as X = [x1, x2…xN]. Also if the channel matrix has M columns, as there are receiving antennas with index j. Then the received signal vector is Y = [y1, y2…yM]. These vectors are extended later to matrixes by inserting K samples into each column.

Channel estimation procedure:

In order to estimate the channel characteristics, we expand each transmitted and received signals in time and write into signal matrix columns K discrete samples in time. The signals matrixes get K rows and as many columns as we have antennas, N or M. In general a MIMO system involves multiple antennas at the transmitting end and multiple receivers at the receiving end. Figure 1.4 show a general representation of a 3 x 3 general MIMO system.

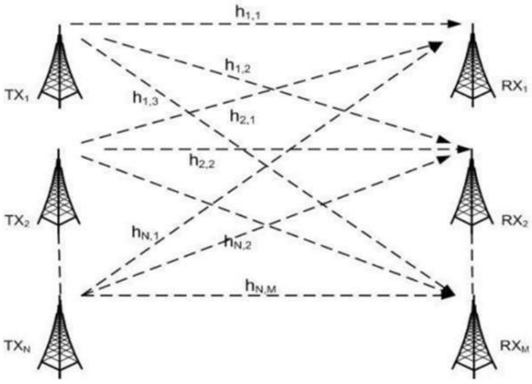


Figure 1.4 A general representation of a 3 x 3 general MIMO system

In order to determine the characteristic of the channel, both the transmitted signal X and the received signal Y have to be known. If the transmitted and received signals are of the form. If the channel transfer matrix Wc can be determine, then it means the transmitted signals can also be determine because the received signal Y is known. We may write the expression between these vector signals as Y = X Wc. In order to determine the transmitted signals, the channel transfer matrix, Wc -1 have to be inverted and then multiplied with the received signal matrix Y. X = Y Wc -1 The channel transfer matrix that we have to solve with the help of known training signal X and the received signal Y has the form. The channel transfer matrix is calculated periodically with the help of the known training signals and remains constant over information transmission time. It is then recalculated when new information is being transmitted. The channel characteristics in equation are defined for each signal path at discrete time instants 1,2,...K. However, we need to derive Wc and this can be derived with the help of the transmitted signals X and the corresponding known received signals Y measured at the receiver. Equation expresses the channel matrix Wc which do not need to be a square matrix. Because, for example if we have a 4x4 MIMO system with 100 samples, we do not need to have a 100x100 channel matrix. However, 5 samples can be transmitted from the four (4) antennas at a time. The next section explains the procedures required in order to estimate the channel transfer matrix.

Channel identification algorithm:

To model the transfer channel a common space matrix (orthonormal basis matrix Ux) is first generated and then used to map both the transmitted and received signals that vary in space (multiple antennas) and in time (samples in time). This common orthonormal basis matrix is obtained by decomposing either the transmitted signal X or the received signal Y. Hence, MIMO channel problem can be solved using four step approach under a condition where there is no noise (N = 0) and if the transmitted signals are orthogonal. These steps are summarized as follows:

(1) Finding an orthonormal basis Ux of the transmitted signal matrix X using the Gram-Schmidt procedure.

(2) In the K-dimensional signal vector space spanned by Ux, we express the N column vectors of the transmitted signal X by the projection onto the orthogonal axes of Ux. Wx = Ux X.

(3) In the K-dimensional signal vector space spanned by Ux, we express the M column vectors of the received signal matrix Y by their projection onto the orthogonal axes of Ux. Wy = Ux Y (4.2.2)

(4) Calculate the inverse or the pseudo inverse of the Fourier coefficients of the transmitted signal Wx and find an estimate of channel transfer matrix Oc = Wx -1 Wy or Oc = Wx + Wy.

OFDM (orthogonal frequency division multiplexing)

OFDM-Orthogonal Frequency Division Multiplexing, is a form of signal modulation that divides a high data rate modulating stream placing them onto many slowly modulated narrowband close-spaced subcarriers, and in this way is less sensitive to frequency selective fading. OFDM is a modulation format that is being used for many of the latest wireless and telecommunications standard.

OFDM is a form of multicarrier modulation. An OFDM signal consists of a number of closely spaced modulated carriers. When modulation of any form – voice, data, etc. is applied to a carrier, then sidebands spread out either side. It is necessary for a receiver to be able to receive the whole signal to be able to successfully demodulate the data. As a result when signals are transmitted close to one another they must be spaced so that the receiver can separate them using a filter and there must be a guard band between them. This is not the case with OFDM. Although the sidebands from each carrier overlap, they can still be received without the interference that might be expected because they are orthogonal to each another. This is achieved by having the carrier spacing equal to the reciprocal of the symbol period.

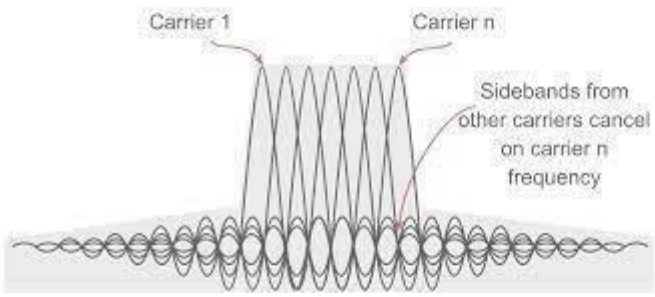


Fig: 1.5 OFDM Signal

The data to be transmitted on an OFDM signal is spread across the carriers of the signal, each carrier taking part of the payload. This reduces the data rate taken by each carrier. The lower data rate has the advantage that interference from reflections is much less critical. This is achieved by adding a guard band time or guard interval into the system. This ensures that the data is only sampled when the signal is stable and no new delayed signals arrive that would alter the timing and phase of the signal. The distribution of the data across a large number of carriers in the OFDM signal has some further advantages. Nulls caused by multi-path effects or interference on a given frequency only affect a small number of the carriers, the remaining ones being received correctly. By using error-coding techniques, which does mean adding further data to the transmitted signal, it enables many or all of the corrupted data to be reconstructed within the receiver. This can be done because the error correction code is transmitted in a different part of the signal.

OFDM advantages

- Immunity to selective fading: One of the main advantages of OFDM is that is more resistant to frequency selective fading than single carrier systems because it divides the overall channel into multiple narrowband signals that are affected individually as flat fading sub-channels.
- Resilience to interference: Interference appearing on a channel may be bandwidth limited and in this way will not affect all the sub-channels. This means that not all the data is lost.
- Spectrum efficiency: Using close-spaced overlapping sub-carriers, a significant OFDM advantage is that it makes efficient use of the available spectrum.
- Resilient to ISI: Another advantage of OFDM is that it is very resilient to inter-symbol and inter-frame interference. This results from the low data rate on each of the sub-channels.
- Resilient to narrow-band effects: Using adequate channel coding and interleaving it is possible to recover symbols lost due to the frequency selectivity of the channel and narrow band interference. Not all the data is lost.
- Simpler channel equalization: One of the issues with CDMA systems was the complexity of the channel equalization which had to be applied across the whole channel. An advantage of OFDM is that using multiple sub-channels, the channel equalization becomes much simpler.

OFDM disadvantages

Whilst OFDM has been widely used, there are still a few disadvantages to its use which need to be addressed when considering its use.

- High peak to average power ratio: An OFDM signal has a noise like amplitude variation and has a relatively high large dynamic range, or peak to average power ratio. This impacts the RF amplifier efficiency as the amplifiers need to be linear and accommodate the large amplitude variations and these factors mean the amplifier cannot operate with a high efficiency level.
- Sensitive to carrier offset and drift: Another disadvantage of OFDM is that is sensitive to carrier frequency offset and drift. Single carrier systems are less sensitive.

Applications of OFDM

- Digital Audio Broadcasting (DAB) and
- Digital Video Broadcasting over the terrestrial network: Digital Terrestrial Television Broadcasting (DTTB). In the DTTB OFDM transmission standard, about 2,000 to 8,000 subcarriers are used.
- UMTS. The UMTS Forum is selecting an appropriate radio solution for the third generation mobile standard, as a successor to GSM. OFDM is one of the five competing proposals.
- Wireless LANs. OFDM is used in HIPERLAN Phase II, which supports 20 Mbit/s in propagation environments with delay spreads up to 1 msecond.

Key features of OFDM

- The OFDM scheme differs from traditional FDM in the following interrelated ways:
- Multiple carriers (called subcarriers) carry the information stream
- The subcarriers are orthogonal to each other.
- A guard interval is added to each symbol to minimize the channel delay spread and inter symbol interference.

Data on OFDM

The data to be transmitted on an OFDM signal is spread across the carriers of the signal, each carrier taking part of the payload. This reduces the data rate taken by each carrier. The lower data rate has the advantage that interference from reflections is much less critical. This is achieved by adding a guard band time or guard interval into the system. This ensures that the data is only sampled when the signal is stable and no new delayed signals arrive that would alter the timing and phase of the signal.

Guard interval on OFDM signals

The distribution of the data across a large number of carriers in the OFDM signal has some further advantages. Nulls caused by multi-path effects or interference on a given frequency only affect a small number of the carriers, the remaining ones being received correctly. By using error-coding techniques, which does mean adding further data to the transmitted signal, it enables many or all of the corrupted data to be reconstructed within the receiver. This can be done because the error correction code is transmitted in a different part of the signal.

MIMO-OFDM SYSTEM

It combines multiple-input, multiple-output (MIMO) technology, which multiplies capacity by transmitting different signals over multiple antennas, and orthogonal frequency-division multiplexing (OFDM), which divides a radio channel into a large number of closely spaced sub channels to provide more reliable communications at high speeds.

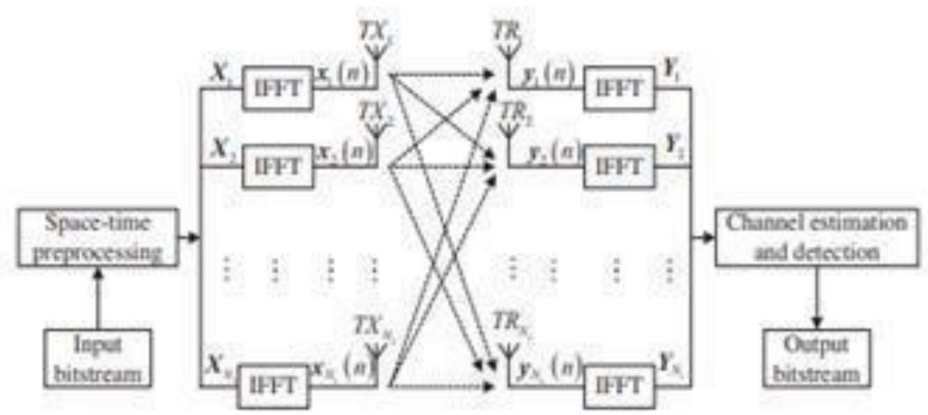


Fig: 1.6 The model of MIMO-OFDM system

The block diagram of the MIMO-OFDM transceiver is shown. We consider a MIMO system employing transmit and receive antennas. As seen from, for the transmission of each frame, a total of information bits enter the MIMO-OFDMtransmitter. These bits are first split into groups and the corresponding bits are processed in each branch of the transmitter by the OFDM index modulators. The incoming information bits are used to form the OFDM block, in each branch of the transmitter, where is the size of the fast Fourier transform (FFT). According to the OFDM principle, these bits are split into groups each containing bits, which are used to form OFDM sub blocks, of length N.

The corresponding bits, only out of available subcarriers are selected as active by the index selector at each sub block ,while the remaining subcarriers are inactive and set to zero. On the other hand, the remaining bits are mapped onto the considered M-ary signal constellation. Therefore, unlike classical MIMO-OFDM, contains some zero terms whose positions carry information for MIMO-OFDM. In this study, active subcarrier index selection is performed by the reference look-up tables at OFDM index modulators of the transmitter. The two active subcarriers out of four available sub carriers according to the reference look-up table of size. The OFDM index modulators in each branch of the transmitter obtain the OFDM sub blocks first and then concatenate these sub blocks to form the main OFDM blocks. In order to transmit the elements of the sub blocks from uncorrelated channels, block inter leavers are employed at the transmitter. The block interleaved OFDM frames are processed by the inverse FFT (IFFT) operators to obtain. We assume that the time- domain OFDM symbols are normalized to have unit energy, i.e., for all. After the addition of cyclic prefix of samples, parallel-to-serial and digital-to analog conversions, the resulting signals sent simultaneously from transmit antennas over a frequency selective Rayleigh fading MIMO channel, where represents the -tap wireless channel between the transmit antenna and the receive antenna ,whose elements are independent and identically distributed with. Assuming the wireless channels remain constant during the transmission of a MIMO-OFDM frame and , after removal of the cyclic prefix and performing FFT operations in each branch of the receiver, the input-output relationship of the MIMO-OFDM scheme in the frequency domain is obtained as for , where is the vector of the received signals for receive antenna , represents the frequency response of the wireless channel between the transmit antenna and receive antenna , and is the vector of noise samples. The elements of and follow and distributions, respectively, where denotes the variance of the noise samples in the frequency domain, which is related to the variance of the noise samples in the time domain as. We define the signal-to-noise ratio (SNR) as where [joules/bit] is the average transmitted energy per bit. The spectral efficiency of the MIMO-OFDM scheme is [bits/s/Hz], which is equal to times that of the OFDM scheme.

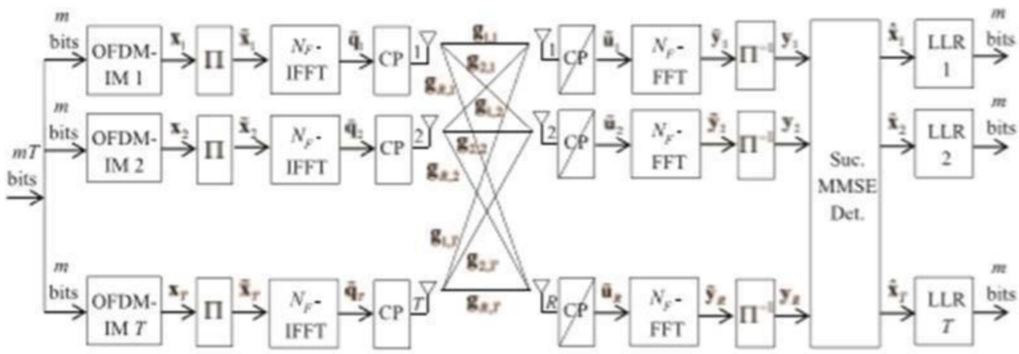


Fig: 1.7 The MIMO-OFDM transceiver

As seen from, the ML detector has to make a joint search overall transmit antennas due the interference between the sub blocks of different transmit antennas. Since has different realizations, the total decoding complexity of the ML detector in , in terms of complex multiplications (CMs), is per sub block, which becomes impractical for higher order modulations and MIMO systems. Instead of the exponentially increasing decoding complexity of the ML detector, we propose a novel MMSE detection and LLR calculation based detector, which has a linear decoding complexity as that of classical MIMO-OFDM with MMSE detection. For the detection of the corresponding OFDM sub blocks of different transmit antennas, the following MIMO signal model is obtained from for subcarrier of sub block where is the received signal vector, is the corresponding channel matrix which contains the channel coefficients between transmit and receive antennas and assumed to be perfectly known at the receiver, is the data vector which contains the simultaneously transmitted symbols from all transmit antennas and can have zero terms due to index selection in each branch of the transmitter and is the noise vector. For classical MIMO-OFDM, the data symbols can be simply recovered after processing the received signal vector with the MMSE detector. On the other hand, due to the index information carried by the sub blocks of the proposed scheme, it is not possible to detect the transmitted symbols by only processing for a given subcarrier in the MIMO-OFDM scheme. Therefore, successive MMSE detections are performed for the proposed scheme using the MMSE filtering matrix.

CHAPTER-2

LITERATURE SURVEY

Beixiong Zheng et al. "Multiple-Input Multiple-Output OFDM with Index Modulation"

Low-Complexity Detector Design", in this paper proposed two low-complexity detectors derived from the SMC theory for the MIMO-OFDMIM system. The first proposed sub block-wise detector draws samples at the sub block level, exhibiting near-optimal performance for the MIMO-OFDM system. The second proposed subcarrier-wise detector draws samples at the subcarrier level, exhibiting substantially reduced complexity with a marginal performance loss. An effective legality examination method has been also developed to couple with the subcarrier wise detector. Computer simulation and numerical results have validated the outstanding performance and the low complexity of both proposed detectors.

Ertugrul Basar et al. "Multiple-Input Multiple-Output OFDM with Index Modulation"

A novel scheme called MIMO-OFDM with index modulation has been proposed as an alternative multicarrier transmission technique for 5G networks. It has been shown via extensive computer simulations that the proposed scheme can provide significant BER performance improvements over classical MIMO-OFDM for several different configurations.

The following points remain unsolved in this study:

- performance analysis,
- the selection of optimal N and K values,
- diversity techniques for MIMO-OFDM-IM, and
- Implementation scenarios for high mobility.

In this study, the recently proposed MIMO-OFDM-IM scheme has been investigated for next generation 5G wireless networks. For the MIMO-OFDM-IM scheme, new detector types such as ML, near-ML, simple MMSE, MMSE-LLR-OSIC detectors have been proposed and their ABEP have been theoretically examined. It has been shown via extensive computer simulations that MIMO-OFDM-IM scheme provides an interesting trade-off between complexity, spectral efficiency and error performance compared to classical MIMO-OFDM scheme and it can be considered as a possible candidate for 5G wireless networks.

The main features of MIMO-OFDM-IM can be summarized as follows:

- better BER performance,
- flexible system design with variable number of active OFDM subcarriers and
- Better compatibility to higher MIMO setups.

However, interesting topics such as diversity methods, generalized OFDM-IM cases, high mobility implementation and transmit antenna indices selection still remain to be investigated for the MIMO-OFDM-IM scheme.

Ertugrul Bas et al. "Performance of Multiple-Input Multiple-Output OFDM with Index Modulation"

This paper, proposed ML and near-ML detectors for the recently introduced MIMO-OFDM-IM scheme to improve its error performance compared to MMSE based detection. The ABEP upper bound of the MIMO-OFDM-IM scheme with ML detection has been derived and it has been shown that the derived theoretical upper bound can be used as an efficient tool to predict the BER performance of the MIMO-OFDMIM scheme. It has been shown via computer simulations that MIMO- FDM-IM scheme can provide significant improvements in BER performance over classical MIMO-OFDM using different type of detectors and MIMO configurations.

Beixiong Zheng et al. "Low-Complexity ML Detector and Performance Analysis for OFDM with In-Phase/Quadrature Index Modulation".

In this letter, they've planned a low-complexity detector supported the milliliter criterion, that dispenses with a priori data of the noise variance and also the potential realizations of the active subcarrier indices. supported the framework of OFDM-I/Q-IM using the planned milliliter detector, the straight line ABEP and also the actual coding gain achieved by OFDMI/Q-IM are derived, that absolutely matches the simulation results. Moreover, the exact coding gain including the spectral efficiency price has provided a clear plan of a basic trade-off between the system

Performance and also the spectral efficiency of OFDM-I/Q-IM by the adjustment of the quantity of active subcarriers.

Sheng Wu et al. "Low-Complexity Iterative Detection for Large-Scale Multiuser MIMO-OFDM Systems Using Approximate Message Passing".

For the detection of large-scale multiuser MIMO-OFDM systems, we have proposed a range of low-complexity approximate message passing algorithms that can offer desirable tradeoff between performance and complexity. It is verified through extensive simulations that our proposed approximate message passing algorithms can achieve near optimal. Compared with existing turbo detection algorithms, the proposed schemes can achieve or even outperform the performance of some complex algorithms, such as the iterative decoding based on STS-SD and MMSESIC. In addition, the number of iterations required to achieve near-optimal performance is small and does not increase with the system dimension.

CHAPTER 3

PROPOSED METHOD

The proposed system of Application of MIMO-OFDM system in UAV communication system is shown with the help of a block diagram in the below figure. The methodology is carried out as shown in steps and each step of this system is discussed in detail in section below.

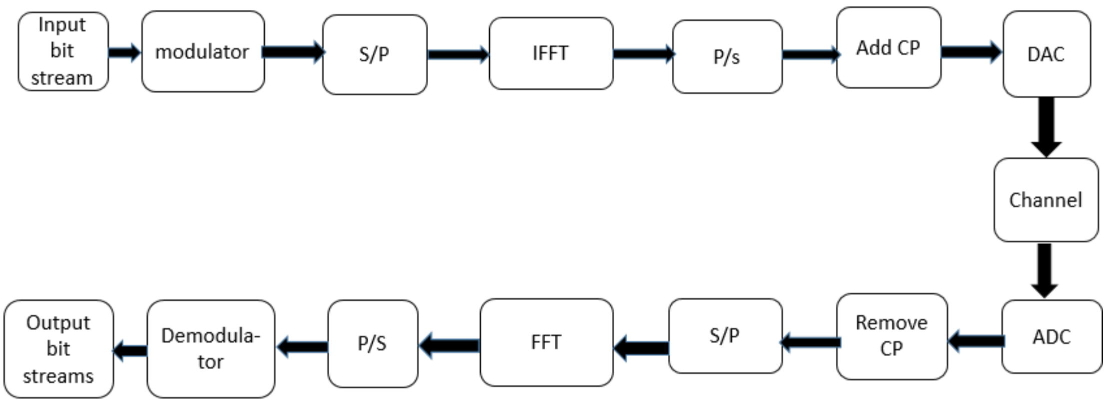


Fig: 3.1 Block diagram of MIMO-OFDM system

3.1 Modulator and demodulator

Quadrature Phase Shift Keying (QPSK) is a form of phase modulation technique, used in the above figure as modulator in which two information bits (combined as one symbol) are modulated at once, selecting one of the four possible carrier phase shift states.

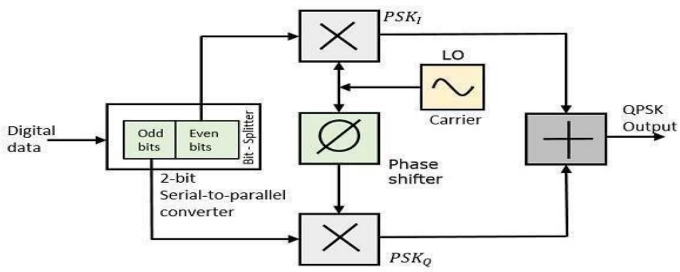


Fig: 3.2 Block diagram of QPSK modulator

The QPSK Demodulator is used as demodulator which uses two product demodulator circuits with local oscillator, two band pass filters, two integrator circuits, and a 2-bit parallel to serial converter.

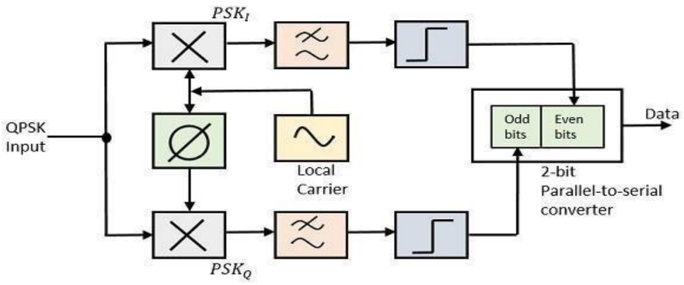


Fig: 3.3 Block diagram of QPSK demodulator

3.2 Serial to parallel converter(s/p)

A serial to parallel converter is a digital circuit where we feed the input data serially, and read the outputs in parallel fashion. A 4-bit serial-to-parallel shift register is one of the simplest types of circuits utilizing four D-type flip-flops. e.g. 2 bits/word for QPSK, and shifted into a parallel format. The data is then transmitted in parallel by assigning each data word to one carrier in the transmission.

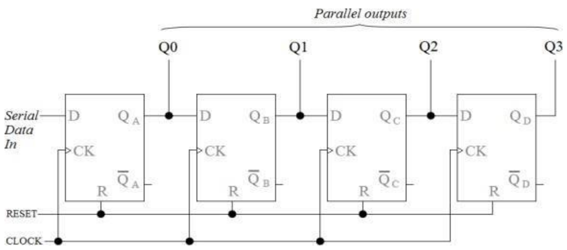


Fig: 3.4 serial to parallel converter

Parallel to serial converter (p/s)

A conversion process in which the stream of data elements received all at once is converted and sent as a stream of data at one bit at a time.

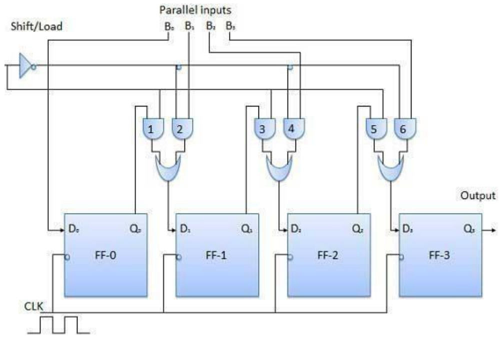


Fig: 3.5 Parallel to serial converter

3.3 FFT and IFFT

- FFT stands for Fast Fourier Transform.
- FFT converts time domain vector signal to frequency domain vector signal.
- It is used before demodulator block in the OFDM Receiver.

$$X(k) = \sum_{n=0}^{N-1} X(n) * e^{-i*2*pi*n*k/N} \quad \text{....... FFT Equation}$$

Where,

$X(n)$ -> Time Domain Samples

$X(k)$ -> Freq. Domain Samples

N -> FFT Size

k -> 0,1,2,........,N-1

Fig 3.6 FFT Equation

$$X(n) = \frac{1}{N} * \sum_{k=0}^{N-1} X(k) * e^{i*2*pi*n*k/N} \quad \text{... IFFT Equation}$$

Where,

$X(k)$ -> Freq. Domain Samples

$X(n)$ -> Time Domain Samples

N -> FFT Size

k -> 0,1,2,........,N-1

Fig 3.7 IFFT Equation

- IFFT stands for Inverse Fast Fourier Transform.
- IFFT converts frequency domain vector signal to time domain vector signal.
- It is used after the modulator block in the OFDM Transmitter

3.4 Add CP and Remove CP

The cyclic prefix provides a guard interval to eliminate inter symbol interference from the previous symbol. It repeats the end of the symbol so the linear convolution of a frequency-selective multipath channel can be modeled as circular

1convolution, which in turn may transform to the frequency domain via a discrete Fourier transform. This approach accommodates simple frequency domain processing, such as channel estimation and equalization. The cyclic prefix is created so that each OFDM symbol is preceded by a copy of the end part of that same symbol. Different OFDM cyclic prefix lengths are available in various systems. For example within LTE a normal length and an extended length are available and after Release 8 a third extended length is also included, although not normally used.

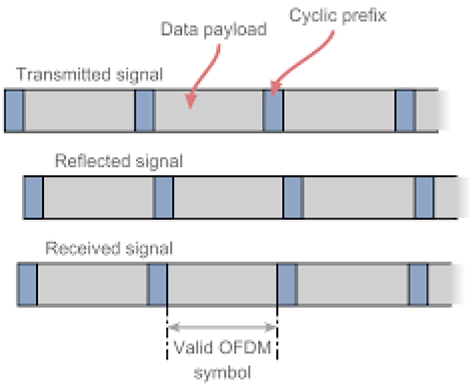


Fig: 3.8 cyclic prefix

Advantages of cyclic prefix

- Provides robustness: The addition of the cyclic prefix adds robustness to the OFDM signal. The data that is retransmitted can be used if required.
- Reduces inter-symbol interference: The guard interval introduced by the cyclic prefix enables the effects of inter-symbol interference to be reduced.

Disadvantages of cyclic prefix

- Reduces data capacity: As the cyclic prefix re-transmits data that is already being transmitted, it takes up system capacity and reduces the overall data rate.
- The use of a cyclic prefix is standard within OFDM and it enables the performance to be maintained even under conditions when levels of reflections and multipath propagation are high.

3.5 Analog to digital (ADC) and digital to analog converters (DAC)

Analogue to Digital Converter, or ADC, is a data converter which allows digital circuits to interface with the real world by encoding an analogue signal into a binary code

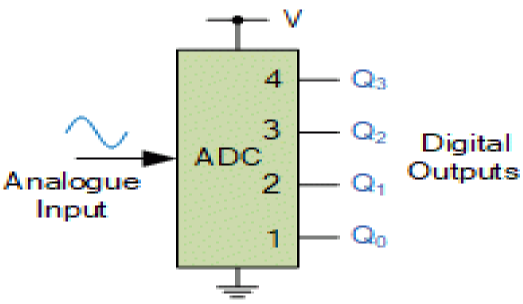


Fig: 3.9Analogue to Digital Converter

A Digital to Analog Converter (DAC) converts a digital input signal into an analog output signal. The digital signal is represented with a binary code, which is a combination of bits 0 and 1.

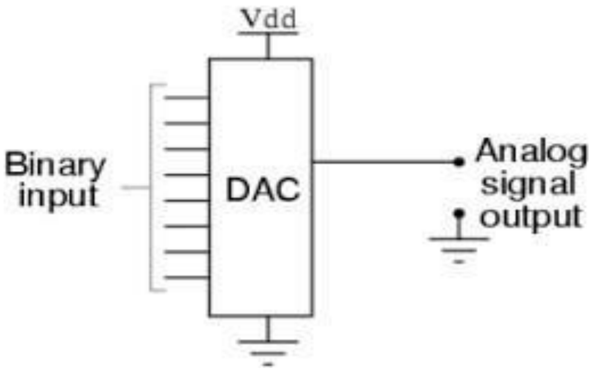


Fig: 3.10Digital to Analog Converter

The performance of ADCs and DACs mainly depends on their Resolution and Speed.

The Resolution of a converter is expressed in the number of Bit. For an ADC, the Resolution states the number of intervals or levels which can be divided from a certain analog input range. An n-bit ADC has the resolution of 1 / 2n. For example, the Resolution of a 16-bit ADC is 1 / 65536, since 216 = 65536. If the measuring voltage range is 10 V, then this input range can be resolved into 10 V / 65536 = 0.153 mV precision.

The Speed of a converter is expressed by the Sampling Frequency. It is the number of times that the converter samples the analog signal, its unit is Hertz (Hz). In audio signal processing, Sampling Frequencies of 44 kHz, 22 kHz and 11 kHz are mostly used. Using 44 kHz Sampling Frequency means the converter is sampling the analog audio signal and doing analog to digital conversion at 44000 times per second.

ADCs are used virtually everywhere, whenever an analog signal has to be transported, it is processed and stored in digital form. They are always used together with different transducers to convert physical sense and measurement such as temperature, pressure, humidity, speed, vibration, sound, picture etc. in digital signal for further processing by microprocessor.

3.6 FLOW CHART

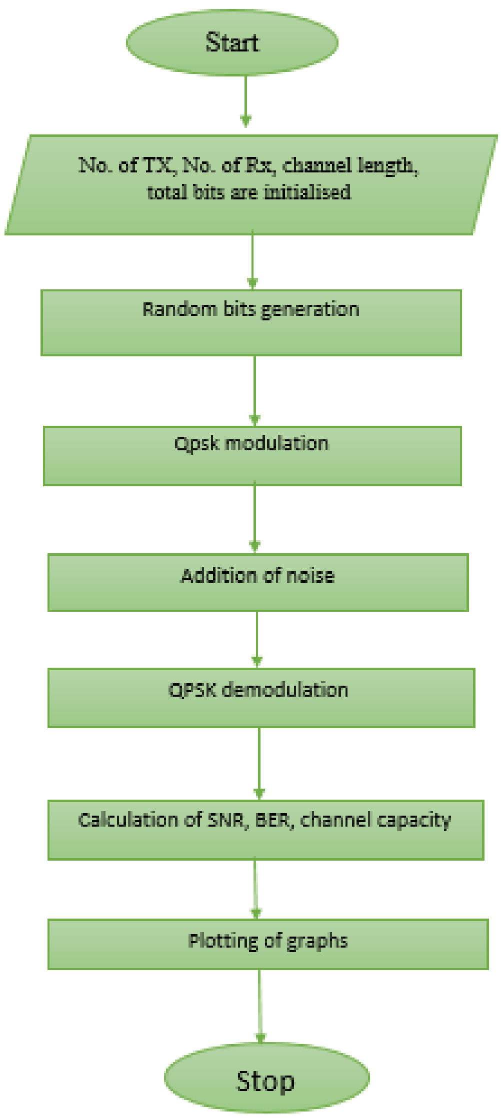


Fig: 3.11 flowchart of the program

3.7 ALGORITHM

- Firstly we will initialize the input data such as no.of transmitter ,receivers , channel length, total no.of bits etc.
- Then we create a iterative loop until it takes total no. of bits.

At the transmitter side

- We generate random bits from total no. of bits and modulate using QPSK modulation.
- We do the IFFT analysis.

At the receiver side

- We do the FFT analysis and extract real part of the signal.
- Next step is to count for the errors.
- Then calculate signal to noise ratio (SNR), channel capacity and bit error rate (BER) of the signal.
- Plotting the graph of SNR vs channel capacity.
- Plotting the semi log graph of SNR vs bit error rate.

CHAPTER 4

SIMULATION RESULTS AND ANALYSIS

The experimental environment of this project is based on MATLAB R2013a. The MIMO-OFDM communication system is constructed with MATLAB software. The Channel capacity and the bit error rate are used as main factors to prove the result.

Channel capacity

Channel capacity is a maximum information rate that a channel can transmit. It is measured in bits per second (bps). Channel capacity is a rough value as measuring takes into account only the whole amount of data transferred but leaves out of account communication quality.

It depends on three factors:

- The bandwidth available.
- The level of signal use.
- The quality of the channel (level of noise present).

Bit error rate

The bit error rate (BER) is the number of bit errors per unit time. The bit error ratio (also BER) is the number of bit errors divided by the total number of transferred bits during a studied time interval. Bit error ratio is a unit less performance measure, often expressed as a percentage**.** The BER may be improved by choosing a strong signal strength (unless this causes cross-talk and more bit errors), by choosing a slow and robust modulation scheme or line coding scheme, and by applying channel coding schemes such as redundant forward error correction codes**.**

If the medium between the transmitter and receiver is good and the signal to noise ratio is high, then the bit error rate will be very small - possibly insignificant and having no noticeable effect on the overall system However if noise can be detected, then there is chance that the bit error rate will need to be considered**.**

Signal-to-noise ratio (**SNR** or **S/N**)

It is a measure used in science and engineering that compares the level of a desired signal to the level of background noise. SNR is defined as the ratio of signal power to the noise power, often expressed in decibels. A ratio higher than 1:1 (greater than 0 dB) indicates more signal than noise. SNR, bandwidth, and channel capacity of a communication channel are connected by the Shannon–Hartley theorem.

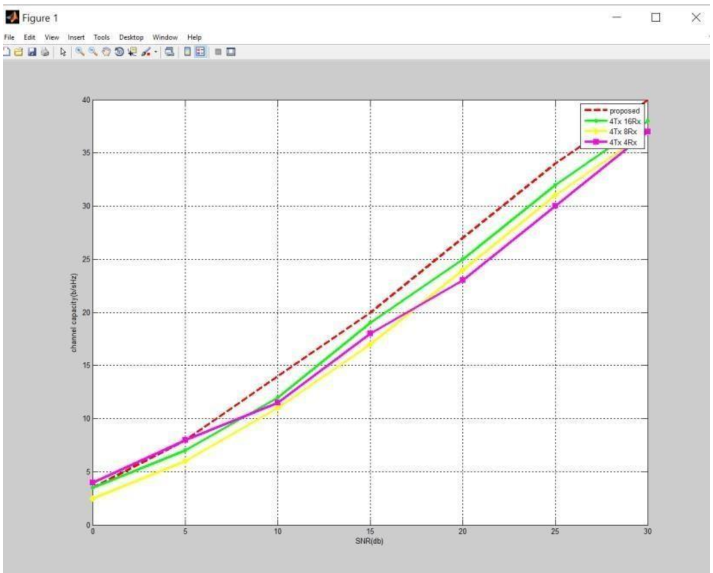


Fig: 4.1

Fig. 5.1 shows the channel capacity for different numbers of antennas. It can be seen from the figure that as the number of antennas increases, the channel capacity of the system increases.

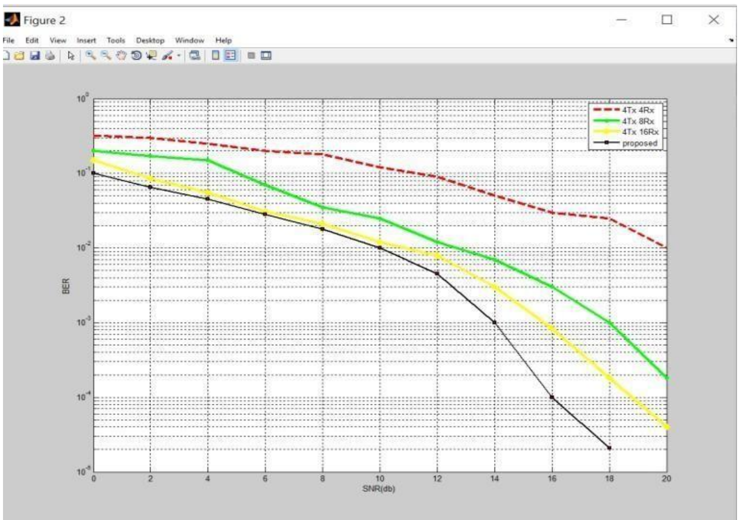


Fig: 4.2

Fig: 5.2 shows the bit error rate for different numbers of antennas. It can be seen from the figure that as the number of antennas increases, the bit error rate of the system decreases.

CHAPTER 5
CONCLUSION AND FUTURE ENHANCEMENT

Improving the communication quality of UAV communication network is an important topic in the field of UAV communication. In this paper, the characteristics of UAV communication network are analyzed. Based on the technical principle and system model of MIMO-OFDM, the MIMO-OFDM communication system is constructed. The simulation results show that MIMO-OFDM technology can effectively improve the channel capacity and reduce the bit error rate, which provides an effective solution for improving the communication quality of UAV communication network.

Acceptance Letter for online oral presentation & journal publication

Online Mega international Conference "Smart Modernistic in Electronics and Communication" *(ICSMEC–21) on 02nd & 03rd July 2021*

Dear G.Samyuktha, **K.Vijay**, **K.Kusuma, Dr.A.Chaithanya Krishna**:

Title of the paper: APPLICATION OF MIMO-OFDM TECHNOLOGY IN UAV COMMUNICATION NETWORK

Paper ID: ICSMEC21-0035

With heartiest congratulations we are pleased to inform you that based on the recommendations of the expert reviewers, your paper has been accepted for journal publication and oral presentation in online mega International Conference on "Smart Modernistic in Electronics and Communication" ***(ICSMEC-21).***

ICSMEC-21 conference received over 400+ submissions from different parts of the world and regions so far, reviewed by international experts; the acceptance ratio is controlled below 20%. Your paper will be submitted for **Journal publications free of cost** in the appropriate given Journals after registration. **You will be given registration kit, International conference proceedings with ISBN (Soft copy), and international certificate of publication.**

You are now required to pay the conference registration fees (Faculty / Research Scholars: Rs.3500/-) as mentioned below A/C details on or before 5 working day after receiving this email else your paper will not be considered for presentation as well as publication in the Journal.

For PAYTM users

For NEFT/RTGS/IMPS

Account Details: (Current Account)

A/C Name: PRINCIPAL, St. Martin's Engineering College
Bank Name: SBI, Kukatpally, Hyderabad
Account No: 30667640268
IFSC code: SBIN0011664
Swift Code: SBININBB725 (Applicable for foreign author)

After payment, please send the registration form, payment proof and final paper in.doc/.docx format.

Your interest in *ICSMEC-21* is very much appreciated. Look forward to meeting you on the day of conference.

Yours sincerely,
ICSMEC- 21 Organizing Committees
icsmec21@smec.ac.in

(Organized by Department of ECE)

REFERENCES

[1] F. Xiong, A. Li, H. Wang and L. Tang, "An SDN-MQTT Based Communication System for Battlefield UAV Swarms", IEEE Commun. Mag., vol. 57, Aug. 2019, pp. 41–47, doi: 10.1109/MCOM.2019.1900291.

[2] A. K. Gupta and A. Chockalingam, "Performance of MIMO Modulation Schemes with Imaging Receivers in Visible Light Communication", J. Lightwave Technol., vol. 36, Jan. 2018, pp. 1912–1927, doi: 10.1109/JLT.2018.2795698.

[3] X. Sun, X. Gao, G.Y. Li and W. Han, "Single-Site Localization Based on a New Type of Fingerprint for Massive MIMO-OFDM Systems", IEEE Trans. Veh. Technol., vol. 67, Mar. 2018, pp. 6134– 6145, doi: 10.1109/TVT.2018.2813058.

[4] A.J. Al-Askery, C.C. Tsimenidis, S. Boussakta and J.A. Chambers, "Performance Analysis of Coded Massive MIMO-OFDM Systems Using Effective Matrix Inversion", IEEE Trans. Commun., vol. 65, Sep. 2017, pp. 5244–5256, doi: 10.1109/TCOMM.2017.2749370.

[5] W. Zhang, H. Li, W. Kong, Y. Fan and W. Cheng, "Structured Compressive Sensing Based Block-Sparse Channel Estimation for MIMO-OFDM Systems", Wirel. Pers. Commun., vol. 108, May. 2019, pp. 2279-2309, doi: 10.1007/s11277-019-06522-8.

[6] H. Singh and S. Bansal, "Channel estimation with ISFLA based pilot pattern optimization for MIMO OFDM system", AEU-Int. J. Electron. Commun. vol. 81, Jul. 2017, pp. 143–149, doi: 10.1016/j.aeue.2017.07.024.

[7] P. Pattanayak and P. Kumar, "Quantized feedback scheduling for MIMO-OFDM broadcast networks with subcarrier clustering", Ad Hoc Netw., vol. 65, Oct. 2017, pp. 26–37, doi: 10.1016/j.adhoc.2017.07.007.

[8] M. Sun and D. Park, "Pilot Reduction Using Spatial and Temporal Correlation in Massive MIMO-OFDM Systems", Wirel. Pers. Commun., vol. 97, Jun. 2017, pp. 2541–2550, doi: 10.1007/s11277- 017-4622-4.

APPENDIX-A
MATLAB

What Is MATLAB?

MATLAB® is a excessive-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment wherein issues and answers are expressed in familiar mathematical notation. Typical uses encompass

Typical uses of MATLAB:

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data evaluation, exploration, and visualization
- Scientific and engineering pictures
- Application improvement, along with graphical consumer interface constructing.

Introduction

MATLAB is a immoderate-overall performance language for technical computing. It integrates computation, visualization, and programming in an clean-to-use surroundings wherein troubles and answers are expressed in acquainted mathematical notation. MATLAB stands for matrix laboratory, and become written first off to offer smooth get entry to matrix software application advanced by using LINPACK (linear system bundle) and EISPACK (Eigen system package deal) responsibilities.

MATLAB is consequently constructed on a basis of sophisticated matrix software program application in which the important element is array that does not require pre dimensioning which to resolve many technical computing issues, especially people with matrix and vector formulations, in a fragment of time. MATLAB talents a own family of programs specific answers referred to as toolboxes.

Very important to most customers of MATLAB, toolboxes permit analyzing and utilizing specialized era. These are complete collections of MATLAB features (M-documents) that growth the MATLAB environment to remedy unique commands of problems. Areas in which toolboxes are available embody sign processing, manipulate device, neural networks, fuzzy appropriate judgment, wavelets, simulation and plenty of others.

To resolve many technical computing issues, especially people with matrix and vector formulations, in a fragment of time.

Typical uses of MATLAB encompass: Math and computation, Algorithm improvement, Data acquisition, Modeling, simulation, prototyping, Data evaluation, exploration, visualization, Scientific and engineering pics, Application development, along with graphical character interface constructing. Boom the MATLAB environment to remedy unique commands of problems. Areas where in toolboxes are to be had embody signal processing, control device, neural networks, fuzzy appropriate judgment, wavelets, simulation and lots of others.

A.1 Basic Building Blocks of MATLAB

The fundamental building block of MATLAB is MATRIX. The essential information type is the array. Vectors, scalars, real matrices and complicated matrix are dealt with as particular magnificence of this fundamental information kind. The built in functions are optimized for vector operations. No size statements are required for vectors or arrays.

MATLAB Windows

The MATLAB works based totally on five windows: Command window, Workspace window, Current directory window, Command history window, Editor Window, Graphics window and Online-help window.

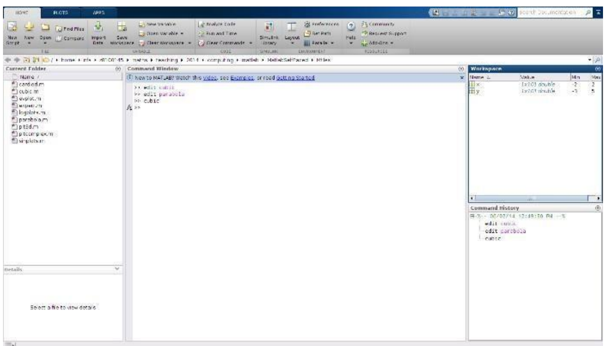


Fig: A.1 matlab window

Command Window

The command window is in which the person sorts MATLAB instructions and expressions on the spark off (>>) and in which the output of those commands is displayed. It is opened when the utility program is released. All instructions inclusive of user-written applications are typed in this window at MATLAB activate for execution.

Work Space Window

MATLAB defines the workspace as the set of variables that the person creates in a piece consultation. The workspace browser indicates these variables and a few records about them. Double clicking on a variable inside the workspace browser launches the Array Editor, which can be used to gain facts.

A.2 Three Current Directory Window

The present day Directory tab suggests the contents of the present day listing, whose route is hooked up inside the present day listing window. For example, within the domestic windows working tool the path is probably as follows: C:MATLABWork, indicating that directory “paintings” is a subdirectory of the precept list “MATLAB”; it's far mounted in pressure C. Clicking at the arrow in the contemporary listing window shows a listing of currently used paths. MATLAB uses a are in search of for route to discover M-files and different MATLAB related documents. Any document run in MATLAB need to live inside the modern listing or in a list that is on trying to find path.

Four Command History Window

The Command History Window consists of a file of the instructions someone has entered within the command window, which includes each modern and previous MATLAB periods. Previously entered MATLAB instructions may be selected and re-accomplished from the command records window through right clicking on a command or collection of commands. This is beneficial to pick out various alternatives further to executing the instructions and is useful characteristic while experimenting with numerous commands in a piece consultation.

Editor Window

The MATLAB editor is every a text editor specialized for growing M-documents and a graphical MATLAB debugger. The editor can seem in a window through the use of itself, or it may be a sub window inside the desktop. In this window you can clearly write, edit, create and preserve applications in documents called M-files.

MATLAB editor window has numerous pull-down menus for responsibilities collectively with saving, viewing, and debugging documents. Because it performs a few smooth assessments and additionally makes use of shade to differentiate among several factors of code, this newsletter editor is recommended because the device of desire for writing and improving M-functions.

Graphics or Figure Window

The output of all photo instructions typed inside the command window is seen in this window.

Online Help Window

MATLAB presents online assist for all its constructed in features and programming language constructs. The primary manner to get help on-line is to apply the MATLAB assist browser, opened as a separate window each with the useful resource of clicking on the question mark image (?) at the desktop toolbar, or with the aid of typing help browser at the activate inside the command window. The assist Browser is a web browser blanketed into the MATLAB computing tool that shows a HypertextMarkupLanguage (HTML) documents. The Help Browser includes planes, the assist navigator pane, used to find out statistics, and the display pane, used to view the statistics. Self-explanatory tabs other than navigator pane are used to carry out are seeking.

MATLAB Files

MATLAB has 3 varieties of files for storing facts. They are: M-documents and MAT-files.

These are wide spread ASCII text record with ‘m’ extension to the document call and growing own matrices using M-files, which are text documents containing MATLAB code. MATLAB editor or any other text editor is used to create a file containing the same statements which are typed at the MATLAB command line and save the record underneath a call that leads to .M. There are sorts of M-

Documents:

1. Script Files

It is an M-file with a hard and fast of MATLAB commands in it and is achieved by way of typing call of record on the command line. These documents paintings on worldwide variables presently found in that surroundings.

2. Function Files

A characteristic report is also an M-file besides that the variables in a feature record are all local. This type of files starts with a characteristic definition line.

MAT-Files

These are binary information documents with .Mat extension to the record which can be created through MATLAB when the records is saved. The information written in a unique format that only MATLAB can examine. These are located into MATLAB with ‘load’ command.

A.3 The MATLAB System:

The MATLAB machine consists of five important components:

Development Environment:

This is the set of tools and centers that assist you use MATLAB functions and files. Many of those gear are graphical person interfaces. It consists of the MATLAB computer and Command Window, a command history, an editor and debugger, and browsers for viewing assist, the workspace, files, and the quest path.

The MATLAB Mathematical Function:

This is a large series of computational algorithms ranging from elementary features like sum, sine, cosine, and complex mathematics, to greater sophisticated features like matrix inverse, matrix Eigen values, Bessel functions, and rapid Fourier transforms.

The MATLAB Language:

This is a excessive-stage matrix/array language with manage glide statements, capabilities, statistics systems, input/output, and item-orientated programming features. It lets in both "programming in the small" to rapidly create quick and dirty throw-away packages, and "programming within the massive" to create complete big and complicated software applications.

A.4.Four Graphics:

MATLAB has large centers for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It consists of high-degree features for two-dimensional and 3-dimensional statistics visualization, picture processing, animation, and presentation pics. It also includes low-level functions that let you fully customize the appearance of photographs as well as to build complete graphical consumer interfaces to your MATLAB packages.

THE MATLAB Application Program Interface (API):

This is a library that allows you to put in writing C and FORTRAN packages that have interaction with MATLAB. It consists of centers for calling exercises from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Workspace list the variables and describes their matrix size clear erases variables and functions from memory

Clear erases variables and functions from memory.

Clear x erases the matrix 'x' from your workspace close by itself, closes the current figure window figure creates an empty figure window.

Hold on holds the current plot and all axis properties so that subsequent graphing.

A.5 SOME BASIC PLOTCOMMANDS:

Kinds of plots:

Plot(x, y)	creates a Cartesian plot of the vectors x &y
Plot(y)	creates a plot of y vs.
Semilogy(x, y)	plots log(x) vs y.
Xlabel ('x-axis label')	generate labels along x-axis.
Ylabel ('y-axis label')	generate labels along y-axis.
Title	allows you to put a title on the graph.
Grid on	allows you to put the grid lines on the graph.

MATLAB provides eight basic color options for drawing graphs. The following table shows the colors and their codes

Table: A.1 Colors and codes used in MATLAB

CODE	**COLOUR**
w	White
k	Black
b	Blue
r	Red
c	Cyan
g	Green
m	Magenta
y	Yellow

A.6 ALGEBRIC OPERATIONS IN MATLAB:

Scalar Calculations:

+ Addition

- Subtraction

* Multiplication

/ Division

^ Exponentiation

Array products: Recall that addition and subtraction of matrices involved addition or subtraction of the character factors of the matrices. Sometimes it's far favored to definitely multiply or divide every element of an matrix by way of the corresponding element of any other matrix 'array operations”.

Array or detail-by way of-element operations are accomplished when the operator is preceded via a '.' (Period):

A.* B multiplies each detail of a by using the

respective element of b

A. / B divides each element of a by using the

respective detail of b

A. B divides every detail of b by using the respective detail of a

A. ^B raise each detail of a with the aid of the respective b element

A.7 MATLAB WORKING ENVIRONMENT

MATLAB DESKTOP

MATLAB Desktop is the principle MATLAB software program window. The laptop includes 5 sub domestic home windows, the command window, the workspace browser, the current list window, the command history window, and one or greater determine home windows, which can be tested most effective when the individual displays a photo.

The command window is in which the person kinds MATLAB instructions and expressions at the spark off (>>) and wherein the output of those commands is displayed. MATLAB defines the workspace as the set of variables that the patron creates in a work session. The cutting-edge Directory tab above the workspace tab suggests the contents of the present-day listing, whose path is shown in the current list window.

For example, in the domestic home windows running device the course is probably as follows: C: MATLAB Work, indicating that listing “art work” is a subdirectory of the primary listing “MATLAB”; WHICH IS INSTALLED IN DRIVE C.

Listing. MATLAB uses a are seeking for route to locate M-files and other MATLAB related files, which is probably arrange in directories within the laptop report system. Any document run in MATLAB want to are living inside the cutting-edge list or in a list this is on are seeking route. By default, the documents supplied with MATLAB and math works.

The simplest manner to look which directories are soon the quest path, or to characteristic or alter a seek direction, is to choose set path from the File menu the laptop, after which use the set direction communication box. It is right exercise to feature any normally used directories to the search course to avoid time and again having the change the cutting-edge list.

The Command History Window includes a file of the commands a consumer has entered in the command window, along with every modern and previous MATLAB periods. Previously entered MATLAB commands can be determined on and re-accomplished from the command history window by the use of right clicking on a command or sequence of commands.

This motion launches a menu from which to choose out numerous options further to executing the instructions. This is useful to choose out severa options in addition to executing the instructions. This is a beneficial feature at the same time as experimenting with various commands in a piece consultation.

A.7.1 Using the MATLAB Editor to create M-File:

The MATLAB editor is each a textual content editor specialized for growing M-files and a graphical MATLAB debugger. The editor can appear in a window via itself, or it could be a sub window within the laptop. M-files are denoted by using the usage of the extension .M, as in pixel The MATLAB editor window has numerous pull-down menus for tasks together with saving, viewing, and debugging files. Because it plays a few simple assessments and also makes use of coloration to distinguish among several elements of code, this newsletter editor is recommended due to the fact the tool of desire for writing and modifying M- abilities.

To open the editor, kind edit at the prompt opens the M-record filename in an editor window, equipped for enhancing. As stated earlier, the report must be inside the modern-day listing, or in a listing within the are looking for route.

A.8.Three Getting Help:

The most critical manner to get assist online is to use the MATLAB help browser, opened as a separate window either through clicking on the question mark symbol (?) at the computer toolbar, or through typing help browser on the set off in the command window. The help Browser is an internet browser incorporated into the MATLAB laptop that presentations a Hypertext Markup Language (HTML) files. The Help Browser includes two panes, the help navigator pane, used to find records, and the display pane, used to view the statistics. Self-explanatory tabs apart from navigator pane are used to carry out a search

.

Appendix B

INTRODUCTION TO DIGITAL COMMUNICATION

B.1 what is Digital Communication?

Digital communication systems, by definition, are communication systems that use such a digital sequence as an interface between the source and the channel input and similarly between the channel output and final destination.

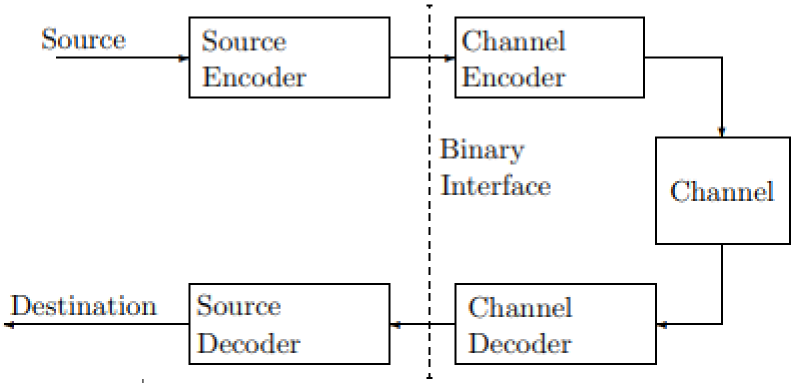


Fig: B.1block diagram of digital communication

Placing a binary interface between source and channel. The source en• coder converts the source output to a binary sequence and the channel encoder (often called a modulator) processes the binary sequence for transmission over the channel. The channel decoder (demodulator) recreates the incoming binary sequence (hopefully reliably), and the source decoder recreates the source output

WHY DIGITAL?

The telegraph, invented in the mid-nineteenth century, was the forerunner of digital communications. However, it is now that we can emphatically say digital is the pervasive technology of the twenty-first century and beyond, as the first generation of cellular phones in the late seventies was the last major analog communication invention. During the past three decades, communication networks, systems, and devices have all moved toward dig• ital. The primary examples are wireless networks, Internet, MP3 players, smart phones, HDTV, GPS, and satellite TV and radio. Digital communication technology will continue to bring about intelligent infrastructures and sophisticated end-user devices, through which a host of applications in entertainment, education, information, and business will be provided. The burgeoning field of digital communications will thus continue to affect almost all aspects of our contemporary life.

A basic definition of digital is the transmission of a message using binary digits (bits) or symbols from a finite alphabet during a finite time interval (bit or symbol duration). A bit or symbol occurring in each interval is mapped onto a continuous-time waveform that is then sent across the channel. Over any finite interval, the continuous-time waveform at the channel output belongs to a finite set of possible waveforms. This is in contrast to analog communications, where the output can assume any possible waveform. Digital can bring about many significant benefits, of course, at the expense of few shortcomings, for there is no free lunch in digital communications.

B.1.1 Advantages of Digital

- **Design efficiency:** Digital is inherently more efficient than analog in exchanging power for bandwidth, the two premium resources in communications. Since an essentially unlimited range of signal conditioning and processing options are available to the designer, effective trade-offs among power, bandwidth, performance, and complexity can be more readily accommodated. For any required performance, there is a three-way trade-off among power, bandwidth, and complexity (i.e., an increase in one means the other two will be reduced).

- **Versatile hardware:** The processing power of digital integrated circuits continues to approximately double every 18 months to 2 years. These programmable processors easily allow the implementation of improved designs or changed requirements. Digital circuits are generally less sensitive to physical effects, such as vibration, aging components, and external temperature. They also allow a greater dynamic range (the difference between the largest and the smallest signal values). Processing is now less costly than precious bandwidth and power resources. This in turn allows considerable flexibility in designing communication systems.

- **New and enhanced services:** In today's widely distributed way of life, Internet services, such as web browsing, e-mailing, texting, e-commerce, streaming and interactive multimedia services, have all become feasible and some even indispensable. It is also easier to integrate different services, with various modalities, into the same transmission scheme or to enhance services through transmission of some additional information, such as playing music or receiving a phone call with all relevant details.

- **Control of quality**: A desired distortion level can be initially set and then kept nearly fixed at that value at every step (link) of a digital communication path. This reconstruction of the digital signal is done by appropriately-spaced regenerative repeaters, which do not allow accumulation of noise and interference. On the other hand, once the analog signal is distorted, the distortion cannot be removed and a repeater in an analog system (i.e., an amplifier) regenerates the distortion together with the signal. In a way, in an analog system, the noises add, whereas in a digital system, the bit error rates add. In other words, with many regenerative repeaters along the path, the impact in an analog sys• tem is a reduction of many decibels (dBs) in the signal-to-noise ratio (SNR), whereas the effect in a digital system is a reduction of only a few dBs in the SNR.

- **Flexibility, compatibility, and switching**: Combining various digital signals and digitized analog signals from different users and applications into streams of different speeds and sizes—along with control and signaling information—can be much easier and more efficient. Signal storage, reproduction, interface with computers, as well as access and search of information in electronic data• bases can also be quite easy and inexpensive. Digital techniques allow the development of communication components with various features that can easily interface with a different component produced by a different manufacturer. Digital transmission brings about the great ability to dynamically switch and route messages of various types, thus offering an array of network connectivity's, including unicast, multicast, narrowcast, and broadcast.

B.2 Source coding

The source encoder in Figure B.1 has the function of converting the input from its original form into a sequence of bits. As discussed before, the major reasons for this almost universal conversion to a bit sequence are as follows: inexpensive digital hardware, standardized interfaces, layering, and the source/channel separation theorem.

The simplest source coding techniques apply to discrete sources and simply involve representing each successive source symbol by a sequence of binary digits. For example, letters from the 27• symbol English alphabet (including a space symbol) may be encoded into 5-bit blocks. Since there are 32 distinct 5-bit blocks, each letter may be mapped into a distinct 5-bit block with a few blocks left over for control or other symbols. Similarly, upper-case letters, lower-case letters, and a great many special symbols may be converted into 8-bit blocks ("bytes") using the standard ASCII code.

Treats coding for discrete sources and generalizes the above techniques in many ways. For example the input symbols might first be segmented into *m*-tuples, which are then mapped into blocks of binary digits. More generally yet, the blocks of binary digits can be generalized into variable-length sequences of binary digits. We shall find that any given discrete source, characterized by its alphabet and probabilistic description, has a quantity called entropy associated with it. Shannon showed that this source entropy is equal to the minimum number of binary digits per source symbol required to map the source output into binary digits in such a way that the source symbols may be retrieved from the encoded sequence.

Some discrete sources generate finite segments of symbols, such as email messages, that are statistically unrelated to other finite segments that might be generated at other times. Other discrete sources, such as the output from a digital sensor, generate a virtually unending sequence of symbols with a given statistical characterization. The most straightforward approach to analog source coding is called analog to digital (A/D) conversion. The source waveform is first sampled at a sufficiently high rate

Each sample is then quantized sufficiently finely for adequate reproduction. For example, in standard voice telephony, the voice waveform is sampled 8000 times per second; each sample is then quantized into one of 256 levels and represented by an

8-bit byte. This yields a source coding bit rate of 64 Kbps.

Beyond the basic objective of conversion to bits, the source encoder often has the further objective of doing this as efficiently as possible— *i.e.*, transmitting as few bits as possible, subject to the need to reconstruct the input adequately at the output. In this case source encoding is often called data compression. For example, modern speech coders can encode telephone-quality speech at bit rates of the order of 6-16 kb/s rather than 64 kb/s.

The problems of sampling and quantization are largely separable. Chapter 3 develops the basic principles of quantization. As with discrete source coding, it is possible to quantize each sample separately, but it is frequently preferable to segment the samples into *n*-tuples and then quantize the resulting *n*-tuples. As shown later, it is also often preferable to view the quantized output as a discrete source output and then to encode the quantized symbols

The purpose of sampling is to convert the analog source into a sequence of real-valued numbers, *i.e.*, into a discrete-time, analog-amplitude source. There are many other ways, beyond sampling, of converting an analog source to a discrete-time source. A general approach, which includes sampling as a special case, is to expand the source waveform into an orthonormal expansion and use the coefficients of that expansion to represent the source output. It forms the basis for the signal space approach to channel encoding/decoding.

B.3 Communication channels

We next discuss the channel and channel coding in a generic digital communication system.

In general, a channel is viewed as that part of the communication system between source and destination that is given and not under the control of the designer. Thus, to a source-code designer, the channel might be a digital channel with binary input and output; to a telephone- line modem designer, it might be a 4 KHz voice channel; to a cable modem designer, it might be a physical coaxial cable of up to a certain length, with certain bandwidth restrictions.

When the channel is taken to be the physical medium, the amplifiers, antennas, lasers, etc. that couple the encoded waveform to the physical medium might be regarded as part of the channel or as as part of the channel encoder. It is more common to view these coupling devices as part of the channel, since their design is quite separable from that of the rest of the channel encoder. This, of course, is another example of layering.

Channel encoding and decoding when the channel is the physical medium (either with or without amplifiers, antennas, lasers, etc.) is usually called (digital) modulation and demodulation respectively. The terminology comes from the days of analog communication where modulation referred to the process of combining a low pass signal waveform with a high frequency sinusoid, thus placing the signal waveform in a frequency band appropriate for transmission and regulatory requirements. The analog signal waveform could modulate the amplitude, frequency, or phase, for example, of the sinusoid, but in any case, the original waveform could be retrieved by demodulation.

As digital communication has increasingly replaced analog communication, the modulation/demodulation terminology has remained, but now refers to the entire process of digital encoding and decoding. In most such cases, the binary sequence is first converted to a baseband waveform and the resulting baseband waveform is converted to bandpass by the same type of procedure used for analog modulation. As will be seen, the challenging part of this problem is the conversion of binary data to baseband waveforms. Nonetheless, this entire process will be referred to as modulation and demodulation, and the conversion of baseband to passband and back will be referred to as frequency conversion.

As in the study of any type of system, a channel is usually viewed in terms of its possible inputs, it’s possible outputs, and a description of how the input affects the output. This description is usually probabilistic. If a channel were simply a linear time-invariant system (*e.g.*, a filter), then it could be completely characterized by its impulse response or frequency response. However, the channels here (and channels in practice) always have an extra ingredient – noise.

Suppose that there were no noise and a single input voltage level could be communicated exactly. Then, representing that voltage level by its infinite binary expansion, it would be possible in principle to transmit an infinite number of binary digits by transmitting a single real number. This is ridiculous in practice, of course, precisely because noise limits the number of bits that can be reliably distinguished.

Channel encoding (modulation)

Modulation is defined as a process through which an information-bearing message (i.e., modulating) signal is used to modify (i.e., modulate) some parameter (e.g., amplitude, frequency, phase) of a periodic (such as a high-frequency sinusoidal) signal known as a carrier wave, individually or in combination, to produce the modulated signal for transmission. This is similar to when a musician modulates a tone by varying its volume (amplitude), its timing (phase), and its pitch (frequency).

We have three distinct modulation categories: amplitude modulation (AM), frequency modulation (FM), and phase modulation (PM). However, in the context of digital communications, the term shift keying is generally used instead of the word modulation. To this effect, we have amplitude-shift keying (ASK), frequency-shift keying (FSK), and phase-shift keying (PSK). There is also another category in digital communications, which is a hybrid of ASK and PSK, known as quadrature amplitude modulation (QAM). It is important to note that PSK and QAM are used in digital communication systems far more often than ASK and FSK. In some cases, the modulated signal is simply related to the message signal, such as with ASK, and in some cases, the relationship is rather complicated, such as with QAM.

A modem is a device that provides two-way communications, and thus performs both modulation in the transmitter and demodulation (i.e., the inverse operation of modulation) in the receiver. Modulation is performed to achieve one or more of the following objectives

Modulation for efficient radio transmission: For radio communications, antennas are needed to radiate (transmit) and receive the modulated signal. The size of the antenna depends on the wavelength and the application. Modulation helps translate the message signal with low frequency components into a signal with much higher frequency components. The resulting modulated signal can thus possess a much smaller wavelength, and that in turn allows much smaller antennas. The signal

Wavelength λ can be found using λ ¼$_{f}^{c}$, where f is the signal frequency in Hertz (Hz), c is the speed of light in meters per second (MPs), and λ in meters (m). As an example, efficient electromagnetic radiation for line-of-sight radio propagation generally requires antennas whose physical dimensions are at least 10% of the signal's wavelength. Therefore, an audio signal with frequency components up to 100 Hz requires an antenna about 300 km, but with modulation at 300 MHz, the antenna needs to be only about 10 cm long. Another example is that for cellular mobile telephones, antennas are typically 25% of the signal wavelength, so for the above-mentioned audio signal, an antenna spanning 750 km would be required, but with modulation at 900 MHz, the equivalent antenna diameter would be only about 8 cm.

Modulation to match channel characteristics: Modulation allows modification of the message signal to a form suited to the characteristics of the transmission channel. The majority of practical channels have bandpass characteristics and modulation translates the frequency components of the low pass message signal to the passband range, so the spectrum of the transmitted signal can match the characteristics of the channel. Applications may include satellite TV, in which baseband video signals with frequency components up to 6 MHz are converted into RF signals operating at 14 GHz, and cellular phones, in which baseband speech signals with frequency components up to 4 kHz are converted into RF signals operating at 900 MHz

Modulation for frequency assignment: Modulation allows many radio and television stations to broadcast simultaneously in a given geographical area. Since each station has a different assigned carrier frequency, the desired broadcast signals can be separated from others by tuning the receiver to select different stations as required.

Modulation for multiplexing: When more than one signal needs to utilize a single channel, modulation may be used to translate different signals to different spectral locations. Applications include FM stereophonic broadcasting, in which the sum of the right-hand and left-hand signals and their differences are accommodated into a single channel, and cable TV, in which a number of TV channels along with the upstream and downstream Internet traffic are all integrated into a single channel.

Modulation to allow common processing: Sometimes the frequency range of the signal to be processed and the frequency range of the processing device do not match, and the processing device is complex. Modulation allows the processing equipment to operate in some fixed frequency range and instead translate the frequency range of the signal to correspond to the fixed frequency range of the processing equipment. This is the case when in a system, such as AM radio, all RF signals coming from various AM radio stations are converted to a certain intermediate-frequency (IF) in a receiver.

Modulation to overcome hardware limitations: The performance, design, and cost of some signal-processing devices, such as filters and amplifiers, often depend on the signal spectral location and the ratio of the highest to lowest signal frequencies. It is generally desirable to keep the ratio of signal bandwidth to its center frequency within 1–10%. Modulation can be therefore used to translate the signal to a location in the frequency domain where design requirements can be better met.

Modulation to reduce noise and interference: The effect of noise and interference cannot be completely eliminated in a communication system. However, by significantly expanding the bandwidth of the transmitted signal, the noise and interference immunity in some cases can be considerably enhanced. In other words, bandwidth increase is traded for noise and interference reduction. Applications include FM radio and spread spectrum techniques.

Modulation to allow design trade-offs: Modulation techniques, employing digital M-ary (vis-a`-vis binary) signaling, can provide a balancing act to optimally achieve digital communication design objectives, such as higher trans• mission rate, lower transmit power, smaller signal bandwidth, lower bit error rate, and more modest complexity. By increasing M, bandwidth can be saved at the expense of an increase in bit error rate. By using trellis-coded modulation, transmission rate can be enhanced, not with an increase in bandwidth but at the expense of increased modem complexity.

Error correction

Frequently the error probability incurred with simple modulation and demodulation techniques is too high. One possible solution is to separate the channel encoder into two layers, first an error-correcting code, and then a simple modulator.

As a very simple example, the bit rate into the channel encoder could be reduced by a factor of 3, and then each binary input could be repeated 3 times before entering the modulator. If at most one of the 3 binary digits coming out of the demodulator were incorrect, it could be corrected by majority rule at the decoder, thus reducing the error probability of the system at a considerable cost in data rate. The scheme above (repetition encoding followed by majority-rule decoding) is a very simple example of error-correction coding. Unfortunately, with this scheme, small error probabilities are achieved only at the cost of very small transmission rates

Digital interface

The interface between the source coding layer and the channel coding layer is a sequence of bits. However, this simple characterization does not tell the whole story. The major complicating factors are as follows:

Unequal rates: The rate at which bits leave the source encoder is often not perfectly matched to the rate at which bits enter the channel encoder.

Errors: Source decoders are usually designed to decode an exact replica of the encoded sequence, but the channel decoder makes occasional errors.

Networks: Encoded source outputs are often sent over networks, traveling serially over several channels; each channel in the network typically also carries the output from a number of different source encoders.

The first two factors above appear both in point-to-point communication systems and in net• works. They are often treated in an ad hoc way in point-to-point systems, whereas they must be treated in a standardized way in networks. The third factor, of course, must also be treated in a standardized way in networks.
The usual approach to these problems in networks is to convert the superficially simple binary interface above into multiple layers.

B.4 GUIDED-TRANSMISSION MEDIA

Information transmission across a communication network is accomplished in the physical layer by means of a transmission medium to convey the energy of a signal from a transmitter to a receiver. In telecommunications, transmission media can be divided into two broad categories: guided (wired) and unguided (radio). Guided-transmission media include twisted-pair cable, coaxial cable, and fiber-optic cable, as shown in and the unguided-transmission medium is the atmosphere or free space, through which electromagnetic waves are propagated to convey information.

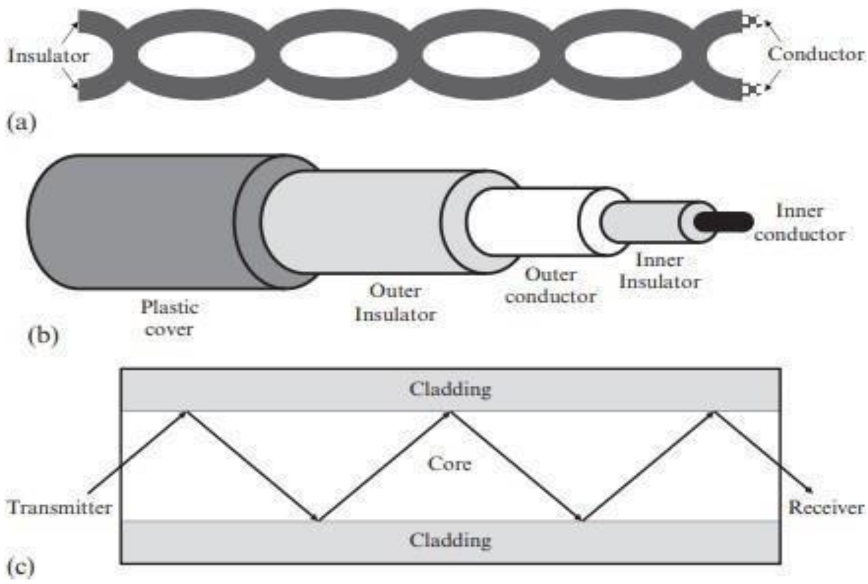


Fig: B.2 transmission media

Twisted-Pair Cable

Twisted-pair was designed and built mainly for speech communications. As shown in fig twisted-pair cable consists of two insulated conducting (typically copper) wires, closely twisted together to reduce the susceptibility to crosstalk (electrical signals from other adjacent wires) and noise. One of the wires is used to carry signals to the receiver, and the other is used only as a ground reference. The receiver detects the information signal by the voltage difference between the two.

A twisted-pair cable can pass a wide range of frequencies. At a given

Frequency, a higher gauge (thicker) wire yields a higher signal attenuation. The attenuation for twisted-pair, measured in dB per distance, sharply increases with frequencies above 100 kHz. Since the attenuation per distance is higher for higher frequencies, the bandwidth of twisted-pair decreases with distance. Depending on the gauge of the unshielded twisted-pair, the attenuation can roughly range between 1–4 dB/km at 4 kHz, between 5–10 dB/km at 100 kHz, and 10dB/km at 1 MHz For digital transmission over a twisted-pair cable, repeaters are required every couple of kilometers.

Twisted-pair cables are used in the telephone network to transmit voice. The local loop, the line connecting subscribers to the central office, consists of unshielded twisted-pair (UTP) cables. To improve voice transmission, the transmission frequencies are limited to 4 kHz to reduce the crosstalk and load• ing coils are added to provide a flatter transfer function. These factors, not the inherent bandwidth of twisted-pair, limit the digital transmission rate over tele• phone lines to below approximately 40 kbps.

However, a digital subscriber line (DSL), which uses a UTP cable, can provide much higher rates for short distances, provided that the user equipment and the interface at the central office are both changed to match the DSL transmission requirements. Besides distance, there are other factors, such as the size of the wire, the signaling type, crosstalk interference from other lines, and the SNR value, which can affect the available bandwidth and in turn the transmission rate. Moreover, twisted-pair cables are used in Ethernet LANs, such as 10Base-T at 10 Mbps, 100Base-TX at 100 Mbps, 1000Base-T at 1 Gbps, capable of carrying various modalities

.

Coaxial Cable

As shown in Figure coaxial cable consists of an inner (central) conductor of solid or stranded wire (usually copper) and an outer conductor of metal foil, braid, or a combination of two, separated by a dielectric insulating material. The outer conductor is also enclosed in an insulating sheath, and the whole cable can be protected by a plastic cover.

As opposed to twisted-pair cables, coaxial cables provide much better immunity to crosstalk and interference, offer much larger bandwidths (hundreds of MHz), but yield higher levels of attenuation. In other words, although coaxial cable has a much higher bandwidth, the signal weakens rapidly and requires the frequent use of repeaters, roughly every kilometer or so. The widest use of coaxial cable is for the distribution of television signals in cable TV systems, and it is also used for local area networks (LANs). Rates up to tens of Mbps are feasible using coaxial cables, with 10 Mbps being the standard. The attenuation can roughly range between 7–27 dB/km at 10 MHz

Fiber-Optic Cable

An optical fiber is a dielectric waveguide that transports light signals just as metallic (twisted-pair or coaxial) cable transports electrical signals. As shown in Figure, an optical fiber consists of a very fine cylinder of glass (core) surrounded by a con• centric layer of glass (cladding). The core has a slightly higher optical density (index of refraction) than the cladding. Therefore, when a ray of light from the core approaches the cladding, the ray is completely reflected back into the core, and the ray of light is guided within the fiber. Information is transmitted by varying the intensity of the light source with the message signal. The light in the fiber is periodically amplified and regenerated by repeaters along the transmission path, and at the receiver, the light intensity is detected by a photodiode.

There are two types of optical fibers, the low-cost multimode fibers and the low-loss single-mode fibers. A multimode fiber has a ray of light that can reach the receiver over multiple paths. Since each path has its own delay, the differences in delays cause interference. This limits the maximum bit rates that are achievable using multimode fibers. In a single-mode fiber where the core of fiber is much narrower, the single mode propagates with low loss and dispersion, thus requiring much fewer repeaters. Sources of noise in fiber-optic cables are photodiodes and electronic amplifiers. Optical fibers have unique characteristics, such as an enormous potential bandwidth, low transmission losses, immunity to electromagnetic interference, small size and weight, ruggedness, and flexibility. Optical fiber transmission systems are widely deployed in backbone net• works, and can provide nearly error-free transmission rates up to several hundred Gbps over tens of kilometers, as typical attenuation is about 0.2–0.5 dB/km.

RADIO TRANSMISSION

Radio encompasses the electromagnetic spectrum in the range of 3 kHz to 300 GHz. Radio transmission uses an unguided medium, and may possess principal benefits, but at the expense of some major shortcomings.

Radio Spectrum

It is imperative to highlight that spectrum is a very scarce commodity, and efficient use of any part of spectrum is of paramount importance. Radio spectrum refers to the part of the electromagnetic spectrum corresponding to radio frequencies i.e., frequencies lower than around 300 GHz (or, equivalently, wavelengths longer than about 1 mm). Frequency assignments and technical standards are set internationally by the International Telecommunications Union (ITU). The Radio communication Sector of ITU (ITU-R) provides frequency assignments and is concerned with the efficient use of the radio frequency spectrum.

Table B.1 shows the frequency bands and some of their major aspects. The frequency bands are designated in logarithmic frequency and the progression of frequency bands has increasingly larger bandwidths. For instance, the MF band (from 0.3–3 MHz) has a bandwidth of 2.7 MHz, whereas the VHF band (from 30–300 MHz) has a bandwidth of 270MHz. Higher bandwidths can generally lend themselves to higher system capacities and transmission rates as well as lower levels of interference and bit error rates, but at the expense of higher signal attenuation and equipment complexity.

Table B.1 Frequency bands and their major aspects

Frequency band	Frequency range	Wavelength range	Transmission media	Propagation mode
Extra-Low Frequency (ELF)	3–30 Hz	100,000–10,000 km	Wire pairs	Ground wave
Super-Low Frequency (SLF)	30–300 Hz	10,000–1,000 km	Wire pairs	Ground wave
Ultra-Low Frequency (ULF)	300–3 kHz	1,000–100 km	Wire pairs	Ground wave
Very-Low Frequency (VLF)	3–30 kHz	100–10 km	Wire pairs	Ground wave
Low Frequency (LF)	30–300 kHz	10–1 km	Wire pairs	Ground wave
Medium Frequency (MF)	0.3–3 MHz	1–0.1 km	Wire pairs & coaxial cable	Ground wave & sky wave
High Frequency (HF)	3–30 MHz	100–10 m	Coaxial cable	Sky wave
Very-High Frequency (VHF)	30–300 MHz	10–1 m	Coaxial cable	Sky wave & line of sight
Ultra-High Frequency (UHF)	0.3–3 GHz	1–0.1 m	Coaxial cable & waveguide	Line of sight
Super-High Frequency (SHF)	3–30 GHz	100–10 mm	Waveguide	Line of sight
Extra-High Frequency (EHF)	30–300 GHz	10–1 mm	Waveguide	Line of sight
Infrared	0.3–430 THz	1 mm–700 nm	Optical fibers	Laser beams
Visible Light	430–750 THz	700–400 nm	Optical fibers	Laser beams
Ultraviolet	0.75–30 PHz	400–10 nm	Optical fibers	Laser beams

Wave Propagation

Like light waves, radio signals by nature travel in a straight line, and therefore propagation beyond line of sight requires a means of deflecting the radio waves. The available methods are reflection (when the radio signal is bounced off a surface), refraction (when the radio signal bends due to a change in medium), diffraction (when the radio signal meets a sharp edge and redirects), and scattering (when the radio signal spreads out). For any type of radio communications, the signal disperses with distance. The signal attenuation in free space is inversely related to the square of the distance that the radio signal must travel as well as the square of the frequency that the radio signal is operating at.

Depending on the frequency and antenna, the radiated energy can propagate in either a unidirectional or omnidirectional fashion. In the former case, a properly-aligned antenna can receive the modulated signal, and in the latter case, any antenna in the area of coverage can receive the signal. In general, radio frequencies below 1 GHz or so are more suitable for omnidirectional applications and above 1 GHz or so are typically tailored for unidirectional applications. Also, at low and medium frequencies, radio waves can penetrate walls. This is viewed as an advantage when a signal is required to be received inside a building and is regarded as a disadvantage when it is required to isolate a communication to just inside or outside a building to reduce the level of interference.

The range of 300 MHz to 300 GHz is known as microwave radio frequencies. Rain attenuation, which refers primarily to the absorption of a microwave frequency signal by atmospheric rain, snow, or ice, is a dominant source of signal degradation. Rain attenuation is a function of many factors, such as location, distance, elevation angle, and frequency. Rain attenuation is directly related to frequency.

Radio waves at different frequencies propagate in different ways, there are three distinct methods for the transmission of radio signals: ground-wave propagation, sky-wave propagation, and line-of-sight propagation.

Ground-wave propagation: It is the dominant mode of propagation for frequencies below 2 MHz (all bands up to and including the lower part of MF band). In this frequency range, the Earth and the ionosphere (the layer of atmosphere where particles exist as ions) act as a waveguide for radio wave propagation. These low-frequency signals propagate (by diffraction) in all directions around the curved surface of the Earth for thousands of kilometers. Distance depends on the amount of power in the signal. Since the ground is not a perfect electrical conductor, ground waves are attenuated rapidly as they follow the Earth's surface. The signal attenuation is a function of time and the frequency band. Also, the atmospheric noise level is rather high. The channel bandwidths available in these frequency bands are rather modest and in turn yield rather low transmission speeds. Typical applications include long-range navigation and maritime communications, radio beacon, and AM radio broadcasting.

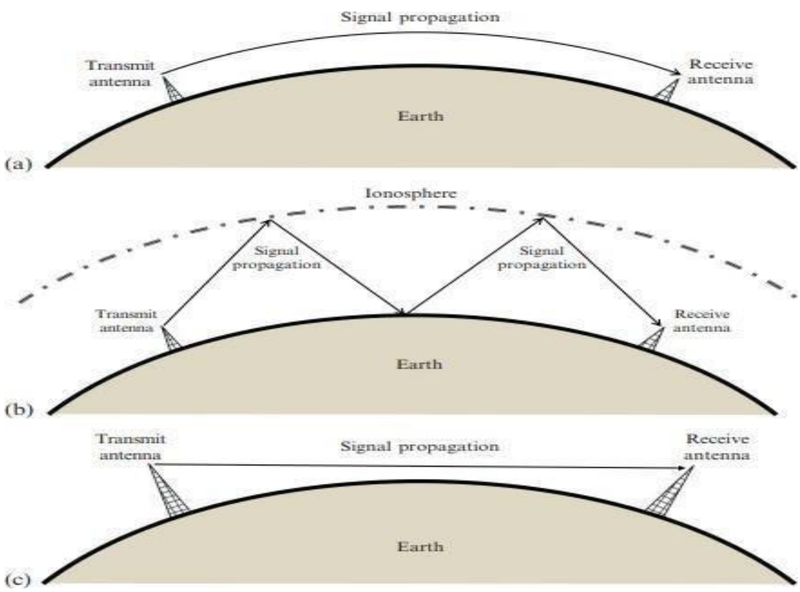


Fig: B.3 propagation of signal

Sky-wave propagation: It is the dominant mode of propagation in the frequency range of 2 MHz to about 30 MHz or in some cases 60 MHz (the upper part of MF band, HF band, and the lower part of VHF band). Sky-wave propagation results when the signal is reflected (bent or refracted) from the ionosphere (ionized layer of the upper atmosphere). In the HF band, signal multipath and fading can be a source of degradation. This impairment may be experienced when listening to a distant radio station at night when the sky-wave propagation is the dominant mode. The angle of reflection and the loss of signal at an ionospheric reflection point depend on the frequency, the time of the day, the season, and the sunspot activity. Ionosphere and troposphere scattering involves large signal propagation losses and requires a rather large amount of transmit power and relatively large antennas. The additive noise is a combination of atmospheric noise and thermal noise. Cosmic noise, which is random noise originating outside the Earth’s atmosphere, impacts sky-way propagation. The typical transmission range in sky-wave propagation can be in hundreds of kilometers. Typical applications include FM radio broadcasting, short-wave broadcasting, amateur radio, and CB radio.

Line-of-sight propagation: From the upper part of the VHF band up to and including the EHF band, signals must be transmitted in straight lines directly from antenna to antenna, hence the term line of sight. The transmit and receive antennas are required to be directional and facing each other. The direct path connecting the antennas in terrestrial communications can be affected by the curvature of the Earth. The distance between the transmitter and the receiver is therefore a function of the heights of the transmit and receive antennas. For this reason, television stations transmitting off-the• air signals or microwave radio relay systems mount their antennas on high towers or buildings to reach a broad coverage area. Specifically, for terrestrial communications, the maximum distance between transmit and receive antennas for direct line-of-sight radio propagation is about D ¼ 17 $H_T + H_R$, where H_T and H_R are the heights of transmit and receive antennas in meters, respectively, and D is the maximum distance in kilometers over which communications between them can take place by direct line-of-sight radio signals. Applications include off-the-air (VHF and UHF) TV broadcasting, which are examples of line-of-sight terrestrial communications, and satellite TV broad• casting and VSAT networks, which are examples of line-of-sight satellite communications.

A Major Project report on

JOINT CHANNEL ESTIMATION AND IMPULSIVE NOISE MITIGATION METHOD FOR OFDM SYSTEMS USING SPARSE BAYESIAN LEARNING

Submitted in partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

IN

Electronics & Communication Engineering

Submitted By

K.Ravichandra (17K81A0422)

M.Vamshy (17K81A0433)

P.Varshini (17K81A0438)

Under the Guidance of

Mr. D. Prasad M.Tech

Assistant Professor

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

UGC AUTONOMOUS

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

Dhulapally, Secunderabad-500 100.

NBA & NAAC A+ Accredited

2020-2021

BONAFIDE CERTIFICATE

Certified that this project report titled "*JOINT CHANNEL ESTIMATION AND IMPLUSIVE NOISE MITIGATION METHOD FOR OFDM SYSTEMS USING SPARSE BAYESIAN LEARNING*", is a *bona fide* work of **1. Mr. K. Ravi Chandra (17K81A0422), 2. Mr. M. Vamshy (17K81A0433), 3. Ms. P. Varshini (17K81A0438), Dr. D. Prasad** who carried out the work under my supervision, for the partial fulfillment of the requirements for the award of the degree of *Bachelor of Technology* in *Electronics and Communication Engineering.* Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion.

Place: Secunderabad

Date:

Dr. B. Hari Krishna,

B.Tech, M.Tech, Ph.D.,

Professor & HOD

Department of ECE

St.Martin's Engineering College,

COUNTER SIGNED

HEAD

Department of ECE

St.Martin's Engineering College

DECLARATION

We declare that this project report titled JOINT CHANNEL ESTIMATION AND IMPLUSIVE NOISE MITIGATION METHOD FOR OFDM SYSTEMS USING SPARSE BAYESIAN LEARNING submitted in partial fulfillment of the degree of B. Tech in Department of electronics and communication engineering record of original work carried out by us under the guidance and supervision of Dr. D. Prasad, and has not formed the name basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

1. Mr. K. Ravi Chandra (17K81A0433)	2.Mr. M. Vamshy (17K81A0433)	3. Ms. P. Varshini (17K81A0438)

Place: Secunderabaad

ACKNOWLEGMENTS

While bringing out this project report to its final form, we came across a number of people whose contributions in various ways helped my field of research and they deserve special thanks. It is a pleasure to convey our gratitude to all of them.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr. Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr. B. HariKrishna**, **Head &Professor**, the Dept. of electronics and communication engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept. project coordinator Dr. A. Soumya** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide, **Mr. D. Prasad**, **Assistant Professor**, Department of ECE and our class project coordinator **Ms. A. Soumya**, **Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

We are highly grateful and obliged to **our college level Project Coordinators Dr. Maalyadri**, **Professor in IT** and **Dr. R. Santosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. K.RAVI CHANDRA
2. M.VAMSHY
3. P.VARSHINI

ABSTRACT

The impulsive noise can deteriorate sharply the performance of orthogonal frequency division multiplexing (OFDM) systems. In this paper, we propose a novel joint channel impulse response estimation and impulsive noise mitigation algorithm based on compressed sensing theory. In this algorithm, both the channel impulse response and the impulsive noise are treated as a joint sparse vector. Then, the sparse Bayesian learning framework is adopted to jointly estimate the channel impulse response, the impulsive noise, and the data symbols, in which the data symbols are regarded as unknown parameters. The Cramér–Rao Bound is derived for the benchmark. Unlike the previous impulsive noise mitigation methods, the proposed algorithm utilizes all subcarriers without any a priori information of the channel and impulsive noise. The simulation results show that the proposed algorithm achieves significant performance improvement on the channel estimation and bit error rate performance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

OFDM	Orthogonal Frequency Division Multiplexing
MIMO	Multiple-Input Multiple-Output
MMSE	Minimum Mean Square Error
SNR	Signal to Noise Ratio
SBL	Sparse Bayesian Learning
FFT	Fast Fourier Transform
QAM	Quadrature Amplitude Modulation
GUI	Graphical User Interface
ISP	Internet Service Providers
AWGN	Additive White Gaussian Noise
PS	Phase Shift Keying
BER	Bit Error Rate

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVES OF THE STUDY

Orthogonal frequency division multiplexing (OFDM) has become the most popular multicarrier signaling format for high-speed wireless communications and has been included in many standards such as Long Term Evolution (LTE), IEEE 802.11 wireless local area network (WLAN) and digital video broadcasting (DVB). Due to its efficient implementation and robustness to the frequency selectivity, OFDM and its combination with multiple-input multiple-output (MIMO) system sun surprisingly appears as a strong alternative for 5G networks. OFDM with index modulation (OFDM-IM) is a recently proposed novel scheme which transmits information not only by M-ary constellation symbols, but also by the indices of the active subcarriers which are activated according to the incoming information bits. Subcarrier index modulation techniques for OFDM have attracted considerable attention from researchers and have been investigated in some recent studies due to interesting trade-offs they offer in error performance and spectral efficiency compared to classical OFDM systems. The bit error probability of OFDM-IM is analytically derived. The spectral efficiency of OFDM-IM is improved by selecting the active subcarriers in a more flexible way in, where index modulation is applied for both in-phase and quadrature components of the subcarriers. The authors deal with the problem of selecting the optimal number of active subcarriers in OFDM-IM. More recently, OFDM-I Mis-combined with coordinate interleaving to achieve additional diversity gains. However, the combination of OFDM-IM and MIMO transmission techniques remains an open and interesting research problem. Inthisstudy, we propose MIMO-OFDM with index modulation (MIMO-OFDM-IM) as an efficient alternative multicarrier transmission scheme for 5G networks by combining MIMO and OFDM-IM transmission techniques. In the proposed scheme, each transmit antenna transmits its own OFDM-IM frame as in Vertical Bell Labs layered space-time (V-BLAST) scheme, and at the receiver side, these OFDM-IM frames are separated and demodulated using a novel and low complexity minimum mean square error (MMSE) detection and log-likelihood ratio (LLR) calculation based detector. It is shown via computer simulations that the MIMO-OFDM-IM scheme achieves significantly better bit error rate (BER) performance than classical V-BLAST type MIMO-OFDM for several MIMO configurations. The rest of the letter is organized as follows. The system model of MIMO-OFDM-IM is presented,receiver structure of the MIMO –OFDM –I M-scheme is given. Simulation results are provided. we propose MIMO-OFDM with index modulation (MIMO-OFDM-IM) as an efficient alternative multicarrier transmission scheme for 5G networks by combining MIMO and OFDM-IM transmission techniques. In the proposed scheme, each transmit antenna transmits its own OFDM-IM frame as in Vertical Bell Labs layered space-time (V-BLAST) scheme, and at the receiver side, these OFDM-IM frames are separated and demodulated using a novel and low complexity minimum mean square error (MMSE) detection and log-likelihood ratio (LLR) calculation baseddetector. Orthogonal frequency division multiplexing (OFDM) has become one among the most common multicarrier transmission techniques for wideband wireless communications in recent years. Because of its benefits like efficient implementation and strength to frequency selective fading channels, OFDM has been enclosed in several standards like long term Evolution (LTE), IEEE 802.11x wireless local area network (LAN), digital video broadcasting (DVB) and IEEE 802.16e-WiMAX. Considering the benefits of multiple-input multiple-output (MIMO) systems over single antenna systems like improved rate and energy efficiency, the mixture of OFDM and MIMO transmission techniques seems as a robust different for future wireless standards like 5G and beyond. OFDM with index modulation (OFDM-IM) can be a new multicarrier transmission method that has been planned as another to classical OFDM. Stimulating from the SM conception, in OFDM-IM, index modulation techniques are applied for the indices of the offered subcarriers of an OFDM system. In OFDM-IM method, only a set of available subcarriers are preferred as active consistent with the data bits, whereas the remaining inactive subcarriers are set zero. In different words, the data is transmitted not only by the information symbols selected from M-ary signal constellations, however additionally by the indices of the active subcarriers. Not like classical OFDM, the quantity of active subcarriers may be adjusted within the OFDM-IM scheme, and this flexibility within the system style provides a stimulating trade-off between error performance and spectral efficiency. Moreover, it's been shown that OFDM-IM has the potential to realize a more robust error performance than classical OFDM for low-to-mid spectral efficiency values. Due to its adjustable range of active subcarriers, OFDM-IM may be a potential candidate not just for high speed wireless communications systems however additionally for machine-to-machine (M2M) communications systems that need low power consumption.

In statistics and signal processing, a minimum mean square error (MMSE) estimator is an estimation method which minimizes the mean square error (MSE), which is a common measure of estimator quality, of the fitted values of a dependent variable. In the Bayesian setting, the term MMSE more specifically refers to estimation with quadratic loss function. In such case, the MMSE estimator is given by the posterior mean of the parameter to be estimated. Since the posterior mean is cumbersome to calculate, the form of the MMSE estimator is usually constrained to be within a certain class of functions. Linear MMSE estimators are a popular choice since they are easy to use, easy to calculate, and very versatile. It has given rise to many popular estimators such as the Wiener–Kolmogorov filter and Kalman filter.

An M-ary transmission is a type of digital modulation where instead of transmitting one bit at a time, two or more bits are transmitted simultaneously. This type of transmission results in reduced channel bandwidth. This process is known as quadrature modulation.

M-ary Encoding. The word binary represents two-bits. ... This is the type of digital modulation technique used for data transmission in which instead of one-bit, two or more bits are transmitted at a time. As a single signal is used for multiple bit transmission, the channel bandwidth is reduced. An M-ary transmission is a type of digital modulation where instead of transmitting one bit at a time, two or more bits are transmitted simultaneously. This type of transmission results in reduced channel bandwidth. However, sometimes, two or more quadrature carriers are used for modulation. Multi-level modulation techniques permit high data rates within fixed bandwidth constraints. A convenient set of signals for M-ary PSK is

$\varphi i(t) = A \cos(\omega ct + \theta i), 0 < t \leq Ts,$

Where the M phase angles are $\theta i = 0, 2\pi M , . . ., 2(M - 1)\pi M$.

For equi probable ones and zeros the PSD for M-ary PSK is S

$\varphi(\omega) = A\ 2\ TsSa2(\omega - \omega c)\ Ts\ 2.$

The symbols in this case are of duration Ts, so the information (or bit) rate Tb satisfies Ts = Tb log2 M. The potential bandwidth efficiency of M-ary PSK can be shown to be fb B = log2 M bps/Hz.

All signals have the same energy Es over the interval (0, Ts), and each signal is correctly demodulated at the receiver if the phase is within ±π/M of the correct phase θi. No information is contained in the energy of the signal. A probability of error calculation involves analysing the received phase at the receiver (in the presence of noise), and comparing it to the actual phases. An exact solution is difficult to compute, but for P< 10−3 an approximate probability of making a symbol error is P 2erfcs 2Es η sin2 π M, M > 2. If a Gray code is used, then the corresponding bit error is approximately Pbe/ log2 M. Stremler provides a table of the SNR requirements of M-ary PSK for fixed error rates. The results indicate that for QPSK (M = 4) has definite advantages over coherent PSK (M = 2) the bandwidth efficiency is doubled for only about a 0.3dB increase in SNR. For higher-rate transmissions in band limited channels the choice M = 8 is often used. Values of M > 8 are seldom used due to excessive power requirements. M-ary PSK requires more complex equipment than BPSK signalling. Carrier recovery is also more complicated. The requirement that the carrier be recovered can be mitigated by using a comparison between the phases of two successive symbols. This leads to M-ary differential PSK, and is in principle similar to DPSK (which is differential PSK for M = 2). For large SNR the probability of error is P=2erfcs 2Es η sin2 π √ 2M .Thus differential detection increases the power requirements by the factor 0 = sin2 π/M sin2 (π/ √ 2M) For m = 4, the increase is required power is about 2.5dB, which may be justified by the saving in equipment complexity.

In telecommunications, orthogonal frequency-division multiplexing (OFDM) is a method of encoding digital data on multiple carrierfrequencies. OFDM has developed into a popular scheme for wideband digital communication, used in applications such as digital television and audio broadcasting, DSL internet access, wireless networks, power line networks, and 4G mobile communications.

In coded orthogonal frequency-division multiplexing (COFDM), forward error correction (convolutional coding) and time/frequency interleaving are applied to the signal being transmitted. This is done to overcome errors in mobile communication channels affected by multipath propagation and Doppler effects. COFDM was introduced by Alard in 1986 for Digital Audio Broadcasting for Eureka Project 147. In practice, OFDM has become used in combination with such coding and interleaving, so that the terms COFDM and OFDM co-apply to common applications.

OFDM is a frequency-division multiplexing (FDM) scheme used as a digital multi-carrier modulation method. OFDM was introduced by Chang of Bell Labs in 1966. Numerous closely spaced orthogonal sub-carrier signals with overlapping spectra are emitted to carry data. Demodulation is based on Fast Fourier Transform algorithms. OFDM was improved by Weinstein and Ebert in 1971 with the introduction of a guard interval, providing better orthogonality in transmission channels affected by multipath propagation. Each sub-carrier (signal) is modulated with a conventional modulation scheme (such as quadrature amplitude modulation or phase-shift keying) at a low symbol rate. This maintains total data rates similar to conventional single-carrier modulation schemes in the same bandwidth.

Conceptually, OFDM is a specialized frequency-division multiplexing (FDM) method, with the additional constraint that all subcarrier signals within a communication channel are orthogonal to one another.

In OFDM, the sub-carrier frequencies are chosen so that the sub-carriers are orthogonal to each other, meaning that cross-talk between the sub-channels is eliminated and inter-carrier guard bands are not required. This greatly simplifies the design of both the transmitter and the receiver; unlike conventional FDM, a separate filter for each sub-channel is not required.

The orthogonality requires that the sub-carrier spacing is Hertz, where T_U seconds is the useful symbol duration (the receiver-side window size), and k is a positive integer, typically equal to 1. This stipulates that each carrier frequency undergoes k more complete cycles per symbol period than the previous carrier. Therefore, with N sub-carriers, the total passband bandwidth will be $B= N*\Delta f$ (Hz).

The orthogonality also allows high spectral efficiency, with a total symbol rate near the Nyquist rate for the equivalent baseband signal (i.e. near half the Nyquist rate for the double-side band physical passband signal). Almost the whole available frequency band can be used. OFDM generally has a nearly 'white' spectrum, giving it benign electromagnetic interference properties with respect to other co-channel users.

In radio, multiple-input and multiple-output, or MIMO, is a method for multiplying the capacity of a radio link using multiple transmission and receiving antennas to exploit multipath propagation. MIMO has become an essential element of wireless communication standards including IEEE802.11n (Wi-Fi), IEEE802.11ac (Wi-Fi), HSPA+ (3G), WiMAX (4G),and Long Term Evolution (LTE4G). More recently, MIMO has been applied to power-line communication for 3-wire installations as part of ITU G.hn standard and HomePlugAV2 specification.

At one time, in wireless the term "MIMO" referred to the use of multiple antennas at the transmitter and the receiver. In modern usage, "MIMO" specifically refers to a practical technique for sending and receiving more than one data signal simultaneously over the same radio channel by exploiting multipath propagation. MIMO is fundamentally different from smart antenna techniques developed to enhance the performance of a single data signal, such as beam forming and diversity.

MIMO is often traced back to 1970s research papers concerning multi-channel digital transmission systems and interference (crosstalk) between wire pairs in a cable bundle: AR Kaye and DA George (1970), Branderburg and Wyner (1974), and W. van Etten (1975, 1976). Although these are not examples of exploiting multipath propagation to send multiple information streams, some of the mathematical techniques for dealing with mutual interference proved useful to MIMO development. In the mid-1980s Jack Salz at Bell Laboratories took this research a step further, investigating multi-user systems operating over "mutually cross-coupled linear networks with additive noise sources" such as time-division multiplexing and dually-polarized radio systems.

Methods were developed to improve the performance of cellular radio networks and enable more aggressive frequency reuse in the early 1990s. Space-division multiple access (SDMA) uses directional or smart antennas to communicate on the same frequency with users in different locations within range of the same base station. An SDMA system was proposed by Richard Roy and Bjorn Ottersten, researchers at ArrayComm, in 1991. Their US patent (No. 5515378 issued in 1996) describes a method for increasing capacity using "an array of receiving antennas at the base station" with a "plurality of remote users."

ArogyaswamiPaulraj and Thomas Kailath proposed an SDMA-based inverse multiplexing technique in 1993. Their US patent (No. 5,345,599 issued in 1994) described a method of broadcasting at high data rates by splitting a high-rate signal "into several low-rate signals" to be transmitted from "spatially separated transmitters" and recovered by the receive antenna array based on differences in "directions-of-arrival." Paulraj was awarded the prestigious Marconi Prize in 2014 for "his pioneering contributions to developing the theory and applications of MIMO antennas. His idea for using multiple antennas at both the transmitting and receiving stations – which is at the heart of the current high speedWiFi and 4G mobile systems – has revolutionized high speed wireless."

In an April 1996 paper and subsequent patent, Greg Raleigh proposed that natural multipath propagation can be exploited to transmit multiple, independent information streams using co-located antennas and multi-dimensional signal processing. The paper also identified practical solutions for modulation (MIMO-OFDM), coding, synchronization, and channel estimation. Later that year (September 1996) Gerard J. Foschini submitted a paper that also suggested it is possible to multiply the capacity of a wireless link using what the author described as "layered space-time architecture."

Greg Raleigh, V. K. Jones, and Michael Pollack founded Clarity Wireless in 1996, and built and field-tested a prototype MIMO system. Cisco Systems acquired Clarity Wireless in 1998. Bell Labs built a laboratory prototype demonstrating its V-BLAST (Vertical-Bell Laboratories Layered Space-Time) technology in 1998. ArogyaswamiPaulraj founded Iospan Wireless in late 1998 to develop MIMO-OFDM products. Iospan was acquired by Intel in 2003. V-BLAST was never commercialized, and neither Clarity Wireless nor Iospan Wireless shipped MIMO-OFDM products before being acquired.

CHAPTER 2

LITERATURE SURVEY

In several applications of wireless communication technology (e.g., vehicular networks, smart grid, and shallow sea underwater networks), the transmission of data signals will be severely deteriorated by the impulsive noise (IN). The sources of impulsive noise are diverse, such as ignition noise in automobiles, switches for electrical equipments, various maritime operations, and so on. Compared to additive white Gaussian noise (AWGN), the impulsive noise arises randomly with short duration and high power impulses.

Orthogonal frequency division multiplexing (OFDM) technology has been widely adopted in most modern wireless communication standards. In conventional OFDM receivers, the time-domain received signal is convertedinto the frequency domain through a discrete Fourier transform (DFT), after which each subcarrier is demodulated independently. Such tone-by-tone demodulation achieves optimal maximum likelihood detection in AWGN and perfect channel state information. When the impulsive noise is present, however, the corresponding frequency-domain noise samples will be highly dependent, and tone-by-tone demodulation is no longer feasible since the complexity of performing joint-detection at the receiver increases exponentially with the number of subcarriers.

Recently, there has been growing interest in developing compressed sensing (CS) based impulsive noise mitigation methods that exploit the time-domain sparsity of impulsive noise. These methods all make use of the information of null tones (i.e., tones that do not carry data or pilots) of the received OFDM symbol to estimate the IN sample and then subtract it from the received signal. Furthermore, some of them have been extended for detecting bursty (i.e., block sparse) impulsive noise by using structured compressed sensing theory. Although these methods show obvious advantages over those based on nonlinear preprocessor, the common drawback of these algorithms is that their performances are mostly limited by the number of null tones. It is worth pointing out that these approaches also assume that the channel state information is already estimated perfectly before the impulsive noise removal and do not consider the severe impact of impulsive noise on the channel estimation.

The performance of the IN estimator can be improved by increasing the number of null tones. However having more null tones means reduced throughput. When the number of null tones is limited, it is desirable to exploit information available in all tones to improve the estimation performance of the impulsive noise. The difficulty for exploiting all tones, however, is how to simultaneously estimate the channel and impulsive noise. An approach for jointly estimating channel and IN is proposed in, but it requires that there is no overlap between the support of impulsive noise and channel impulse response. In, an iterative channel estimation and impulsive noise mitigation algorithm is proposed on the assumption that the length of channel impulse response is known in advance and that the channel is static for several OFDM symbols. In, generalized approximate message passing (GAMP) has been used to jointly estimate the channel taps, the impulse noise samples, symbols, and the unknown bits. This method requires the acquisition of a priori information of the channel and impulsive noise and does not offer rigorous convergence although it is lower in computational complexity. By assuming that the impulsive noise parameter distributions are known at the receiver, a joint channel estimation and data decoding algorithm is developed. By exploiting the sparsity of both them, the orthogonal matching pursuit(OMP) is adopted for joint channel and impulsive noise estimation in underwater acoustic OFDM systems. This algorithm needs to collectthe number and position of IN samples by applying a blanking operation.

In this paper, we propose two novel algorithms based on Sparse Bayesian Learning (SBL) framework to jointly estimate both the channel impulse response and impulsive noise by exploiting the sparsity of both them. Our algorithms can also be categorized as an extension of the method proposed.

CHAPTER 3

CONCLUSION ON REVIEWS

3.1 Introduction to OFDM

OFDM has several significant benefits over single carrier QAM modulation which has led to its adoption in many of the modern wireless standards, including ADSL, European Digital Video Broadcast, IEEE 802.11a/g/n (WiFi), WiMax, and 3GPP Long Term Evolution (LTE). First of all, OFDM enables easy equalization, solving many of the equalization complexity issues we encountered for single carrier QAM by working instead in the frequency domain. Second of all, the ability to control the size of the constellation on each subcarrier in OFDM also allows it to mimic the "waterfilling" construction which maximizes the amount of information which can be reliably transmitted over the channel: subcarriers (frequencies) with better channel gains can utilize higher order modulations and coding rates. A final significant benefit that we will encounter later in the course is that OFDM allows for an especially easy to implement and understand multiple access strategy OFDMA, in which different subcarriers (frequencies) are allocated to different users at different times. To understand how OFDM modems work, we must first revisit several properties of the discrete Fourier Transform, and its efficient implementation the Fast Fourier Transform.

The circular convolution between two length N signals x1[n] and x2[n], n ∈ {0, 1, . . . , N − 1}, is defined as x1[n] * x2[n] = N X−1 =0 x1*x2[((n −1))N] where ((n))N = N − n n< 0 n n ≥ 0 Circularly convolving two length N signals x1[n] and x2[n] is the same thing as multiplying their discrete Fourier transforms x1[n]* x2[n] DF T ←→ X1[k]X2[k] .The underlying principle of OFDM is based on this property of the DFT.

The architecture of an OFDM transmitter and receiver pair is depicted in Fig. 1. First, during the th block, N QAM symbols c[k], k ∈ {0, . . . , N − 1} are mapped to "sub-carriers", one per subcarrier, which are the inputs to a size N IFFT, which computes the DFT shown below efficiently q[n] = N X−1 k=0 c[k]e j 2π N kn, n ∈ {0, . . . , N}

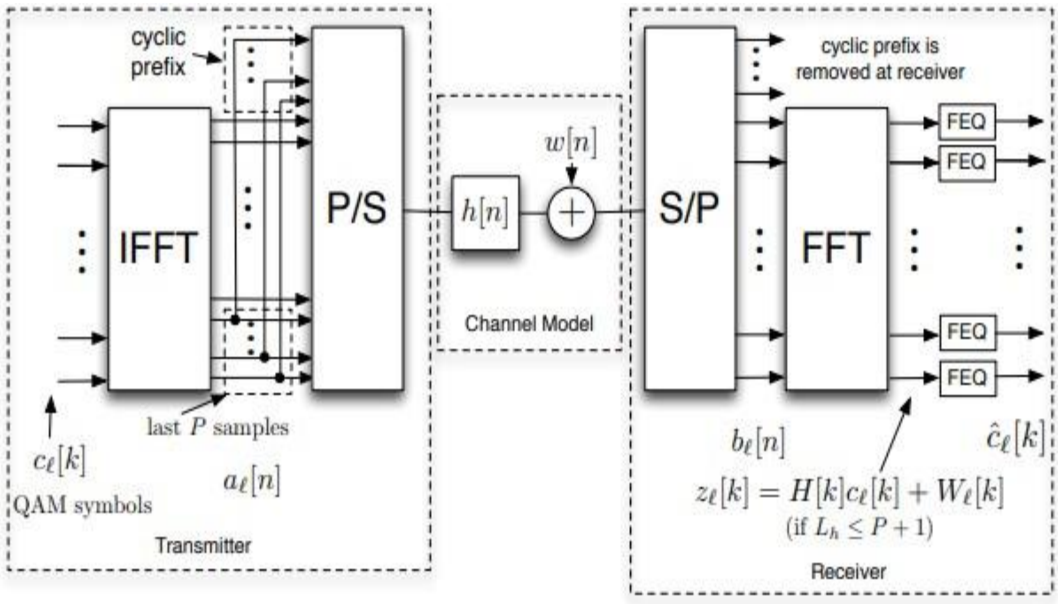
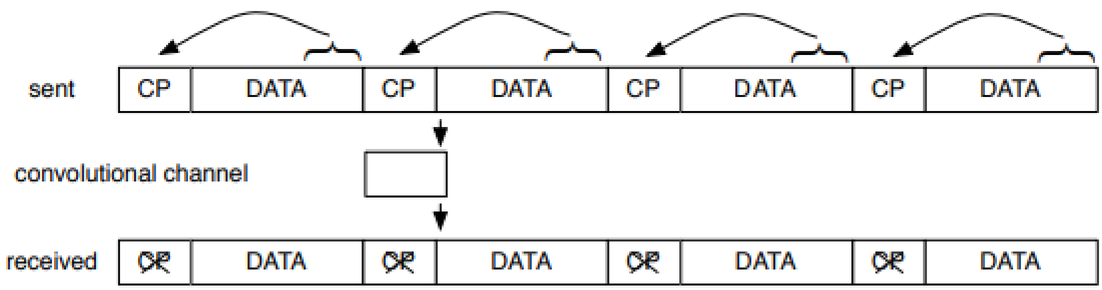


Fig.3.1.1 An OFDM transmitter,channel model and receiver.

These OFDM blocks are then serialized to form xc[n] = a b n N+P c [n mod (N + P)] which is then put through an digital to analog convertor to form the complex baseband signal to be modulated by an ordinary quadrature modulator. As discussed in the previous notes on equalization, on its way from the transmitter to the receiver, the signal is subject to a multipath channel, as well as potentially some additive interference, and at the receiver there is additive white Gaussian noise in the electronics. As discussed in the lecture on quadrature modulation, we can model the effects of this multipath channel by modeling the signal at the receiver, after demodulation and sampling as yc[n] = L Xh−1=0 h + w[n] where Δ indicates the delay before the channel is perceived as having any significant energy, and h[n] is the sampled impulse response of the channel, whose delay spread is Lh samples long, and w[n] is the additive white Gaussian noise. The receiver then proceeds to process this signal yc[n] in the exact opposite manner as at the transmitter. First the blocks are collected b[n] = yc[n + Δ + (N + P)], n ∈ {0, 1 . . . , N + P − 1} (8) then the cyclic prefix is removed r[n] = b[n + P], n ∈ {0, 1, . . . , N − 1} . The Fast Fourier transform is then taken, which calculates the DFT shown below efficiently z[k] = X N n=0 r[n]e −j 2π N kn.



The Purpose of the Cyclic Prefix: Removing Inter-Block & Inter-Carrier Interference. The cyclic prefix allows an OFDM transmitter receiver pair to accomplish to important objectives, provided that the length of the multipath channel is less than or equal to the length of the cyclic prefix plus one:

1) It eliminates interblock interference, and

2) It enables easy equalization, as for the remainder of the block it makes the linear convolution with the ISI channel into a circular convolution. Indeed, as is indicated in Fig. 2, if we consider the output of the channel at times $n \in (\{0, \ldots, P-1\} + k(N + P)$, this will contain a contribution from both the previous OFDM symbol and the next one. However, if the channel length $Lh \leq P + 1$ then these are the only samples containing the inter-block interference, and hence, once we drop these samples at the receiver (when we remove the cyclic prefix), we will have no interblock interference. Additionally, under the same criterion $Lh \leq P + 1$ the remaining samples in each block after dropping the cyclic prefix will be the circular convolution of the channel with the associated transmitted block. Hence, by the circular convolution property of the DFT, when the FFT is taken of these received blocks, they will be the product of the FFT of the channel H[k] and the original QAM symbols, plus the FFT of the additive noise, as depicted in Fig 1. $z[k] = H[k]c[k] + w[k]$. The estimated QAM symbols can then be reconstructed by processing the subcarriers k individually by multiplying them with a frequency domain equalizer F[k] $c[k] = F[k]z[k]$ (12) where the F[k] can selected as $H[k] * \ |H[k]|\ 2+\sigma 2$, which minimizes the average magnitude squared error between c[k] and c[k], where σ 2 is the variance w[k]. These FEQs F[k] can be estimated by placing training information, also called pilot signals or reference signals, agreed upon in advance between the transmitter and receiver, on various subcarriers k which are spread out in frequency. Because the channel is shorter than the cyclic prefix, and the cyclic prefix is typically far shorter than the FFT size, the channel coefficients H[k] vary slowly with frequency k, and hence the channel estimates from the pilot signals can be interpolated to

yield improved channel estimates for other carriers Considerations when Selecting a Cyclic Prefix Length and FFT Size There are two important considerations when selecting a cyclic prefix length and FFT size for an OFDM system in practice. We already noted that the cyclic prefix needs to be longer than one less than the length of the channel (its delay spread). As the cyclic prefix does not bear any extra information itself, it is desirable to select a very large FFT size so that the redundancy is introduces is low. However, the multipath channel must remain fixed over the entire duration of the OFDM symbol (i.e. the entire cyclic prefix and received data block), and this places an upper limit on the length of the FFT based on how quickly the channel is varying.

Bit-Loading & OFDMA Because the OFDM construction turns the wideband multipath channel into multiple parallel lower rate channels associated with evenly spaced frequencies within the band, it enables several intuitive other technologies. As mentioned in the introduction, those carriers k with higher signal to noise ratios |H[k]| 2 σ2 can be given higher order QAM symbols and higher code rates. This process is called bit-loading and adaptive modulation and coding in different contexts. Additionally, using OFDM modems also enable an especially intuitive form of multiple access. If the transmitter wishes to send individual messages to multiple receivers (as in a cellular system) listening to the same wideband channel, one can place information to be transmitted to different users on different subcarriers. Furthermore, since the multipath profiles H[k] of different receivers listening to the same transmitter will be different, if these multipath profiles are reliably fed back to the transmitter, it can also select those subcarriers for different users which they are receiving with high signal to noise ratios. On the other hand, if the multipath profiles are not fed back, or are varying too quickly, the transmitter can diversify which frequencies are used for different users by spreading the carriers/blocks of carriers given to each particular user out in frequency.

Finally, we note that you are likely to encounter at least two other variants of OFDM: discrete multitone (DMT) and SC-FDMA, both of which differ from pure OFDM solely by what is modulated on each of the subcarrier k (bins of the IFFT at the transmitter). Discrete multitone, which is used in ADSL systems, simply selects the blocks of QAM symbols to be conjugate symmetric, so that their IFFT is realvalued, and hence only a single analog to digital convertor is required at the receiver. A SC-FDMA transmitter selects a contiguous sub-block of subcarriers, and places the result of the FFT of a sequence of QAM symbols of these subcarriers. This “spreading” of the inputs to the subcarriers with a FFT enables the peak to average power ratio of the output of the IFFT to be reduced, which allows the battery operated power amplifiers in the analog RF front ends of the transmitters in mobiles to operate in more efficient regimes. SC-FDMA is used in the uplink of LTE (from mobile phones to the base stations), while OFDMA is used in the downlink (from base stations to mobile phones). Note that the overall effect of this operation is the similar to pulse-shaping a single carrier QAM signal, since the zeros on the other subcarriers effectively determine a pulse shape, and which subcarriers are selected determine an effective overall carrier frequency. This is the reason this type of modulation is called "single carrier FDMA" in the LTE standard.

Orthogonal frequency-division multiplexing (OFDM) is a method of digital signal modulationin which a single data stream is split across several separate narrowband channels at different frequencies to reduce interference and crosstalk. The original data stream bits -- that in a conventional single-channel modulation scheme would be sent serially (one after the other) -- are transmitted in parallel (several at once on separate channels) but at lower speed in each substream (a stream within another stream) relative to the original signal. This means symbols sent in the substreams are longer and spaced farther apart. In the original stream, each bit might be represented by a 1-nanosecond (ns) segment of the signal, with 0.25 ns spacing between bits. Splitting the signal across four component streams lets each bit be represented by 4 ns of the signal with 1ns spacing between. This reduces interference among symbols and makes it easier to receive each symbol accurately while maintaining the same throughput (in this example, 4 bits every 5 ns).

OFDM technology was first conceived of in the 1960s and 1970s during research into minimizing interference among channels near each other in frequency and to achieve clean data transmission in situations prone to interference and signal corruption when more conventional modulation schemes are used.

OFDM is used in Wi-Fi, DSL internet access, 4Gwireless communications, and digital televisionand radio broadcast services

Orthogonal Frequency Division Multiplexing (OFDM) is special form of multi-carrier modulation, patented in 1970. It is particularly suited for transmission over a dispersive channel. (See further discussion of MCM over wireless channel.)

In a multipath channel, most conventional modulation techniques are sensitive to intersymbol interference unless the channel symbol rate is small compared to the delay spread of the channel. OFDM is significantly less sensitive to intersymbol interference, because a special set of signals is used to build the composite transmitted signal. The basic idea is that each bit occupies a frequency-time window which ensures little or no distortion of the waveform. In practice, it means that bits are transmitted in parallel over a number of frequency-nonselective channels.

3.2 Coded OFDM

Multi-Carrier Modulation on its own is not the solution to the problems of communication over unreliable multipath channels. The channel time dispersion will excessively attenuate some subcarriers such that the throughput on these sub-channels would be unacceptable small. Only if the joint signal of many subcarriers is processed appropriately, the diversity advantages of MCM can be exploited. The need for coding across subcarriers was addressed by Sari et al. warning against overly enthusiastic pursuit of MCM. The advantages of frequency-domain implementations of equalizers (using an FFT) should not be mistaken for an "inherent" diversity gain of OFDM, which may not exist.

In an OFDM transmitter, blocks of k incoming bits are encoded into n channel bits. Before transmission, an n-point Inverse-FFT operation is performed. When the signals at the I-FFT output are transmitted sequentially, each of the n channel bits appears at a different (subcarrier) frequency. Such coding across subcarriers is necessary. If one subcarrier experiences deep fading, this leads to erasure of the bit on this subcarrier.

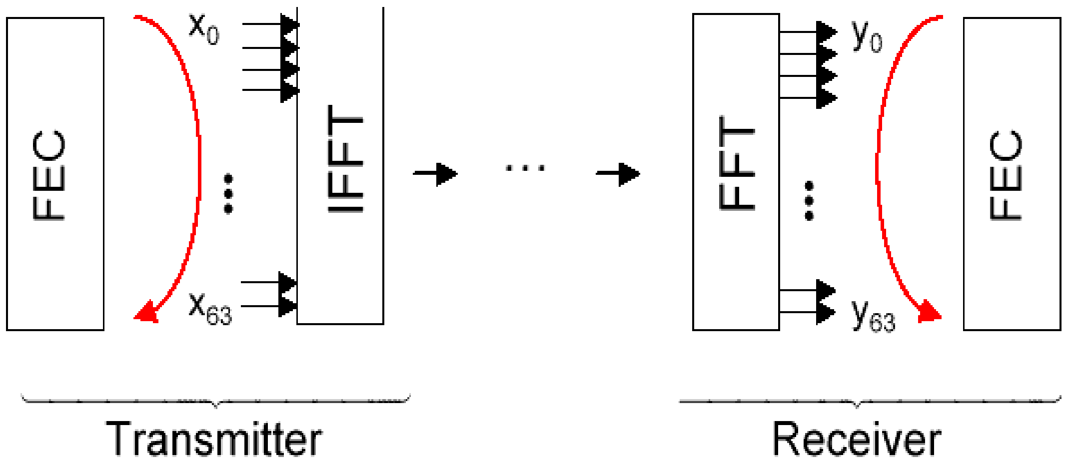


Fig 3.2.1 OFDM Transmitter and receiver

But of course coding across subcarriers is not the only mechanism that can be invoked to combat dispersion or to exploit diversity. Other possibilities are

- Interleaving in frequency or time domain with coding in the other domain,
- The use of different signal constellations at different frequencies, i.e., adapting the subcarrier bit rate to the channel state,
- Signal spreading over various subcarriers, e.g., according to a linear matrix operation, as is proposed in Orthogonal Multi-Carrier Code Division Multiplexing.

If in a point-to-point MCM link, the receiver and the transmitter can cooperate by adaptively distributing of their power budget over the individual subcarriers. For instance, the signal-to-noise ratios selected according to Gallager's water-pouring theorem can (under certain conditions) be proved to be optimum. Efficient loading of the various subcarriers can for instance significantly enhance the performance of MCM over twisted pair telephone subscriber loops with crosstalk from other nearby copper pairs.

Implementational Aspects

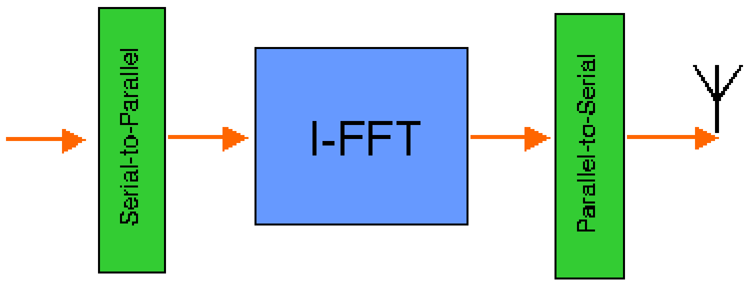


Figure 3.2.2 OFDM transmitter using an (inverse) Fast Fourier Transform (FFT).

- OFDM is not a constant-envelope modulation method. Therefore transmit power amplifiers must be highly linear.

- OFDM receiver performance is very sensitive to phase noise.
- Synchronization to an OFDM signal also requires frame synchronization, to support an FFT operation at the receiver.

Single Frequency Networks:

OFDM allows very efficient frequency reuse. Transmitters broadcasting the same program can use the same frequency in a Single Frequency Network.

Code Division Multiple Access:

OFDM can be combined with CDMA transmission, e.g. in Multi-Carrier CDMA.

3.3 Channel Modeling:

Many channel simulation models follow the narrowband model. Wideband channels are often simulated by extending the model assuming multiple time-delayed resolvable paths. This allows the simulation of the channel impulse response, including its stochastic behavior. To determine the performance of an multicarrier, OFDM or MC-CDMA system, another approach can be to model a set of fading subchannels. Considering a single subcarrier, the channel may be modelled as a narrowband fading channel, for instance with Rician or Rayleigh amplitude distributions. The collection of multiple subcarriers can be modelled as a set of mutually dependent fading channels. In such model, it is important to address correlation of the fading of various subchannels using the models of delay spread and coherence bandwidth. See a discussion of such model. Also: read about the discrete-frequency model for OFDM with Delay spread and Doppler.

OFDM, Orthogonal Frequency Division Multiplexing is a form of signal waveform or modulation that provides some significant advantages for data links.

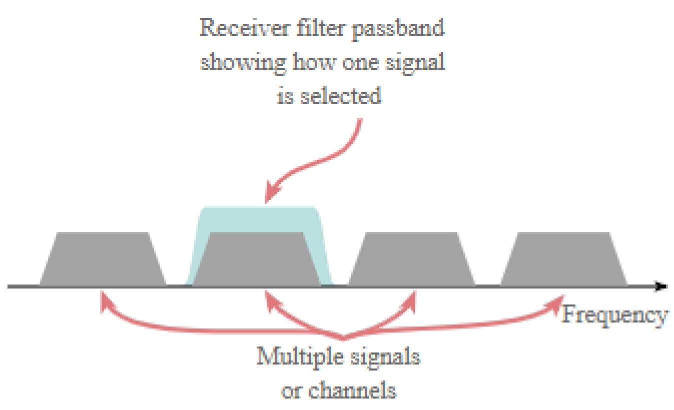
Accordingly, OFDM, Orthogonal Frequency Division Multiplexing is used for many of the latest wide bandwidth and high data rate wireless systems including Wi-Fi, cellular telecommunications and many more.

The fact that OFDM uses a large number of carriers, each carrying low bit rate data, means that it is very resilient to selective fading, interference, and multipath effects, as well providing a high degree of spectral efficiency.

Early systems using OFDM found the processing required for the signal format was relatively high, but with advances in technology, OFDM presents few problems in terms of the processing required.

3.4 What is OFDM?

OFDM is a form of multicarrier modulation. An OFDM signal consists of a number of closely spaced modulated carriers. When modulation of any form - voice, data, etc. is applied to a carrier, then sidebands spread out either side. It is necessary for a receiver to be able to receive the whole signal to be able to successfully demodulate the data. As a result when signals are transmitted close to one another they must be spaced so that the receiver can separate them using a filter and there must be a guard band between them. This is not the case with OFDM. Although the sidebands from each carrier overlap, they can still be received without the interference that might be expected because they are orthogonal to each another. This is achieved by having the carrier spacing equal to the reciprocal of the symbol period.



Traditional-slection if signals on different chanels

Figure 3.4.1 Traditional-selection if signals on different channels.

To see how OFDM works, it is necessary to look at the receiver. This acts as a bank of demodulators, translating each carrier down to DC. The resulting signal is integrated over the symbol period to regenerate the data from that carrier. The same demodulator also demodulates the other carriers. As the carrier spacing equal to the reciprocal of the symbol period means that they will have a whole number of cycles in the symbol period and their contribution will sum to zero - in other words there is no interference contribution.

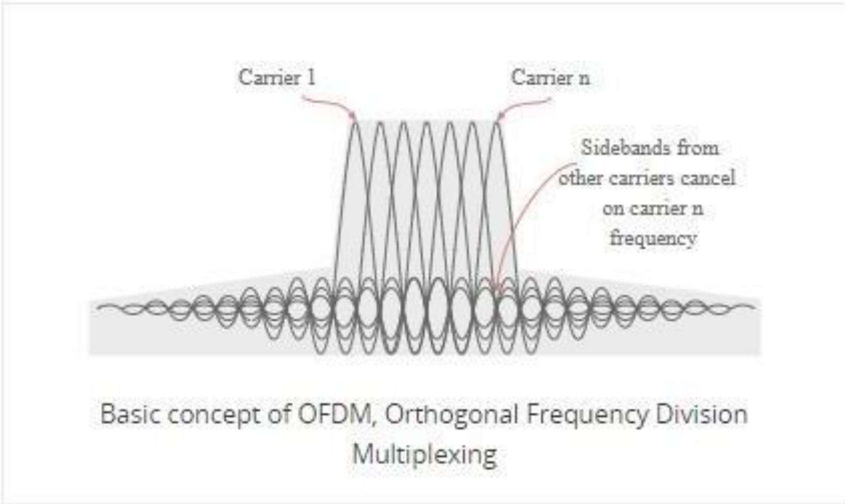


Figure 3.4.2 Basic concept of OFDM.

One requirement of the OFDM transmitting and receiving systems is that they must be linear. Any non-linearity will cause interference between the carriers as a result of inter-modulation distortion. This will introduce unwanted signals that would cause interference and impair the orthogonality of the transmission.

In terms of the equipment to be used the high peak to average ratio of multi-carrier systems such as OFDM requires the RF final amplifier on the output of the transmitter to be able to handle the peaks whilst the average power is much lower and this leads to inefficiency. In some systems the peaks are limited. Although this introduces distortion that results in a higher level of data errors, the system can rely on the error correction to remove them.

3.5 Data on OFDM

The data to be transmitted on an OFDM signal is spread across the carriers of the signal, each carrier taking part of the payload. This reduces the data rate taken by each carrier. The lower data rate has the advantage that interference from reflections is much less critical. This is achieved by adding a guard band time or guard interval into the system. This ensures that the data is only sampled when the signal is stable and no new delayed signals arrive that would alter the timing and phase of the signal.

3.6 Guard interval on OFDM signals

The distribution of the data across a large number of carriers in the OFDM signal has some further advantages. Nulls caused by multi-path effects or interference on a given frequency only affect a small number of the carriers, the remaining ones being received correctly. By using error-coding techniques, which does mean adding further data to the transmitted signal, it enables many or all of the corrupted data to be reconstructed within the receiver. This can be done because the error correction code is transmitted in a different part of the signal.

3.7 Key features of OFDM

The OFDM scheme differs from traditional FDM in the following interrelated ways:

- Multiple carriers (called subcarriers) carry the information stream
- The subcarriers are orthogonal to each other.

3.8 Applications of OFDM:

- Digital Video Broadcasting over the terrestrial network: Digital Terrestrial Television Broadcasting (DTTB). In the DTTB OFDM transmission standard, about 2,000 to 8,000 subcarriers are used.
- UMTS. The UMTS Forum is selecting an appropriate radio solution for the third generation mobile standard, as a successor to GSM. OFDM is one of the five competing proposals.
- Wireless LANs. OFDM is used in HIPERLAN Phase II, which supports 20 Mbit/s in propagation environments with delay spreads up to 1ms.

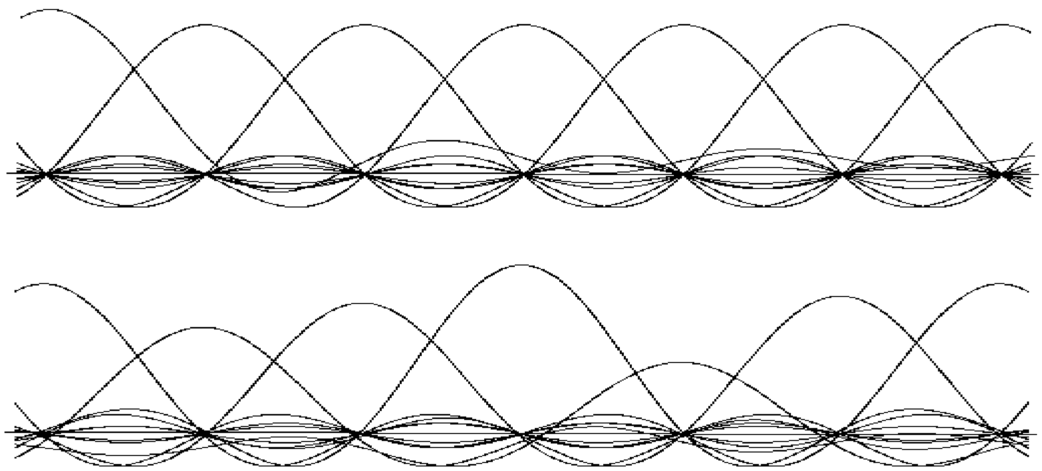


Figure 3.8.1 Signal spectrum of an OFDM signal, which consists of the spectra of many bits, in parallel. Rectangular pulses in time domain produce sinc-functions in frequency domain.

- Above: signal spectrum as transmitted.
- Below: as received in over a dispersive, time-invariant channel.

The effect of multipath scattering on OFDM differs from what happens to other forms of modulation. A qualitative description and mathematical description of OFDM is presented by Dusan Matic. Jean-Paul Linnartz reviews the effects of a Doppler spread and the associated rapid channel variations. Dusan Matic also studied the system design aspects of OFDM at mm-wavelengths.

3.9 OFDM advantages

OFDM has been used in many high data rate wireless systems because of the many advantages it provides.

- Immunity to selective fading: One of the main advantages of OFDM is that is more resistant to frequency selective fading than single carrier systems because it divides the overall channel into multiple narrowband signals that are affected individually as flat fading sub-channels.
- Resilience to interference: Interference appearing on a channel may be bandwidth limited and in this way will not affect all the sub-channels. This means that not all the data is lost.
- Spectrum efficiency: Using close-spaced overlapping sub-carriers, a significant OFDM advantage is that it makes efficient use of the available spectrum.
- Resilient to ISI: Another advantage of OFDM is that it is very resilient to inter-symbol and inter-frame interference. This results from the low data rate on each of the sub-channels.
- Resilient to narrow-band effects: Using adequate channel coding and interleaving it is possible to recover symbols lost due to the frequency selectivity of the channel and narrow band interference. Not all the data is lost.
- Simpler channel equalisation: One of the issues with CDMA systems was the complexity of the channel equalisation which had to be applied across the whole channel. An advantage of OFDM is that using multiple sub-channels, the channel equalization becomes much simpler.

3.10 OFDM disadvantages

Whilst OFDM has been widely used, there are still a few disadvantages to its use which need to be addressed when considering its use.

- High peak to average power ratio: An OFDM signal has a noise like amplitude variation and has a relatively high large dynamic range, or peak to average power ratio. This impacts the RF amplifier efficiency as the amplifiers need to be linear and accommodate the large amplitude variations and these factors mean the amplifier cannot operate with a high efficiency level.
- Sensitive to carrier offset and drift: Another disadvantage of OFDM is that is sensitive to carrier frequency offset and drift. Single carrier systems are less sensitive.

OFDM, orthogonal frequency division multiplexing has gained a significant presence in the wireless market place. The combination of high data capacity, high spectral efficiency, and its resilience to interference as a result of multi-path effects means that it is ideal for the high data applications that have become a major factor in today's communications scene.
Use of cyclic prefix is a key element of enabling the OFDM signal to operate reliably.

The cyclic prefix acts as a buffer region or guard interval to protect the OFDM signals from intersymbol interference. This can be an issue in some circumstances even with the much lower data rates that are transmitted in the multicarrier OFDM signal.

3.11 What is a cyclic prefix?

The basic concept behind the OFDM cyclic prefix is quite straightforward. The cyclic prefix performs two main functions.

- The cyclic prefix provides a guard interval to eliminate intersymbol interference from the previous symbol.

- It repeats the end of the symbol so the linear convolution of a frequency-selective multipath channel can be modeled as circular convolution, which in turn may transform to the frequency domain via a discrete Fourier transform. This approach accommodates simple frequency domain processing, such as channel estimation and equalization.

The cyclic prefix is created so that each OFDM symbol is preceded by a copy of the end part of that same symbol.Different OFDM cyclic prefix lengths are available in various systems. For example within LTE a normal length and an extended length are available and after Release 8 a third extended length is also included, although not normally used.

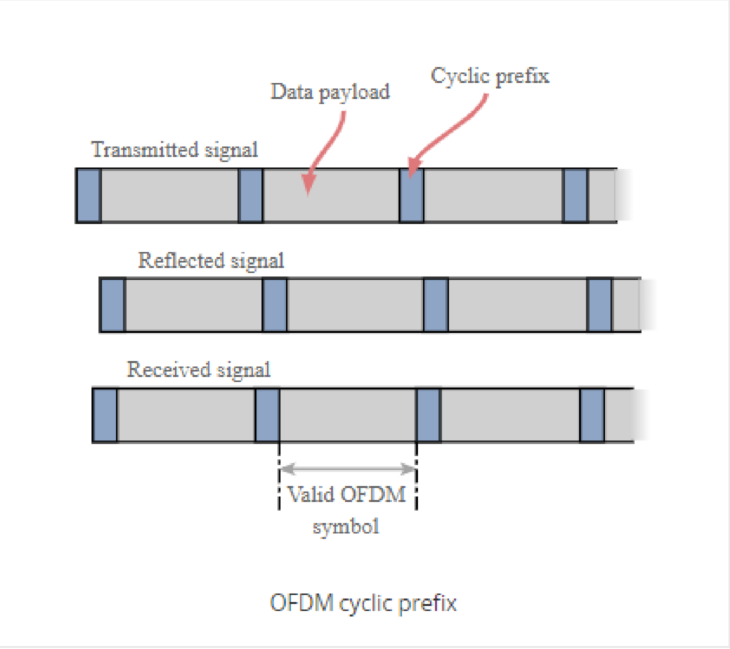


Figure 3.11.1 OFDM Cyclic prefix

3.12 Cyclic prefix advantages and disadvantages

There are several advantages and disadvantages attached to the use for the cyclic prefix within OFDM.

Advantages

- Provides robustness: The addition of the cyclic prefix adds robustness to the OFDM signal. The data that is retransmitted can be used if required.
- Reduces inter-symbol interference: The guard interval introduced by the cyclic prefix enables the effects of inter-symbol interference to be reduced.

Disadvantages

- Reduces data capacity: As the cyclic prefix re-transmits data that is already being transmitted, it takes up system capacity and reduces the overall data rate.

The use of a cyclic prefix is standard within OFDM and it enables the performance to be maintained even under conditions when levels of reflections and multipath propagation are high

One of the key requirements for optimum demodulation of OFDM signals is that there is accurate frequency and timing synchronisation.

Fortunately good OFDM timing and frequency synchronisation is relatively easy to obtain as accurate signals are easy to derive.

Poor timing and frequency synchronisation result in significant levels of degradation to the signal, and as a result this aspect of the signal chain is key to optimum performance.

3.13 OFDM synchronization basics:

OFDM offers many advantages in terms of resilience to fading, reflections and the like. OFDM also offers a high level of spectrum efficiency. However to reap the rewards, it is necessary that the OFDM system operates correctly, and to achieve this, it is necessary for the OFDM synchronization to be effective.

There are a number of areas in which the OFDM synchronisation is critical to the operation of the system:

- OFDM synchronization in terms of frequency offset: It is necessary that the frequencies are accurately tracked to ensure that orthogonality is maintained.
- OFDM synchronisation in terms of clock accuracy: It is necessary that the sampling occurs at the correct time interval to ensure that the samples are synchronized and data errors are minimised.

In order to ensure that the OFDM system works to its optimum, it is necessary to ensure that there are schemes in place to ensure the OFDM synchronization is within the required limits.

3.14 Frequency offset OFDM synchronization:

It is particularly important that the demodulator in an OFDM receiver is able to synchronize accurately with the carriers within the OFDM signal. Offsets may arise for a number of reasons including any frequency errors between the transmitter and the receiver and also as a result of Doppler shifts if there is movement between the transmitter and receiver.

If the frequency synchronisation is impaired, then the orthogonality of the carriers is reduced within the demodulation process and error rates increase. Accordingly it is essential to maintain orthogonality to reduce errors and maintain the performance of the link.

First look at the way that sampling should occur. With the demodulator in synchronisation, all the contributions from the other carriers sum to zero as shown. On this way all the carriers are orthogonal and the error rate is at its minimum.

OFDM demodulation showing how the interference from other carriers is minimized at the centre of the carrier frequency.
If a situation is encountered where the OFDM synchronisation for the frequency aspects are poor, then the demodulator will centre its samples away from the peak of the signal, and also at a point where the contributions from the other signals do not sum to zero. This will lead to a degradation of the signal which could in turn lead to an increase in the number of bit errors.

OFDM demodulation where interference is high due to poor frequency synchronization.

Clock offset OFDM synchronization

It is also necessary to maintain OFDM synchronization in terms of the clock. Gain if the clock synchronisation is not accurate, sampling will be offset and again orthogonality will be reduced, and data errors will increase.

When looking at OFDM synchronization with regard to the clock offset, the carrier spacing used within the receiver for sampling the received signal will be based upon the internal clock rate. If this differs from that used within the transmitter, it will be found that even if the first carrier within the multiplex is correct, then there will be a growing discrepancy with each carrier away from the first one. Even small levels of discrepancy will cause the error rate to increase.

OFDM demodulation with clock synchronization misalignment: When using OFDM it is necessary to ensure that the synchronisation for both timing and frequency is accurate. By ensuring accurate synchronisation, it is possible to perform the optimum demodulation of the signal. Any misalignment causes the receiver to start to pick up the unwanted interference signals. Fortunatelyit is relatively easy to obtain accurate synchronisation signals as these are available from the network, and short term synchronisation can be generated internally.

In telecommunications, orthogonal frequency-division multiplexing (OFDM) is a method of encoding digital data on multiple carrierfrequencies. OFDM has developed into a popular scheme for wideband digital communication, used in applications such as digital television and audio broadcasting, DSL internet access, wireless networks, power line networks, and 4G mobile communications.

In coded orthogonal frequency-division multiplexing (COFDM), forward error correction (convolutional coding) and time/frequency interleaving are applied to the signal being transmitted. This is done to overcome errors in mobile communication channels affected by multipath propagation and Doppler effects. COFDM was introduced by Alard in 1986 [1] [2][3] for Digital Audio Broadcasting for Eureka Project 147. In practice, OFDM has become used in combination with such coding and interleaving, so that the terms COFDM and OFDM co-apply to common applications.[4]

OFDM is a frequency-division multiplexing (FDM) scheme used as a digital multi-carrier modulation method. OFDM was introduced by Chang of Bell Labs in 1966.[5][6][7] Numerous closely spaced orthogonal sub-carrier signals with overlapping spectra are emitted to carry data.[8] Demodulation is based on Fast Fourier Transform algorithms. OFDM was improved by Weinstein and Ebert in 1971 with the introduction of a guard interval, providing better orthogonality in transmission channels affected by multipath propagation.[9] Each sub-carrier (signal) is modulated with a conventional modulation scheme (such as quadrature amplitude modulation or phase-shift keying) at a low symbol rate. This maintains total data rates similar to conventional single-carrier modulation schemes in the same bandwidth.

The main advantage of OFDM over single-carrier schemes is its ability to cope with severe channel conditions (for example, attenuation of high frequencies in a long copper wire, narrowband interference and frequency-selective fading due to multipath) without complex equalization filters. Channel equalization is simplified because OFDM may be viewed as using many slowly modulated narrowband signals rather than one rapidly modulated wideband signal. The low symbol rate makes the use of a guard interval between symbols affordable, making it possible to eliminate intersymbol interference (ISI) and use echoes and time-spreading (in analog television visible as ghosting and blurring, respectively) to achieve a diversity gain, i.e. a signal-to-noise ratio improvement. This mechanism also facilitates the design of single frequency networks (SFNs) where several adjacent transmitters send the same signal simultaneously at the same frequency, as the signals from multiple distant transmitters may be re-combined constructively, sparing interference of a traditional single-carrier system.

3.15 Orthogonality

Conceptually, OFDM is a specialized frequency-division multiplexing (FDM) method, with the additional constraint that all subcarrier signals within a communication channel are orthogonal to one another.

In OFDM, the sub-carrier frequencies are chosen so that the sub-carriers are orthogonal to each other, meaning that cross-talk between the sub-channels is eliminated and inter-carrier guard bands are not required. This greatly simplifies the design of both the transmitter and the receiver; unlike conventional FDM, a separate filter for each sub-channel is not required.

The orthogonality requires that the sub-carrier spacing is Hertz, where T_U seconds is the useful symbol duration (the receiver-side window size), and k is a positive integer, typically equal to 1. This stipulates that each carrier frequency undergoes k more complete cycles per symbol period than the previous carrier. Therefore, with N sub-carriers, the total passband bandwidth will be B = N*Δf (Hz).

The orthogonality also allows high spectral efficiency, with a total symbol rate near the Nyquist rate for the equivalent baseband signal (i.e. near half the Nyquist rate for the double-side band physical passband signal). Almost the whole available frequency band can be used. OFDM generally has a nearly 'white' spectrum, giving it benign electromagnetic interference properties with respect to other co-channel users.

A simple example: A useful symbol duration T_U = 1 ms would require a sub-carrier spacing of (or an integer multiple of that) for orthogonality. N = 1,000 sub-carriers would result in a total passband bandwidth of NΔf = 1 MHz. For this symbol time, the required bandwidth in theory according to Nyquist is (half of the achieved bandwidth required by our scheme), where R is the bit rate and where N = 1,000 samples per symbol by FFT. If a guard interval is applied (see below), Nyquist bandwidth requirement would be even lower. The FFT would result in N = 1,000 samples per symbol. If no guard interval was applied, this would result in a base band complex valued signal with a sample rate of 1 MHz, which would require a baseband bandwidth of 0.5 MHz according to Nyquist. However, the passband RF signal is produced by multiplying the baseband signal with a carrier waveform (i.e., double-sideband quadrature amplitude-modulation) resulting in a passband bandwidth of 1 MHz. A single-side band (SSB) or vestigial sideband (VSB) modulation scheme would achieve almost half that bandwidth for the same symbol rate (i.e., twice as high spectral efficiency for the same symbol alphabet length). It is however more sensitive to multipath interference.

OFDM requires very accurate frequency synchronization between the receiver and the transmitter; with frequency deviation the sub-carriers will no longer be orthogonal, causing inter-carrier interference (ICI) (i.e., cross-talk between the sub-carriers). Frequency offsets are typically caused by mismatched transmitter and receiver oscillators, or by Doppler shift due to movement. While Doppler shift alone may be compensated for by the receiver, the situation is worsened when combined with multipath, as reflections will appear at various frequency offsets, which is much harder to correct. This effect typically worsens as speed increases, and is an important factor limiting the use of OFDM in high-speed vehicles. In order to mitigate ICI in such scenarios, one can shape each sub-carrier in order to minimize the interference resulting in a non-orthogonal subcarriers overlapping.[12] For example, a low-complexity scheme referred to as WCP-OFDM (Weighted Cyclic Prefix Orthogonal Frequency-Division Multiplexing) consists of using short filters at the transmitter output in order to perform a potentially non-rectangular pulse shaping and a near perfect reconstruction using a single-tap per subcarrier equalization. Other ICI suppression techniques usually increase drastically the receiver complexity.

3.16 Implementation using the FFT algorithm:

The orthogonality allows for efficient modulator and demodulator implementation using the FFT algorithm on the receiver side, and inverse FFT on the sender side. Although the principles and some of the benefits have been known since the 1960s, OFDM is popular for wideband communications today by way of low-cost digital signal processing components that can efficiently calculate the FFT.

The time to compute the inverse-FFT or FFT transform has to take less than the time for each symbol, which for example for DVB-T (FFT 8k) means the computation has to be done in 896 μs or less.

For an 8192-point FFT this may be approximated to:

MIPS = Million instructions per second

The computational demand approximately scales linearly with FFT size so a double size FFT needs double the amount of time and vice versa.[15]:83 As a comparison an Intel Pentium III CPU at 1.266 GHz is able to calculate a 8192 point FFT in 576 μs using FFTW. Intel Pentium M at 1.6 GHz does it in 387 μs Intel Core Duo at 3.0 GHz does it in 96.8 μs.

Guard interval for elimination ofintersymbol interference

One key principle of OFDM is that since low symbol rate modulation schemes (i.e., where the symbols are relatively long compared to the channel time characteristics) suffer less from intersymbol interference caused by multipath propagation, it is advantageous to transmit a number of low-rate streams in parallel instead of a single high-rate stream. Since the duration of each symbol is long, it is feasible to insert a guard interval between the OFDM symbols, thus eliminating the intersymbol interference.

The guard interval also eliminates the need for a pulse-shaping filter, and it reduces the sensitivity to time synchronization problems.

The cyclic prefix, which is transmitted during the guard interval, consists of the end of the OFDM symbol copied into the guard interval, and the guard interval is transmitted followed by the OFDM symbol. The reason that the guard interval consists of a copy of the end of the OFDM symbol is so that the receiver will integrate over an integer number of sinusoid cycles for each of the multipaths when it performs OFDM demodulation with the FFT.

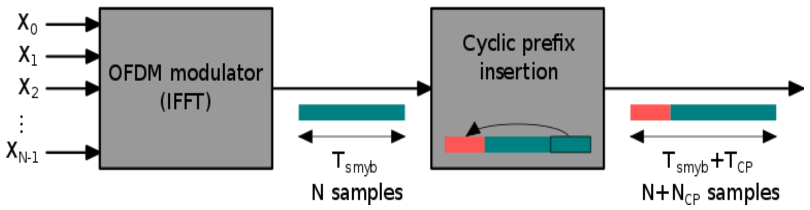


Figure 3.14.1 OFDM modulator and cyclic prefix insertion

In some standards such as Ultra wideband, in the interest of transmitted power, cyclic prefix is skipped and nothing is sent during the guard interval. The receiver will then have to mimic the cyclic prefix functionality by copying the end part of the OFDM symbol and adding it to the beginning portion.

3.17 Transmitter

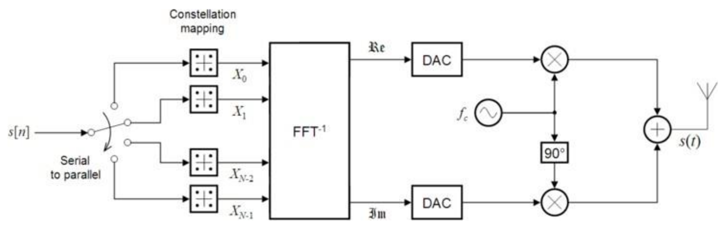


Figure 3.17.1 Transmitter

An OFDM carrier signal is the sum of a number of orthogonal sub-carriers, with baseband data on each sub-carrier being independently modulated commonly using some type of quadrature amplitude modulation (QAM) or phase-shift keying (PSK). This composite baseband signal is typically used to modulate a main RF carrier.

OFDM is used by many power line devices to extend digital connections through power wiring. Adaptive modulation is particularly important with such a noisy channel as electrical wiring. Some medium speed smart metering modems, "Prime" and "G3" use OFDM at modest frequencies (30–100 kHz) with modest numbers of channels (several hundred) in order to overcome the intersymbol interference in the power line environment. The IEEE 1901 standards include two incompatible physical layers that both use OFDM. The ITU-T G.hn standard, which provides high-speed local area networking over existing home wiring (power lines, phone lines and coaxial cables) is based on a PHY layer that specifies OFDM with adaptive modulation and a Low-Density Parity-Check (LDPC) FEC code.

An inverse FFT is computed on each set of symbols, giving a set of complex time-domain samples. These samples are then quadrature-mixed to passband in the standard way. The real and imaginary components are first converted to the analogue domain using digital-to-analogue converters (DACs); the analogue signals are then used to modulate cosine and sine waves at the carrier frequency, respectively. These signals are then

summed to give the transmission signal. .

3.18 Receiver

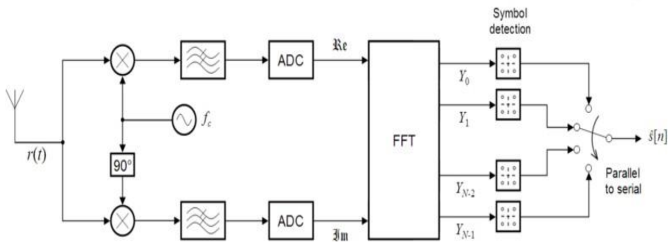


Figure 3.18.1 Receiver

The receiver picks up the signal, which is then quadrature-mixed down to baseband using cosine and sine waves at the carrier frequency. This also creates signals centered on, so low-pass filters are used to reject these. The baseband signals are then sampled and digitised using analog-to-digital converters (ADCs), and a forward FFT is used to convert back to the frequency domain.

This returns parallel streams, each of which is converted to a binary stream using an appropriate symbol detector. These streams are then re-combined into a serial stream, , which is an estimate of the original binary stream at the transmitter.

3.19 Wireless local area networks (LAN) and metropolitan area networks (MAN)

OFDM is extensively used in wireless LAN and MAN applications, including IEEE 802.11a/g/n and WiMAX.

IEEE 802.11a/g/n, operating in the 2.4 and 5 GHz bands, specifies per-stream airside data rates ranging from 6 to 54 Mbit/s. If both devices can use "HT mode" (added with 802.11n), the top 20 MHz per-stream rate is increased to 72.2 Mbit/s, with the option of data rates between 13.5 and 150 Mbit/s using a 40 MHz channel. Four different modulation schemes are used: BPSK, QPSK, 16-QAM, and 64-QAM, along with a set of error correcting rates (1/2–5/6). The multitude of choices allows the system to adapt the optimum data rate for the current signal conditions.

Wireless personal area networks (PAN)

OFDM is also now being used in the WiMedia/Ecma-368 standard for high-speed wireless personal area networks in the 3.1–10.6 GHz ultrawideband spectrum (see MultiBand-OFDM).

Terrestrial digital radio and television broadcasting:

Much of Europe and Asia has adopted OFDM for terrestrial broadcasting of digital television (DVB-T, DVB-H and T-DMB) and radio (EUREKA 147 DAB, Digital Radio Mondiale, HD Radio and T-DMB).

By Directive of the European Commission, all television services transmitted to viewers in the European Community must use a transmission system that has been standardized by a recognized European standardization body, and such a standard has been developed and codified by the DVB Project, Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television. Customarily referred to as DVB-T, the standard calls for the exclusive use of COFDM for modulation. DVB-T is now widely used in Europe and elsewhere for terrestrial digital TV.

SDARS

The ground segments of the Digital Audio Radio Service (SDARS) systems used by XM Satellite Radio and Sirius Satellite Radio are transmitted using Coded OFDM (COFDM).The word "coded" comes from the use of forward error correction (FEC).

CHAPTER 4

SOFTWARE USED

4.1. Introduction to MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

4.2 THE MATLAB SYSTEM:

The MATLAB system consists of five main parts

- Development Environment:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- The MATLAB Mathematical Function Library:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- The MATLAB Language:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both ―programming in the small‖ to rapidly create quick and dirty throw-away programs, and ―programming in the large‖ to create large and complex application programs.

Graphics:

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for twodimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

The MATLAB Application Program Interface (API):

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using IMAGE PROCESSING toolbox.

4.3. GRAPHICAL USER INTERFACE (GUI):

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based Mfunction is executed.
- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type

Guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted,

GUIDE opens a new (i.e., blank) window.

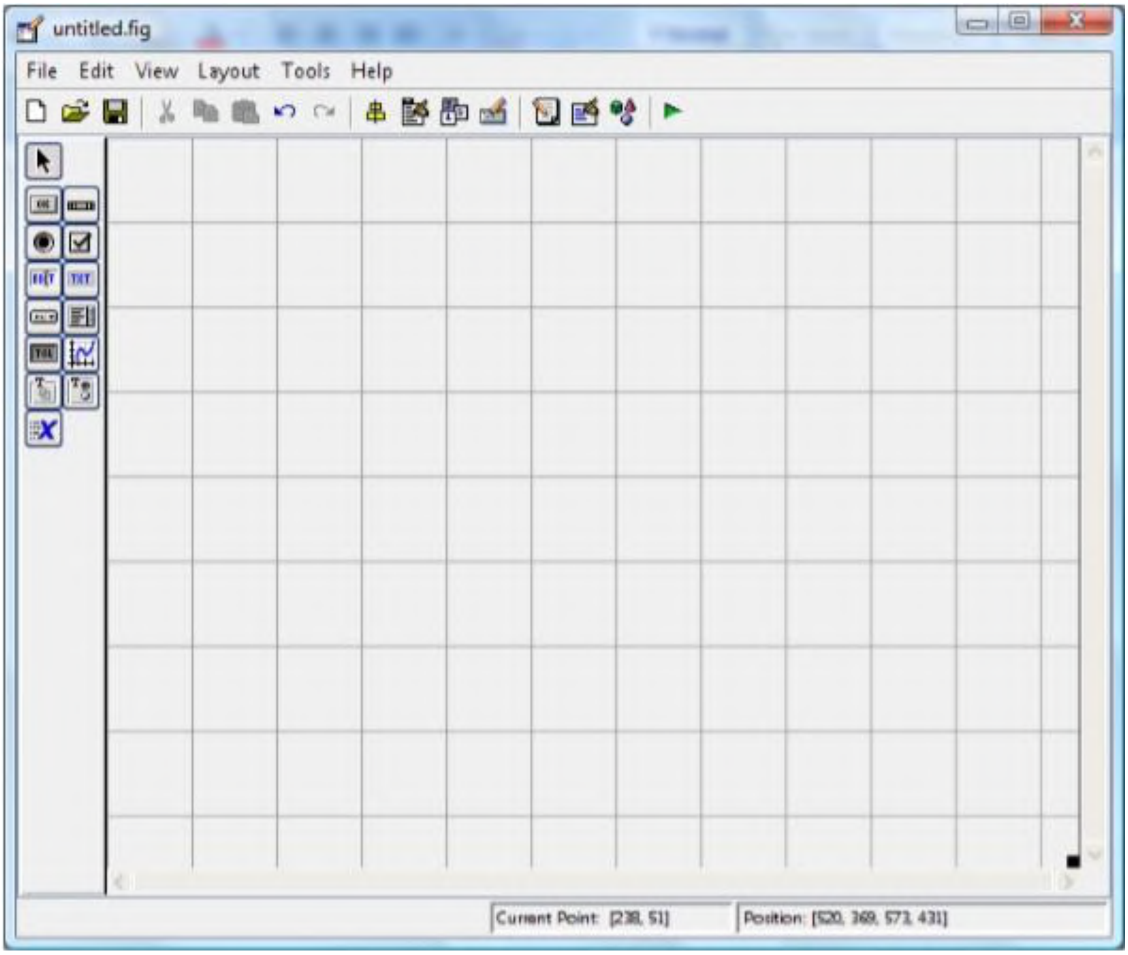


Figure 4.3.1:matlab procedure

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

4.4 GETTING STARTED

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

4.4.1 INTRODUCTION - describes the components of the MATLAB system.

4.4.2 DEVELOPMENT ENVIRONMENT - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

4.4.3 MANIPULATING MATRICES - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

4.4.4 GRAPHICS - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images. 2.4.5 Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

4.5 DEVELOPMENT ENVIRONMENT

4.5.1INTRODUCTION

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

4.5.2 STARTING MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.On a UNIX platform, to start MATLAB, type matlab

At the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

4.5.3 QUITTING MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

4.5.4 MATLAB DESKTOP

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

4.5.5 DESKTOP TOOLS

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacsmagik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type helpbrowser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find Mfiles and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directorythat is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and who’s.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to runMATLAB functions. The Editor/Debugger provides a graphical user interface for basic textediting, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB .

CHAPTER 5

PROJECT IMPLEMENTATION

5.1 Introduction to OFDM System

We consider an OFDM system with N subcarriers where M subcarriers are used for carrying pilot and $N - M$ subcarriers are used for sending data. In some case, we also considerthe system which has null subcarriers and then these subcarriers can also be considered as pilot subcarriers with setting pilot symbols as zero. At the transmitter, data symbols which are mapped from the source bits and known pilot symbols are joined as the frequency-domain OFDM symbol $x = (x_0, x_1, \ldots, x_{N-1})^T$. The OFDM modulator, as an inverse discrete Fourier transformation (IDFT), converts the frequency-domain OFDM symbols into the time-domain OFDM signals to which the cyclic prefix (CP) is prepended before feeding into the wireless channel. Assuming that inter-symbol interference is avoided by simply discarding the cyclic prefix at the receiver, the received time-domain signal is expressed as

$$r = HF^* x + u$$

Where H is a $N \times N$ circulant matrix whose first column is formed by the zero-padded channel impulse response vector $h = (h_0, h_1, \ldots, h_{L-1})^T$, L is the length of channel impulse response. F is the unitary N-point discrete Fourier transform (DFT) matrix with (m, n) element $[F]_{m,n} = \sqrt{\frac{1}{N}} e^{-j2\pi mn/N}$ with $m, n \in \{0, 1, \ldots, N-1\}$ and F^* is its conjugate transpose. $u = i + g$ is the additive noise term which includesi, denoting impulsive noise component, and g, denoting AWGN component.

$$y = Fr = FHF^* x + Fi + Fg = 3x + Fi + n$$

Where 3,diag($\check{h}$) is a diagonal matrix with the channel frequency response $\check{h}$ as its diagonal elements. The channel frequency response $\check{h}$ is the DFT of the channel impulse response h, namely $\check{h} = \sqrt{N} F_L h$, where $F_L \in C^{N\times L}$ is the submatrix selected from the first L columns of matrix F. n is the frequency-domain background noise vector which is still AWGN since F is unitary matrix. Gaussian-Mixture (GM), Middleton Class A (MCA), and Symmetric alpha stable are the three IN models most widely adopted in the literature. In this work, we use GM model to simulate the impulsive noise. In particular, aKcomponent GM is accepted for performance analysis. The probability density function(pdf) of a time-domain impulsive noise sample u is expressed as

$$p(\boldsymbol{u}) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k f_k(u_i)$$

5.2 JOINT CHANNEL AND IN ESTIMATION ALGORITHM USING PILOT SUBCARRIERS:

Let P and D denote the index set corresponding to the pilot subcarriers and data subcarriers, respectively. The part pertaining to the pilot subcarriers in can be written as

$$\boldsymbol{y}_P = \boldsymbol{\Phi}_P \boldsymbol{w} + \boldsymbol{n}_P$$

Where yP is a M × 1 vector containing the elements of y sampled at pilot locations, 8P is a M×(L+N) submatrix of 8 consisting of the rows corresponding to the pilot locations, and nP is also a M × 1 vector consisting of the components of n sampled at pilot locations. Since 8P is a known flat matrix and w is a sparse vector, we may use CS theory to estimate w directly.

Many CS algorithms have been proposed in the literatures. In practice, different CS algorithms will have different requirements on the matrix and the sparsity for a reliable recovery. In this paper, we adopt SBL to solve the problem. Compared with the greedy CS algorithm, the SBL algorithms have shown super recovery performance when the recovering signal is less sparsity or the measurement matrix is higher coherence, which could well fit our problem. Another advantage of SBL is that it is capable of handling partially unknown dictionary matrices by virtue of the EM framework, which leads to the solution. In the following, we derive the algorithm for joint channel and IN estimation using pilots based on SBL framework (JCI).For observation model , SBL imposes firstly a parameterized Gaussian prior on the vector w, given by

$$p(\boldsymbol{w}; 0, \boldsymbol{\Gamma}) = \prod_{i=0}^{N+L-1} (\pi \gamma_i)^{-1} \exp\left(-\frac{|w_i|^2}{\gamma_i}\right)$$

The computational complexity of the JCI is dominated by the matrix multiplication and inversion operations in, which has a complexity of O(M(N + L) 2) per iteration. After given the initial values of data symbols, the JCIS using all tones to estimate channel and IN has a complexity of O(N(N +L) 2) per iteration. So the total complexity of JCIS is O((N + M)(N + L) 2). Compared with JCI, the complexity of JCIS is higher. This means JCIS improves the performance at

TABLE 1. Parameters for simulations.

Parameter	Notation	Value
Subcarrier Interval	Δf	15kHz
Sampling Frequency	f_s	3.84MHz
OFDM symbol Duration	T_s	$83.3\mu s$
Guard Interval Duration	T_g	$16.6\mu s$
Total Number of Subcarriers	N	256
Cyclic Prefix Length	N_G	64
Channel Length	L	64

Figure 6.2.1 Parameters for simulations.

The expense of increasing complexity, which will be verified by simulation results in the next Section.

CHAPTER 6

SIMULATION AND RESULTS

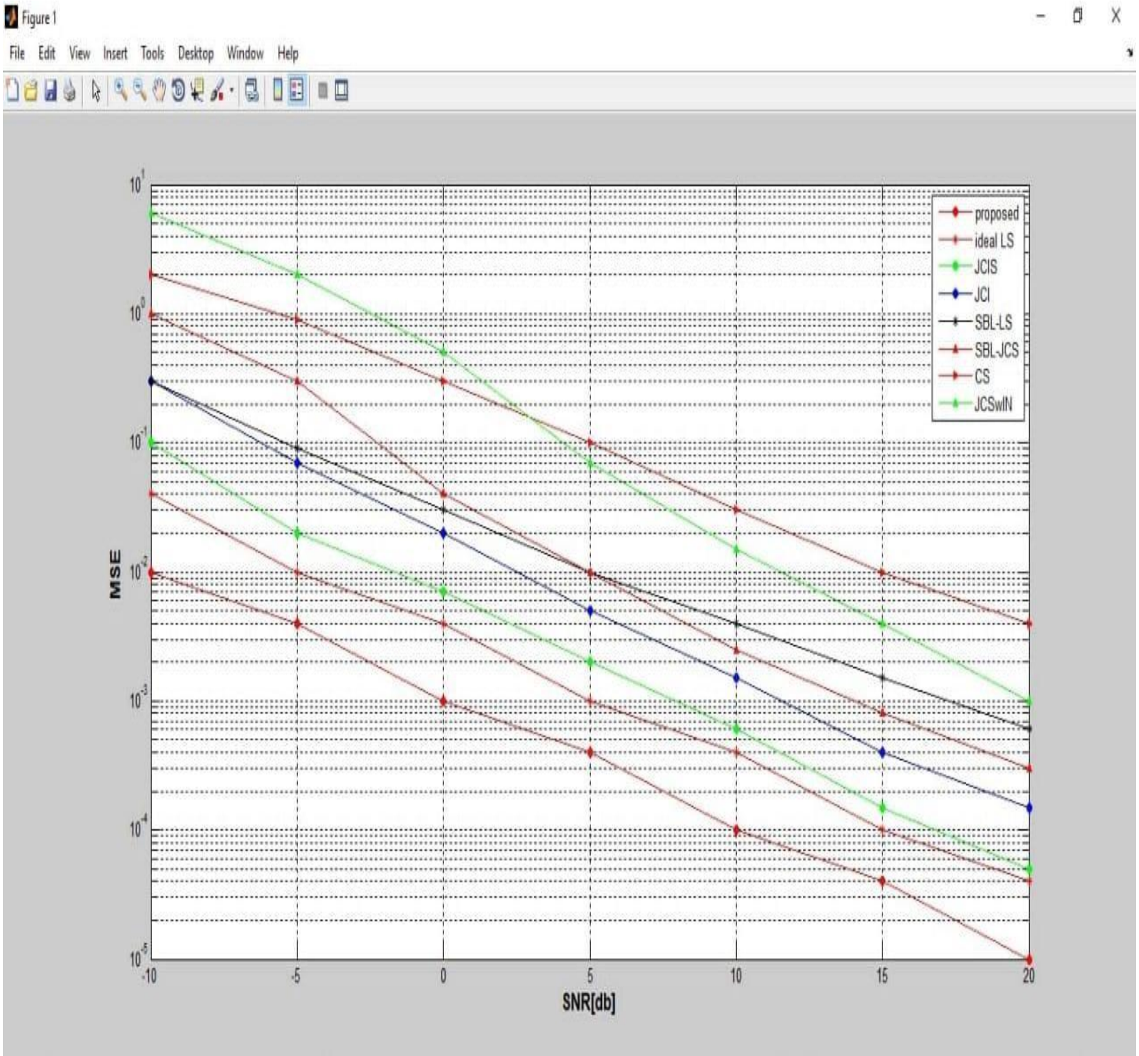


Figure 1:The MSE of channel estimation versus SNR. Number of pilot subcarriers is 44 and number of null subcarriers is 50.

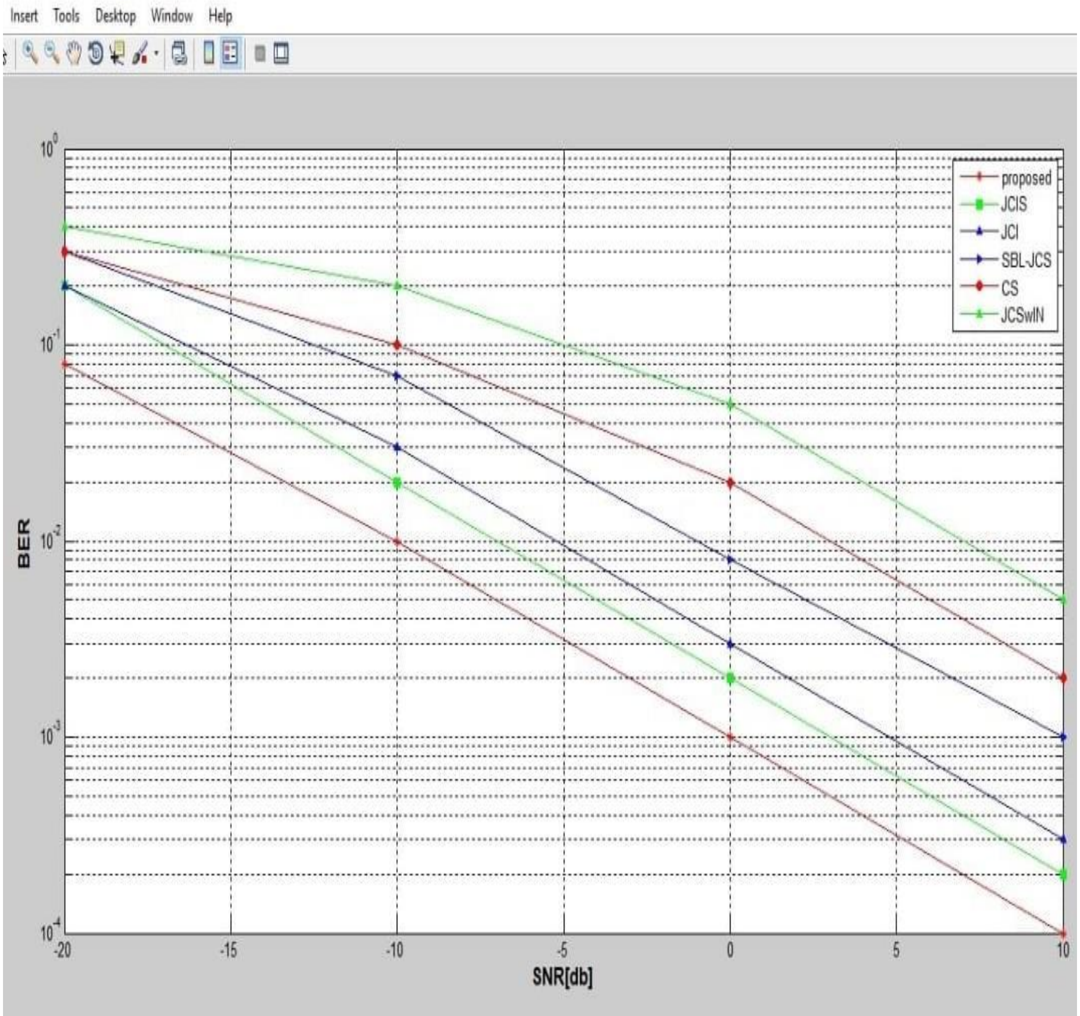


Figure 2: BER versus SNR in uncoded OFDM system. Number of pilot subcarriers 44 and number of null subcarriers 50.

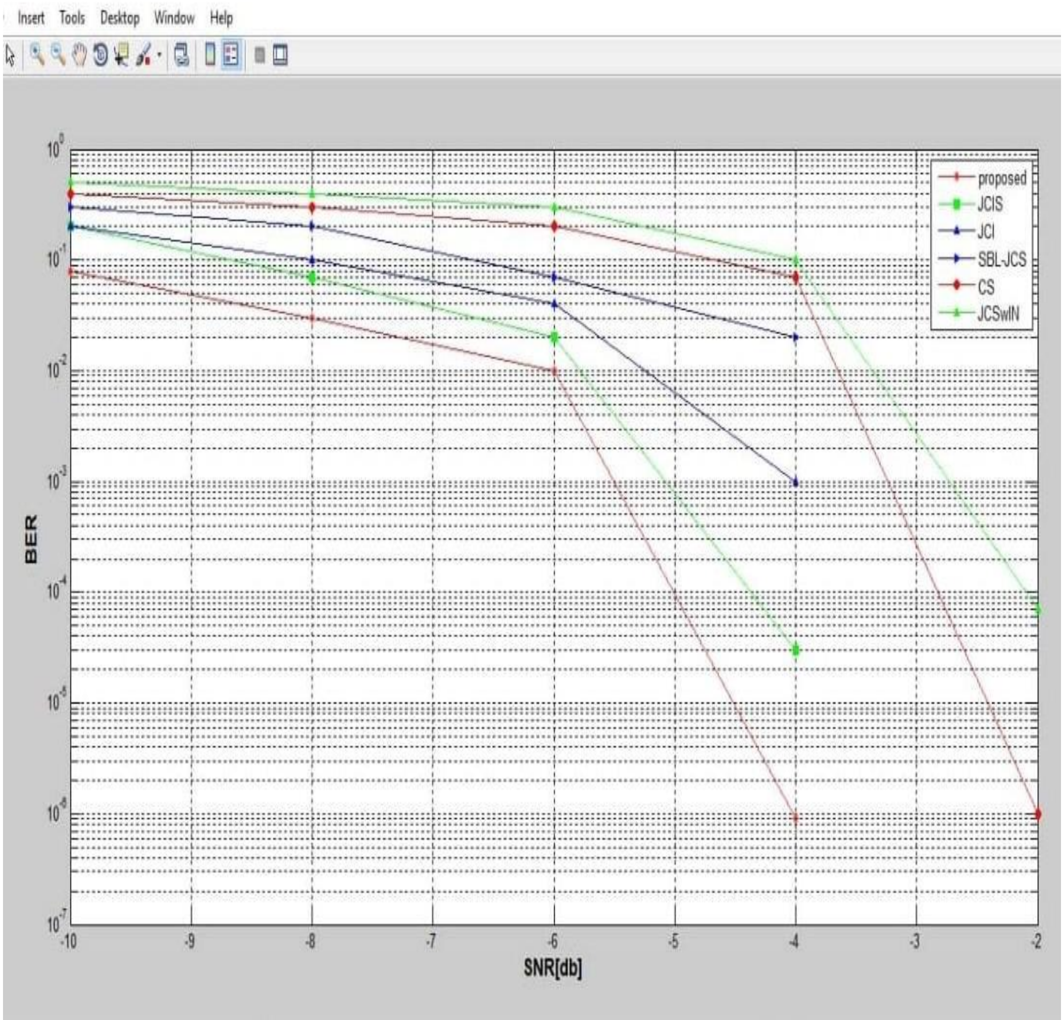


Figure 3: BER versus SNR in coded OFDM system. Number of pilot subcarriers 44 and number of null subcarriers 50

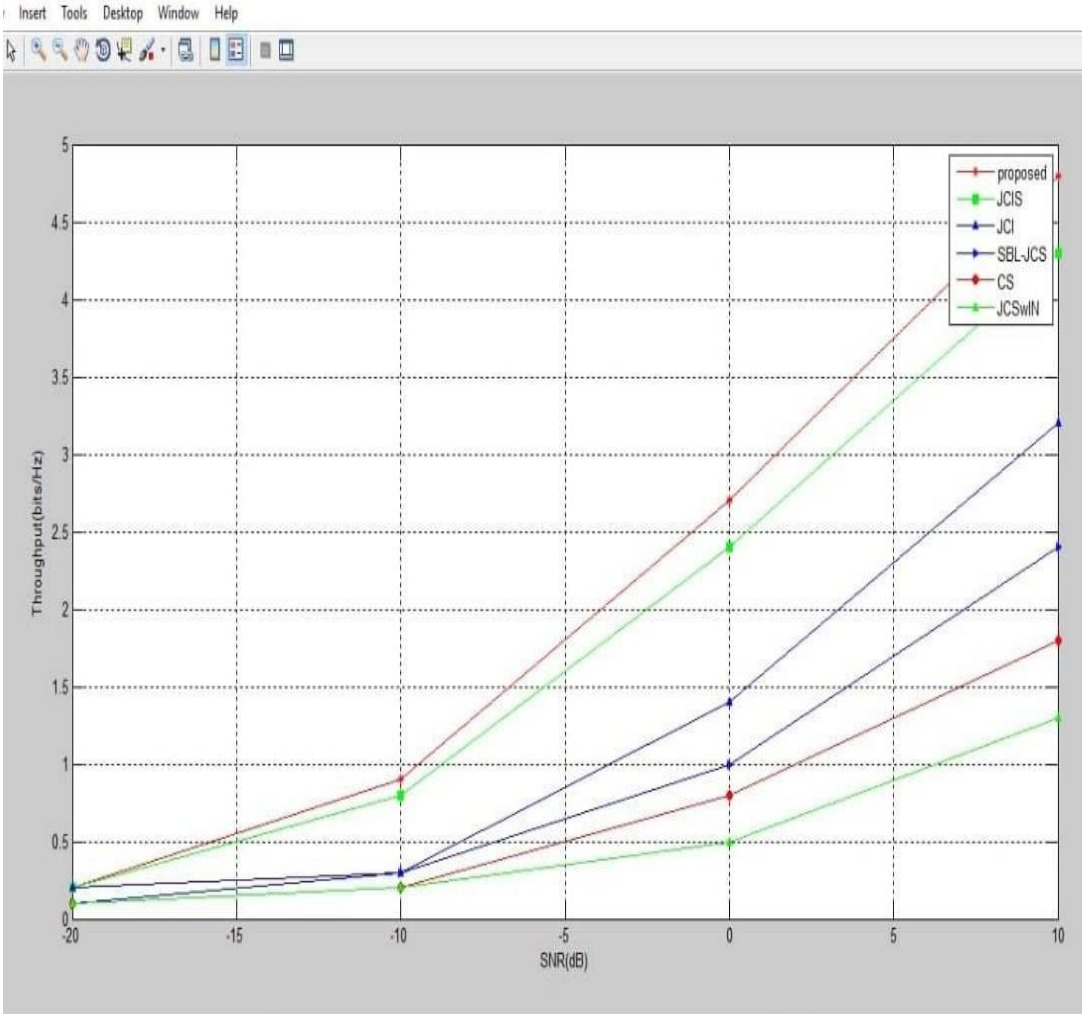


Figure 4: System throughput versus SNR.

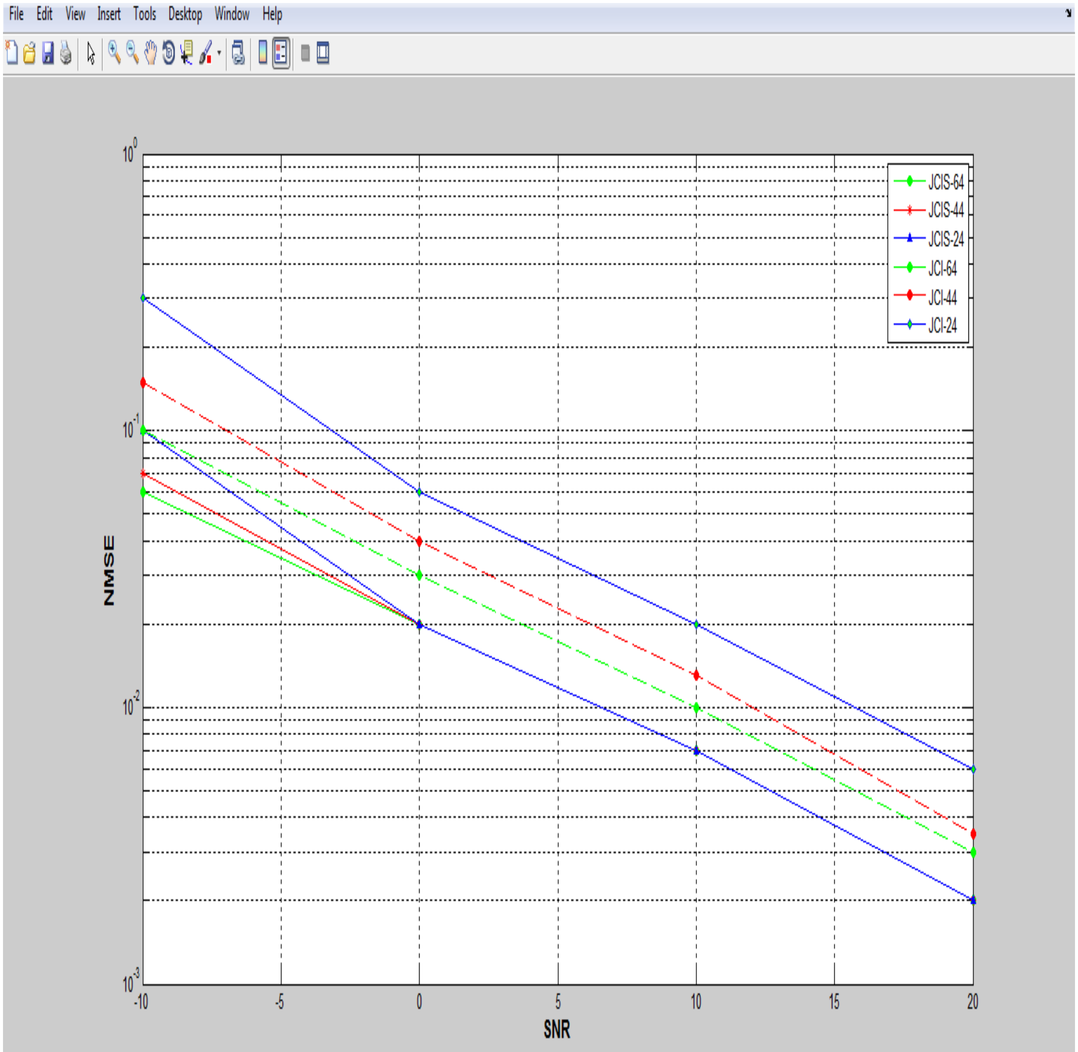


Figure 5: The MSE Performance Comparision between JCI and JCIS with various number of pilot tones.

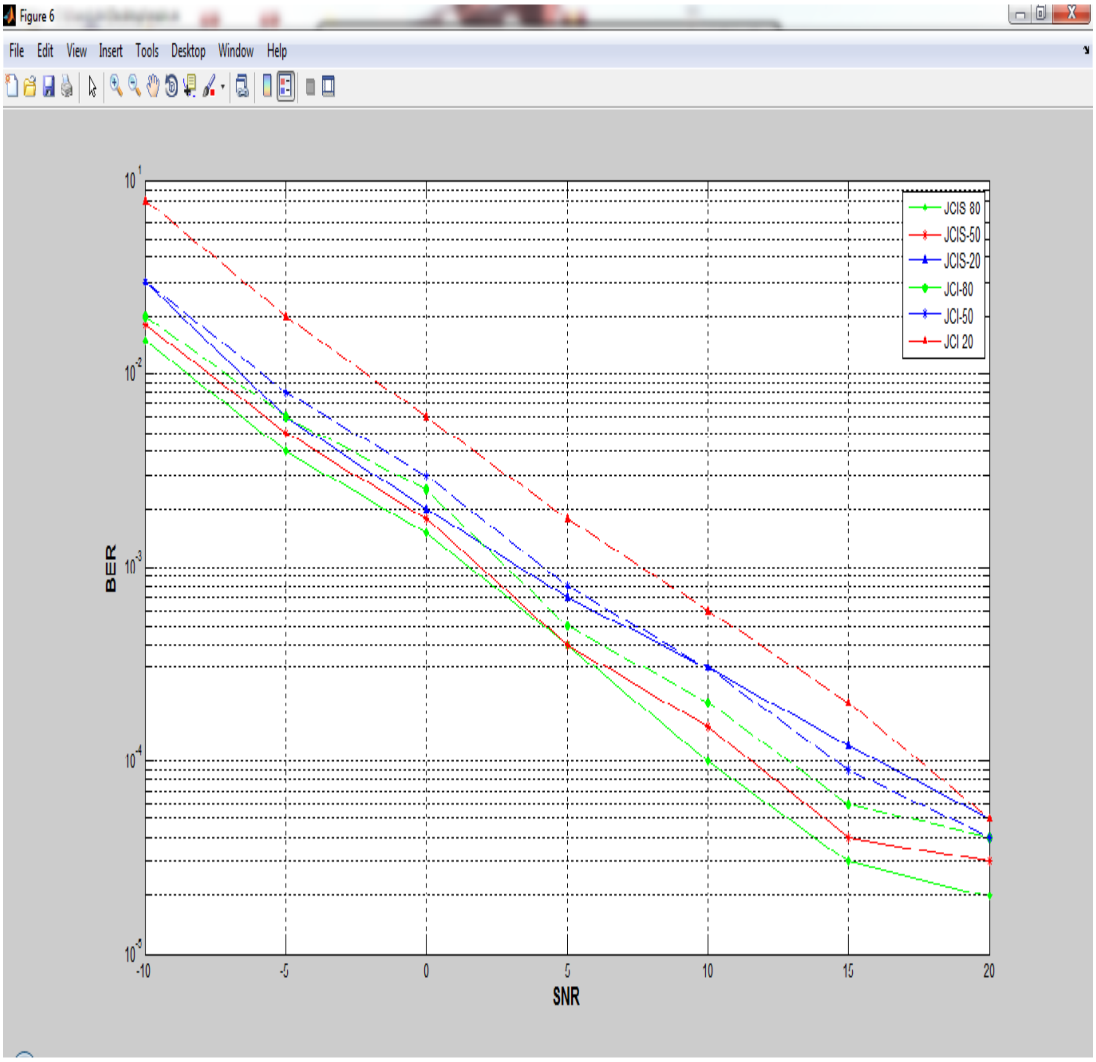


Figure 6: BER Versus SNR ratio with various number of null tones.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

In this paper, we consider the joint sparse channel estimation, impulsive noise mitigation, and data detection for OFDM systems. By observing the sparsity of channel and impulsive noise in the time domain, we construct an expanded sparse vector to represent the channel and impulsive noise together. To estimate the augmented vector, JCI algorithm is proposed which uses only the pilot and null subcarriers. Furthermore JCIS algorithm is developed to improve the performance of channel estimation and impulsive noise cancelation, which apply the data detection simultaneously. We derive the analytical expression of BCRB for JCIS algorithm as well. The MSE performance of our proposed scheme outperforms the conventional methods and is close to the lower bound. Moreover, simulation results show our methods can have a good BER performance with fewer pilot and null subcarriers and obtain better spectral efficiency.

PUBLICATION

St. MARTIN'S ENGINEERING COLLEGE
An Autonomous Institute
A Non Minority College | Approved by AICTE | Affiliated to JNTUH, Hyderabad |
NAAC-Accredited 'A+' Grade | 2(f) & 12(B) status (UGC) ISO 9001:2008 Certified |
NBA Accredited | SIRO (DSIR) | UGC-Paramarsh | Recognized Remote Center of IIT, Bombay
Dhulapally, Secunderabad – 500100, Telangana State, India.
www.smec.ac.in

Acceptance Letter for online oral presentation & journal publication

Online Mega international Conference "Smart Modernistic in Electronics and Communication" *(ICSMEC–21) on 02nd & 03rd July 2021*

Dear K.Ravi, M.Vamshy, P.Varshini,D.Prasad:

Title of the paper: JOINT CHANNEL ESTIMATION AND IMPULSIVE NOISE MITIGATION METHOD FOR OFDM SYSTEMS USING SPARSE BAYESIAN LEARNING

Paper ID: ICSMEC21-0023
With heartiest congratulations we are pleased to inform you that based on the recommendations of the expert reviewers, your paper has been accepted for journal publication and oral presentation in online mega International Conference on "Smart Modernistic in Electronics and Communication" ***(ICSMEC-21).***

ICSMEC-21 conference received over 400+ submissions from different parts of the world and regions so far, reviewed by international experts; the acceptance ratio is controlled below 20%. Your paper will be submitted for **Journal publications free of cost** in the appropriate given Journals after registration. **You will be given registration kit, International conference proceedings with ISBN (Soft copy), and international certificate of publication.**

You are now required to pay the conference registration fees (Faculty / Research Scholars: Rs.3500/-) as mentioned below A/C details on or before 5 working day after receiving this email else your paper will not be considered for presentation as well as publication in the Journal.

For NEFT/RTGS/IMPS
Account Details: (Current Account)
A/C Name: PRINCIPAL, St. Martin's Engineering College
Bank Name: SBI, Kukatpally, Hyderabad
Account No: 30667640268
IFSC code: SBIN0011664
Swift Code: SBININBB725 (Applicable for foreign author)

For PAYTM users

After payment, please send the registration form, payment proof and final paper in.doc/.docx format.
Your interest in *ICSMEC-21* is very much appreciated. Look forward to meeting you on the day of conference.

Yours sincerely,
ICSMEC- 21 Organizing Committees
icsmec21@smec.ac.in

(Organized by Department of ECE)

REFERENCES

[1] S. F. Hasan, X. Ding, N. H. Siddique, and S. Chakraborty, ‘‘Measuring disruption in vehicular communications,’’ IEEE Trans. Veh. Technol., vol. 60, no. 1, pp. 148–159, Jan. 2011.

[2] S. W. Lai and G. G. Messier, ‘‘Using the wireless and PLC channels for diversity,’’ IEEE Trans. Commun., vol. 60, no. 12, pp. 3865–3875, Dec. 2012.

[3] Z. Wang, S. Zhou, J. Catipovic, and P. Willett, ‘‘Asynchronous multiuser reception for OFDM in underwater acoustic communications,’’ IEEE Trans. Wireless Commun., vol. 12, no. 3, pp. 1050–1061, Mar. 2013.

[4] S. Liu, F. Yang, W. Ding, and J. Song, ‘‘Double kill: Compressivesensing-based narrow-band interference and impulsive noise mitigation for vehicular communications,’’ IEEE Trans. Veh. Technol., vol. 65, no. 7, pp. 5099–5109, Jul. 2016.

[5] F. Sacuto, F. Labeau, and B. L. Agba, ‘‘Wide band time-correlated model for wireless communications under impulsive noise within power substation,’’ IEEE Trans. Wireless Commun., vol. 13, no. 3, pp. 1449–1461, Mar. 2014.

[6] X. Kuai, H. Sun, S. Zhou, and E. Cheng, ‘‘Impulsive noise mitigation in underwater acoustic OFDM systems,’’ IEEE Trans. Veh. Technol., no. 65, no. 10, pp. 8190–8202, Oct. 2016.

[7] IEEE Standard for Information technology–Local and metropolitan area networks–Specific requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6:Wireless Access in Vehicular Environments, IEEE Standard 802.11p Std., Jul. 2010.

[8] D. Tse and P. Viswanath, Fundamentals of Wireless Communication. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[9] Y. Liu, Z. Tan, H. Hu, L. J. Cimini, and G. Y. Li, ‘‘Channel estimation for OFDM,’’ IEEE Commun. Surveys Tuts., vol. 16, no. 4, pp. 1891–1908, 4th Quart., 2014.

[10] A. Mahmood, M. Chitre, and M. A. Armand, ‘‘Detecting OFDM signals in alpha-stable noise,’’ IEEE Trans. Commun., vol. 62, no. 10, pp. 3571–3583, Oct. 2014.

[11] S. V. Zhidkov, ‘‘Performance analysis and optimization of OFDM receiver with blanking nonlinearity in impulsive noise environment,’’ IEEE Trans. Veh. Technol., vol. 55, no. 1, pp. 234–242, Jan. 2006.

[12] S. V. Zhidkov, ‘‘Analysis and comparison of several simple impulse noise mitigation schemes for OFDM receivers,’’ IEEE Trans. Commun., vol. 56, no. 1, pp. 5–9, Jan. 2008.

[13] U. Epple and M. Schnell, ‘‘Advanced blanking nonlinearity for mitigating impulsive interference in OFDM systems,’’ IEEE Trans. Veh. Technol., vol. 66, no. 1, pp. 146–158, Jan. 2017.

[14] F. H. Juwono, Q. Guo, Y. Chen, L. Xu, D. D. Huang, and K. P. Wong, ‘‘Linear combining of nonlinear preprocessors for OFDM-based powerline communications,’’ IEEE Trans. Smart Grid, vol. 7, no. 1, pp. 253–260, Jan. 2016.

[15] N. Rožić, P. Banelli, D. Begušić, and J. Radić, ‘‘Multiple-threshold estimators for impulsive noise suppression in multicarrier communications,’’ IEEE Trans. Signal Process., vol. 66, no. 6, pp. 1619–1633, Mar. 2018.

[16] D. Darsena, G. Gelli, F. Melito, and F. Verde, ‘‘ICI-free equalization in OFDM systems with blanking preprocessing at the receiver for impulsive noise mitigation,’’ IEEE Signal Process. Lett., vol. 22, no. 9, pp. 1321–1325, Sep. 2015.

[17] G. Caire, T. Y. Al-Naffouri, and A. K. Narayanan, ‘‘Impulse noise cancellation in OFDM: An application of compressed sensing,’’ in Proc. IEEE Int. Symp. Inf. Theory, Toronto, ON, Canada, Jul. 2008, pp. 1293–1297.

[18] J. Lin, M. Nassar, and B. L. Evans, ‘‘Impulsive noise mitigation in powerline communications using sparse Bayesian learning,’’ IEEE J. Sel. Areas Commun., vol. 31, no. 7, pp. 1172–1183, Jul. 2013.

[19] T. Y. Al-Naffouri, A. A. Quadeer, and G. Caire, ‘‘Impulse noise estimation and removal for OFDM systems,’’ IEEE Trans. Commun., vol. 62, no. 3, pp. 976–989, Mar. 2014.

[20] L. Lampe, ‘‘Bursty impulse noise detection by compressed sensing,’’ in Proc. IEEE Int. Symp. Power Line Commun. Appl., Udine, Italy, Apr. 2011, pp. 29–34.

[21] M. Korki, J. Zhang, C. Zhang, and H. Zayyani, ‘‘Block-sparse impulsive noise reduction in OFDM systems—A novel iterative Bayesian approach,’’ IEEE Trans. Commun., vol. 64, no. 1, pp. 271–284, Jan. 2016.

[22] M. F. Duarte and Y. C. Eldar, ‘‘Structured compressed sensing: From theory to applications,’’ IEEE Trans. Signal Process., vol. 59, no. 9, pp. 4053–4085, Sep. 2011.

[23] X. Jiang, T. Kirubarajan, and W.-J. Zeng, ‘‘Robust sparse channel estimation and equalization in impulsive noise using linear programming,’’ Signal Process., vol. 93, no. 5, pp. 1095–1105, May 2013.

[24] A. Mehboob, L. Zhang, J. Khangosstar, and K. Suwunnapuk, ‘‘Joint channel and impulsive noise estimation for OFDM based power line communication systems using compressed sensing,’’ in Proc. IEEE 17th Int. Symp. Power Line Commun. Appl., Johannesburg, South Africa, Mar. 2013, pp. 203–208.

[25] Y.-R. Chien, ‘‘Iterative channel estimation and impulsive noise mitigation algorithm for OFDM-based receivers with application to power-line communications,’’ IEEE Trans. Power Del., vol. 30, no. 6, pp. 2435–2442, Dec. 2015.

A

PROJECT REPORT

On

SEWAGE MONITORING SYSTEM USING IoT

Submitted by

1) KSS Pankaj (17K81A0425) 2) S Sai Krishna (17K81A0444)

3) T Supriya (17K81A0453) 4) Y Sai Ganesh (17K81A0460)

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Dr. B. Hari Krishna, Ph.D.,

Professor and HoD

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

An Autonomous Institute

Dhulapally, Secunderabad – 500 100

JUNE 2021

BONAFIDE CERTIFICATE

This is to certify that the project entitled Sewage Monitoring System using IoT, is being submitted by **KSS Pankaj (17K81A0425), S Sai Krishna (17K81A0444), T Supriya (17K81A0453), and Y Sai Ganesh (17K81A0460)** in partial fulfilment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** IN **ELECTRONICS AND COMMUNICATION ENGINEERING** is a record of bonafide work carried out by them. The result embodied in this report has been verified and found satisfactory.

Dr. B. Hari Krishna
Department of ECE

Head of the Department
Dr. B. Hari Krishna
Department of ECE

Internal Examiner

External Examiner

Place:

Date:

DECLARATION

We, the students of **Bachelor of Technology** in Department of Electronics and Communication Engineering, session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled **Sewage Monitoring System using IoT** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted to any university for the award of any degree.

KSS Pankaj (17K81A0425)
S Sai Krishna (17K81A0444)
T Supriya (17K81A0453)
Y Sai Ganesh (17K81A0460)

// ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to our College Management for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to the principal **Dr. P. Santosh Kumar Patra**, for his invaluable encouragement, suggestions, and support from an early stage of this research and for providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr. B. Hari Krishna,** Professor & Head, the Department of Electronics & Communication Engineering for his advice, supervision, and vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our internal project coordinator **A. Soumya**, and the project coordinator of our department **Dr. A. Anand** for his involvement with originality has triggered and nourished the intellectual maturity that will help us for a long time to come.

1. KSS Pankaj
2. S Sai Krishna
3. T Supriya
4. Y Sai Ganesh

ABSTRACT

The main objective of this project is to develop a Cloud integrated IoT-based sewage level monitoring system for a solid-liquid separator. The need to develop such a system is that there is a requirement to monitor sludge and liquid levels continuously. As a part of this product development, an interface is created with digital outputs, and the communication medium is the Internet. Having a simple yet optimised software-hardware interface and minimal hardware components, this system is not only very cost-effective but also very efficient with respect to the existing sewage level monitoring models. As technology continues to evolve and costs come down, these tools will provide more in-depth information and should be considered by designers, regulators, installers, and service providers. Sensors can be installed in septic tanks, pump tanks, and drain fields. The sensors are connected to a hub that uses the internet service in the home or business. Data from the sensors can be viewed in real-time from any internet-connected device, and all historical data is accessible for maintenance, decision-making, or time-of-property-transfer inspections. Warnings and critical alert notifications can be notified to the septic professional, system owner, regulator, or any other interested stakeholders. For product development, NodeMCU and HC-SR04 Ultrasonic Sensor Module are used. HTML and CSS are used to design the Webpages, while PHP is used as a connection between the Microcontroller (ESP8266) and the Cloud Database. The Cloud Database uses the standard Relational Database Model for storing data.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF OUTPUT SCREENS

1. INTRODUCTION

1.1 OVERVIEW OF THE PROJECT:

Sewage Management is a very big problem today. There are multiple cons of unseen septic systems. Gases get developed, which may sometimes cause the chambers to explode, which may lead to the spread of several pathogens. Also, the sludge and the slime may overflow. The main objective is to develop a Cloud integrated IoT-based sewage level monitoring system. In the conventional case, the issue is that the sludge level must be continuously monitored, so that manually or automatically the sludge tank or chamber can be emptied so that the sludge chamber/ tank won't overflow and it could be reused. This can't be done as one doesn't wish to open the sludge chamber using their bare hands. Thus this task needs to be done remotely, and monitored as a graph or a digital value representing the solid level.

So as a part of this product development, a device is made which contains sensor(s) to calculate the level of the sludge, communicating with a controller which sends this data to the internet in digital format, and finally this data can be seen as entries in a cloud database, and on a website.

An interface is created with both analog and digital representations of level values, and a ready made cloud server is used, and the communication medium is the Internet.

1.2 OBJECTIVES OF THE STUDY:

With the increasing number of Sewage Explosion cases and the severities of it, there is an urgent need to have an efficient Sewage management system.

So the main objective of this project is to develop a Cloud integrated IoT-based sewage level monitoring system for a solid-liquid separator, and to make such a model as economical and simple as possible.

1.3 SCOPE OF THE STUDY:

This model could be installed in any Sludge Chamber/ Septic System, for Remote Monitoring.

This model can be further developed by integrating a liquid level monitoring system with it.

1.4 MATERIAL REQUIREMENT:

For materialising this project, primarily, a sensor and a microcontroller are needed. Also for the software development, a cloud server with database connectivity, and IDEs (Integrated Development Environments) are needed.

So, the hardware requirements include:

1. HC-SR04 Ultrasonic Sensor Module

2. ESP8266 NodeMCU Board.

Likewise, we have Software Requirements as well.

The software requirements include:

1. Arduino IDE

2. Code Editor for HTML, CSS, and PHP

3. Azure Data Studio Cloud Server

4. MySQL Database Connectivity

5. HTML, CSS, PHP supported Web Browser

1.5 PROCUREMENT OF EQUIPMENT:

Procuring the hardware tools was done due to their availability in the market, and software tools that are all open-source. The Hardware is purchased, assembled, connected to the Software via code in IDE and a Server that connects the entire system to the Cloud via Internet.

1.6 ORGANIZATION OF CHAPTERS:

1.6.0 INTRODUCTION

1.6.1 LITERATURE SURVEY

1.6.2 PROJECT DESIGN

1.6.3 PROJECT IMPLEMENTATION

1.6.4 PROJECT TESTING

1.6.5 CONCLUSION AND FUTURE ENHANCEMENT

2. LITERATURE SURVEY AND EXISTING MODELS

2.0 LITERATURE SURVEY ON SEWAGE MANAGEMENT:

In the articles about sewer gas explosion in public lavatory[1], emission of gases from the septic systems[2] & [3], estimated health risks due to these recreational waters[4], it is very clear how fatal the unmonitored sewage systems might be. Also from the article named Quantification of methane emissions from latrines, septic tanks, and stagnant, open sewers in the world[5], we can get the approximate quantified information about the amount of carbon emission. It says, open sewers and on-site wastewater treatment systems, including latrines and septic sewage tanks around the world may be a significant methane (CH_4) source that has previously received little attention. Combined global CH_4 emissions from latrines, septic sewage tanks, and stagnant, open sewers are estimated at 29 teragram per year (Tg/yr), with lower and upper bound ranges of 14 and 49 Tg/yr. Major uncertainties in the estimates are associated with the degree to which wastewater is treated in a specific system; the amount of wastewater that is discharged into stagnant, open sewers; and the degree to which anaerobic decomposition takes place. Latrines in rural areas of developing countries such as China and India are believed to be the single most significant source of CH_4, accounting for approximately 12 Tg/yr. Total CH_4 emissions from stagnant, open sewers are estimated at 10 Tg/yr. Considering these risks, many sewage monitoring, altering, and maintenance systems were developed. For instance, the paper on an IoT based sewage monitoring system[6] takes gases as references to estimate the fatality. And moreover to deal with the entire model, the components increase the costs and commercialising this prototype may not be an indispensable idea. So, keeping in mind all these factors, a comparatively inexpensive model of Sewage monitoring system is developed based on the fact that in a large flat-bottomed tank, the sludge rests and the sludge eventually gets flattened down to a nearly single surface level. So here, based on the observations from the articles, Obstacle detection using Ultrasonic Sensors[7], HC SR04 Ultrasonic Sensor[8], and Investigating the resolution ability of the HC-SR04 Ultrasonic Sensor[9], and the pricing of such products, HC-SR04 Ultrasonic sensor is considered to be a good fit for making an inexpensive yet efficient product. And based on the paper, Distance measurement using Ultrasonic Sensor and Node MCU[10], a basic level detecting system had been developed. Further, the model has to be used in a more complex system which communicates with the Cloud. So, the basic idea of this model was based on the conference papers, Internet of Things (IoT) enabled water monitoring system[11], and IoT Based Water Level Control System[12] using the Blynk IoT platform. Based on the articles, How to upload NodeMCU Temperature, Humidity data on ThingSpeak using Arduino IDE[13], ESP8266 Plot Sensor readings to Web Server in Real-Time Chart[14], and Great Projects: ESP8266 DHT11 SD Card WebServer[15], an integration between ESP8266/ NodeMCU, Thing Speak, and Web Server was made.

2.1 REVIEW OF RELATED LITERATURE:

In the existing system, we are using Water Level Monitoring Systems that are confined to only Water/ Liquid Level Monitoring. There is no perfect Cloud-and-IoT integration in any of the reviewed systems, which is actually essential for our project. But the reviewed literature provides a basic idea on how to connect ESP8266/ NodeMCU module to the Ultrasonic Sensor Module, and the Internet. Also, the papers and articles reviewed provide information on how to connect the NodeMCU board to the Cloud Server via the PHP functions and the Internet.

2.2 CONCLUSION ON REVIEWS:

In the existing system, we are using Water Level Monitoring Systems that are confined to only Water/ Liquid Level Monitoring. And in the conventional level monitoring and networking systems, we are having lengthy codes/ programs, larger delays, and the overall performance of the system is not upto the mark for commercial use. Also, the ThingSpeak and Blynk applications can not be used in commercial applications, as they are third party open-source tools.

So, keeping in mind the practical applications and the necessities, we couldn’t move forward with the existing systems.

3. PROJECT DESIGN

3.0 OVERVIEW OF THE DESIGN:

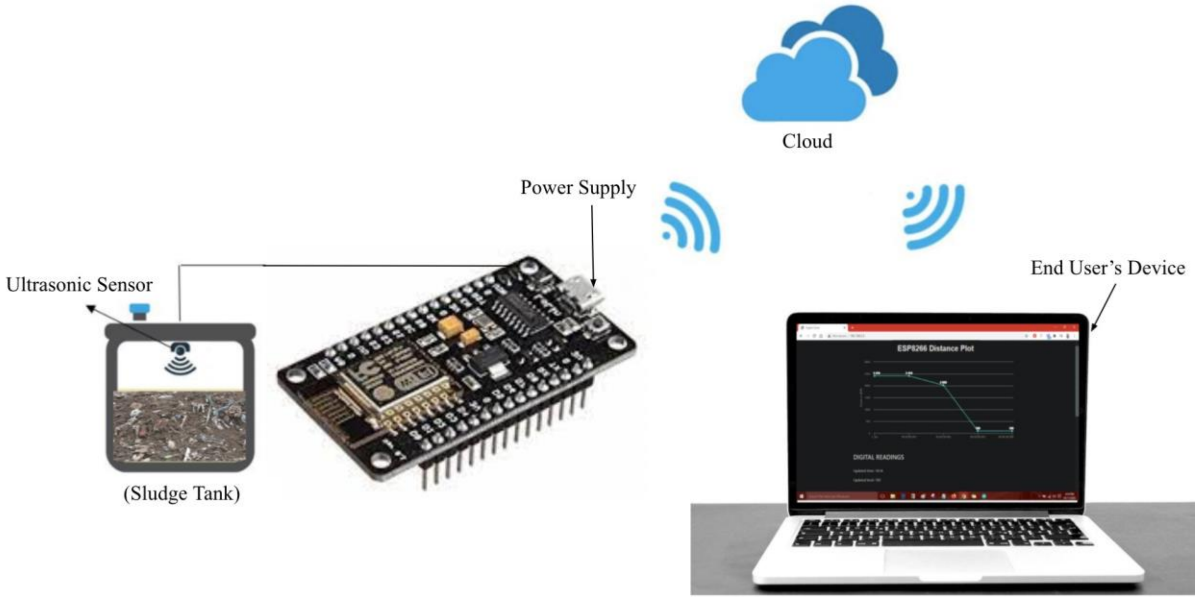


Fig 1: Schematic Block Diagram of the Project

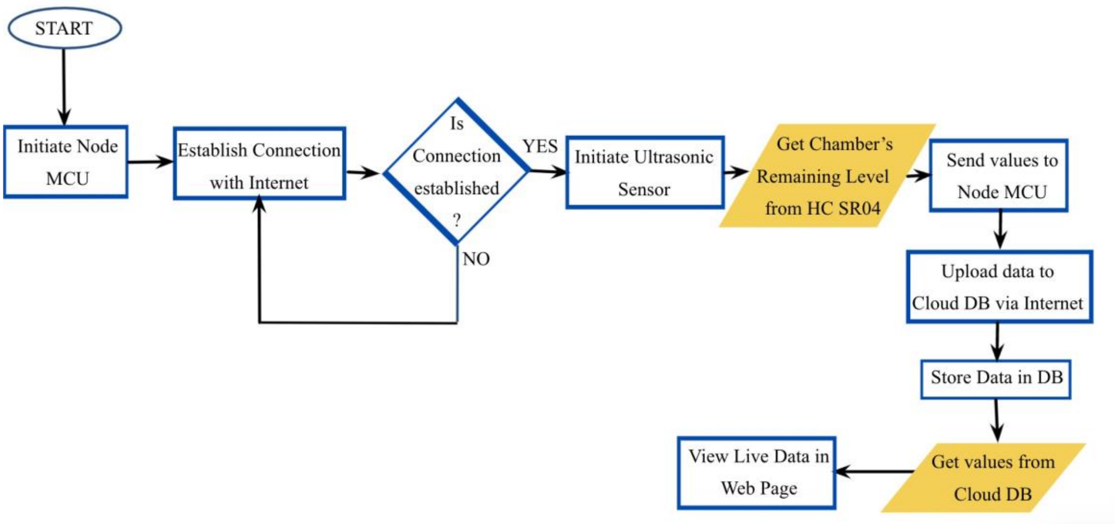


Fig 2: Flow Chart of the Project

3.0.1 Embedded Systems:

An embedded system is a computer system designed to perform one or a few dedicated functions often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. By contrast, a general-purpose computer, such as a personal computer (PC), is designed to be flexible and to meet a wide range of end-user needs. Embedded systems control many devices in common use today.

Embedded systems are controlled by one or more main processing cores that are typically either microcontrollers or digital signal processors (DSP). The key characteristic, however, is being dedicated to handle a particular task, which may require very powerful processors. For example, air traffic control systems may usefully be viewed as embedded, even though they involve mainframe computers and dedicated regional and national networks between airports and radar sites. (Each radar probably includes one or more embedded systems of its own.)

Since the embedded system is dedicated to specific tasks, design engineers can optimise it to reduce the size and cost of the product and increase the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

Physically embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

In general, "embedded system" is not a strictly definable term, as most systems have some element of extensibility or programmability. For example, handheld computers share some elements with embedded systems such as the operating systems and microprocessors which power them, but they allow different applications to be loaded and peripherals to be connected. Moreover, even systems which don't expose programmability as a primary feature generally need to support software updates. On a continuum from "general purpose" to "embedded", large application systems will have subcomponents at most points even if the system as a whole is "designed to perform one or a few dedicated functions", and is thus appropriate to call "embedded". A modern example of embedded system is shown in fig: 2.1.

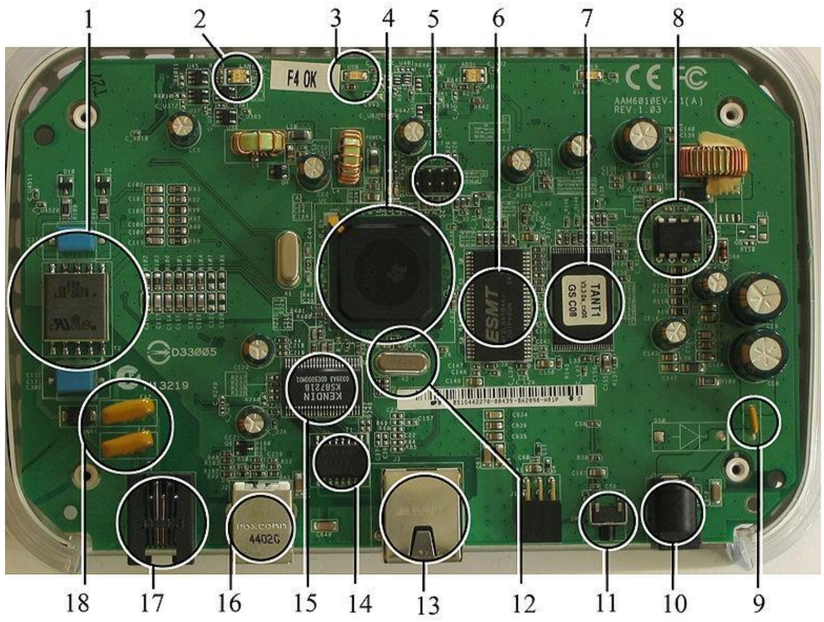


Fig 3: A modern example of embedded system

Labeled parts include microprocessor (4), RAM (6), flash memory (7).Embedded systems programming is not like normal PC programming. In many ways, programming for an embedded system is like programming PC 15 years ago. The hardware for the system is usually chosen to make the device as cheap as possible. Spending an extra dollar a unit in order to make things easier to program can cost millions. Hiring a programmer for an extra month is cheap in comparison. This means the programmer must make do with slow processors and low memory, while at the same time battling a need for efficiency not seen in most PC applications. Below is a list of issues specific to the embedded field.

3.0.2 History of Embedded Systems:

In the earliest years of computers in the 1930–40s, computers were sometimes dedicated to a single task, but were far too large and expensive for most kinds of tasks performed by embedded computers of today. Over time however, the concept of programmable controllers evolved from traditional electromechanical sequencers, via solid state devices, to the use of computer technology.

One of the first recognizably modern embedded systems was the Apollo Guidance Computer, developed by Charles Stark Draper at the MIT Instrumentation Laboratory. At the project's inception, the Apollo guidance computer was considered the riskiest item in the Apollo project as it employed the then newly developed monolithic integrated circuits to reduce the size and weight. An early mass-produced embedded system was the Autonetics D-17 guidance computer for the Minuteman missile, released in 1961. It was built from transistor logic and had a hard disk for main memory. When the Minuteman II went into production in 1966, the D-17 was replaced with a new computer that was the first high-volume use of integrated circuits.

3.0.3 Embedded Tools:

Embedded development makes up a small fraction of total programming. There's also a large number of embedded architectures, unlike the PC world where 1 instruction set rules, and the Unix world where there's only 3 or 4 major ones. This means that the tools are more expensive. It also means that they're lowering featured, and less developed. On a major embedded project, at some point you will almost always find a compiler bug of some sort.

Debugging tools are another issue. Since you can't always run general programs on your embedded processor, you can't always run a debugger on it. This makes fixing your program difficult. Special hardware such as JTAG ports can overcome this issue in part. However, if you stop on a breakpoint when your system is controlling real world hardware (such as a motor), permanent equipment damage can occur. As a result, people doing embedded programming quickly become masters at using serial IO channels and error message style debugging.

3.0.4 Embedded Systems - Resources:

To save costs, embedded systems frequently have the cheapest processors that can do the job. This means your programs need to be written as efficiently as possible. When dealing with large data sets, issues like memory cache misses that never matter in PC programming can hurt you. Luckily, this won't happen too often- use reasonably efficient algorithms to start, and optimise only when necessary. Of course, normal profilers won't work well, due to the same reason debuggers don't work well.

Memory is also an issue. For the same cost savings reasons, embedded systems usually have the least memory they can get away with. That means their algorithms must be memory efficient (unlike in PC programs, you will frequently sacrifice processor time for memory, rather than the reverse). It also means you can't afford to leak memory. Embedded applications generally use deterministic memory techniques and avoid the default "new" and "malloc" functions, so that leaks can be found and eliminated more easily. Other resources programmers expect may not even exist. For example, most embedded processors do not have hardware FPUs (Floating-Point Processing Unit). These resources either need to be emulated in software, or avoided altogether.

3.0.5 Real Time Issues with Embedded Systems:

Embedded systems frequently control hardware, and must be able to respond to them in real time. Failure to do so could cause inaccuracy in measurements, or even damage hardware such as motors. This is made even more difficult by the lack of resources available. Almost all embedded systems need to be able to prioritise some tasks over others, and to be able to put off/skip low priority tasks such as UI in favour of high priority tasks like hardware control.

3.0.6 Need For Embedded Systems:

The uses of embedded systems are virtually limitless, because every day new products are introduced to the market that utilises embedded computers in novel ways. In recent years, hardware such as microprocessors, microcontrollers, and FPGA chips have become much cheaper. So when implementing a new form of control, it's wiser to just buy the generic chip and write your own custom software for it. Producing a custom-made chip to handle a particular task or set of tasks costs far more time and money. Many embedded computers even come with extensive libraries, so that "writing your own software" becomes a very trivial task indeed. From an implementation viewpoint, there is a major difference between a computer and an embedded system. Embedded systems are often required to provide Real-Time response. The main elements that make embedded systems unique are its reliability and ease in debugging.

3.0.7 Embedded Systems - Debugging:

Embedded debugging may be performed at different levels, depending on the facilities available. From simplest to most sophisticate they can be roughly grouped into the following areas:

- Interactive resident debugging, using the simple shell provided by the embedded operating system (e.g. Forth and Basic)
- External debugging using logging or serial port output to trace operation using either a monitor in flash or using a debug server like the Remedy Debugger which even works for heterogeneous multi core systems.
- An in-circuit debugger (ICD), a hardware device that connects to the microprocessor via a JTAG or Nexus interface. This allows the operation of the microprocessor to be controlled externally, but is typically restricted to specific debugging capabilities in the processor.
- An in-circuit emulator replaces the microprocessor with a simulated equivalent, providing full control over all aspects of the microprocessor.
- A complete emulator provides a simulation of all aspects of the hardware, allowing all of it to be controlled and modified and allowing debugging on a normal PC.
- Unless restricted to external debugging, the programmer can typically load and run software through the tools, view the code running in the processor, and start or stop its operation. The view of the code may be as assembly code or source-code.

Because an embedded system is often composed of a wide variety of elements, the debugging strategy may vary. For instance, debugging a software(and microprocessor) centric embedded system is different from debugging an embedded system where most of the processing is performed by peripherals (DSP, FPGA, co-processor). An increasing number of embedded systems today use more than one single processor core. A common problem with multi-core development is the proper synchronisation of software execution. In such a case, the embedded system design may wish to check the data traffic on the busses between the processor cores, which requires very low-level debugging, at signal/bus level, with a logic analyser, for instance.

3.0.8 Reliability of Embedded Systems:

Embedded systems often reside in machines that are expected to run continuously for years without errors and in some cases recover by them if an error occurs. Therefore the software is usually developed and tested more carefully than that for personal computers, and unreliable mechanical moving parts such as disk drives, switches or buttons are avoided.

Specific reliability issues may include:

- The system cannot safely be shut down for repair, or it is too inaccessible to repair. Examples include space systems, undersea cables, navigational beacons, bore-hole systems, and automobiles.
- The system must be kept running for safety reasons. "Limp modes" are less tolerable. Often backups are selected by an operator. Examples include aircraft navigation, reactor control systems, safety-critical chemical factory controls, train signals, engines on single-engine aircraft.
- The system will lose large amounts of money when shut down: Telephone switches, factory controls, bridge and elevator controls, funds transfer and market making, automated sales and service.

A variety of techniques are used, sometimes in combination, to recover from errors—both software bugs such as memory leaks, and also soft errors in the hardware:

- Watchdog timer that resets the computer unless the software periodically notifies the watchdog
- Subsystems with redundant spares that can be switched over to
- software "limp modes" that provide partial function
- Designing with a Trusted Computing Base (TCB) architecture[6] ensures a highly secure & reliable system environment
- An Embedded Hypervisor is able to provide secure encapsulation for any subsystem component, so that a compromised software component cannot interfere with other subsystems, or privileged-level system software. This encapsulation keeps faults from propagating from one subsystem to another, improving reliability. This may also allow a subsystem to be automatically shut down and restarted on fault detection.
- Immunity Aware Programming

3.0.9 Explanation of Embedded Systems:

3.0.9.0 Software Architecture:

There are several different types of software architecture in common use.

- Simple Control Loop:

In this design, the software simply has a loop. The loop calls subroutines, each of which manages a part of the hardware or software.

- Interrupt Controlled System:

Some embedded systems are predominantly interrupt controlled. This means that tasks performed by the system are triggered by different kinds of events. An interrupt could be generated for example by a timer in a predefined frequency, or by a serial port controller receiving a byte. These kinds of systems are used if event handlers need low latency and the event handlers are short and simple.

Usually these kinds of systems run a simple task in a main loop also, but this task is not very sensitive to unexpected delays. Sometimes the interrupt handler will add longer tasks to a queue structure. Later, after the interrupt handler has finished, these tasks are executed by the main loop. This method brings the system close to a multitasking kernel with discrete processes.

- Cooperative Multitasking:

A non-preemptive multitasking system is very similar to the simple control loop scheme, except that the loop is hidden in an API. The programmer defines a series of tasks, and each task gets its own environment to “run” in. When a task is idle, it calls an idle routine, usually called “pause”, “wait”, “yield”, “nop” (stands for no operation), etc. The advantages and disadvantages are very similar to the control loop, except that adding new software is easier, by simply writing a new task, or adding to the queue-interpreter.

- Primitive Multitasking:

In this type of system, a low-level piece of code switches between tasks or threads based on a timer (connected to an interrupt). This is the level at which the system is generally considered to have an "operating system" kernel. Depending on how much functionality is required, it introduces more or less of the complexities of managing multiple tasks running conceptually in parallel.

As any code can potentially damage the data of another task (except in larger systems using an MMU) programs must be carefully designed and tested, and access to shared data must be controlled by some synchronisation strategy, such as message queues, semaphores or a non-blocking synchronisation scheme.

Because of these complexities, it is common for organisations to buy a real-time operating system, allowing the application programmers to concentrate on device functionality rather than operating system services, at least for large systems; smaller systems often cannot afford the overhead associated with a generic real time system, due to limitations regarding memory size, performance, and/or battery life.

- Microkernels And Exokernels:

A microkernel is a logical step up from a real-time OS. The usual arrangement is that the operating system kernel allocates memory and switches the CPU to different threads of execution. User mode processes implement major functions such as file systems, network interfaces, etc.

In general, microkernels succeed when the task switching and intertask communication is fast, and fail when they are slow. Exokernels communicate efficiently by normal subroutine calls. The hardware, and all the software in the system are available to, and extensible by application programmers. Based on performance, functionality, requirement the embedded systems are divided into three categories:

3.0.9.1 Stand Alone Embedded System:

These systems takes the input in the form of electrical signals from transducers or commands from human beings such as pressing of a button etc.., process them and produces desired output. This entire process of taking input, processing it and giving output is done in standalone mode. Such embedded systems comes under stand alone embedded systems

Eg: microwave oven, air conditioner etc..

3.0.9.2 Real-time embedded systems:

Embedded systems which are used to perform a specific task or operation in a specific time period those systems are called as real-time embedded systems. There are two types of real-time embedded systems.

- Hard Real-time embedded systems:

These embedded systems follow an absolute dead line time period i.e.., if the tasking is not done in a particular time period then there is a cause of damage to the entire equipment.

Eg: consider a system in which we have to open a valve within 30 milliseconds. If this valve is not opened in 30 ms this may cause damage to the entire equipment. So in such cases we use embedded systems for doing automatic operations.

- Soft Real Time embedded systems:

These embedded systems follow a relative dead line time period i.e.., if the task is not done in a particular time that will not cause damage to the equipment.

Eg: Consider a TV remote control system, if the remote control takes a few milliseconds delay it will not cause damage either to the TV or to the remote control. These systems which will not cause damage when they are not operated at considerable time period those systems comes under soft real-time embedded systems.

3.0.9.3 Network communication embedded systems:

A wide range network interfacing communication is provided by using embedded systems.

Eg:

- Consider a web camera that is connected to the computer with internet can be used to spread communication like sending pictures, images, videos etc.., to another computer with internet connection throughout anywhere in the world.
- Consider a web camera that is connected at the door lock.

Whenever a person comes near the door, it captures the image of a person and sends to the desktop of your computer which is connected to internet. This gives an alerting message with image on to the desktop of your computer, and then you can open the door lock just by clicking the mouse. Fig: 2.2 show the network communications in embedded systems.

Fig 4: Network communication embedded systems

3.0.9.4 Different types of processing units:

The central processing unit (c.p.u) can be any one of the following microprocessor, microcontroller, digital signal processing.

- Among these Microcontroller is of low cost processor and one of the main advantage of microcontrollers is, the components such as memory, serial communication interfaces, analog to digital converters etc.., all these are built on a single chip. The numbers of external components that are connected to it are very less according to the application.
- Microprocessors are more powerful than microcontrollers. They are used in major applications with a number of tasking requirements. But the microprocessor requires many external components like memory, serial communication, hard disk, input output ports etc.., so the power consumption is also very high when compared to microcontrollers.
- Digital signal processing is used mainly for the applications that particularly involved with processing of signals.

3.0.10 APPLICATIONS OF EMBEDDED SYSTEMS:

3.0.10.0 Consumer applications:

At home we use a number of embedded systems which include microwave oven, remote control, vcd players, dvd players, camera etc….

Fig 5: Automatic coffee maker equipment

3.0.10.1 Office automation:

We use systems like fax machine, modem, printer etc…

Fig 6: Fax machine **Fig 7: Printing machine**

3.0.10.2 Industrial automation:

Today a lot of industries are using embedded systems for process control. In industries we design the embedded systems to perform a specific operation like monitoring temperature, pressure, humidity ,voltage, current etc.., and basing on these monitored levels we do control other devices, we can send information to a centralized monitoring station.

Fig 8: Robot

In critical industries where human presence is avoided there we can use robot**s** which are programmed to do a specific operation.

3.0.10.3 Computer networking:

Embedded systems are used as bridges routers etc..

Fig 9: Computer networking

3.0.10.4 Tele communications:

Cell phones, web cameras etc.

Fig 10: Cell Phone **Fig 11: Web camera**

3.1 HARDWARE DESCRIPTION:

3.1.0 ESP8266/ NodeMCU:

NodeMCU is a low-cost open source IoT platform. It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which was based on the ESP-12 module.

NodeMCU is an open source firmware for which open source prototyping board designs are available. The name "NodeMCU" combines "node" and "MCU" (micro-controller unit). The term "NodeMCU" strictly speaking refers to the firmware rather than the associated development kits.

Both the firmware and prototyping board designs are open source.

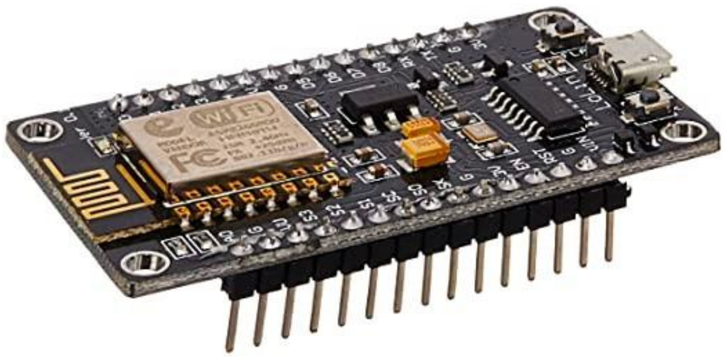


Fig 12: ESP 8266/ Node MCU

The firmware uses the Lua scripting language. The firmware is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS. Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit ESP32 has also been implemented.

The prototyping hardware typically used is a circuit board functioning as a dual in-line package (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on breadboards. The design was initially based on the ESP-12 module of the ESP8266, which is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications.

3.1.1 Regulated power supply:

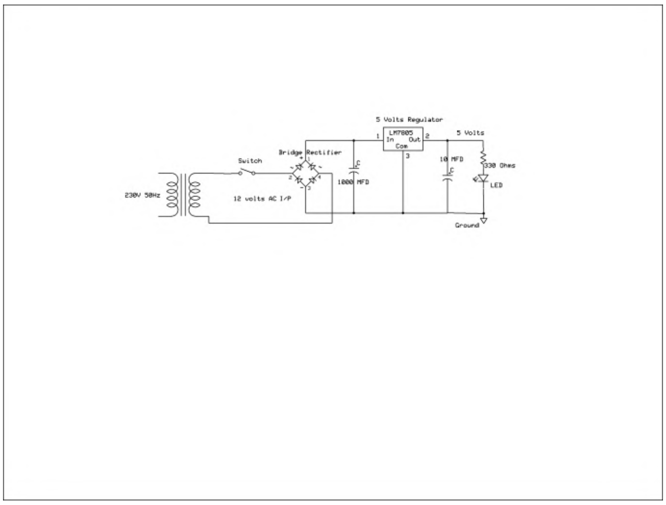


Fig 13: Circuit diagram of Regulated Power Supply with LED connection

The components mainly used in above figure are:

- 230V AC MAINS
- TRANSFORMER
- BRIDGE RECTIFIER (DIODES)
- CAPACITOR
- VOLTAGE REGULATOR (IC 7805)
- RESISTOR
- LED (LIGHT EMITTING DIODE)

The detailed explanation of each and every component mentioned above is as follows:

Transformation: The process of transforming energy from one device to another is called transformation. For transforming energy we use transformers.

Transformers: A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors without changing its frequency. A varying current in the first or primary winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the secondary winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

If a load is connected to the secondary, an electric current will flow in the secondary winding and electrical energy will be transferred from the primary circuit through the transformer to the load. This field is made up from lines of force and has the same shape as a bar magnet.

If the current is increased, the lines of force move outwards from the coil. If the current is reduced, the lines of force move inwards.

If another coil is placed adjacent to the first coil then, as the field moves out or in, the moving lines of force will "cut" the turns of the second coil. As it does this, a voltage is induced in the second coil. With the 50 Hz AC mains supply, this will happen 50 times a second. This is called MUTUAL INDUCTION and forms the basis of the transformer.

The input coil is called the PRIMARY WINDING; the output coil is the SECONDARY WINDING. Fig: 3.3.4 shows step-down transformer. The voltage induced in the secondary is determined by the TURNS RATIO.

$$\frac{\text{primary voltage}}{\text{secondary voltage}} = \frac{\text{number of primary turns}}{\text{number of secondary turns}}$$

For example, if the secondary has half the primary turns; the secondary will have half the primary voltage.

Another example is if the primary has 5000 turns and the secondary has 500 turns, then the turn's ratio is 10:1.

If the primary voltage is 240 volts then the secondary voltage will be x 10 smaller = 24 volts. Assuming a perfect transformer, the power provided by the primary must equal the power taken by a load on the secondary. If a 24-watt lamp is connected across a 24 volt secondary, then the primary must supply 24 watts.

To aid magnetic coupling between primary and secondary, the coils are wound on a metal CORE. Since the primary would induce power, called EDDY CURRENTS, into this core, the core is LAMINATED. This means that it is made up from metal sheets insulated from each other. Transformers to work at higher frequencies have an iron dust core or no core at all.

Note that the transformer only works on AC, which has a constantly changing current and moving field. DC has a steady current and therefore a steady field and there would be no induction.

Some transformers have an electrostatic screen between primary and secondary. This is to prevent some types of interference being fed from the equipment down into the mains supply, or in the other direction. Transformers are sometimes used for IMPEDANCE MATCHING.

We can use the transformers as step up or step down.

Step Up transformer:

In case of step up transformer, primary windings are every less compared to secondary winding.

Because of having more turns secondary winding accepts more energy, and it releases more voltage at the output side.

Step down transformer:

Incase of step down transformer, Primary winding induces more flux than the secondary winding, and secondary winding is having less number of turns because of that it accepts less number of flux, and releases less amount of voltage.

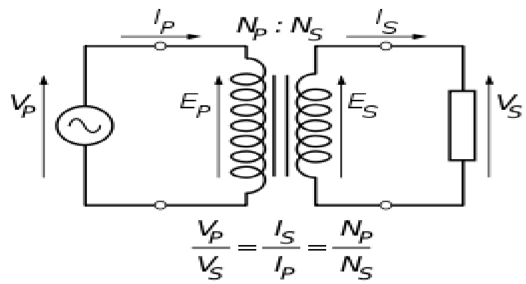


Fig 14: Step-Down Transformer

Battery power supply:

A battery is a type of linear power supply that offers benefits that traditional line-operated power supplies lack: mobility, portability and reliability. A battery consists of multiple electrochemical cells connected to provide the voltage desired. Fig: 3.3.5 shows Hi-Watt 9V battery

Fig 15: Hi-Watt 9V Battery

The most commonly used dry-cell battery is the carbon-zinc dry cell battery. Dry-cell batteries are made by stacking a carbon plate, a layer of electrolyte paste, and a zinc plate alternately until the desired total voltage is achieved. The most common dry-cell batteries have one of the following voltages: 1.5, 3, 6, 9, 22.5, 45, and 90. During the discharge of a carbon-zinc battery, the zinc metal is converted to a zinc salt in the electrolyte, and magnesium dioxide is reduced at the carbon electrode. These actions establish a voltage of approximately 1.5 V.

The lead-acid storage battery may be used. This battery is rechargeable; it consists of lead and lead/dioxide electrodes which are immersed in sulphuric acid. When fully charged, this type of battery has a 2.06-2.14 V potential (A 12 volt car battery uses 6 cells in series). During discharge, the lead is converted to lead sulphate and the sulphuric acid is converted to water. When the battery is charging, the lead sulphate is converted back to lead and lead dioxide A nickel-cadmium battery has become more popular in recent years. This battery cell is completely sealed and rechargeable. The electrolyte is not involved in the electrode reaction, making the voltage constant over the span of the batteries long service life. During the charging process, nickel oxide is oxidised to its higher oxidation state and cadmium oxide is reduced. The nickel-cadmium batteries have many benefits. They can be stored both charged and uncharged. They have a long service life, high current availabilities, constant voltage, and the ability to be recharged. Fig: 3.3.6 shows pencil battery of 1.5V.

Fig 16: Pencil Battery of 1.5V

RECTIFICATION:

The process of converting an alternating current to a pulsating direct current is called as rectification. For rectification purpose we use rectifiers.

Rectifiers:

A rectifier is an electrical device that converts alternating current (AC) to direct current (DC), a process known as rectification. Rectifiers have many uses including as components of power supplies and as detectors of radio signals. Rectifiers may be made of solid-state diodes, vacuum tube diodes, mercury arc valves, and other components.

A device that it can perform the opposite function (converting DC to AC) is known as an inverter.

When only one diode is used to rectify AC (by blocking the negative or positive portion of the waveform), the difference between the term diode and the term rectifier is merely one of usage, i.e., the term rectifier describes a diode that is being used to convert AC to DC. Almost all rectifiers comprise a number of diodes in a specific arrangement for more efficiently converting AC to DC than is possible with only one diode. Before the development of silicon semiconductor rectifiers, vacuum tube diodes and copper (I) oxide or selenium rectifier stacks were used.

Bridge full wave rectifier:

The Bridge rectifier circuit is shown in fig: 3.3.7, which converts an ac voltage to dc voltage using both half cycles of the input ac voltage. The Bridge rectifier circuit is shown in the figure. The circuit has four diodes connected to form a bridge. The ac input voltage is applied to the diagonally opposite ends of the bridge. The load resistance is connected between the other two ends of the bridge.

For the positive half cycle of the input ac voltage, diodes D1 and D3 conduct, whereas diodes D2 and D4 remain in the OFF state. The conducting diodes will be in series with the load resistance R_L and hence the load current flows through R_L.

For the negative half cycle of the input ac voltage, diodes D2 and D4 conduct whereas, D1 and D3 remain OFF. The conducting diodes D2 and D4 will be in series with the load resistance R_L and hence the current flows through R_L in the same direction as in the previous half cycle. Thus a bi-directional wave is converted into a unidirectional wave.

Input Output

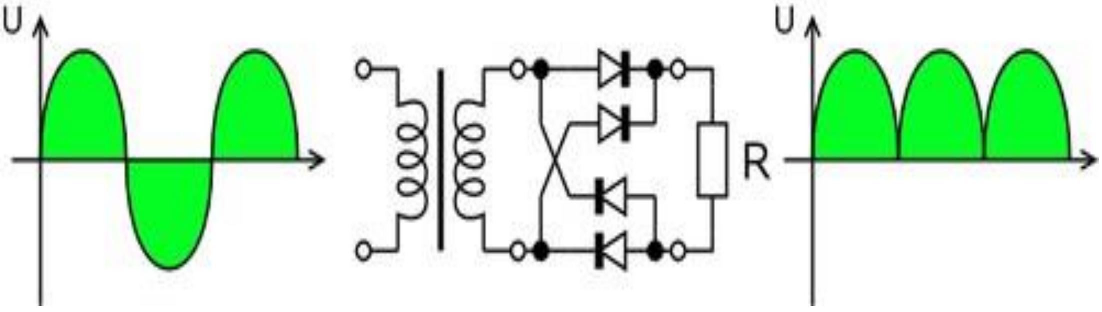


Fig 17: Bridge rectifier: a full-wave rectifier using 4 diodes

DB107:

Now -a -days Bridge rectifier is available in IC with a number of DB107. In our project we are using an IC in place of bridge rectifier. The picture of DB 107 is shown in fig: 3.3.8.

Features:

- Good for automation insertion
- Surge overload rating - 30 amperes peak
- Ideal for printed circuit board
- Reliable low cost construction utilizing molded
- Glass passivated device
- Polarity symbols molded on body
- Mounting position: Any
- Weight: 1.0 gram

Fig 18: DB107

Filtration:

The process of converting a pulsating direct current to a pure direct current using filters is called as filtration.

Filters:

Electronic filters are electronic circuits, which perform signal-processing functions, specifically to remove unwanted frequency components from the signal, to enhance wanted ones.

Introduction to Capacitors:

The Capacitor or sometimes referred to as a Condenser is a passive device, and one which stores energy in the form of an electrostatic field which produces a potential (static voltage) across its plates. In its basic form a capacitor consists of two parallel conductive plates that are not connected but are electrically separated either by air or by an insulating material called the Dielectric. When a voltage is applied to these plates, a current flows charging up the plates with electrons giving one plate a positive charge and the other plate an equal and opposite negative charge. This flow of electrons to the plates is known as the Charging Current and continues to flow until the voltage across the plates (and hence the capacitor) is equal to the applied voltage Vcc. At this point the capacitor is said to be fully charged and this is illustrated below. The construction of capacitor and an electrolytic capacitor are shown in figures 3.3.9 and 3.3.10 respectively.

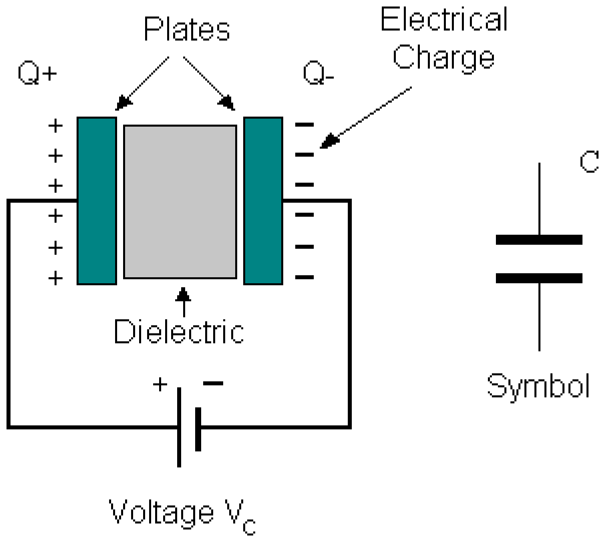


Fig 19: Construction Of a Capacitor

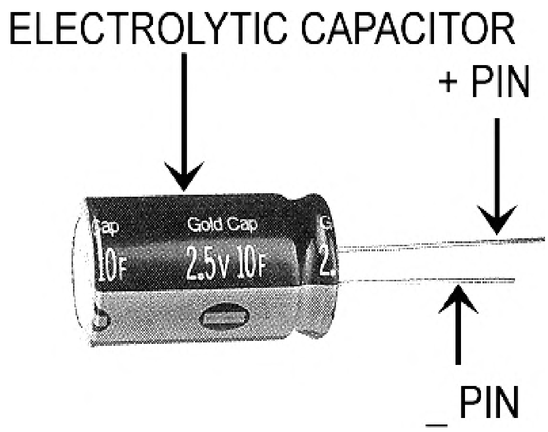


Fig 20: Electrolytic Capacitor

Units of Capacitance:

Microfarad (µF) 1µF = 1/1,000,000 = 0.000001 = 10^{-6} F

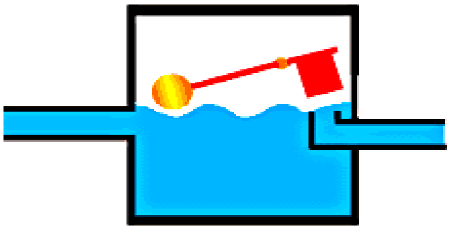
Nanofarad (nF) 1nF = 1/1,000,000,000 = 0.000000001 = 10^{-9} F

Pico farad (pF) 1pF = 1/1,000,000,000,000 = 0.000000000001 = 10^{-12} F

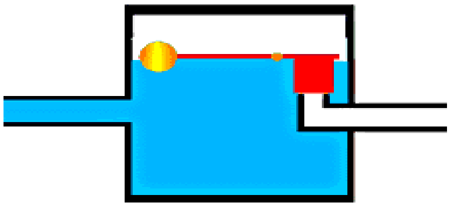
Operation of Capacitor:

Think of water flowing through a pipe. If we imagine a capacitor as being a storage tank with an inlet and an outlet pipe, it is possible to show approximately how an electronic capacitor works.

First, let's consider the case of a "coupling capacitor" where the capacitor is used to connect a signal from one part of a circuit to another but without allowing any direct current to flow.



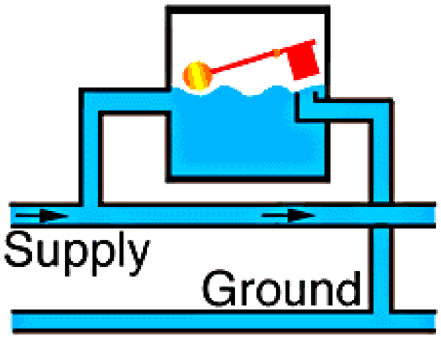
If the current flow is alternating between zero and a maximum, our "storage tank" capacitor will allow the current waves to pass through.



However, if there is a steady current, only the initial short burst will flow until the "floating ball valve" closes and stops further flow.

So a coupling capacitor allows "alternating current" to pass through because the ball valve doesn't get a chance to close as the waves go up and down. However, a steady current quickly fills the tank so that all flow stops.

A capacitor will pass alternating current but (apart from an initial surge) it will not pass d.c.



Where a capacitor is used to decouple a circuit, the effect is to "smooth out ripples". Any ripples, waves or pulses of current are passed to ground while d.c. Flows smoothly.

Regulation:

The process of converting a varying voltage to a constant regulated voltage is called as regulation. For the process of regulation we use voltage regulators.

Voltage Regulator:

A voltage regulator (also called a ‘regulator’) with only three terminals appears to be a simple device, but it is in fact a very complex integrated circuit. It converts a varying input voltage into a constant ‘regulated’ output voltage. Voltage Regulators are available in a variety of outputs like 5V, 6V, 9V, 12V and 15V. The LM78XX series of voltage regulators are designed for positive input. For applications requiring negative input, the LM79XX series is used. Using a pair of ‘voltage-divider’ resistors can increase the output voltage of a regulator circuit.

It is not possible to obtain a voltage lower than the stated rating. You cannot use a 12V regulator to make a 5V power supply. Voltage regulators are very robust. These can withstand over-current draw due to short circuits and also over-heating. In both cases, the regulator will cut off before any damage occurs. The only way to destroy a regulator is to apply reverse voltage to its input. Reverse polarity destroys the regulator almost instantly. Fig: 3.3.11 shows voltage regulator.

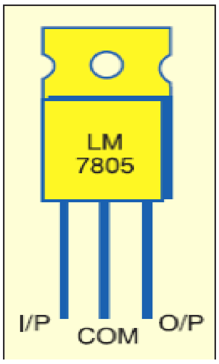


Fig 21: Voltage Regulator

Resistors:

A resistor is a two-terminal electronic component that produces a voltage across its terminals that is proportional to the electric current passing through it in accordance with Ohm's law:

$$V = IR$$

Resistors are elements of electrical networks and electronic circuits and are ubiquitous in most electronic equipment. Practical resistors can be made of various compounds and films, as well as resistance wire (wire made of a high-resistivity alloy, such as nickel/ chrome).

The primary characteristics of a resistor are the resistance, the tolerance, maximum working voltage and the power rating. Other characteristics include temperature coefficient, noise, and inductance. Less well-known is critical resistance, the value below which power dissipation limits the maximum permitted current flow, and above which the limit is applied voltage. Critical resistance is determined by the design, materials and dimensions of the resistor.

Resistors can be made to control the flow of current, to work as Voltage dividers, to dissipate power and it can shape electrical waves when used in combination of other components. Basic unit is ohms.

Theory of operation:

Ohm's law:

The behaviour of an ideal resistor is dictated by the relationship specified in Ohm's law:

$$V = IR$$

Ohm's law states that the voltage (V) across a resistor is proportional to the current (I) through it where the constant of proportionality is the resistance (R).

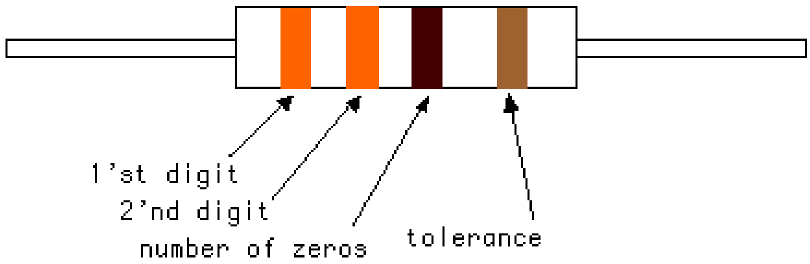
Power dissipation:

The power dissipated by a resistor (or the equivalent resistance of a resistor network) is calculated using the following:

$$P = I^2 R = IV = \frac{V^2}{R}$$

Fig 22: Resistor

Resistor color Coding



Digit	color	Tolerance	color
0	Black	20%	nothing
1	Brown	10%	Silver
2	Red	5%	Gold
3	Orange	2%	Red
4	Yellow	1%	Brown
5	Green		
6	Blue		
7	Violet		
8	Grey		
9	White		

Fig 23: Color Bands In Resistor

3.1.2 HC-SR04 Ultrasonic Distance Measurement Sensor:

Ultrasonic transducers and ultrasonic sensors are devices that generate or sense ultrasound energy. They can be divided into three broad categories: transmitters, receivers and transceivers. Transmitters convert electrical signals into ultrasound, receivers convert ultrasound into electrical signals, and transceivers can both transmit and receive ultrasound.

In a similar way to radar and sonar, ultrasonic transducers are used in systems which evaluate targets by interpreting the reflected signals. For example, by measuring the time between sending a signal and receiving an echo the distance of an object can be calculated. Passive ultrasonic sensors are basically microphones that detect ultrasonic noise that is present under certain conditions.

The design of transducer can vary greatly depending on its use: those used for medical diagnostic purposes, for example the range-finding applications listed above, are generally lower power than those used for the purpose of changing the properties of the liquid medium, or targets immersed in the liquid medium, through chemical, biological or physical (e.g. erosive) effects. The latter class include ultrasonic probes and ultrasonic baths, which apply ultrasonic energy to agitate particles, clean, erode, or disrupt biological cells, in a wide range of materials.

Fig 24: An Ultrasonic Distance/Depth Measuring Sensor (HC-SR04)

3.2 SOFTWARE DESCRIPTION:

3.2.0 Arduino IDE (Integrated Development Environment):

Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module. It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process. It is easily available for operating systems like Mac, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.

A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more. Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code. The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.

The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module. This environment supports both C and C++ languages.

What exactly is Arduino?

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Why use Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50

Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

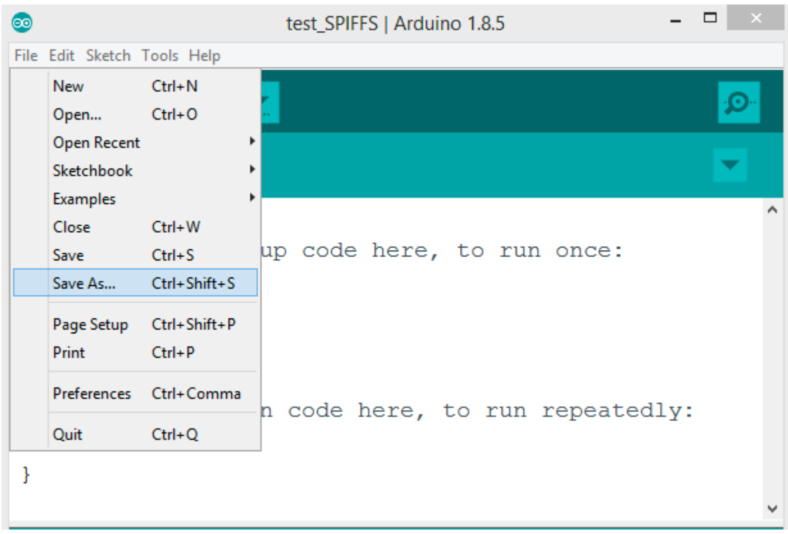
Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

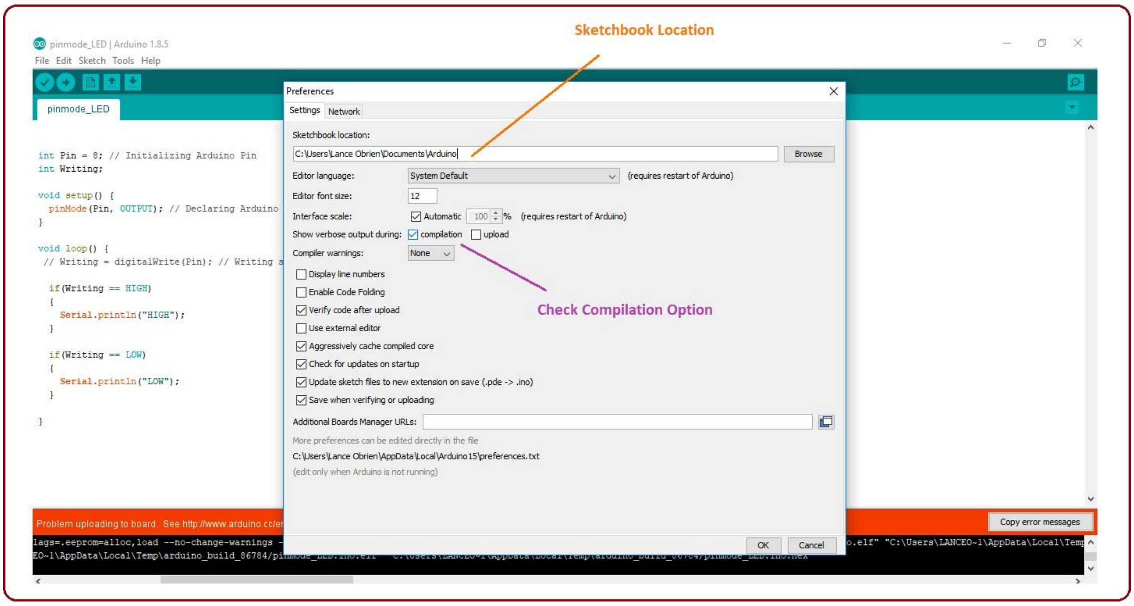
Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

The IDE environment is mainly distributed into three sections:

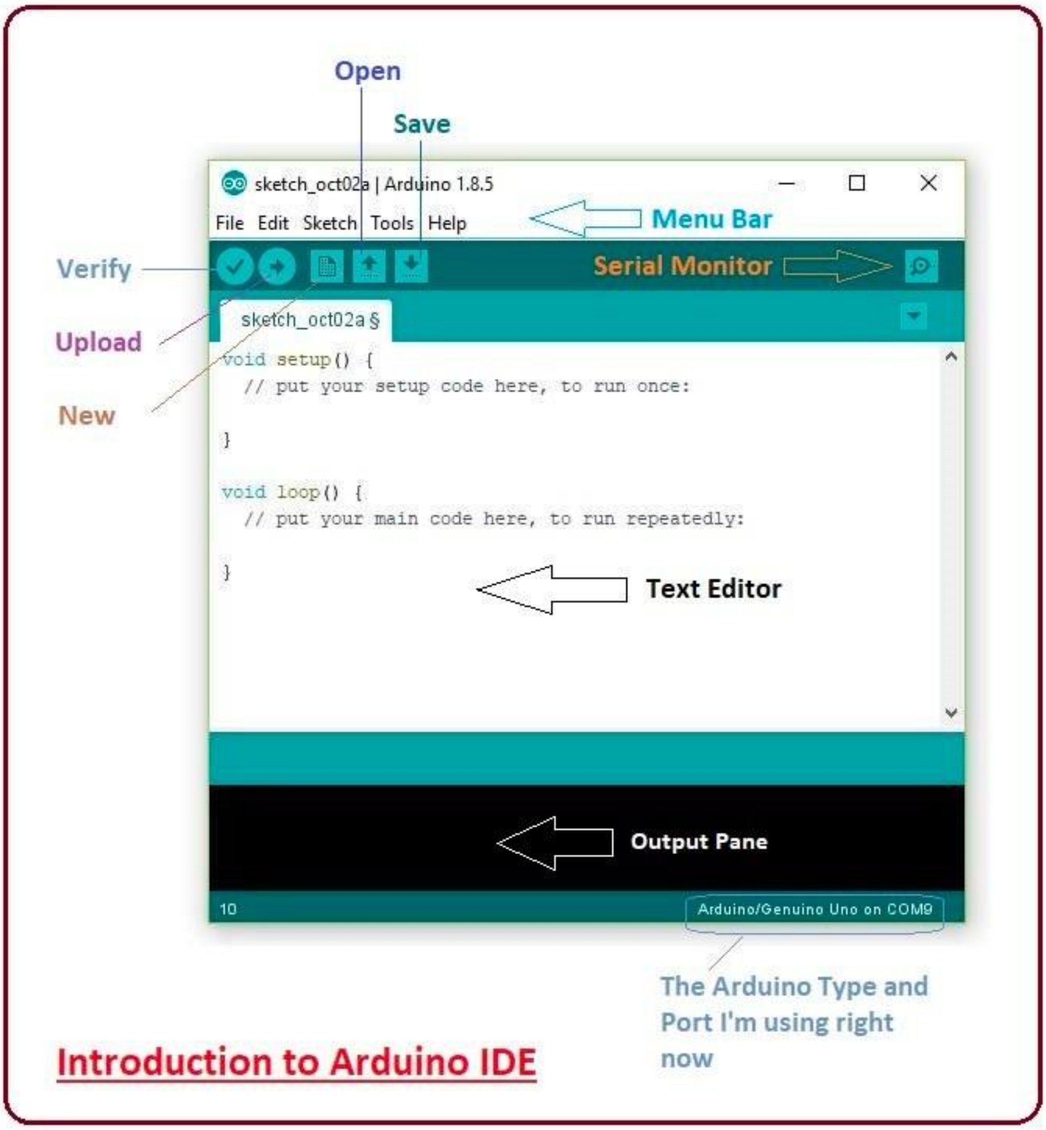
- Menu Bar
- Text Editor
- Output Pane
- File: We can open a new window for writing the code or open an existing one. Following table shows the number of further subdivisions the file option is categorised into.



As we go to the preference section and check the compilation section, the Output Pane will show the code compilation as we click the upload button.



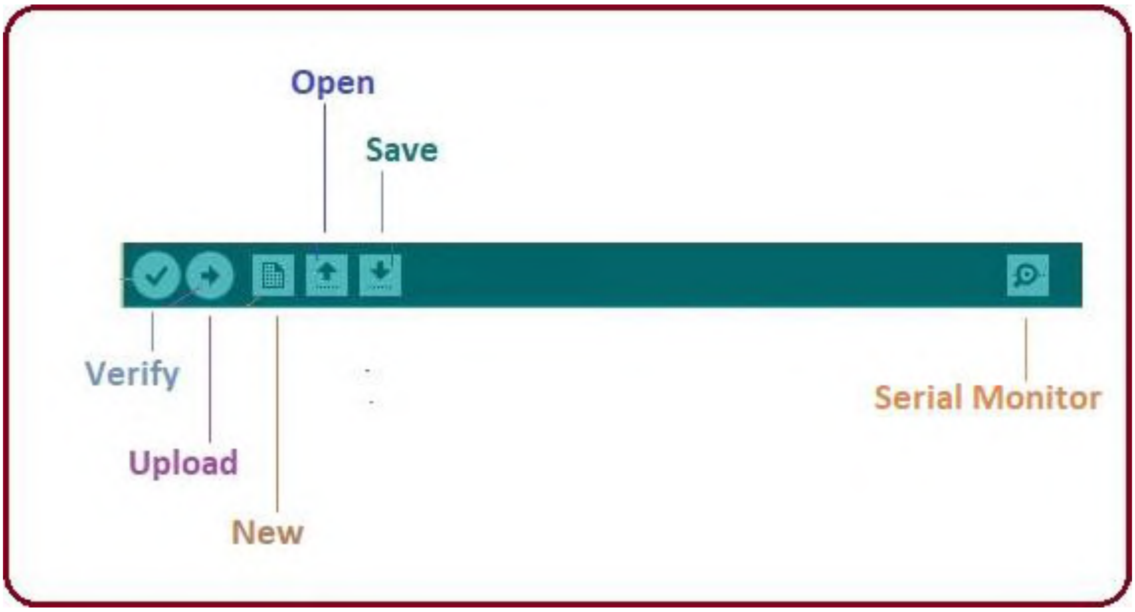
- And at the end of compilation, it will show us the hex file it has generated for the recent sketch that will send to the Arduino Board for the specific task we aim to achieve.



- **Edit** - Used for copying and pasting the code with further modification for font
- **Sketch** - For compiling and programming

- **Tools** - Mainly used for testing projects. The Programmer section in this panel is used for burning a boot-loader to the new microcontroller.
- **Help** - In case you are feeling skeptical about software, complete help is available from getting started to troubleshooting.

The **Six Buttons** appearing under the Menu tab are connected with the running program as follow.



The check mark appearing in the circular button is used to verify the code. Click this once you have written your code.

The arrow key will upload and transfer the required code to the Arduino board.

The dotted paper is used for creating a new file.

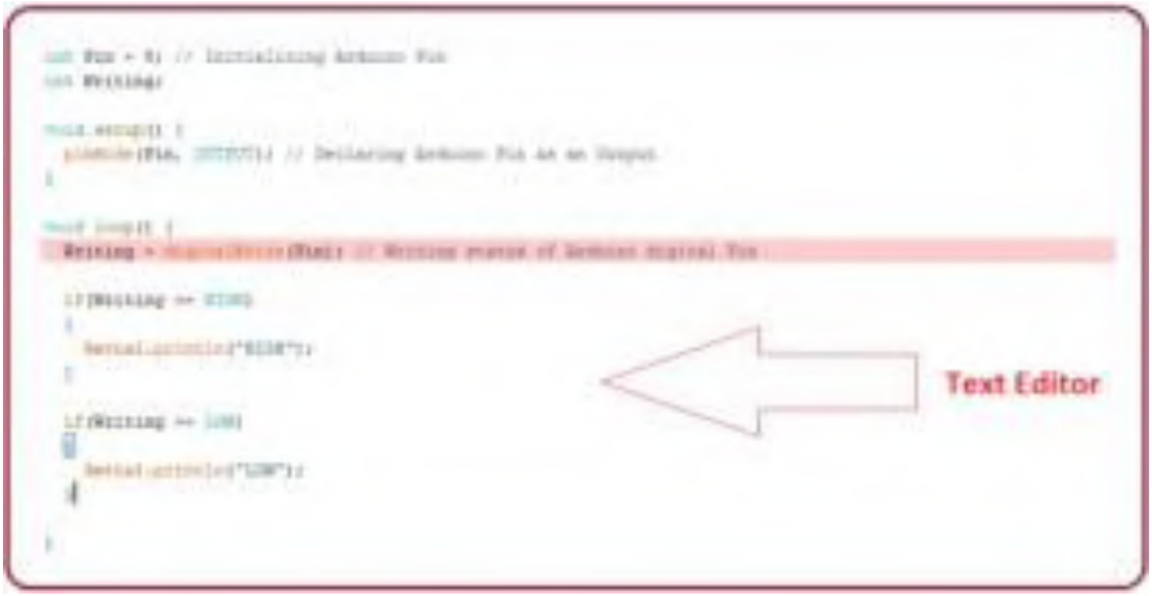
The upward arrow is reserved for opening an existing Arduino project.

The downward arrow is used to save the current running code.

The button appearing on the top right corner is a **Serial Monitor** - A separate pop-up window that acts as an independent terminal and plays a vital role for sending and receiving the Serial Data. You can also go to the Tools panel and select Serial Monitor, or pressing Ctrl+Shift+M all at once will open it instantly. The Serial Monitor will actually help to debug the written Sketches where you can get a hold of how your program is operating. Your Arduino Module should be connected to your computer by USB cable in order to activate the Serial Monitor.

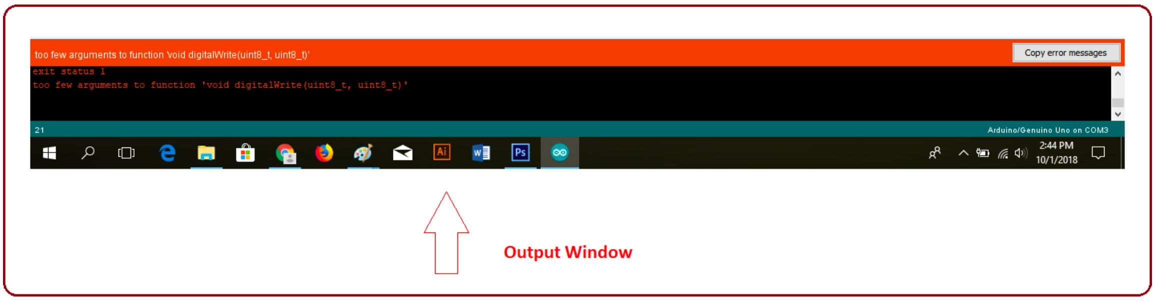
We need to select the baud rate of the Arduino Board we are using right now. For my Arduino Uno Baud Rate is 9600, as we write the following code and click the Serial Monitor, the output will show as the image below.

The main screen below the Menu bard is known as a simple text editor used for writing the required code.



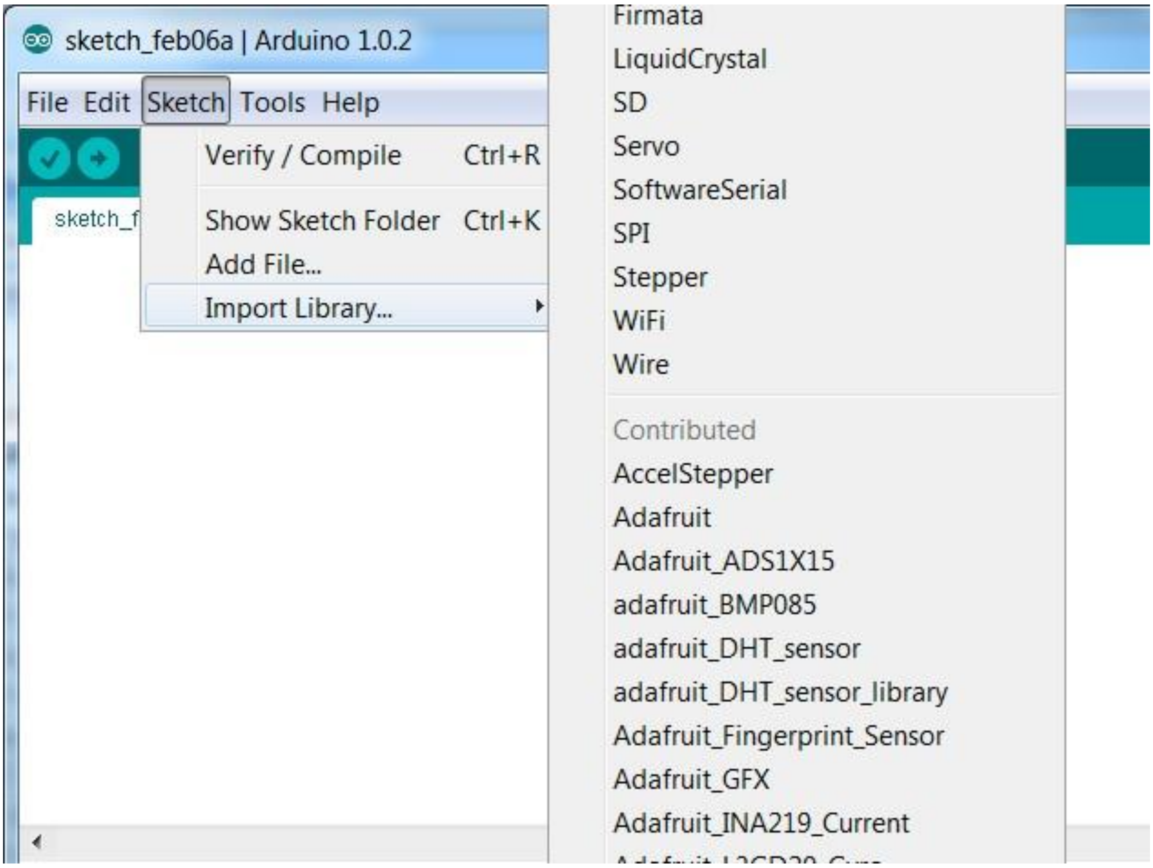
The bottom of the main screen is described as an Output Pane that mainly highlights the compilation status of the running code: the memory used by the code, and errors occurred in the program. You need to fix those errors before you intend to upload the hex file into your Arduino

Module.



More or less, Arduino C language works similar to the regular C language used for any embedded system microcontroller, however, there are some dedicated libraries used for calling and executing specific functions on the board.**Libraries**

Libraries are very useful for adding the extra functionality into the Arduino Module. There is a list of libraries you can add by clicking the Sketch button in the menu bar and going to Include Library.



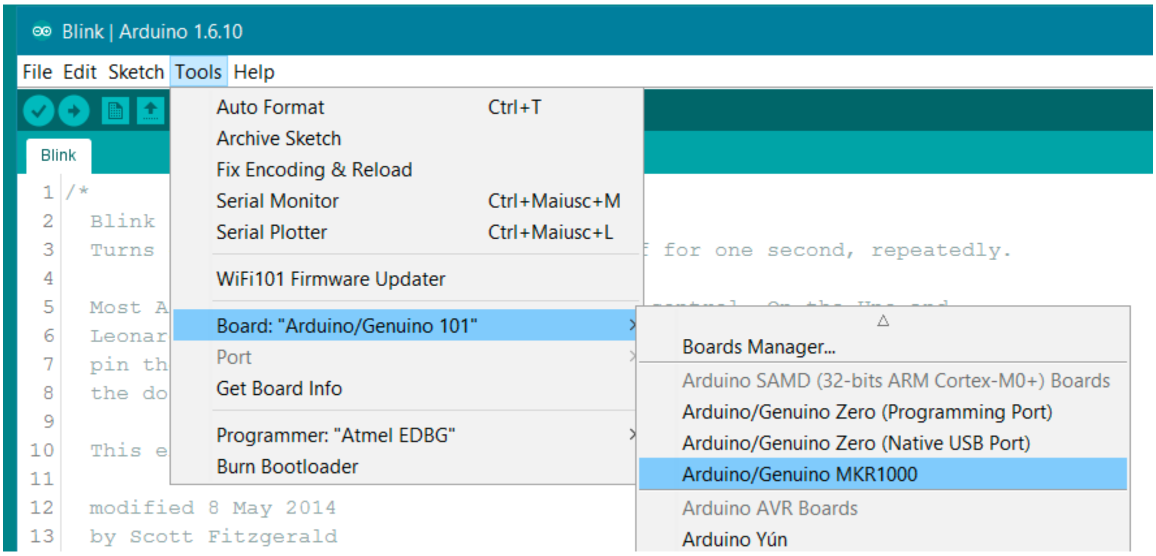
As we click the Include Library and Add the respective library it will on the top of the sketch with a #include sign. Suppose, I Include the EEPROM library, it will appear on the text editor as

#include <EEPROM.h>.

Most of the libraries are preinstalled and come with the Arduino software. However, we can also download them from the external sources.**Making Pins Input or Output**

The digitalRead and digitalWrite commands are used for addressing and making the Arduino pins as an input and output respectively. These commands are text sensitive i.e. we need to write them down the exact way they are given like digitalWrite starting with small "d" and write with capital "W". Writing it down with Digitalwrite or digitalwrite won't be calling or addressing any function.**How to Select the Board**

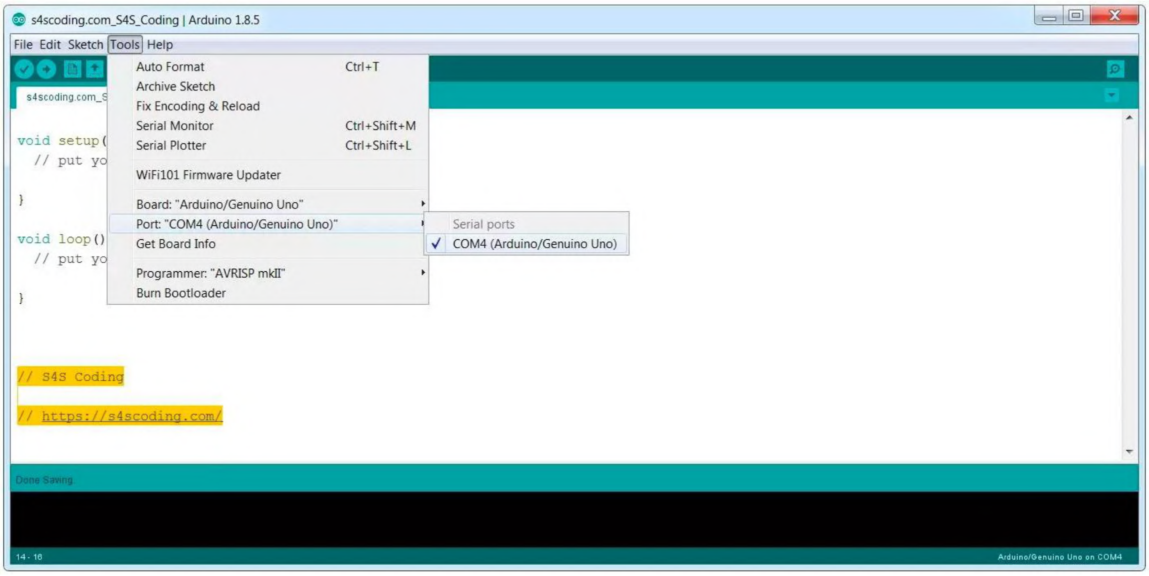
In order to upload the sketch, we need to select the relevant board we are using and the ports for that operating system. As we click the Tools on the Menu, it will open like the figure below.



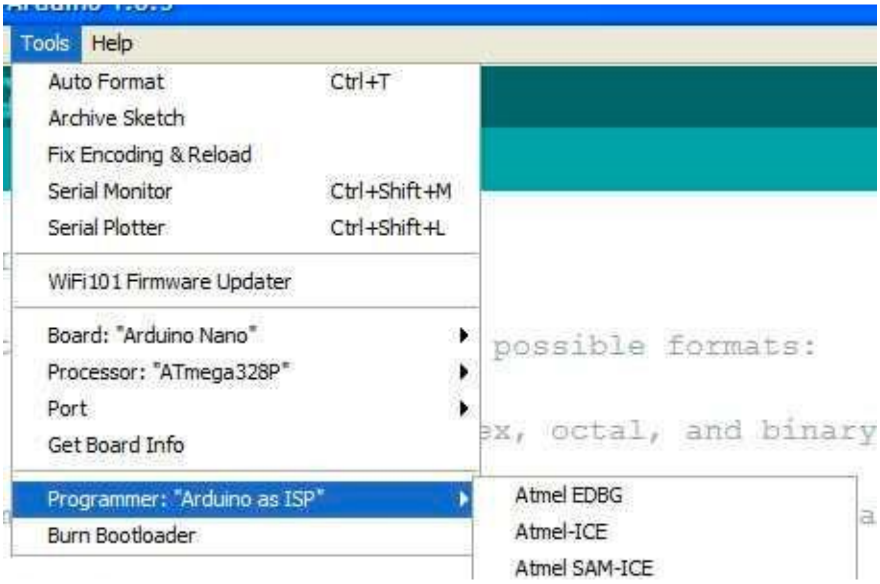
- Just go to the "Board" section and select the board we aim to work on. Similarly, COM1, COM2, COM4, COM5, COM7 or higher are reserved for the serial and USB board. We can look for the USB serial device in the ports section of the Windows Device Manager.

Following figure shows the COM4, indicating the Arduino Uno with COM4 port at the right bottom corner of the screen.

- After correct selection of both Board and Serial Port, click the verify and then upload button appearing in the upper left corner of the six button section or we can go to the Sketch section and press verify/compile and then upload.



- The sketch is written in the text editor and is then saved with the file extension .ino.



It is important to note that the recent Arduino Modules will reset automatically as we compile and press the upload button the IDE software, however, older version may require the physical reset on the board.

- Once we upload the code, TX and RX LEDs will blink on the board, indicating the desired program is running successfully.
- As we go to the Tools section, we will find a boot-loader at the end. It is very helpful to burn the code directly into the controller, setting us free from buying the external burner to burn the required code.

When we buy the new Arduino Module, the bootloader is already installed inside the controller. However, if we intend to buy a controller and put in the Arduino module, we need to burn the bootloader again inside the controller by going to the Tools section and selecting the burn bootloader.

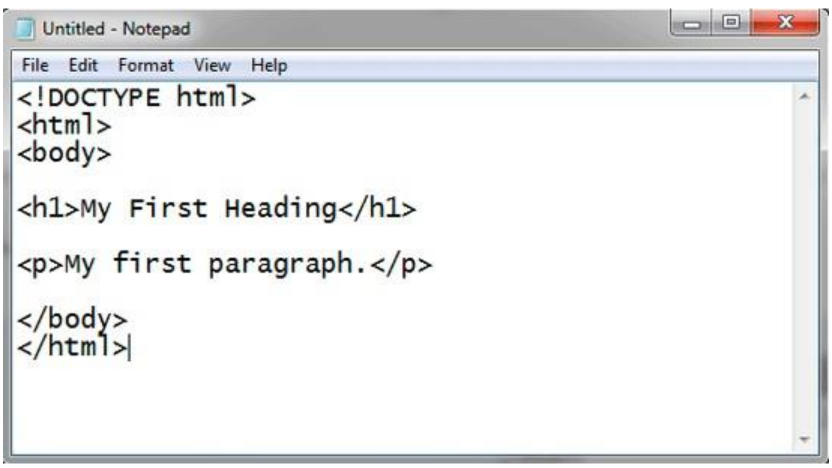
3.2.1 HTML, CSS, and PHP:

HTML (Hyper Text Markup Language):

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.



HTML can embed programs written in a scripting language such as JavaScript, which affects the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the

HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

Markup:

HTML markup consists of several key components, including those called tags (and their attributes), character-based data types, character references and entity references. HTML tags most commonly come in pairs like <h1> and </h1>, although some represent empty elements and so are unpaired, for example <img>. The first tag in such a pair is the start tag, and the second is the end tag (they are also called opening tags and closing tags).

Another important component is the HTML document type declaration, which triggers standards mode rendering.

The following is an example of the classic "Hello, World!" program:

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is a title</title>
  </head>
  <body>
    <div>
        <p>Hello world!</p>
    </div>
  </body>
</html>
```

The text between <html> and </html> describes the web page, and the text between <body> and </body> is the visible page content. The markup text <title>This is a title</title> defines the browser page title, and the tag <div> defines a division of the page used for easy styling.

The Document Type Declaration <!DOCTYPE html> is for HTML5. If a declaration is not included, various browsers will revert to "quirks mode" for rendering.

Elements:

HTML documents imply a structure of nested HTML elements. These are indicated in the document by HTML tags, enclosed in angle brackets thus: <p>.

In the simple, general case, the extent of an element is indicated by a pair of tags: a "start tag" <p> and "end tag" </p>. The text content of the element, if any, is placed between these tags.

Tags may also enclose further tag markup between the start and end, including a mixture of tags and text. This indicates further (nested) elements, as children of the parent element.

The start tag may also include attributes within the tag. These indicate other information, such as identifiers for sections within the document, identifiers used to bind style information to the presentation of the document, and for some tags such as the <img> used to embed images, the reference to the image resource in the format like this: <img src="example.com/example.jpg">

Some elements, such as the line break <br>, or <br /> do not permit any embedded content, either text or further tags. These require only a single empty tag (akin to a start tag) and do not use an end tag.

Many tags, particularly the closing end tag for the very commonly used paragraph element <p>, are optional. An HTML browser or other agent can infer the closure for the end of an element from the context and the structural rules defined by the HTML standard. These rules are complex and not widely understood by most HTML coders.

The general form of an HTML element is therefore: <tag attribute1="value1" attribute2="value2">"content"</tag>. Some HTML elements are defined as empty elements and take the form <tag attribute1="value1" attribute2="value2">. Empty elements may enclose no content, for instance, the <br> tag or the inline <img> tag. The name of an HTML element is the name used in the tags. Note that the end tag's name is preceded by a slash character, /, and that in empty elements the end tag is neither required nor allowed. If attributes are not mentioned, default values are used in each case.

Paragraphs:

`<p>Paragraph 1</p> <p>Paragraph 2</p>`

Line breaks: `<br>`. The difference between `<br>` and `<p>` is that `<br>` breaks a line without altering the semantic structure of the page, whereas `<p>` sections the page into paragraphs. The element `<br>` is an empty element in that, although it may have attributes, it can take no content and it may not have an end tag.

```
<p>This <br> is a paragraph <br> with <br> line breaks</p>
```

This is a link in HTML. To create a link the `<a>` tag is used. The href attribute holds the URL address of the link.

```
<a href="https://www.wikipedia.org/">A link to Wikipedia!</a>
```

Inputs:

There are many possible ways a user can give input/s like:

```
<input type="text" /> <!-- This is for text input -->

<input type="file" /> <!-- This is for uploading files -->

<input type="checkbox" /> <!-- This is for checkboxes —>
```

Comments:

```
<!-- This is a comment -->
```

Comments can help in the understanding of the markup and do not display in the webpage.

There are several types of markup elements used in HTML:

Structural markup indicates the purpose of text

For example, <h2>Golf</h2> establishes "Golf" as a second-level heading. Structural markup does not denote any specific rendering, but most web browsers have default styles for element formatting. Content may be further styled using Cascading Style Sheets (CSS). [72]

Presentational markup indicates the appearance of the text, regardless of its purpose

For example, bold text indicates that visual output devices should render "boldface" in bold text, but gives little indication what devices that are unable to do this (such as aural devices that read the text aloud) should do. In the case of both bold text and <i>italic text</i>, there are other elements that may have equivalent visual renderings but that are more semantic in nature, such as strong text and emphasized text respectively. It is easier to see how an aural user agent should interpret the latter two elements. However, they are not equivalent to their presentational counterparts: it would be undesirable for a screen-reader to emphasize the name of a book, for instance, but on a screen such a name would be italicized. Most presentational markup elements have become deprecated under the HTML 4.0 specification in favor of using CSS for styling.

Hypertext markup makes parts of a document into links to other documents

An anchor element creates a hyperlink in the document and its href attribute sets the link's target URL. For example, the HTML markup Wikipedia, will render the word "Wikipedia" as a hyperlink. To render an image as a hyperlink, an img element is inserted as content into the a element. Like br, img is an empty element with attributes but no content or closing tag. <img src="image.gif" alt="descriptive text" width="50" height="50" border="0">.

Attributes:

Most of the attributes of an element are name-value pairs, separated by = and written within the start tag of an element after the element's name. The value may be enclosed in single or double quotes, although values consisting of certain characters can be left unquoted in HTML (but not XHTML). Leaving attribute values unquoted is considered unsafe. In contrast with name-value pair attributes, there are some attributes that affect the element simply by their presence in the start tag of the element,[6] like the ismap attribute for the img element.

There are several common attributes that may appear in many elements :

- The id attribute provides a document-wide unique identifier for an element. This is used to identify the element so that stylesheets can alter its presentational properties, and scripts may alter, animate or delete its contents or presentation. Appended to the URL of the page, it provides a globally unique identifier for the element, typically a sub-section of the page. For example, the ID "Attributes" in https://en.wikipedia.org/wiki/HTML#Attributes.

- The class attribute provides a way of classifying similar elements. This can be used for semantic or presentation purposes. For example, an HTML document might semantically use the designation <class="notation"> to indicate that all elements with this class value are subordinate to the main text of the document. In presentation, such elements might be gathered together and presented as footnotes on a page instead of appearing in the place where they occur in the HTML source. Class attributes are used semantically in microformats. Multiple class values may be specified; for example <class="notation important"> puts the element into both the notation and the important classes.
- An author may use the style attribute to assign presentational properties to a particular element. It is considered better practice to use an element's id or class attributes to select the element from within a stylesheet, though sometimes this can be too cumbersome for a simple, specific, or ad hoc styling.
- The title attribute is used to attach subtextual explanation to an element. In most browsers this attribute is displayed as a tooltip.
- The lang attribute identifies the natural language of the element's contents, which may be different from that of the rest of the document. For example, in an English-language document:<p>Oh well, c'est la vie, as they say in France.</p>

The abbreviation element, abbr, can be used to demonstrate some of these attributes:

<abbr id="anId" class="jargon" style="color:purple;" title="Hypertext Markup Language">HTML</abbr>

This example displays as HTML; in most browsers, pointing the cursor at the abbreviation should display the title text "Hypertext Markup Language."

Most elements take the language-related attribute dir to specify text direction, such as with "rtl" for right-to-left text in, for example, Arabic, Persian or Hebrew.

Data types:

HTML defines several data types for element content, such as script data and stylesheet data, and a plethora of types for attribute values, including IDs, names, URIs, numbers, units of length, languages, media descriptors, colors, character encodings, dates and times, and so on. All of these data types are specializations of character data.

Document type declaration:

HTML documents are required to start with a Document Type Declaration (informally, a "doctype"). In browsers, the doctype helps to define the rendering mode—particularly whether to use quirks mode.

The original purpose of the doctype was to enable parsing and validation of HTML documents by SGML tools based on the Document Type Definition (DTD). The DTD to which the DOCTYPE refers contains a machine-readable grammar specifying the permitted and prohibited content for a document conforming to such a DTD. Browsers, on the other hand, do not implement HTML as an application of SGML and by consequence do not read the DTD.

HTML5 does not define a DTD; therefore, in HTML5 the doctype declaration is simpler and shorter:

```
<!DOCTYPE html>
```

An example of an HTML 4 doctype:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "https://www.w3.org/TR/html4/strict.dtd">
```

This declaration references the DTD for the "strict" version of HTML 4.01. SGML-based validators read the DTD in order to properly parse the document and to perform validation. In modern browsers, a valid doctype activates standards mode as opposed to quirks mode.

In addition, HTML 4.01 provides Transitional and Frameset DTDs, as explained below. Transitional type is the most inclusive, incorporating current tags as well as older or "deprecated" tags, with the Strict DTD excluding deprecated tags. Frameset has all tags necessary to make frames on a page along with the tags included in transitional type.

Semantic HTML:

Semantic HTML is a way of writing HTML that emphasizes the meaning of the encoded information over its presentation (look). HTML has included semantic markup from its inception,[82] but has also included presentational markup, such as `<font>`, `<i>` and `<center>` tags. There are also the semantically neutral span and div tags. Since the late 1990s, when Cascading Style Sheets were beginning to work in most browsers, web authors have been encouraged to avoid the use of presentational HTML markup with a view to the separation of presentation and content.[83]

In a 2001 discussion of the Semantic Web, Tim Berners-Lee and others gave examples of ways in which intelligent software "agents" may one day automatically crawl the web and find, filter and correlate previously unrelated, published facts for the benefit of human users.[84] Such agents are not commonplace even now, but some of the ideas of Web 2.0, mashups and price comparison websites may be coming close. The main difference between these web application hybrids and Berners-Lee's semantic agents lies in the fact that the current aggregation and hybridization of information is usually designed in by web developers, who already know the web locations and the API semantics of the specific data they wish to mash, compare and combine.

An important type of web agent that does crawl and read web pages automatically, without prior knowledge of what it might find, is the web crawler or search-engine spider. These software agents are dependent on the semantic clarity of web pages they find as they use various techniques and algorithms to read and index millions of web pages a day and provide web users with search facilities without which the World Wide Web's usefulness would be greatly reduced.

In order for search-engine spiders to be able to rate the significance of pieces of text they find in HTML documents, and also for those creating mashups and other hybrids as well as for more automated agents as they are developed, the semantic structures that exist in HTML need to be widely and uniformly applied to bring out the meaning of published text.

Presentational markup tags are deprecated in current HTML and XHTML recommendations. The majority of presentational features from previous versions of HTML are no longer allowed as they lead to poorer accessibility, higher cost of site maintenance, and larger document sizes.

Good semantic HTML also improves the accessibility of web documents (see also Web Content Accessibility Guidelines). For example, when a screen reader or audio browser can correctly ascertain the structure of a document, it will not waste the visually impaired user's time by reading out repeated or irrelevant information when it has been marked up correctly.

Delivery:

HTML documents can be delivered by the same means as any other computer file. However, they are most often delivered either by HTTP from a web server or by email.

HTTP:

The World Wide Web is composed primarily of HTML documents transmitted from web servers to web browsers using the Hypertext Transfer Protocol (HTTP). However, HTTP is used to serve images, sound, and other content, in addition to HTML. To allow the web browser to know how to handle each document it receives, other information is transmitted along with the document. This meta data usually includes the MIME type (e.g., text/ html or application/xhtml+xml) and the character encoding (see Character encoding in HTML).

In modern browsers, the MIME type that is sent with the HTML document may affect how the document is initially interpreted. A document sent with the XHTML MIME type is expected to be well-formed XML; syntax errors may cause the browser to fail to render it. The same document sent with the HTML MIME type might be displayed successfully, since some browsers are more lenient with HTML.

The W3C recommendations state that XHTML 1.0 documents that follow guidelines set forth in the recommendation's Appendix C may be labeled with either MIME Type. XHTML 1.1 also states that XHTML 1.1 documents should be labeled with either MIME type.

HTML e-mail:

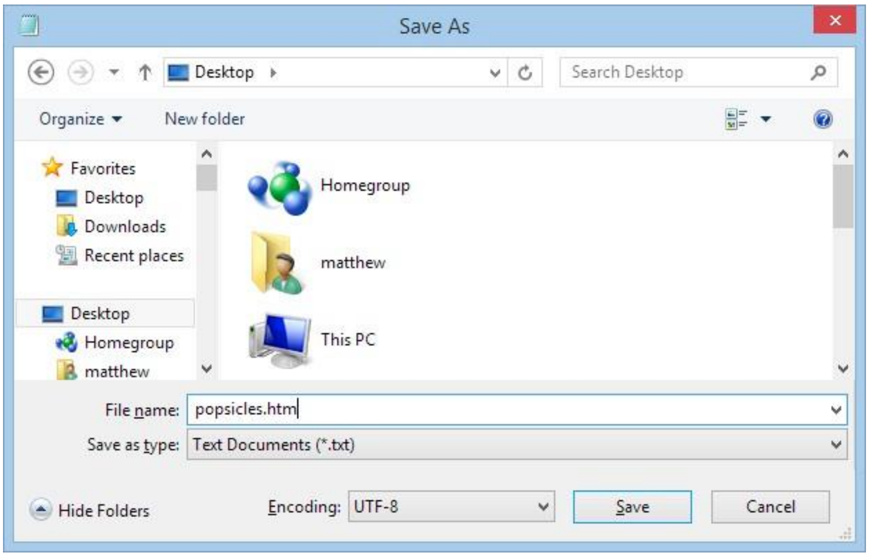
Most graphical email clients allow the use of a subset of HTML (often ill-defined) to provide formatting and semantic markup not available with plain text. This may include typographic information like coloured headings, emphasized and quoted text, inline images and diagrams. Many such clients include both a GUI editor for composing HTML e-mail messages and a rendering engine for displaying them. Use of HTML in e-mail is criticized by some because of compatibility issues, because it can help disguise phishing attacks, because of accessibility issues for blind or visually impaired people, because it can confuse spam filters and because the message size is larger than plain text.

Naming conventions:

The most common filename extension for files containing HTML is .html. A common abbreviation of this is .htm, which originated because some early operating systems and file systems, such as DOS and the limitations imposed by FAT data structure, limited file extensions to three letters.

HTML Application:

An HTML Application (HTA; file extension ".hta") is a Microsoft Windows application that uses HTML and Dynamic HTML in a browser to provide the application's graphical interface. A regular HTML file is confined to the security model of the web browser's security, communicating only to web servers and manipulating only web page objects and site cookies. An HTA runs as a fully trusted application and therefore has more privileges, like creation/editing/removal of files and Windows Registry entries. Because they operate outside the browser's security model, HTAs cannot be executed via HTTP, but must be downloaded (just like an EXE file) and executed from local file system.



CSS (Cascaded Style Sheets):

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

Example:

```
body {
  background-color: lightblue;
}

h1 {
```

```
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```

PHP (Hypertext Preprocessor):

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

Example:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Example</title>
    </head>
    <body>
        <?php
          echo "Hi, I'm a PHP script!”;
        ?>
    </body>
</html>
```

Instead of lots of commands to output HTML (as seen in C or Perl), PHP pages contain HTML with embedded code that does "something" (in this case, output "Hi, I'm a PHP script!"). The PHP code is enclosed in special start and end processing instructions <?php and ?> that allow you to jump into and out of "PHP mode."

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer. Don't be afraid reading the long list of PHP's features. You can jump in, in a short time, and start writing simple scripts in a few hours.

Although PHP's development is focused on server-side scripting, you can do much more with it. PHP is mainly focused on server-side scripting, so you can do anything any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies. But PHP can do much more. There are three main areas where PHP scripts are used.

- **Server-side scripting:** This is the most traditional and main target field for PHP. You need three things to make this work: the PHP parser (CGI or server module), a web server and a web browser. You need to run the web server, with a connected PHP installation. You can access the PHP program output with a web browser, viewing the PHP page through the server. All these can run on your home machine if you are just experimenting with PHP programming.

- **Command line scripting:** You can make a PHP script to run it without any server or browser. You only need the PHP parser to use it this way. This type of usage is ideal for scripts regularly executed using cron (on *nix or Linux) or Task Scheduler (on Windows). These scripts can also be used for simple text processing tasks.

- **Writing desktop applications:** PHP is probably not the very best language to create a desktop application with a graphical user interface, but if you know PHP very well, and would like to use some advanced PHP features in your client-side applications you can also use PHP-GTK to write such programs. You also have the ability to write cross-platform applications this way. PHP-GTK is an extension to PHP, not available in the main distribution.

PHP can be used on all major operating systems, including Linux, many Unix variants (including HP-UX, Solaris and OpenBSD), Microsoft Windows, macOS, RISC OS, and probably others. PHP also has support for most of the web servers today. This includes Apache, IIS, and many others. And this includes any web server that can utilise the FastCGI PHP binary, like lighttpd and nginx. PHP works as either a module, or as a CGI processor.

So with PHP, you have the freedom of choosing an operating system and a web server. Furthermore, you also have the choice of using procedural programming or object oriented programming (OOP), or a mixture of them both.

With PHP you are not limited to output HTML. PHP's abilities includes outputting images, PDF files and even Flash movies (using libswf and Ming) generated on the fly. You can also output easily any text, such as XHTML and any other XML file. PHP can autogenerate these files, and save them in the file system, instead of printing it out, forming a server-side cache for your dynamic content.

One of the strongest and most significant features in PHP is its support for a wide range of databases. Writing a database-enabled web page is incredibly simple using one of the database specific extensions (e.g., for mysql), or using an abstraction layer like PDO, or connect to any database supporting the Open Database Connection standard via the ODBC extension. Other databases may utilise cURL or sockets, like CouchDB.

PHP also has support for talking to other services using protocols such as LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (on Windows) and countless others. You can also open raw network sockets and interact using any other protocol. PHP has support for the WDDX complex data exchange between virtually all Web programming languages. Talking about interconnection, PHP has support for instantiation of Java objects and using them transparently as PHP objects.

PHP has useful text processing features, which includes the Perl compatible regular expressions (PCRE), and many extensions and tools to parse and access XML documents. PHP standardises all of the XML extensions on the solid base of libxml2, and extends the feature set adding SimpleXML, XMLReader and XMLWriter support.

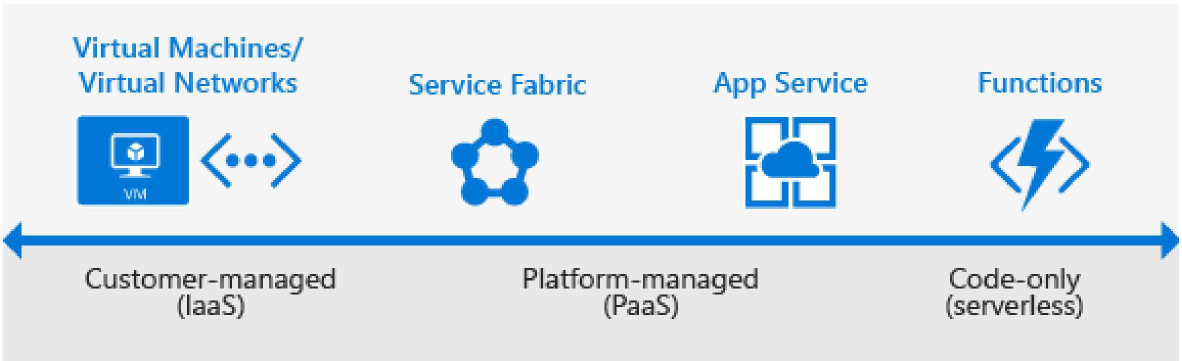
And many other interesting extensions exist, which are categorised both alphabetically and by category. And there are additional PECL extensions that may or may not be documented within the PHP manual itself, like XDebug.

3.2.2 Azure Cloud:

Azure is a complete cloud platform that can host your existing applications and streamline new application development. Azure can even enhance on-premises applications. Azure integrates the cloud services that you need to develop, test, deploy, and manage your applications, all while taking advantage of the efficiencies of cloud computing. By hosting your applications in Azure, you can start small and easily scale your application as your customer demand grows. Azure also offers the reliability that's needed for high-availability applications, even including failover between different regions. The Azure portal lets you easily manage all your Azure services. You can also manage your services programmatically by using service-specific APIs and templates.

Application hosting:

Azure provides several cloud-based compute offerings to run your application so that you don't have to worry about the infrastructure details. You can easily scale up or scale out your resources as your application usage grows. Azure offers services that support your application development and hosting needs. Azure provides Infrastructure as a Service (IaaS) to give you full control over your application hosting. Azure's Platform as a Service (PaaS) offerings provide the fully managed services needed to power your apps. There's even true server-less hosting in Azure where all you need to do is write your code.



Azure App Service:

When you want the quickest path to publish your web-based projects, consider Azure App Service. App Service makes it easy to extend your web apps to support your mobile clients and publish easily consumed REST APIs. This platform provides authentication by using social providers, traffic-based autoscaling, testing in production, and continuous and container-based deployments. You can create web apps, mobile app back ends, and API apps. Because all three app types share the App Service runtime, you can host a website, support mobile clients, and expose your APIs in Azure, all from the same project or solution.

App Service has been designed with DevOps in mind. It supports various tools for publishing and continuous integration deployments. These tools include GitHub webhooks, Jenkins, Azure DevOps, TeamCity, and others.

When to use: Use App Service when you're migrating existing web applications to Azure, and when you need a fully-managed hosting platform for your web apps. You can also use App Service when you need to support mobile clients or expose REST APIs with your app.

Azure Virtual Machines:

As an Infrastructure as a Service (IaaS) provider, Azure lets you deploy to or migrate your application to either Windows or Linux VMs. Together with Azure Virtual Network, Azure Virtual Machines supports the deployment of Windows or Linux VMs to Azure. With VMs, you have total control over the configuration of the machine. When using VMs, you're responsible for all server software installation, configuration, maintenance, and operating system patches.

Because of the level of control that you have with VMs, you can run a wide range of server workloads on Azure that don't fit into a PaaS model. These workloads include database servers, Windows Server Active Directory, and Microsoft SharePoint.

Azure Functions (serverless):

Rather than worrying about building out and managing a whole application or the infrastructure to run your code, you could just write your code and have it run in response to events or on a schedule. Azure Functions is a "server-less"-style offering that lets you write just the code you need. With Functions, you can trigger code execution with HTTP requests, webhooks, cloud service events, or on a schedule. You can code in your development language of choice, such as C#, F#, Node.js, Python, or PHP. With consumption-based billing, you pay only for the time that your code executes, and Azure scales as needed.

Azure Service Fabric:

Azure Service Fabric is a distributed systems platform. This platform makes it easy to build, package, deploy, and manage scalable and reliable micro-services. It also provides comprehensive application management capabilities such as:

- Provisioning
- Deploying
- Monitoring

- Upgrading/patching
- Deleting

Apps, which run on a shared pool of machines, can start small and scale to hundreds or thousands of machines as needed. Service Fabric supports WebAPI with Open Web Interface for .NET (OWIN) and ASP.NET Core. It provides SDKs for building services on Linux in both .NET Core and Java.

Azure Spring Cloud:

Azure Spring Cloud is a server-less micro-services platform that enables you to build, deploy, scale and monitor your applications in the cloud. Use Spring Cloud to bring modern micro-service patterns to Spring Boot apps, eliminating boilerplate code to quickly build robust Java apps.

- Leverage managed versions of Spring Cloud Service Discovery and Config Server, while we ensure those critical components are running in optimum conditions.
- Focus on building your business logic and we will take care of your service runtime with security patches, compliance standards and high availability.
- Manage application lifecycle (e.g.: deploy, start, stop, scale) on top of Azure Kubernetes Service.
- Easily bind connections between your apps and Azure services such as Azure Database for MySQL and Azure Cache for Redis.
- Monitor and troubleshoot microservices and applications using enterprise-grade unified monitoring tools that offer deep insights on application dependencies and operational telemetry.

Hosted storage and data access:

Most applications must store data, so however you decide to host your application in Azure, consider one or more of the following storage and data services.

- **Azure Cosmos DB:** A globally distributed, multi-model database service. This database enables you to elastically scale throughput and storage across any number of geographical regions with a comprehensive SLA.
- **Azure Storage:** Offers durable, highly available storage for blobs, queues, files, and other kinds of non-relational data. Storage provides the storage foundation for VMs.

- **Azure SQL Database:** An Azure-based version of the Microsoft SQL Server engine for storing relational tabular data in the cloud. SQL Database provides predictable performance, scalability with no downtime, business continuity, and data protection.

You can use Azure Data Factory to move existing on-premises data to Azure. If you aren't ready to move data to the cloud, Hybrid Connections in Azure App Service lets you connect your App Service hosted app to on-premises resources. You can also connect to Azure data and storage services from your on-premises applications.

Docker support:

Docker containers, a form of OS virtualization, let you deploy applications in a more efficient and predictable way. A containerized application works in production the same way as on your development and test systems. You can manage containers by using standard Docker tools. You can use your existing skills and popular open-source tools to deploy and manage container-based applications on Azure.

Azure provides several ways to use containers in your applications.

- **Azure Kubernetes Service:** Lets you create, configure, and manage a cluster of virtual machines that are preconfigured to run containerized applications. To learn more about Azure Kubernetes Service, see Azure Kubernetes Service introduction.
- **Docker Machine:** Lets you install and manage a Docker Engine on virtual hosts by using docker-machine commands.
- **Custom Docker image for App Service:** Lets you use Docker containers from a container registry or a customer container when you deploy a web app on Linux.

Authentication:

It's crucial to not only know who is using your applications, but also to prevent unauthorized access to your resources. Azure provides several ways to authenticate your app clients.

- **Azure Active Directory (Azure AD):** The Microsoft multitenant, cloud-based identity and access management service. You can add single-sign on (SSO) to your applications by integrating with Azure AD. You can access directory properties by using the Azure AD Graph API directly or the Microsoft Graph API. You can integrate with Azure AD support for the OAuth2.0 authorization framework and Open ID Connect by using native HTTP/REST endpoints and the multiplatform Azure AD authentication libraries.

- **App Service Authentication:** When you choose App Service to host your app, you also get built-in authentication support for Azure AD, along with social identity providers—including Facebook, Google, Microsoft, and Twitter.

Monitoring:

With your application up and running in Azure, you need to monitor performance, watch for issues, and see how customers are using your app. Azure provides several monitoring options.

- **Application Insights:** An Azure-hosted extensible analytics service that integrates with Visual Studio to monitor your live web applications. It gives you the data that you need to improve the performance and usability of your apps continuously. This improvement occurs whether you host your applications on Azure or not.
- **Azure Monitor:** A service that helps you to visualize, query, route, archive, and act on the metrics and logs that you generate with your Azure infrastructure and resources. Monitor is a single source for monitoring Azure resources and provides the data views that you see in the Azure portal.

DevOps integration:

Whether it's provisioning VMs or publishing your web apps with continuous integration, Azure integrates with most of the popular DevOps tools. You can work with the tools that you already have and maximize your existing experience with support for tools like:

- Jenkins
- GitHub
- Puppet
- Chef
- TeamCity

- Ansible
- Azure DevOps

Command-line interfaces and PowerShell:

Azure provides two ways to manage your applications and services from the command line. You can use tools like Bash, Terminal, the command prompt, or your command-line tool of choice. Usually, you can do the same tasks from the command line as in the Azure portal—such as creating and configuring virtual machines, virtual networks, web apps, and other services.

- **Azure Command-Line Interface (CLI):** Lets you connect to an Azure subscription and program various tasks against Azure resources from the command line.
- **Azure PowerShell:** Provides a set of modules with cmdlets that enable you to manage Azure resources by using Windows PowerShell.

Azure portal:

The Azure portal is a web-based application. You can use the Azure portal to create, manage, and remove Azure resources and services. It includes:

- A configurable dashboard
- Azure resource management tools
- Access to subscription settings and billing information. For more information, see the Azure portal overview.

REST APIs:

Azure is built on a set of REST APIs that support the Azure portal UI. Most of these REST APIs are also supported to let you programmatically provision and manage your Azure resources and applications from any Internet-enabled device.

APIs:

Along with REST APIs, many Azure services also let you programmatically manage resources from your applications by using platform-specific Azure SDKs, including SDKs for the following development platforms:

- .NET
- Node.js
- Java

- PHP
- Python
- Ruby
- Go

Services such as Mobile Apps and Azure Media Services provide client-side SDKs to let you access services from web and mobile client apps.

Azure Resource Manager:

Running your app on Azure likely involves working with multiple Azure services. These services follow the same life cycle and can be thought of as a logical unit. For example, a web app might use Web Apps, SQL Database, Storage, Azure Cache for Redis, and Azure Content Delivery Network services. Azure Resource Manager lets you work with the resources in your application as a group. You can deploy, update, or delete all the resources in a single, coordinated operation.

Along with logically grouping and managing related resources, Azure Resource Manager includes deployment capabilities that let you customize the deployment and configuration of related resources. For example, you can use Resource Manager deploy and configure an application. This application can consist of multiple virtual machines, a load balancer, and a database in Azure SQL Database as a single unit.

You develop these deployments by using an Azure Resource Manager template, which is a JSON-formatted document. Templates let you define a deployment and manage your applications by using declarative templates, rather than scripts. Your templates can work for different environments, such as testing, staging, and production. For example, you can use templates to add a button to a GitHub repo that deploys the code in the repo to a set of Azure services with a single click.

When to use: Use Resource Manager templates when you want a template-based deployment for your app that you can manage programmatically by using REST APIs, the Azure CLI, and Azure PowerShell.

3.2.3 Azure Databases:

Azure SQL:

Azure SQL is a family of managed, secure, and intelligent products that use the SQL Server database engine in the Azure cloud.

- **Azure SQL Database:** Support modern cloud applications on an intelligent, managed database service, that includes serverless compute.
- **Azure SQL Managed Instance:** Modernize your existing SQL Server applications at scale with an intelligent fully managed instance as a service, with almost 100% feature parity with the SQL Server database engine. Best for most migrations to the cloud.
- **SQL Server on Azure VMs:** Lift-and-shift your SQL Server workloads with ease and maintain 100% SQL Server compatibility and operating system-level access.

Azure SQL is built upon the familiar SQL Server engine, so you can migrate applications with ease and continue to use the tools, languages, and resources you're familiar with. Your skills and experience transfer to the cloud, so you can do even more with what you already have.

Learn how each product fits into Microsoft's Azure SQL data platform to match the right option for your business requirements. Whether you prioritize cost savings or minimal administration, this article can help you decide which approach delivers against the business requirements you care about most.

Overview:

In today's data-driven world, driving digital transformation increasingly depends on our ability to manage massive amounts of data and harness its potential. But today's data estates are increasingly complex, with data hosted on-premises, in the cloud, or at the edge of the network. Developers who are building intelligent and immersive applications can find themselves constrained by limitations that can ultimately impact their experience. Limitations arising from incompatible platforms, inadequate data security, insufficient resources and price-performance barriers create complexity that can inhibit app modernisation and development.

One of the first things to understand in any discussion of Azure versus on-premises SQL Server databases is that you can use it all. Microsoft's data platform leverages SQL Server technology and makes it available across physical on-premises machines, private cloud environments, third-party hosted private cloud environments, and the public cloud.

Fully managed and always up to date:

Spend more time innovating and less time patching, updating, and backing up your databases. Azure is the only cloud with evergreen SQL that automatically applies the latest updates and patches so that your databases are always up to date—eliminating end-of-support hassle. Even complex tasks like performance tuning, high availability, disaster recovery, and backups are automated, freeing you to focus on applications.

Protect your data with built-in intelligent security:

Azure constantly monitors your data for threats. With Azure SQL, you can:

- Remediate potential threats in real time with intelligent advanced threat detection and proactive vulnerability assessment alerts.
- Get industry-leading, multi-layered protection with built-in security controls including T-SQL, authentication, networking, and key management.
- Take advantage of the most comprehensive compliance coverage of any cloud database service.

Azure SQL Database:

Azure SQL Database is a relational database-as-a-service (DBaaS) hosted in Azure that falls into the industry category of *Platform-as-a-Service (PaaS)*.

- Best for modern cloud applications that want to use the latest stable SQL Server features and have time constraints in development and marketing.
- A fully managed SQL Server database engine, based on the latest stable Enterprise Edition of SQL Server. SQL Database has two deployment options built on standardised hardware and software that is owned, hosted, and maintained by Microsoft.

With SQL Server, you can use built-in features and functionality that requires extensive configuration (either on-premises or in an Azure virtual machine). When using SQL Database, you pay-as-you-go with options to scale up or out for greater power with no interruption. SQL Database has some additional features that are not available in SQL Server, such as built-in high availability, intelligence, and management.

Azure SQL Database offers the following deployment options:

- As a *single database* with its own set of resources managed via a logical SQL server. A single database is similar to a contained database in SQL Server. This option is optimised for modern application development of new cloud-born applications. Hyperscale and serverless options are available.
- An *elastic pool*, which is a collection of databases with a shared set of resources managed via a logical SQL server. Single databases can be moved into and out of an elastic pool. This option is optimised for modern application development of new cloud-

born applications using the multi-tenant SaaS application pattern. Elastic pools provide a cost-effective solution for managing the performance of multiple databases that have variable usage patterns.

Azure SQL Managed Instance:

Azure SQL Managed Instance falls into the industry category of *Platform-as-a-Service (PaaS)*, and is best for most migrations to the cloud. SQL Managed Instance is a collection of system and user databases with a shared set of resources that is lift-and-shift ready.

- Best for new applications or existing on-premises applications that want to use the latest stable SQL Server features and that are migrated to the cloud with minimal changes. An instance of SQL Managed Instance is similar to an instance of the Microsoft SQL Server database engine offering shared resources for databases and additional instance-scoped features.
- SQL Managed Instance supports database migration from on-premises with minimal to no database change. This option provides all of the PaaS benefits of Azure SQL Database but adds capabilities that were previously only available in SQL Server VMs. This includes a native virtual network and near 100% compatibility with on-premises SQL Server. Instances of SQL Managed Instance provide full SQL Server access and feature compatibility for migrating SQL Servers to Azure.

SQL Server on Azure VM:

SQL Server on Azure VM falls into the industry category *Infrastructure-as-a-Service (IaaS)* and allows you to run SQL Server inside a fully managed virtual machine (VM) in Azure.

- SQL Server installed and hosted in the cloud runs on Windows Server or Linux virtual machines running on Azure, also known as an infrastructure as a service (IaaS). SQL virtual machines are a good option for migrating on-premises SQL Server databases and applications without any database change. All recent versions and editions of SQL Server are available for installation in an IaaS virtual machine.
- Best for migrations and applications requiring OS-level access. SQL virtual machines in Azure are lift-and-shift ready for existing applications that require fast migration to the cloud with minimal changes or no changes. SQL virtual machines offer full

administrative control over the SQL Server instance and underlying OS for migration to Azure.

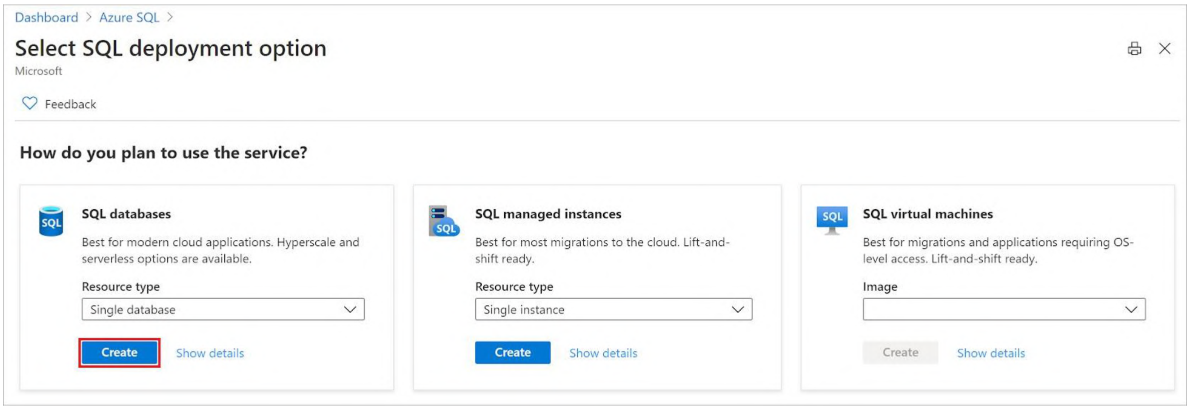
- The most significant difference from SQL Database and SQL Managed Instance is that SQL Server on Azure Virtual Machines allows full control over the database engine. You can choose when to start maintenance/patching, change the recovery model to simple or bulk-logged, pause or start the service when needed, and you can fully customise the SQL Server database engine. With this additional control comes the added responsibility to manage the virtual machine.
- Rapid development and test scenarios when you do not want to buy on-premises non-production SQL Server hardware. SQL virtual machines also run on standardised hardware that is owned, hosted, and maintained by Microsoft. When using SQL virtual machines, you can either pay-as-you-go for a SQL Server license already included in a SQL Server image or easily use an existing license. You can also stop or resume the VM as needed.
- Optimised for migrating existing applications to Azure or extending existing on-premises applications to the cloud in hybrid deployments. In addition, you can use SQL Server in a virtual machine to develop and test traditional SQL Server applications. With SQL virtual machines, you have the full administrative rights over a dedicated SQL Server instance and a cloud-based VM. It is a perfect choice when an organisation already has IT resources available to maintain the virtual machines. These capabilities allow you to build a highly customised system to address your application's specific performance and availability requirements.

AZURE PORTAL: General Steps to Follow:

To create a single database in the Azure portal:

1. Browse to the Select SQL Deployment option page.

2. Under SQL databases, leave Resource type set to Single database, and select Create.



3. On the Basics tab of the Create SQL Database form, under Project details, select the desired Azure Subscription.

4. For Resource group, select Create new, enter *myResourceGroup*, and select OK.

5. For Database name enter *mySampleDatabase*.

6. For Server, select Create new, and fill out the New server form with the following values:

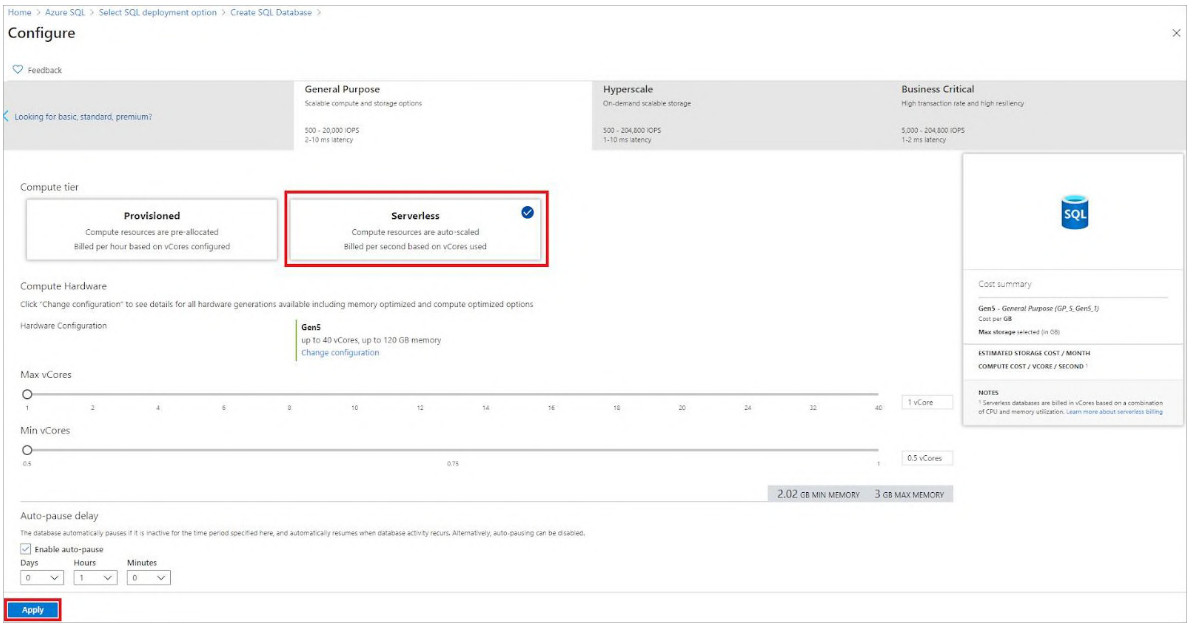
- **Server name:** Enter *mysqlserver*, and add some characters for uniqueness. We can't provide an exact server name to use because server names must be globally unique for all servers in Azure, not just unique within a subscription. So enter something like mysqlserver12345, and the portal lets you know if it is available or not.
- **Server admin login:** Enter *azureuser*.
- **Password:** Enter a password that meets requirements, and enter it again in the Confirm password field.
- **Location:** Select a location from the dropdown list.

7. Select OK.

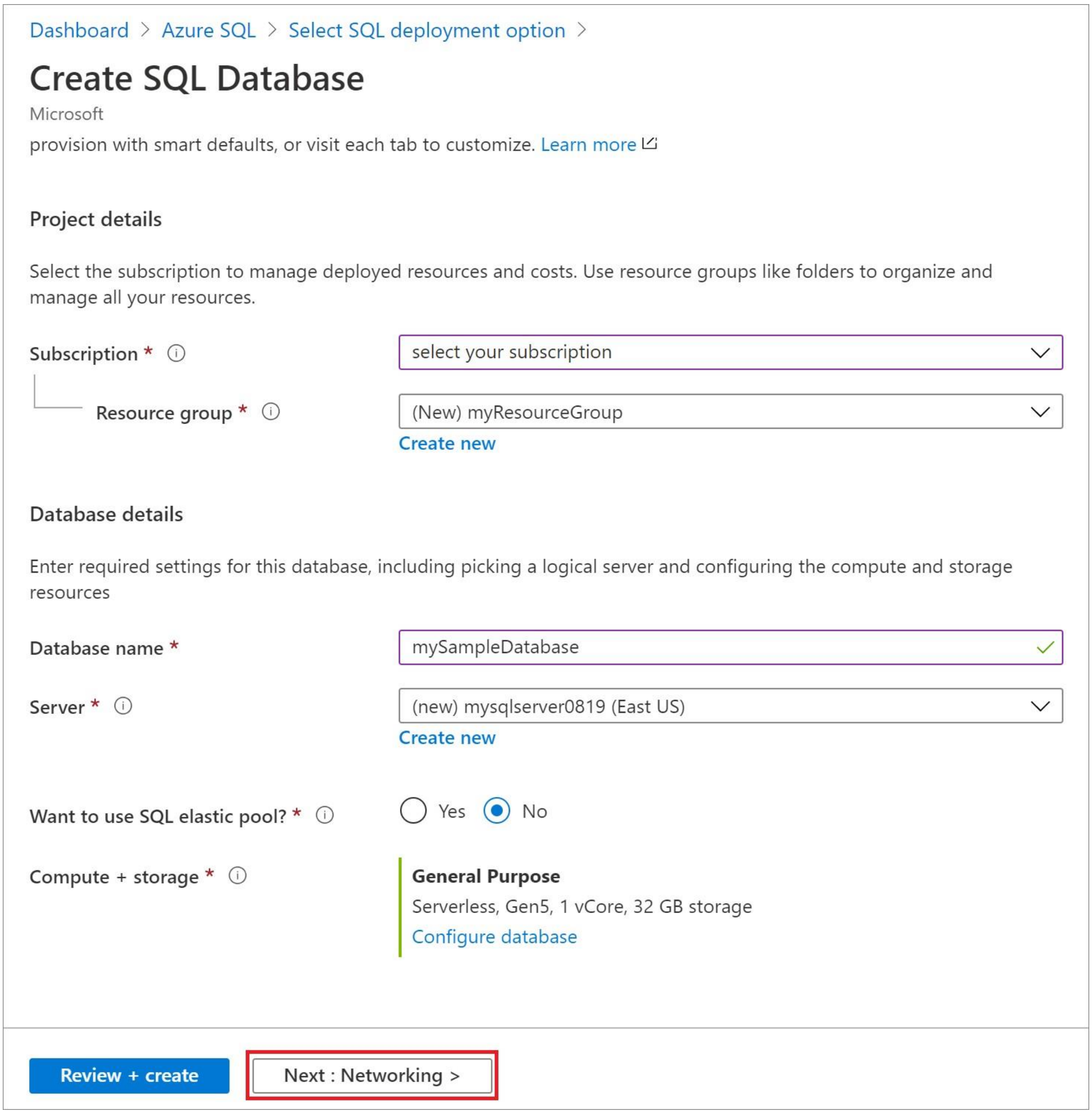
8. Leave Want to use SQL elastic pool set to No.

9. Under Compute + storage, select Configure database.

10. This quickstart uses a serverless database, so select Serverless, and then select Apply.



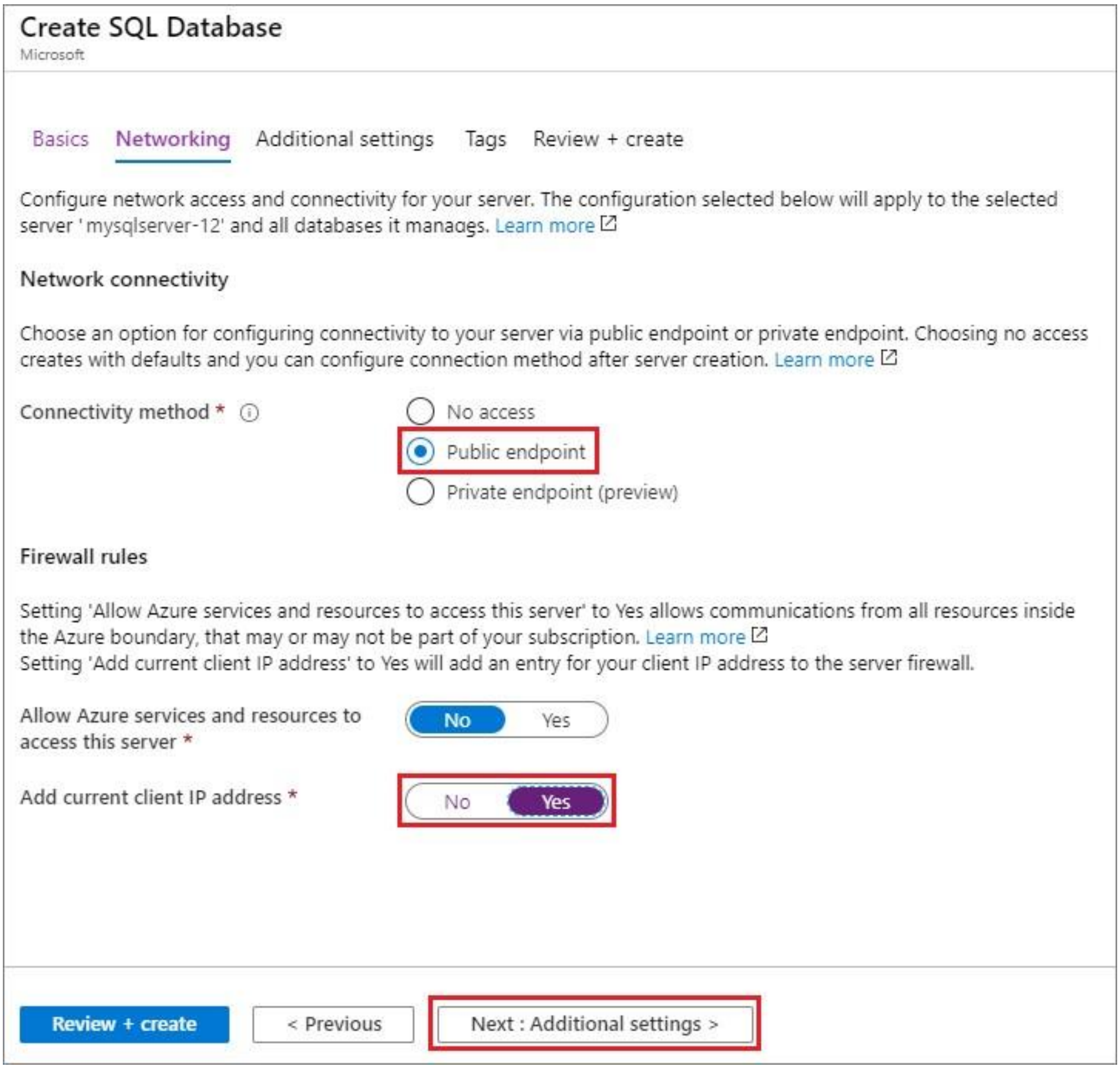
11. Select Next: Networking at the bottom of the page.



12. On the Networking tab, for Connectivity method, select Public endpoint.

13. For Firewall rules, set Add current client IP address to Yes. Leave Allow Azure services and resources to access this server set to No.

14. Select Next: Additional settings at the bottom of the page.

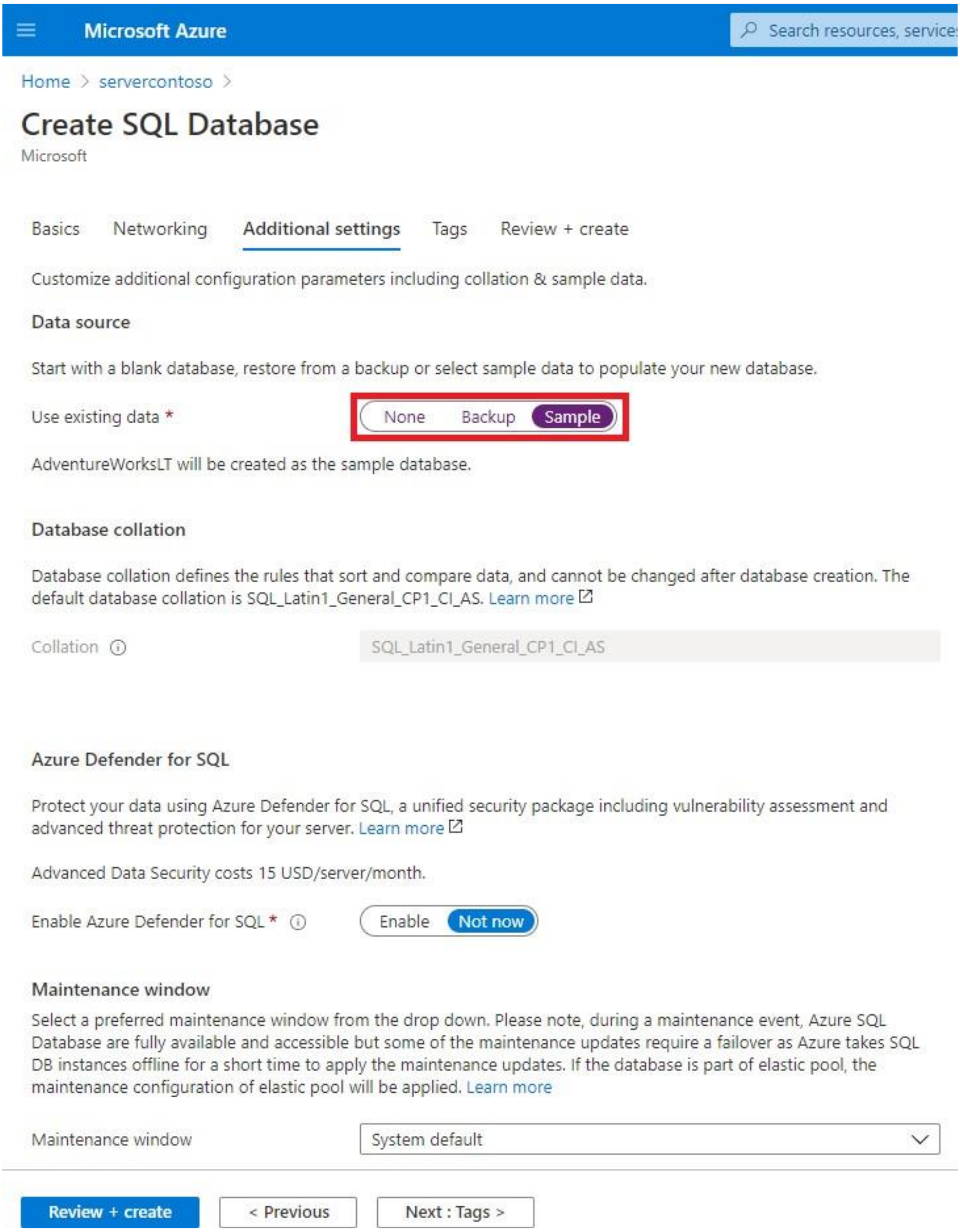


15. On the Additional settings tab, in the Data source section, for Use existing data, select Sample. This creates an AdventureWorksLT sample database so there's some tables and data to query and experiment with, as opposed to an empty blank database.

16. Optionally, enable Azure Defender for SQL.

17. Optionally, set the maintenance window so planned maintenance is performed at the best time for your database.

18. Select Review + create at the bottom of the page:



19. On the Review + create page, after reviewing, select Create.

Query the database:

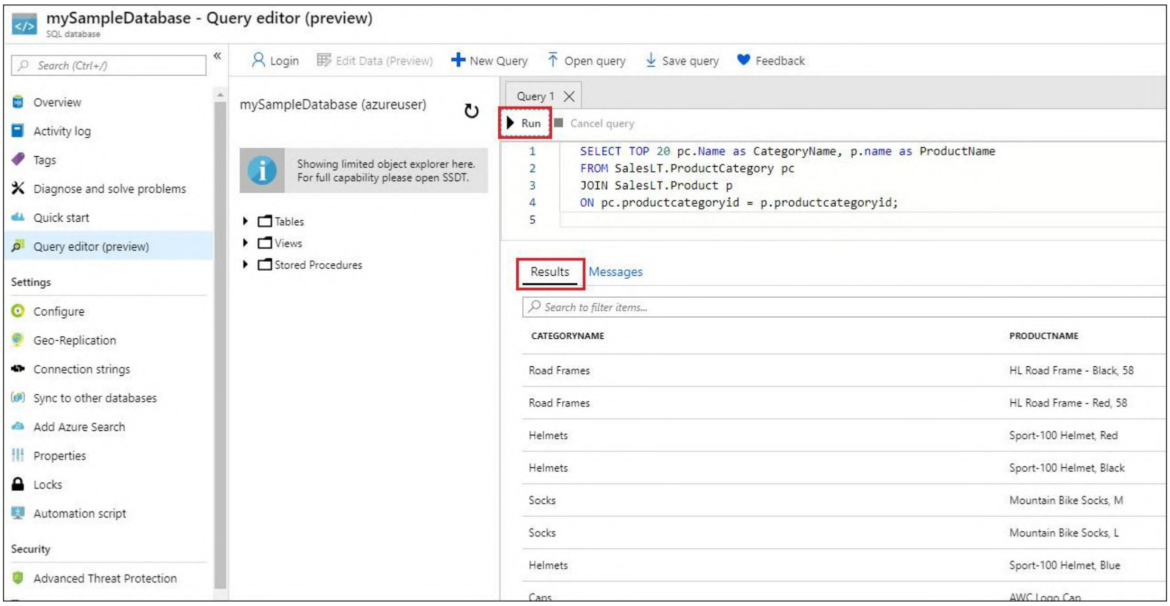
Once your database is created, you can use the Query editor (preview) in the Azure portal to connect to the database and query data.

1. In the portal, search for and select SQL databases, and then select your database from the list.

2. On the page for your database, select Query editor (preview) in the left menu.

3. Enter your server admin login information, and select OK.

4. Select Run, and then review the query results in the Results pane.



5. Close the Query editor page, and select OK when prompted to discard your unsaved edits.

Next Steps:

Connect and query your database using different tools and languages:

- SQL Server Management Studio
- Azure Data Studio

4. PROJECT IMPLEMENTATION

4.0 SYSTEM DESCRIPTION

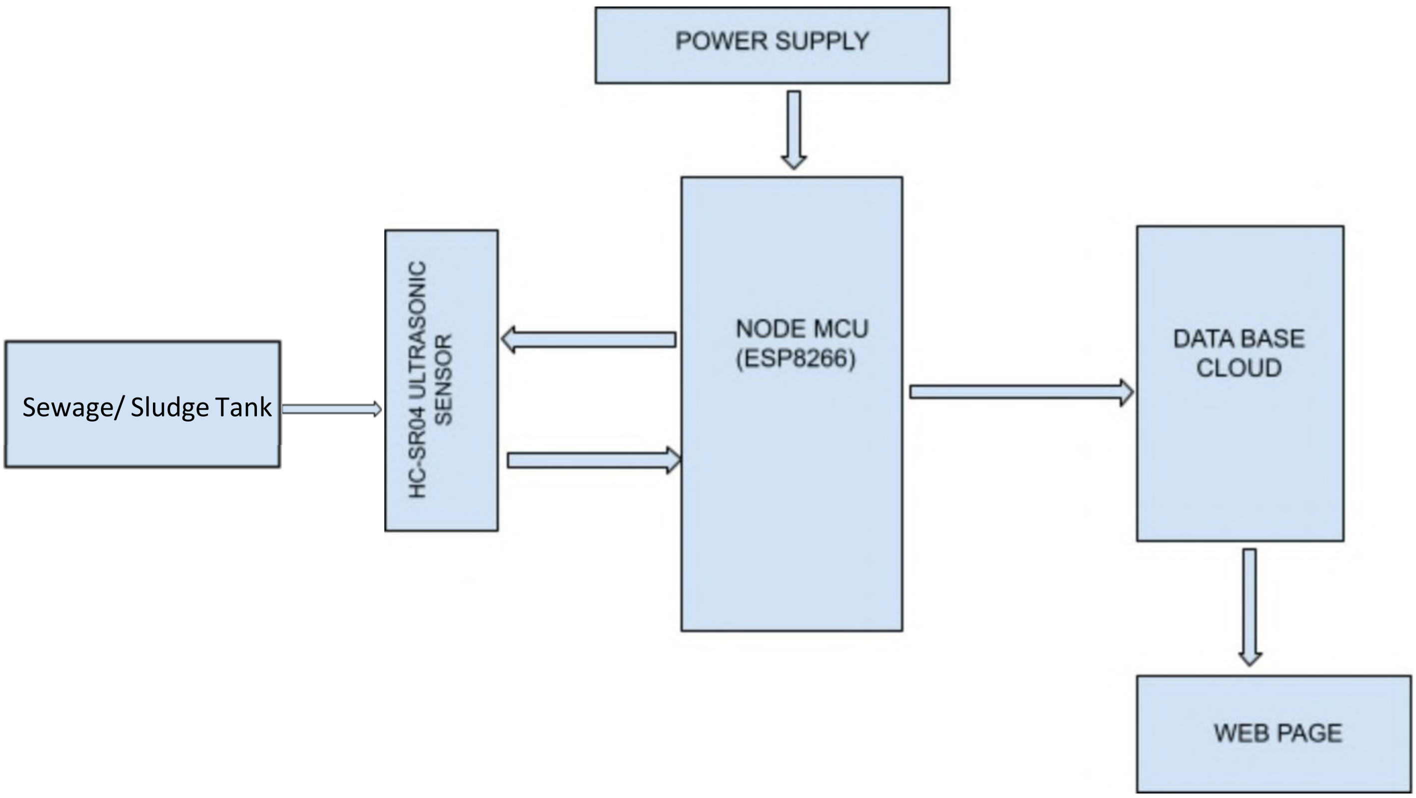


Fig 25: Block Schematic for the Project

- First, on powering the set, the NodeMCU starts connecting to a network (previously setup in the code).
- HC-SR04 Ultrasonic Sensor detects the remaining level, and continuously sends these values to Node MCU.
- Node MCU sends the data to the Cloud DB, through WiFi module which uses PHP Read and PHP Write.
- This data is stored in the Cloud, in an SQL database, which is retrieved using a Webpage. HTML’s High-charts class in the Webpage processes the data, and displays it as a Chart, and as Digital Readings.

4.1 CIRCUIT DIAGRAM

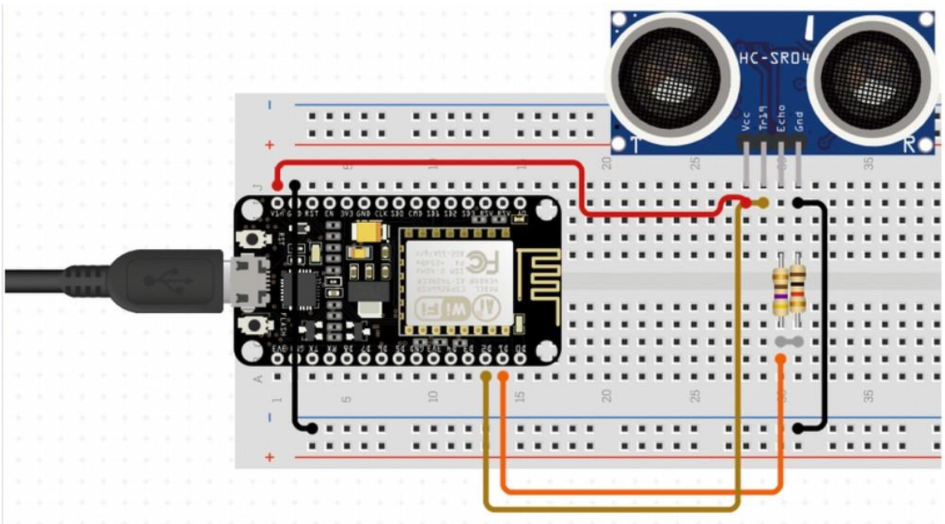


Fig 26: Circuit Schematic for the Project

4.2 RESULTS

The project "**Sewage Level Monitoring System using IoT**" is done. This system provides continuous monitoring-both analog and digital to our sludge tanks. Here is a low-cost, efficient circuit to monitor your sludge level in the chamber/tank. An Ultrasonic distance measuring sensor and an ESP8266 module, which are easily available in the market, can be used to build this device. Now there's no concern that the sludge-tank overflows. To perform this entire task using ESP8266 microcontroller, code is written in embedded C language (for Networking) and HTML (for Monitoring over the Web or App).

Fig 27: Final Model

5. PROJECT TESTING

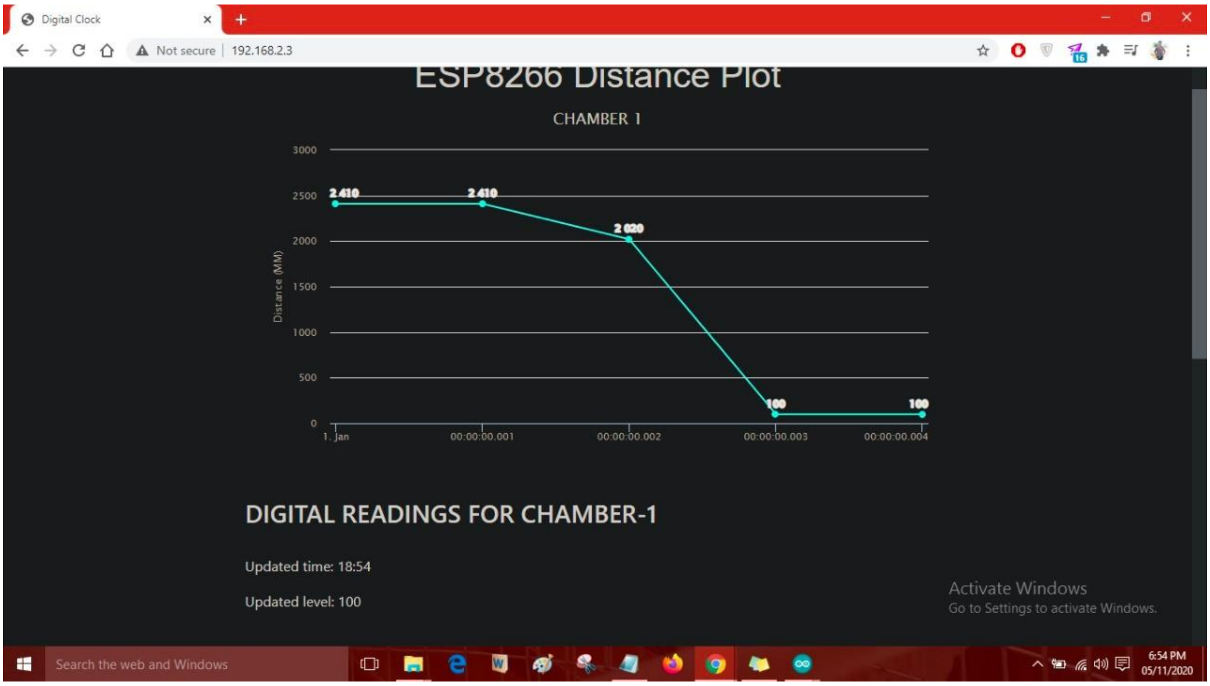


Fig 28: Output Screen for Webpage displaying the level

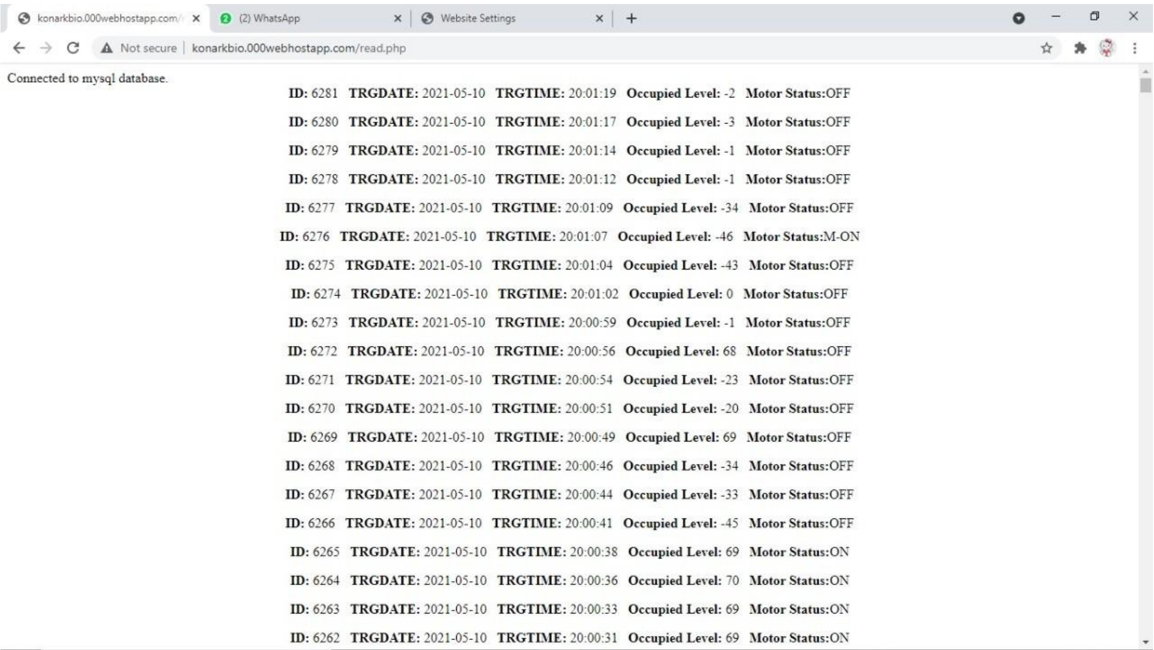


Fig 29: Output Screen for Cloud DB page displaying the levels with respective timestamp

6. CONCLUSION AND FUTURE ENHANCEMENT

6.0 Conclusion:

Integrating features of the hardware components used have been developed in it. Presence of every module has been reasoned out and placed carefully, thus contributing to the best working of the unit. Secondly, using highly advanced IC's with the help of growing technology, the project has been successfully implemented. Thus the project has been successfully designed and tested.

6.1 Future Scope:

Our project "**Sewage Level Monitoring System using IoT**" consists of NodeMCU microcontroller. As of now, an Ultrasonic distance measuring sensor is interfaced to the Arduino microcontroller.

6.1.0 GSM Message Alert System as a Backup for Active Internet Connection:

But this model could be further enhanced by integrating an Optical Ultrasonic sensor (for maximum precision) or even a Radar system. And integrating a GSM module helps in sending emergency SMS alerts in case of chances of overflowing, if any. As an ESP8266 module consists of 17 (D0 to D16) [all are usable] GPIO pins, hence 8 slave devices (Sensors) could be connected to the same master (NodeMCU). Hence more number of tanks could be monitored using a single controller, making the effective cost of the setup lesser than ever.

Fig 30: GSM Module (SIM 900)

6.1.1 Liquid Level Monitoring System:

Also, the same approach could be used for Liquid Sewage's Level Monitoring.

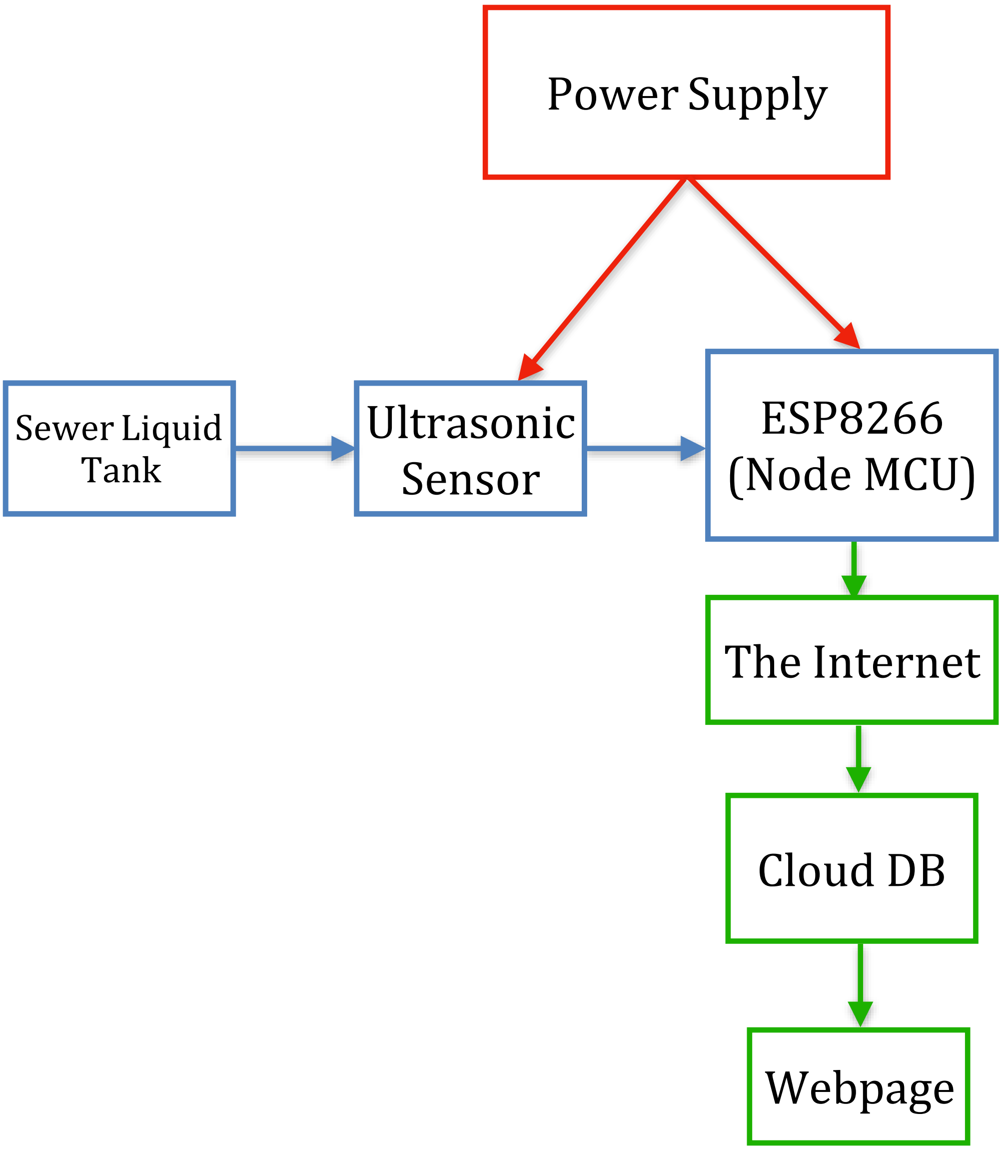


Fig 31: Block Schematic for the Liquid Level Monitoring System

PUBLICATION

Submitted paper in the Conference ICSMEC-21 with Paper ID (ICSMEC21-0074) and got Acceptance for the Paper.

REFERENCES

[1] K Vagholkar, P Vagholkar, “Sewer Gas Explosion In A Public Lavatory: An Unheard Cause Of Genital Burns”, The Internet Journal of Surgery Volume 31 Number 1, ISPUB.com / IJS/31/1/14730

[2] Libia R. Diaz-Valbuena, Harold L. Leverenz, Christopher D. Cappa, George Tchobanoglous, William R. Horwath, and Jeannie L. Darby, “Methane, Carbon Dioxide, and Nitrous Oxide Emissions from Septic Tank Systems”, Environmental Science & Technology 2011 *45* (7), 2741-2747, DOI: 10.1021/es1036095

[3] Loi Tan Huynh, Hidenori Harada, Shigeo Fujii, Lien Pham Hong Nguyen, Thu-Huong Thi Hoang, and Hai Trung Huynh, “Greenhouse Gas Emissions from Blackwater Septic Systems”, Environmental Science & Technology 2021 *55* (2), 1209-1217, DOI: 10.1021/ acs.est.0c03418

[4] Jeffrey A.Soller, Mary E.Schoen, Timothy Bartrand, John E.Ravenscroft, and Nicholas J.Ashbolt, “Estimated human health risks from exposure to recreational waters impacted by human and non-human sources of fecal contamination”, Water Research Volume 44, Issue 16, September 2010, Pages 4674-4691

[5] Doorn M., Liles D., Thorneloe S. (2000) Quantification of methane emissions from latrines, septic tanks, and stagnant, open sewers in the world. In: van Ham J., Baede A.P.M., Meyer L.A., Ybema R. (eds) Non-CO2 Greenhouse Gases: Scientific Understanding, Control, and Implementation. Springer, Dordrecht. https://doi.org/ 10.1007/978-94-015-9343-4_4

[6] Pendharkar, Anushka and Chillapalli, Jyothi and Dhakate, Kanksha and Gogoi, Subhalaxmi and Jadhav, Yogesh, IoT Based Sewage Monitoring System (June 26, 2020). Proceedings of the International Conference on Recent Advances in Computational Techniques (IC-RACT) 2020, Available at SSRN: https://ssrn.com/abstract=3697395 or http://dx.doi.org/10.2139/ssrn.3697395

[7] D. S. Vidhya, Delicia Perlin Rebelo, Cecilia Jane D’Silva, Linford William Fernandes, and Clarissa Joella Costa, “Obstacle Detection using Ultrasonic Sensors”, IJIRST/ Volume 2 / Issue 11/ 057, April 2016

[8] Elijah J. Morgan, “HCSR04 Ultrasonic Sensor”, HC-SR04’s User’s Manual, Tempero Systems, Nov. 16, 2014

[9] Nadhir Ibrahim Abdulkhaleq, Ihsan Jabbar Hasan, and Nahla Abdul Jalil Salih, “Investigating the resolution ability of the HC-SRO4 ultrasonic sensor”, 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* 745 012043

[10] S.Sunitha, “Distance Measurement using Ultrasonic Sensor and NodeMCU”, International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056 Volume: 04 Issue: 06, June -2017 www.irjet.net p-ISSN: 2395-0072

[11] T. Perumal, M. N. Sulaiman, and C. Y. Leong, "Internet of Things (IoT) enabled water monitoring system," 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), 2015, pp. 86-87, doi: 10.1109/GCCE.2015.7398710.

[12] S. Sachio, A. Noertjahyana, and R. Lim, "IoT Based Water Level Control System," 2018 3rd Technology Innovation Management and Engineering Science International Conference (TIMES-iCON), 2018, pp. 1-5, doi: 10.1109/TIMES-iCON.2018.8621630.

[13] Content Development Team, RoboIndia. (25th Jone, 2019), “*NodeMCU Temperature, Humidity data upload on Thingspeak on Arduino IDE*”, https://roboindia.com/tutorials/nodemcu-dht11-thingspeak-data-upload/

[14] Alsan Parajuli. (8th May 2020), “*ESP8266 Plot Sensor readings to Web Server in Real-Time Chart*”. https://theiotprojects.com/esp8266-plot-sensor-readings-to-webserver-in-real-time-chart/

[15] Great Projects. (29th October 2020), “*Great Projects: Esp8266 DHT11 SD Card Web Server*”. https://arduino-projects-free.blogspot.com/2020/10/esp8266-dht11-sd-card-web-server.html

BIBLIOGRAPHY

[1] L. Koval, J. Vanus, and P. Bilik. Distance Measuring by Ultrasonic Sensor. IFAC-PapersOnLine 49-25 (2016), pages 153–158.

[2] Khin Kyawt Kyawt Khaing. Temperature and Humidity Monitoring and Control System with Thing Speak. In International Journal of Scientific Research and Engineering Development-– Volume 2 Issue 5, Sep-Oct 2019.

[3] Christos Stergiou, Kostas E. Psannisa, Byung-Gyu Kim, and Brij Gupta. Secure integration of IoT and Cloud Computing. Future Generation Computer Systems-Volume 78, Part 3, January 2018, Pages 964-975.

[4] Steven Sachio, Agustinus Noertjahyana, and Resmana Lim. IoT Based Water Level Control System. In 2018 3rd IEEE Technology Innovation Management and Engineering Science International Conference (TIMES-iCON)

[5] Dlnya Abdulahad Aziz. Web Server Based Smart Monitoring System Using ESP8266 Node MCU Module. In International Journal Of Scientific & Engineering Research, Volume 9, Issue 6, June-2018. Issn: 2229-5518.

[6] Thinagaran Perumal, Md Nasir Sulaiman, and C. Y. Leong. Internet of Things (IoT) enabled water monitoring systems. In 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE).

[7] Michael Yuan. (12th June 2017), *“Getting to know NodeMCU and its DEVKIT board”.* https://developer.ibm.com/technologies/iot/tutorials/iot-nodemcu-open-why-use/

APPENDICES

This project aims to measure and analyse the real-time levels of sewage. This project attempts to device an IoT technology that shall measure the sludge level while keeping a record of the data history in a Cloud Database. It can be accessed by using Web Browser and Cloud Database on the connected devices of the authorised people who are remotely located.

A

PROJECT REPORT

ON

DAM OPERATION BASED ON WATER LEVEL

Submitted by

1)K. GOPI (17K81A0426)

2) M. SUNIL (17K81A0429)

3) S. KISHORE (15K81A04B5)

In partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Ms.D.Kirtana ,M.Tech
Assistant Professor

DEPARTMENT OF

ELECTRONICS AND COMMUNICATION ENGINEERING

St.MARTIN'S ENGINEERING COLLEGE
(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

AN UGC AUTONOMOUS INSTITUTE

Dhulapally, Secunderabad – 500 100

JUNE 2021

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **KISHORE SINDHU** WITH ROLL NO.**15K81A04B5, KONAGANDLA GOPI** WITH ROLL NO.**17K81A0426**, **MAGAPU SUNIL** WITH ROLL NO.**17K81A0429**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**DAM OPERATION BASED ON WATER LEVEL**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. *Kompally-500014.*

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

St.MARTIN'S ENGINEERING COLLEGE
(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)
An UGC Autonomous Institute
Dhulapally,Secunderabad-500 100
NBA &NAAC A+ Accredited
DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the project entitled **DAM OPERATION BASED ON WATER LEVEL**, is being submitted by1. **K. GOPI (17K81A0426), 2. M. SUNIL (17K81A0429) 3. S.KISHORE (15K81A04B5)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN Electronics And Communication Engineering Department** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Internal Guide
Ms.D.Kirtana M.Tech
Assisstant Professor,
Department of ECE

Head of the Department
Dr.B.Hari Krishna
Department of ECE

Internal Examiner

External Examiner

Place:

Date:

DECLARATION

We, the student of **Bachelor of Technology** in Department of Electronics and Communication Engineering, session: 2017 – 2021, St. Martin's Engineering College, Duhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled '**DAM OPERATION BASED ON WATER LEVEL'** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree

K. GOPI	M.SUNIL	S.KISHORE
(17K81A0426)	(17K81A0429)	(15K81A04B5)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to Dr.B.Hari Krishna, **Head &Professor**, the Dept. of Electronics and Communication engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept. project coordinator Dr.Anand,** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide, **Ms.D.Kirtana Assistant Professor** ,Department of ECE and our class coordinator **Mrs.A.Soumya**, **Assistant Professor**, Department of ECE and for guidance and encouragement in carrying out this project work

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Maalayadri, Professor in IT and Dr.R.Santosh Kumar, Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1) K.GOPI

2) M.SUNIL

3) S.KISHORE

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

We here propose an automatic dam water level monitor and controller system. Our proposed project uses sensors to sense the water level and then opens the dam gate (motor used to demonstrate as dam gate) according to the water level. Our system uses multiple water level sensors (float sensors) for this purpose. The sensor are mounted at three different levels in order to check water level and provide signals accordingly. When water reaches first level it is sensed by it and displayed. When water reaches second level it provides a signal to the microcontroller and it opens the dam gate partially. As soon as the water level reaches the third position, it signals the microcontroller and the microcontroller then signals the motor to run, which is demonstrated as opening the dam gate fully. Thus, our proposed system allows for automatic dam gate opening based on water level sensing.

CHAPTER 1
INTRODUCTION
1.1 PROJECT OVERVIEW:

A Microcontroller is a computer-on-a-chip or a single-chip computer that contains the processor (the CPU), non-volatile memory for the program (ROM or flash), volatile memory for input and output (RAM), a clock and an I/O control unit. Micro suggests that the device is small and controller tells that the device might be used to control objects, processes or events. Another term is Embedded Microcontroller tells that it support circuits are often built into or embedded in the devices for control. You find microcontroller in all kinds of things never days. It is used for measures, controls, stores or displays information by placing microcontroller inside any device. The largest single use for microcontroller in automobiles-car manufactured today includes at least one microcontroller for engine control and more to control additional systems. In desktop computer, you may find microcontrollers inside keyboards, modems, printers, and other peripherals. In test equipment, microcontrollers make things easier to store measurement, to create and store user routines, and to display messages and waveforms. Consumer products like cameras, video recorders, compact-disk players, and ovens. And they are so many applications where we use microcontrollers. A micro controller is similar to the microprocessor inside a personal computer. Examples are Intel's 8086, Zilog's Z80. Both microprocessors and microcontrollers contain CPU. The CPU executes instructions that perform the basic logic, math, and data moving functions of a computer. To make a complete computer, a microprocessor require memory for storing data and programs, and I/O interfaces for connecting external devices like keyboard and displays. In contrast, microcontrollers are a single chip computer because it contains memory and I/O interfaces in addition to the CPU. It tends to limit the amount of memory and interfaces that can fit on single chip, microcontrollers tend to be used in smaller system. Examples of popular microcontrollers are Intel's 8052, 89C052, Motorola's 68HC11 and Zilog's Z8. The following are some of the capabilities of 8051 microcontroller:

- Internal ROM and RAM
- I/O ports with programmable pins
- Timers and counters

Serial data communication Motion control, in electronic terms, means to accurately control the movement of an object based on speed, distance, load, inertia or a combination of all these factors. There are numerous types of motion control systems, including; Stepper

Motor, Linear Step Motor, DC Brush, Brushless Servo and more. This document will concentrate on Step Motor technology. Like many conventional electric motors, a “stepper motor consists of a magnet and coils of wire. Whereas conventional motors spin continuously, a stepper motor moves around one small step at a time (hence the name). A stepper motor is a marvel in simplicity. It has no brushes, or contacts. Basically, it’s a synchronous motor with the magnetic field electronically switched to rotate the armature magnet around. “The stepper motor is an electromagnetic device that converts digital pulses into mechanical shaft or spindle rotation”. The shaft or spindle of a stepper motor rotates in discrete step increments when electrical command pulses are applied to it in the proper sequence. The motors rotation has several direct relationships to these applied input pulses. The sequence of the applied pulses is directly related to the direction of motor shafts rotation. The speed of the motor shafts rotation is directly related to the frequency of the input pulses and the length of rotations applied. The simplest way to think of a stepper motor is a bar magnet and four coils.

CHAPTER 2

EMBEDDED SYSTEMS

PROJECT DESIGN:

2.1 EMBEDDED DESIGN:

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.

The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes. The applications software on such processors is sometimes referred to as firmware

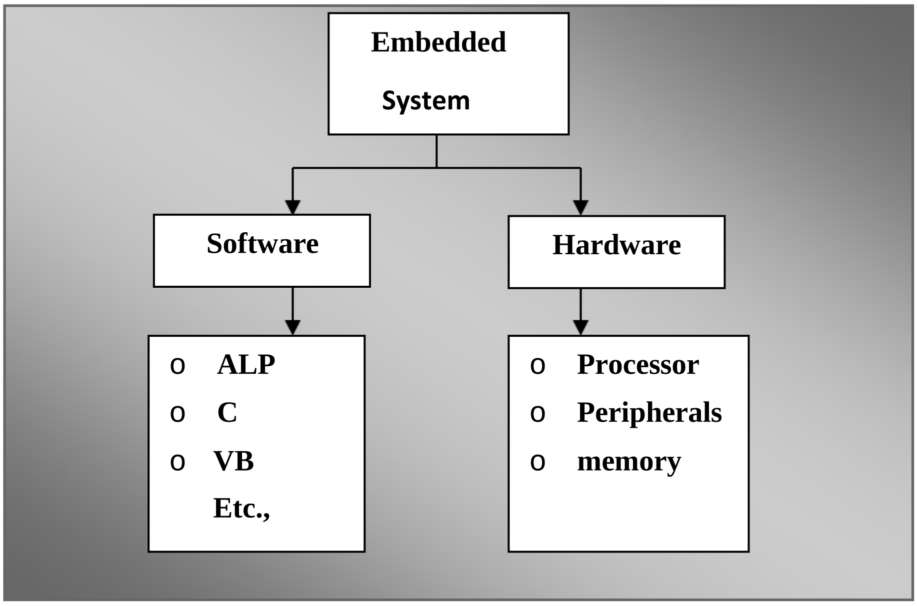


Figure 2.1 Block diagram of Embedded System

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

Memory: It is used to store data or address.

Peripherals: These are the external devices connected

Processor: It is an IC which is used to perform some task

Applications of embedded systems

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

Processors are classified into four types like:

- Micro Processor (µp)
- Micro controller (µc)
- Digital Signal Processor (DSP)
- Application Specific Integrated Circuits (ASIC)

Micro Processor (μp):

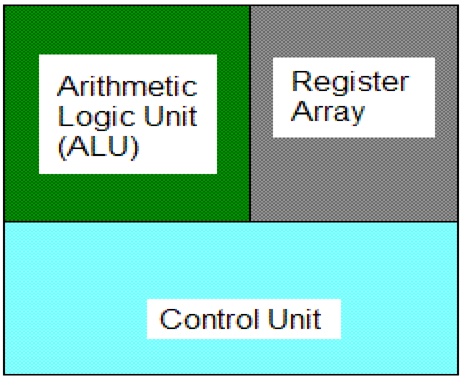
A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Three basic characteristics differentiate microprocessors:

- **Instruction set**: The set of instructions that the microprocessor can execute.
- **Bandwidth :** The number of bits processed in a single instruction.
- **Clock speed :** Given in megahertz (MHz), the clock speed determines how many instructions per second the processor can execute.

In both cases, the higher the value, the more powerful the CPU. For example, a 32-bit microprocessor that runs at 50MHz is more powerful than a 16-bit microprocessor that runs at 25MHz. In addition to bandwidth and clock speed, microprocessors are classified as being either RISC (reduced instruction set computer) or CISC (complex instruction set computer).

A microprocessor has three basic elements, as shown above. The ALU performs all arithmetic computations, such as addition, subtraction and logic operations (AND, OR, etc). It is controlled by the Control Unit and receives its data from the Register Array. The Register Array is a set of registers used for storing data. These registers can be accessed by the ALU very quickly. Some registers have specific functions - we will deal with these later. The Control Unit controls the entire process. It provides the timing and a control signal for getting data into and out of the registers and the ALU and it synchronizes the execution of instructions (we will deal with instruction execution at a later date).



Three Basic Elements of a Microprocessor

Micro Controller (μc):

A microcontroller is a small computer on a single **integrated circuit** containing a processor core, memory, and programmable **input/output** peripherals. Program memory in the form of **NOR flash** or **OTP ROM** is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the **microprocessors** used in **personal computers** or other general purpose applications.

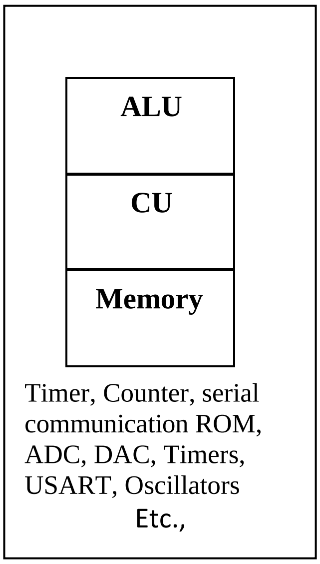


Figure 2.2 Block Diagram of Micro Controller (μc)

Digital Signal Processors (DSPs):

Digital Signal Processors is one which performs scientific and mathematical operation. Digital Signal Processor chips - specialized microprocessors with architectures designed specifically for the types of operations required in digital signal processing. Like a general-purpose microprocessor, a DSP is a programmable device, with its own native instruction code. DSP chips are capable of carrying out millions of floating point operations per second, and like their better-known general-purpose cousins, faster and more powerful versions are continually being introduced. DSPs can also be embedded within complex "system-on-chip" devices, often containing both analog and digital circuitry.

Application Specific Integrated Circuit (ASIC)

ASIC is a combination of digital and analog circuits packed into an IC to achieve the desired control/computation function

ASIC typically contains

- CPU cores for computation and control
- Peripherals to control timing critical functions
- Memories to store data and program

- ➢ Analog circuits to provide clocks and interface to the real world which is analog in nature
- ➢ I/Os to connect to external components like LEDs, memories, monitors etc.

2.2 Computer Instruction Set

There are two different types of computer instruction set there are:

1. RISC (Reduced Instruction Set Computer) and
2. CISC (Complex Instruction Set computer)

2.2.1 Reduced Instruction Set Computer (RISC)

A RISC (reduced instruction set computer) is a microprocessor that is designed to perform a smaller number of types of computer instruction so that it can operate at a higher speed (perform more million instructions per second, or millions of instructions per second). Since each instruction type that a computer must perform requires additional transistors and circuitry, a larger list or set of computer instructions tends to make the microprocessor more complicated and slower in operation.

Besides performance improvement, some advantages of RISC and related design improvements are:

- ➢ A new microprocessor can be developed and tested more quickly if one of its aims is to be less complicated.
- ➢ Operating system and application programmers who use the microprocessor's instructions will find it easier to develop code with a smaller instruction set.
- ➢ The simplicity of RISC allows more freedom to choose how to use the space on a microprocessor.

Higher-level language compilers produce more efficient code than formerly because they have always tended to use the smaller set of instructions to be found in a RISC computer.

2.2.2 RISC characteristics

- ➢ **Simple instruction set:**

In a RISC machine, the instruction set contains simple, basic instructions, from which more complex instructions can be composed.

- ➢ **Same length instructions.**

Each instruction is the same length, so that it may be fetched in a single operation.

- ➢ **1machine-cycleinstructions.**

Most instructions complete in one machine cycle, which allows the processor to handle several instructions at the same time. This pipelining is a key technique used to speed up RISC machines.

2.2.3 Complex Instruction Set Computer (CISC)

CISC, which stands for **Complex Instruction Set Computer,** is a philosophy for designing chips that are easy to program and which make efficient use of memory. Each instruction in a CISC instruction set might perform a series of operations inside the processor. This reduces the number of instructions required to implement a given program, and allows the programmer to learn a small but flexible set of instructions.

The advantages of CISC

At the time of their initial development, CISC machines used available technologies to optimize computer performance.

- Microprogramming is as easy as assembly language to implement, and much less expensive than hardwiring a control unit.
- The ease of micro-coding new instructions allowed designers to make CISC machines upwardly compatible: a new computer could run the same programs as earlier computers because the new computer would contain a superset of the instructions of the earlier computers.
- As each instruction became more capable, fewer instructions could be used to implement a given task. This made more efficient use of the relatively slow main memory.
- Because micro program instruction sets can be written to match the constructs of high-level languages, the compiler does not have to be as complicated.

The disadvantages of CISC

Still, designers soon realized that the CISC philosophy had its own problems, including:

- Earlier generations of a processor family generally were contained as a subset in every new version --- so instruction set & chip hardware become more complex with each generationof computers.
- So that as many instructions as possible could be stored in memory with the least possible wasted space, individual instructions could be of almost any length---this means that different instructions will take different amounts of clock time to execute, slowing down the overall performance of the machine.
- Many specialized instructions aren't used frequently enough to justify their existence --- approximately 20% of the available instructions are used in a typical program.

➢ CISC instructions typically set the condition codes as a side effect of the instruction. Not only does setting the condition codes take time, but programmers have to remember to examine the condition code bits before a subsequent instruction changes them.

2.3 Memory Architecture

There two different type's memory architectures there are:

- Harvard Architecture
- Von-Neumann Architecture

2.3.1 Harvard Architecture

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The **Harvard architecture** is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

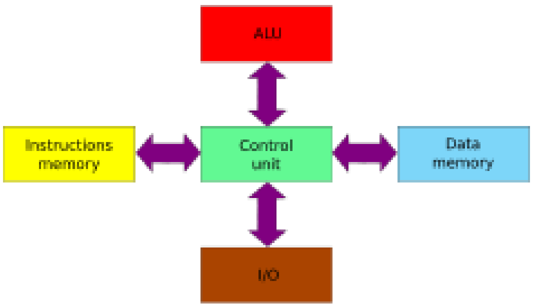


Figure 2.3 Harvard Architecture

Modern uses of the Harvard architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

- Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behavior must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.
- Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

2.3.2 Von-Neumann Architecture

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. The **von Neumann architecture** is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A **stored-program** digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

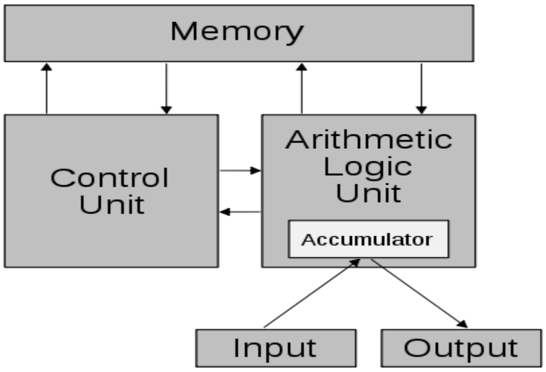


Figure 2.4 Schematic of the Von-Neumann Architecture.

Basic Difference between Harvard and Von-Neumann Architecture

- The primary difference between Harvard architecture and the Von Neumann architecture is in the Von Neumann architecture data and programs are stored in the same memory and managed by the same information handling system.
- Whereas the Harvard architecture stores data and programs in separate memory devices and they are handled by different subsystems.
- In a computer using the Von-Neumann architecture without cache; the central processing unit (CPU) can either be reading and instruction or writing/reading data to/from the memory. Both of these operations cannot occur simultaneously as the data and instructions use the same system bus.
- In a computer using the Harvard architecture the CPU can both read an instruction and access data memory at the same time without cache. This means that a computer with Harvard architecture can potentially be faster for a given circuit complexity because data access and instruction fetches do not contend for use of a single memory pathway.
- Today, the vast majority of computers are designed and built using the Von Neumann architecture template primarily because of the dynamic capabilities and efficiencies gained in designing, implementing, operating one memory system as opposed to two. Von Neumann architecture may be somewhat slower than the contrasting Harvard Architecture for certain specific tasks, but it is much more flexible and allows for many concepts unavailable to Harvard architecture such as self-programming, word processing and so on.

CHAPTER 3

POWER SUPPLY

3.1 Block Diagram

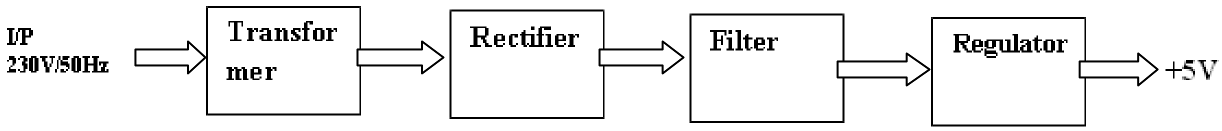
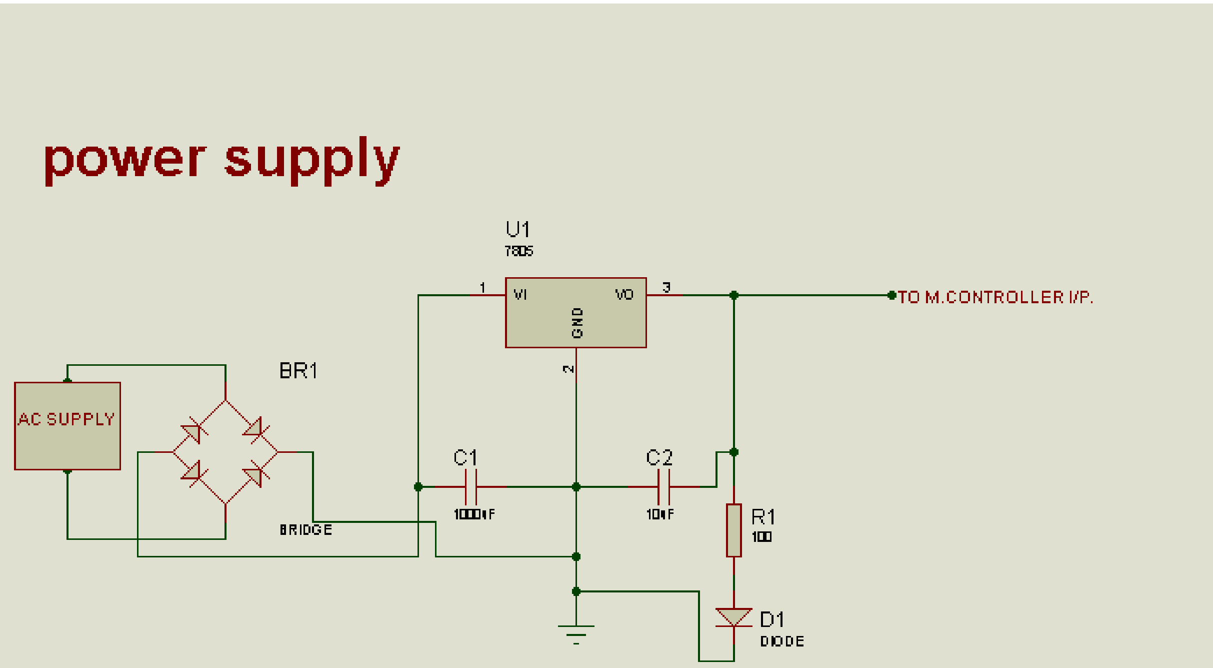


Figure 3.1 Power Supply

3.2 Circuit Diagram



Description

3.2.1 Transformer

A **transformer** is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

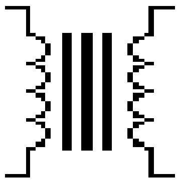


Figure: Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

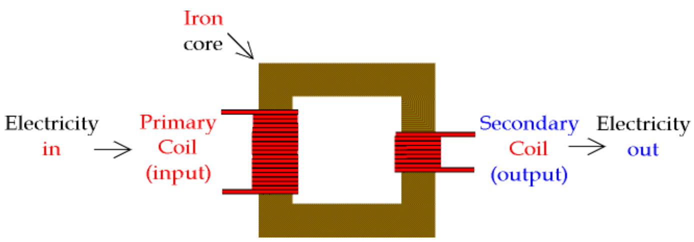
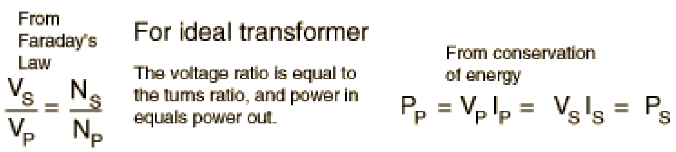


Figure 3.2 Transformer

Basic Principle:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.



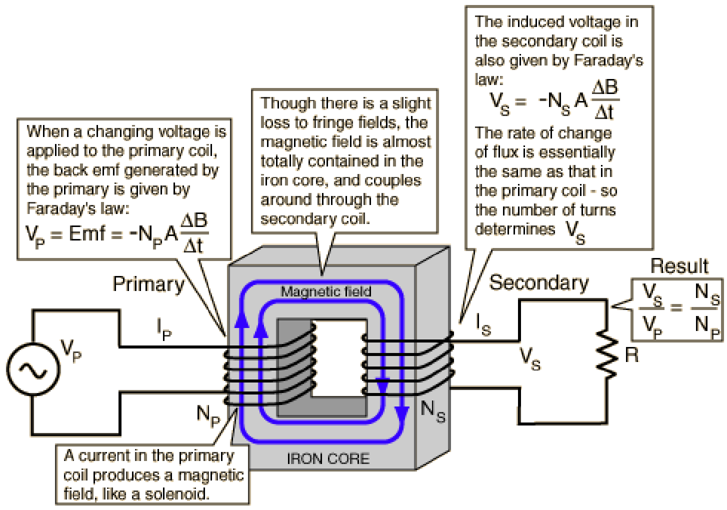


Figure: Basic Principle

Transformer Working

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

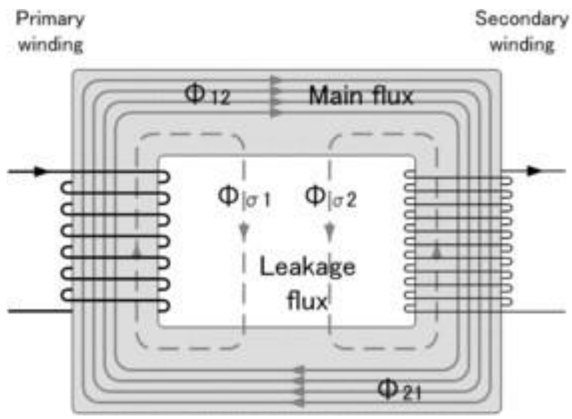


Figure: Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f.

The iron core is laminated to prevent 'eddy currents' flowing in the core. These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

Classification of Transformer:

- Step-Up Transformer
- Step-Down Transformer

Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn’s ratio of the two sets of windings determines the amount of voltage transformation.

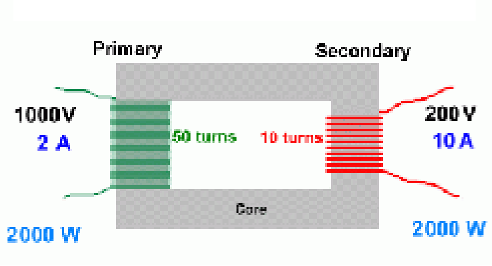


Figure: Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single-phase step-down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase step up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

Step-Up Transformer:

A step-up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply.

A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

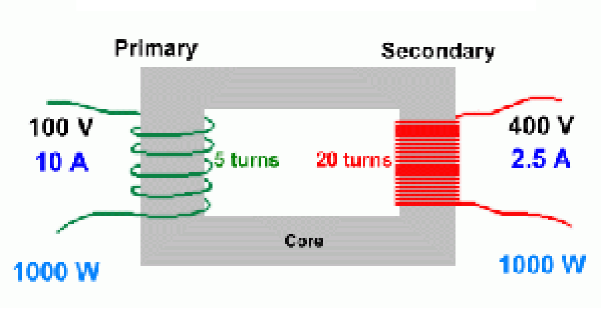


Figure: Step-Up Transformer

Applications

Generally these **Step-Up Transformers** are used in industries applications only.

Turns Ratio and Voltage

The ratio of the number of turns on the primary and secondary coils determines the ratio of the voltages...

$$\frac{V_s}{V_p} = \frac{N_s}{N_p}$$

...where V_p is the primary (input) voltage, V_s is the secondary (output) voltage, N_p is the number of turns on the primary coil, and N_s is the number of turns on the secondary coil.

3.2.2 Diodes

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

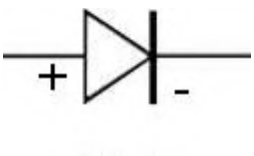


Figure 3.3 Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

3.2.3 Rectifier

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

The Half-wave Rectifier

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure .

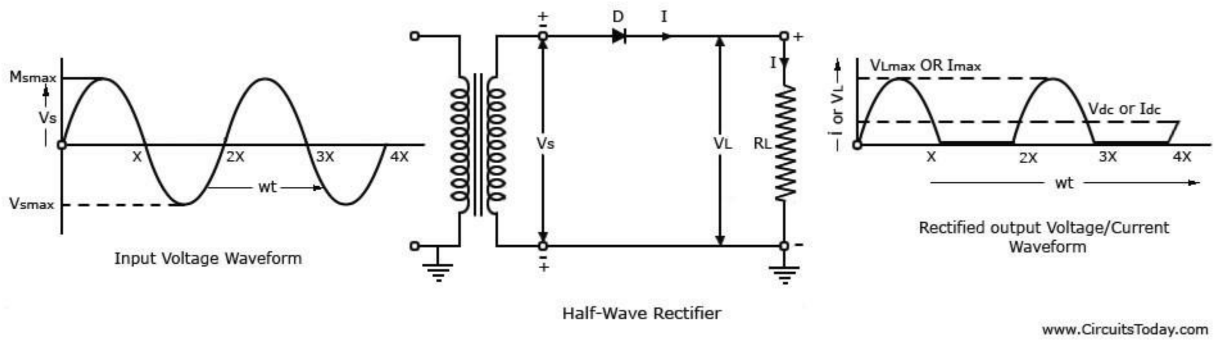


Figure 3.4 Half Wave Rectifier

Figure 2 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is 0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

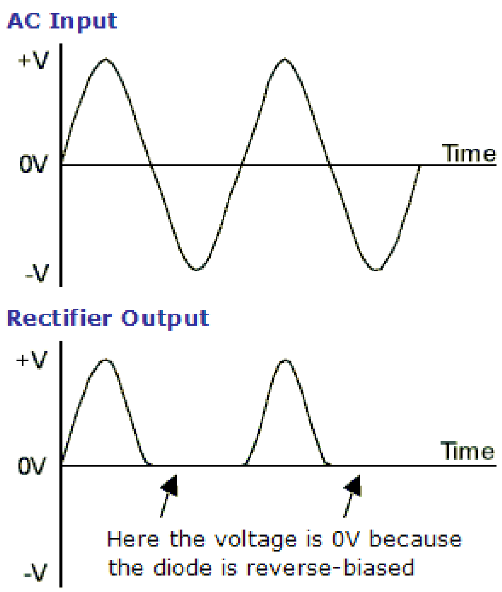


Figure 3.5 Half-Wave Rectification

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

The Full-wave Rectifier

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC. The resulting waveform is shown in figure 4.

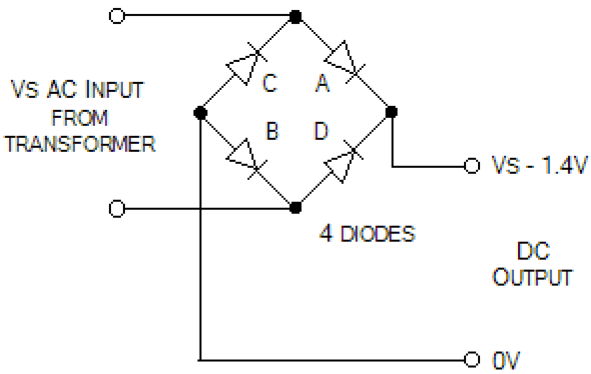


Figure Full-Wave Rectifier

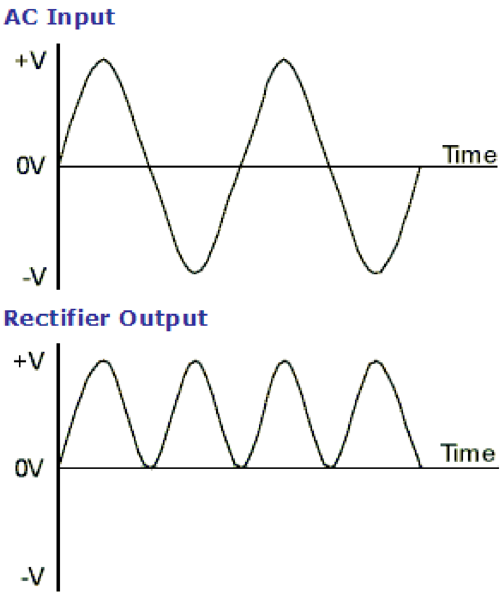


Figure Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased.

While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

Capacitor Filter

The **capacitor-input filter**, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

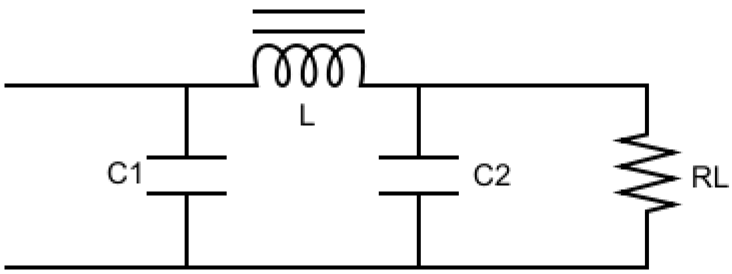


Figure: Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an

appreciable amount of the AC component while the DC component continues its journey to the inductor L

2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result only the DC component appears across the load RL.

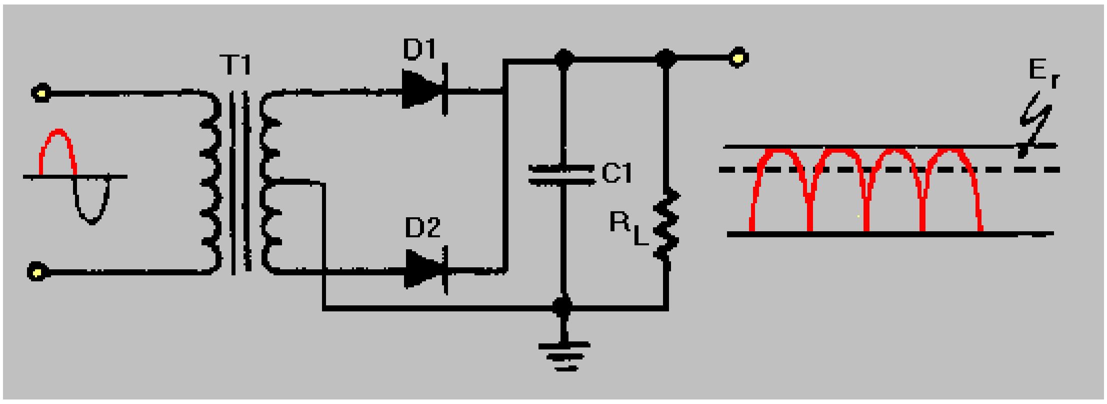


Figure: Centered Tapped Full-Wave Rectifier with a Capacitor Filter

3.2.4 Voltage Regulator:

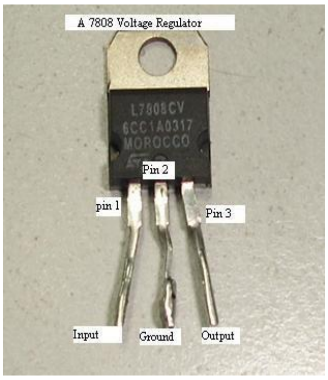
A **voltage regulator** is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx'indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx'indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05' indicates the regulator output is -5V.

These regulators consists the three pins there are



Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

Figure 3.6 Regulator

CHAPTER 4

SOFTWARE

4.1 Proteus

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

4.1.1 About Proteus

It is a software suite containing schematic, simulation as well as PCB designing.

ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.

The designer can also develop 2D drawings for the product.

4.1.2 Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

4.2 Starting New Design

Step 1: Open ISIS software and select New design in File menu

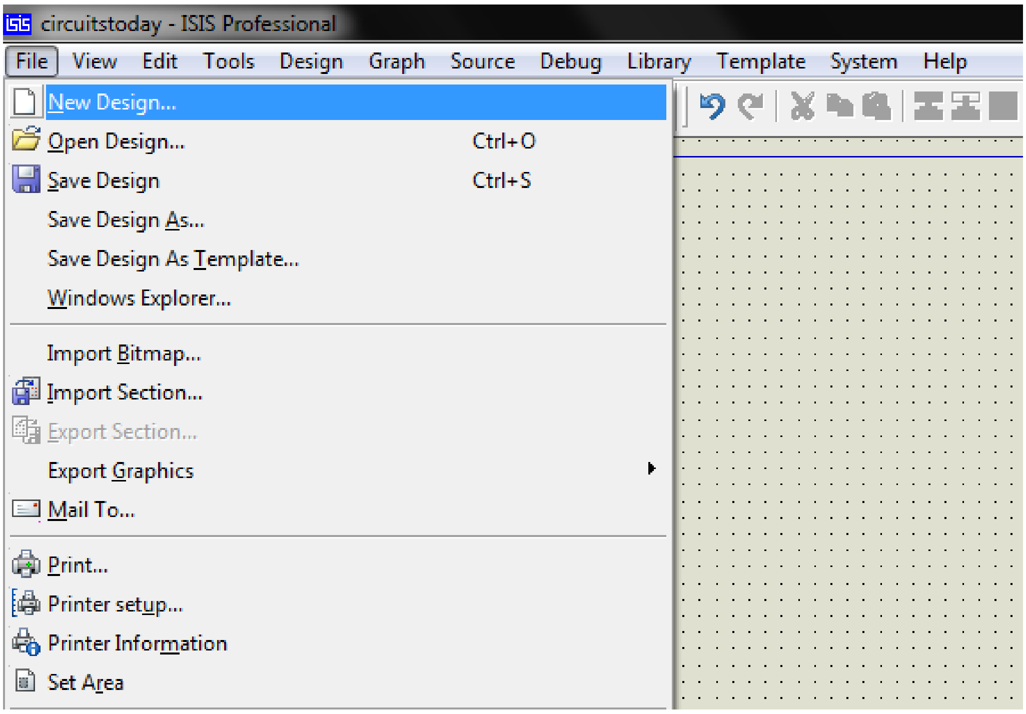


Figure 4.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

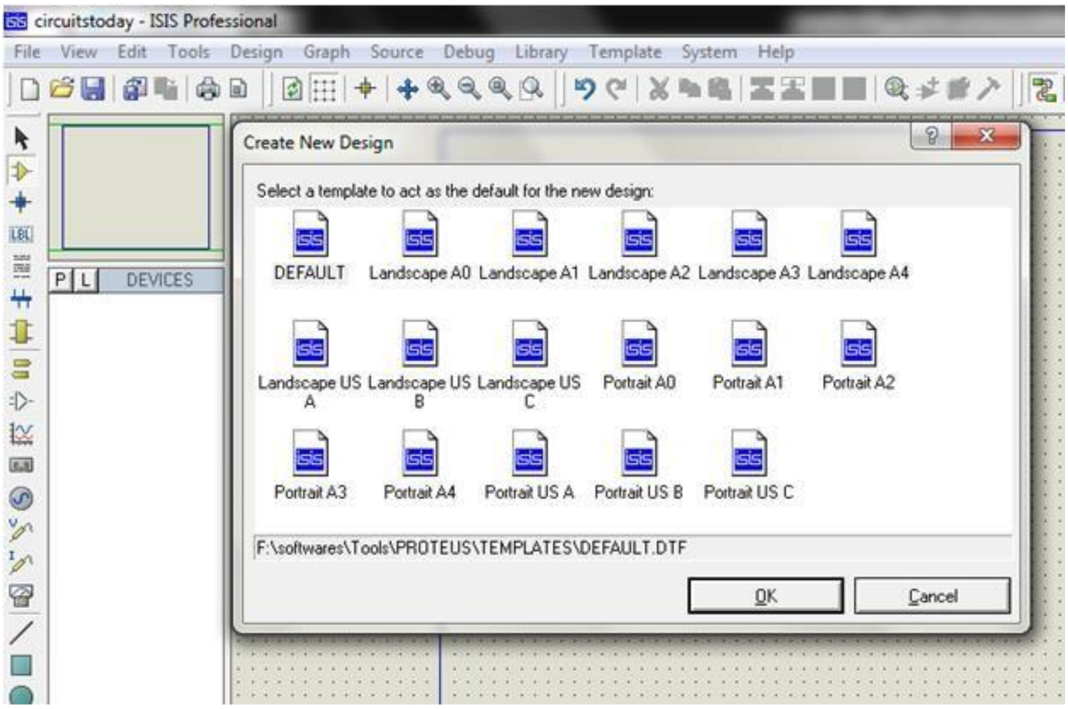


Figure 4.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

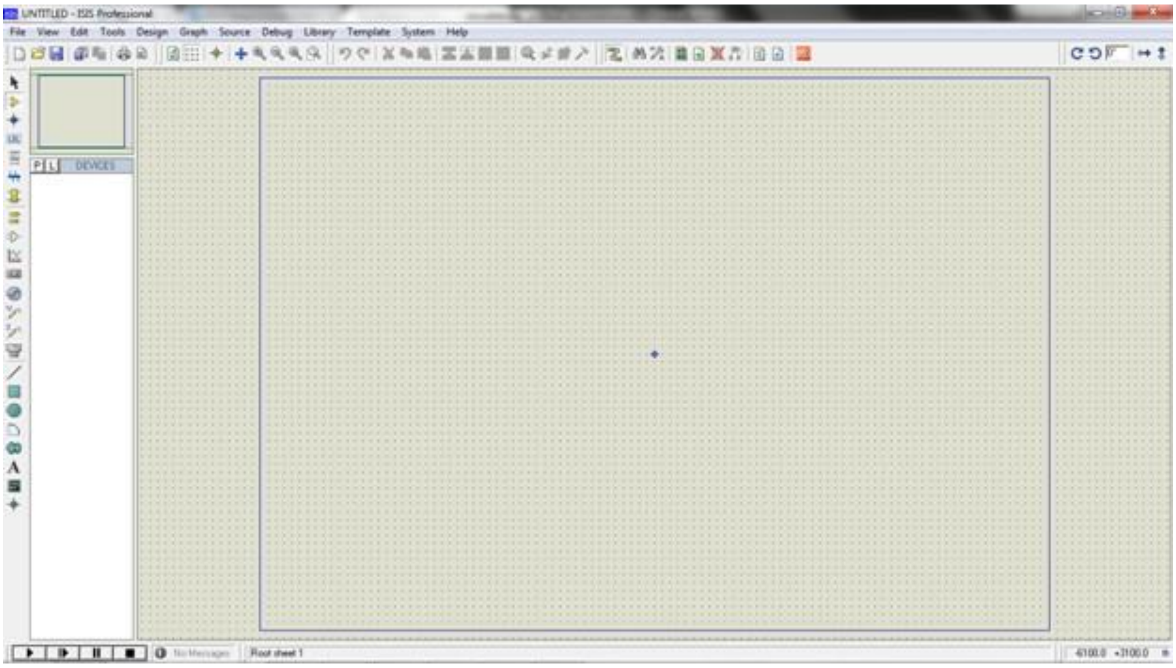


Figure 4.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

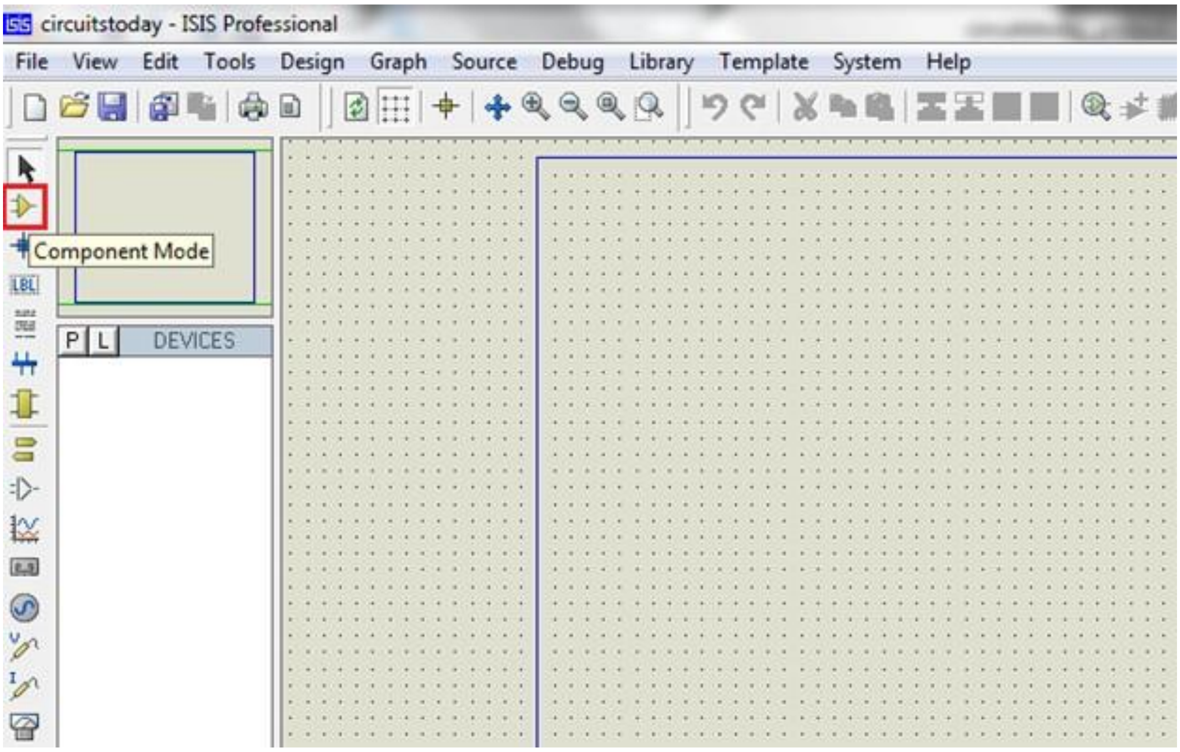


Figure 4.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

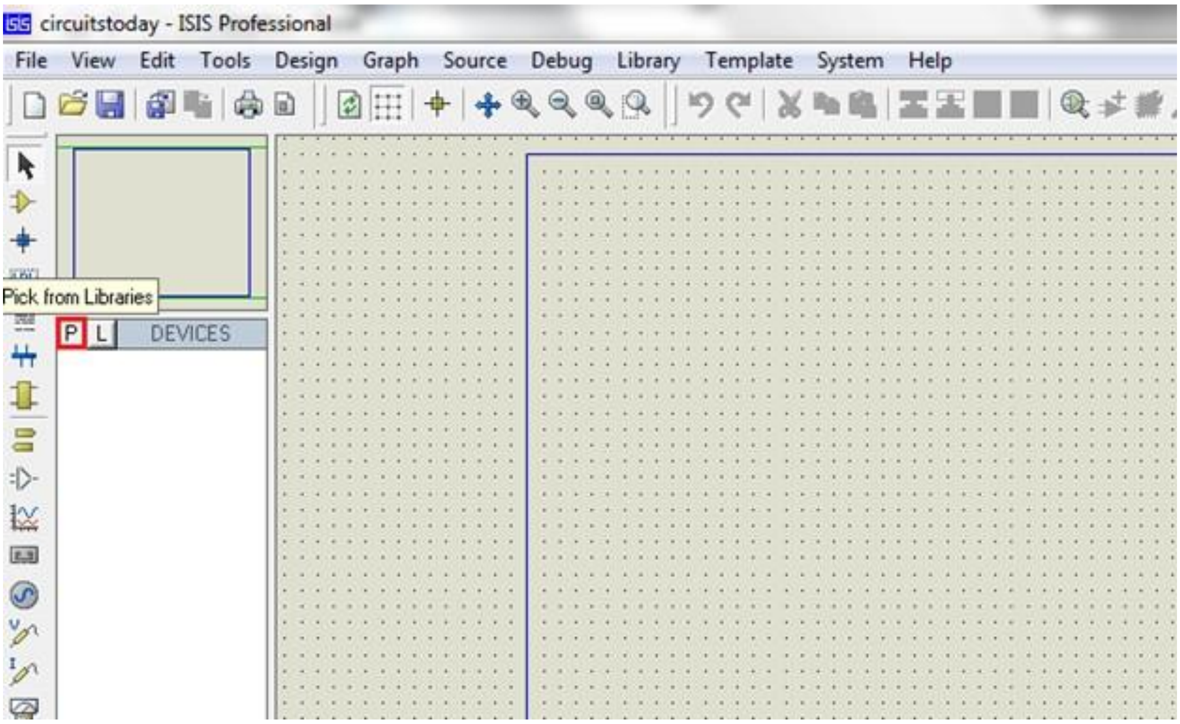


Figure 4.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

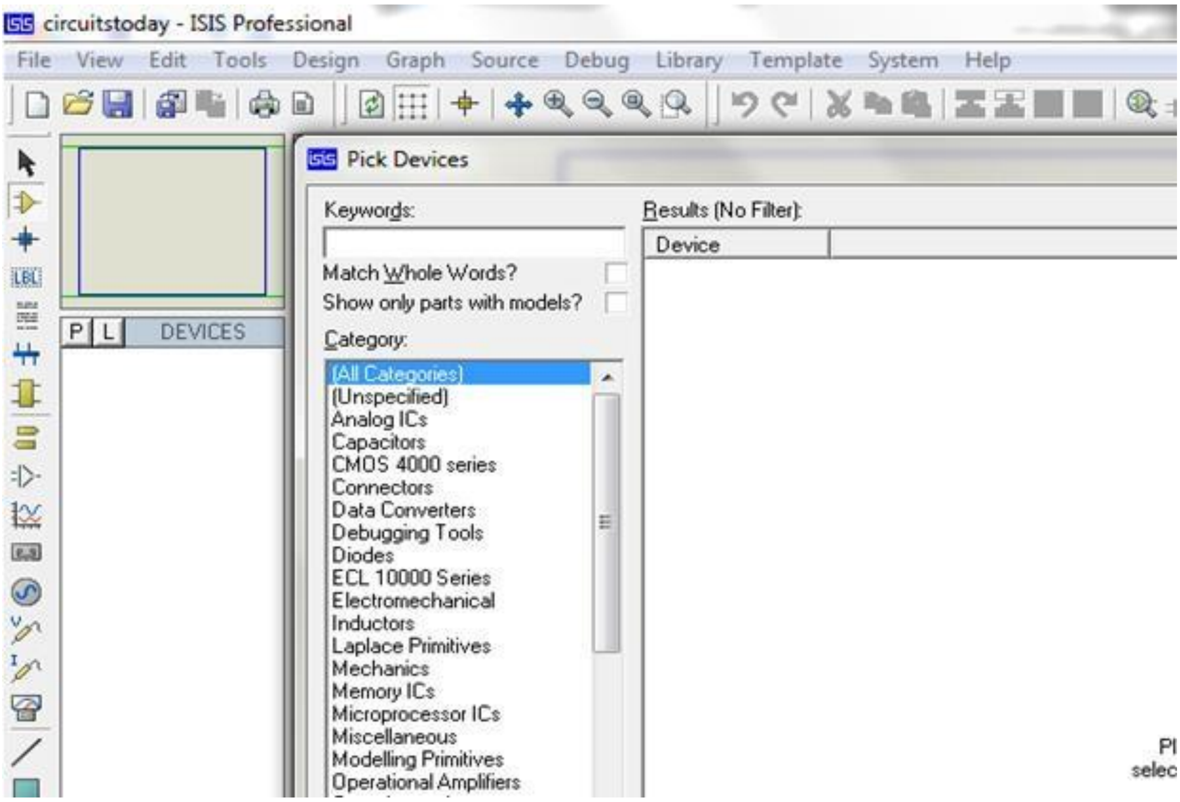


Figure 4.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

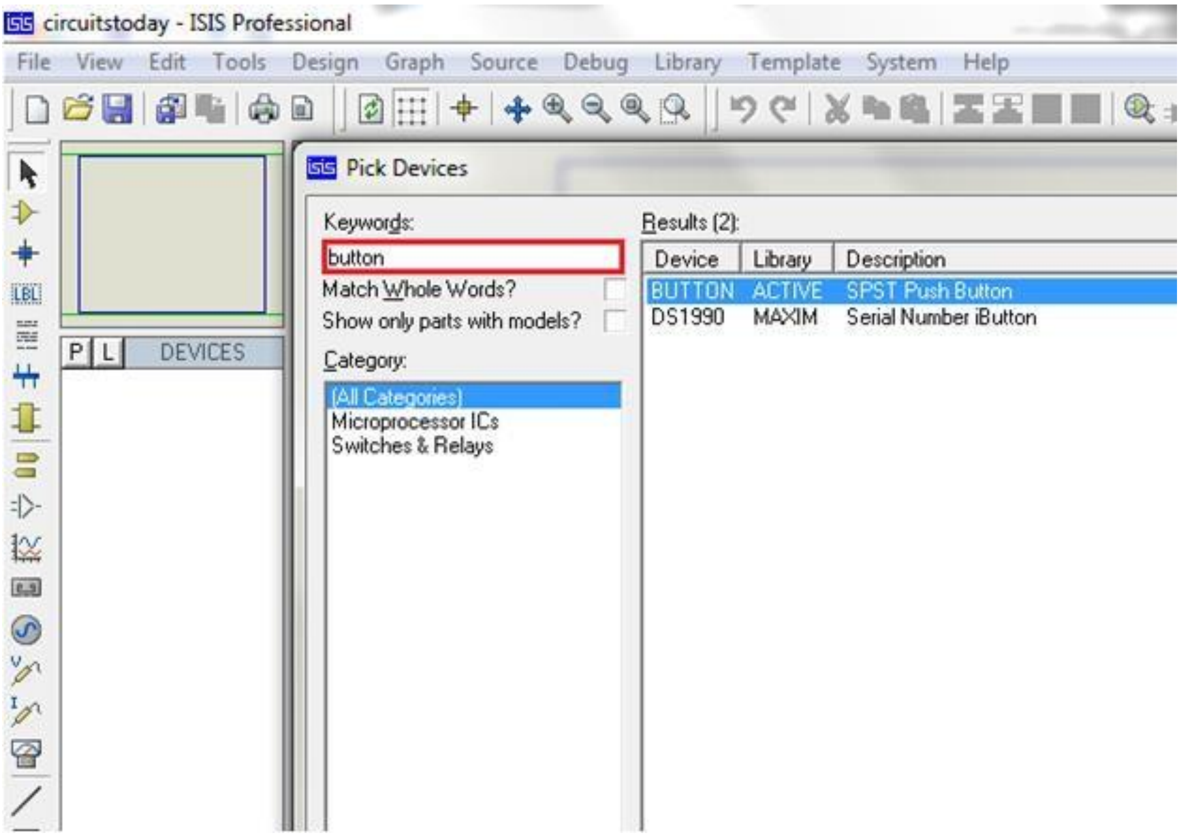


Figure 4.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

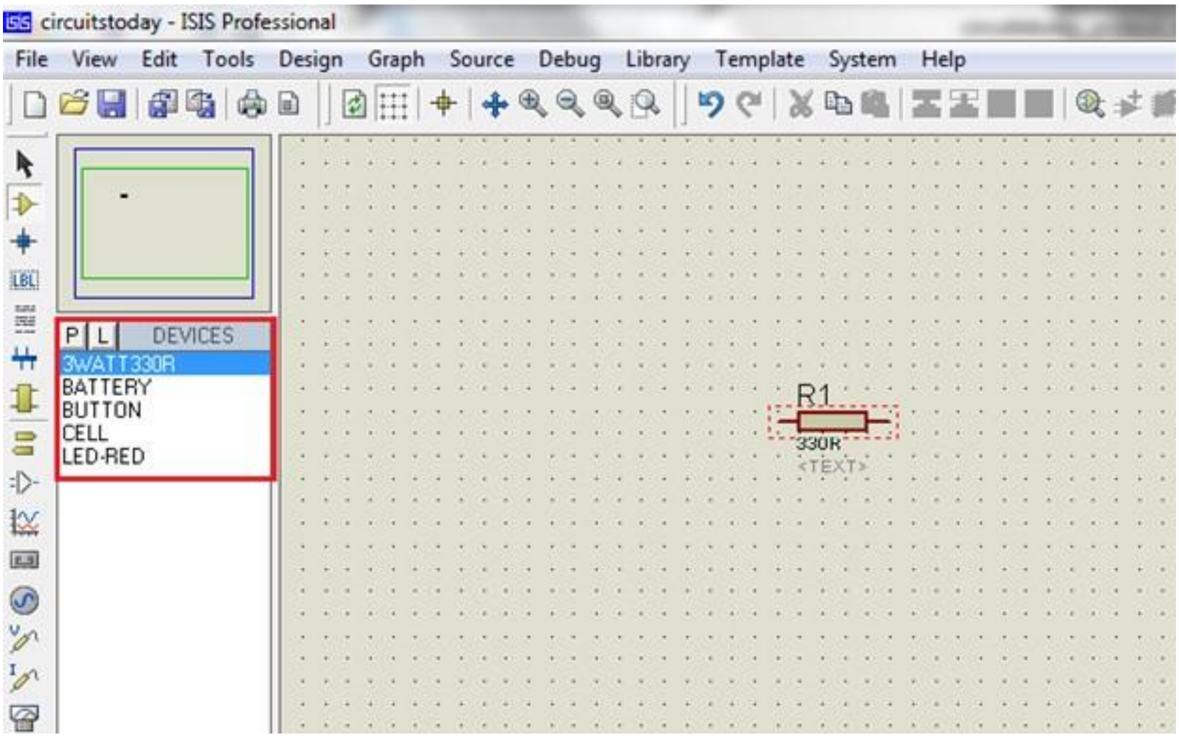


Figure 4.8 Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

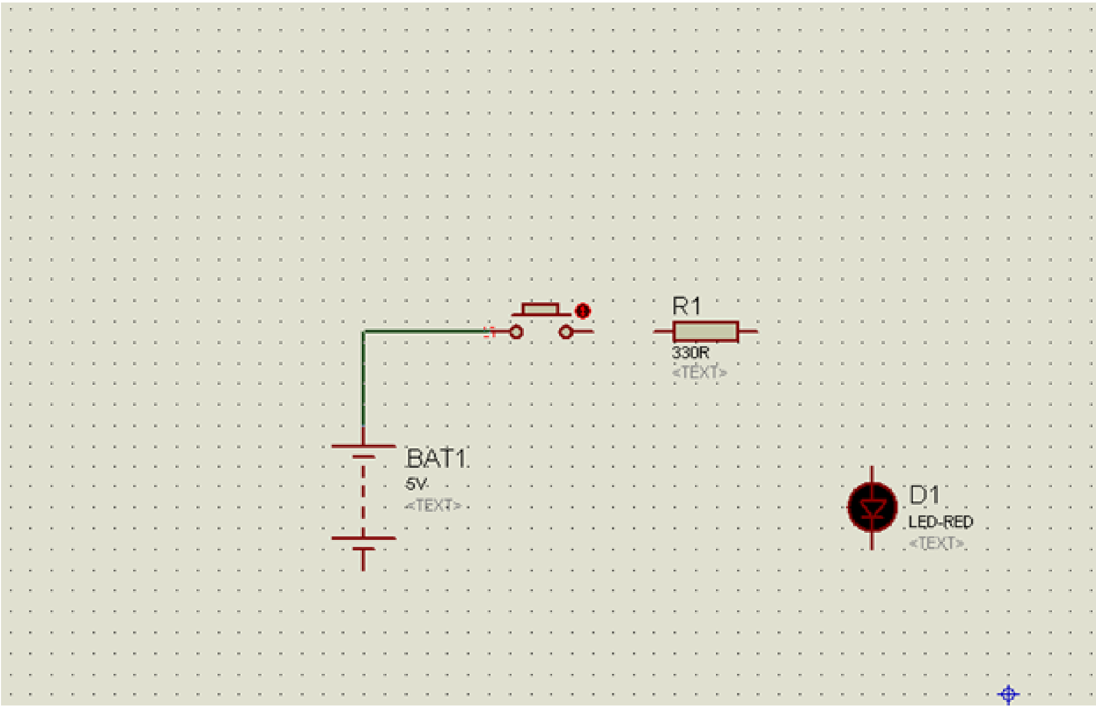


Figure 4.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

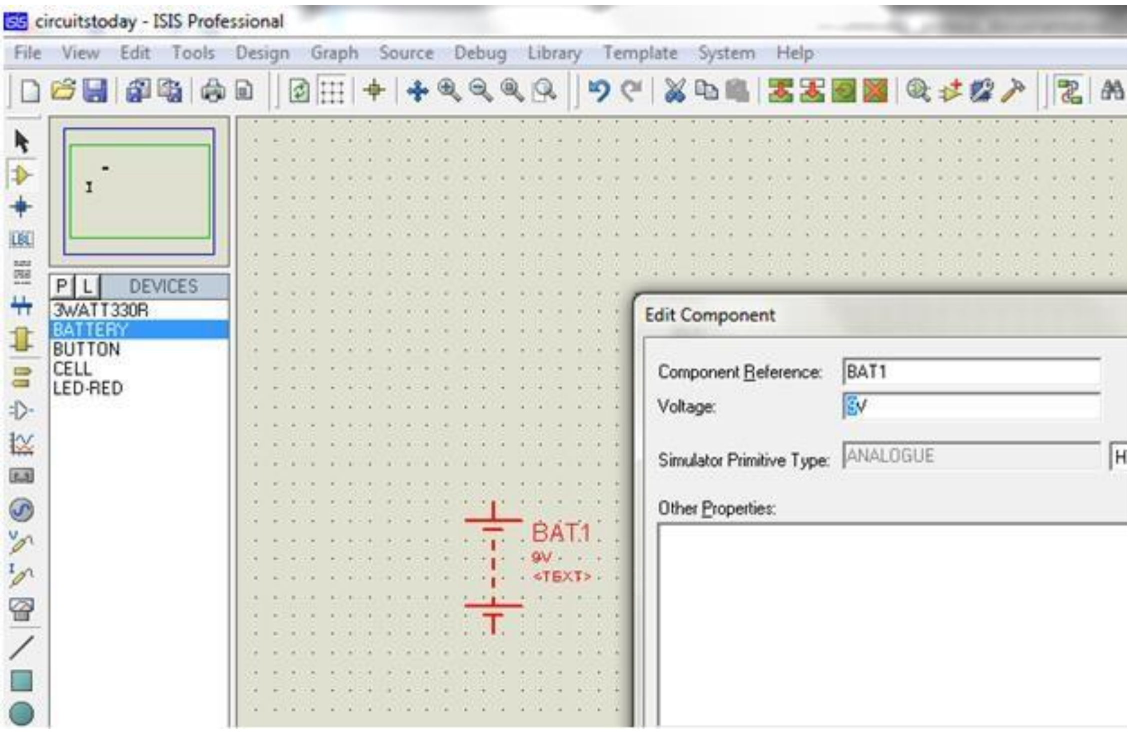


Figure 4.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

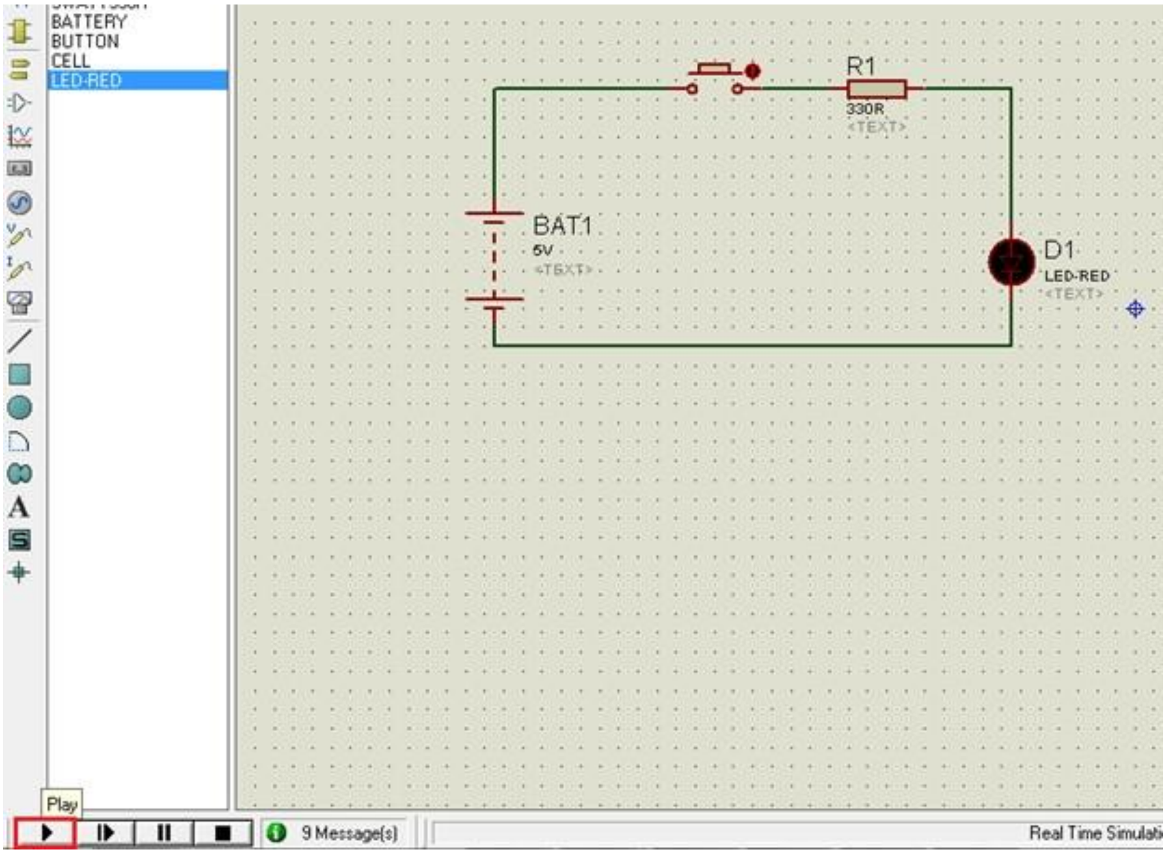


Figure 4.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

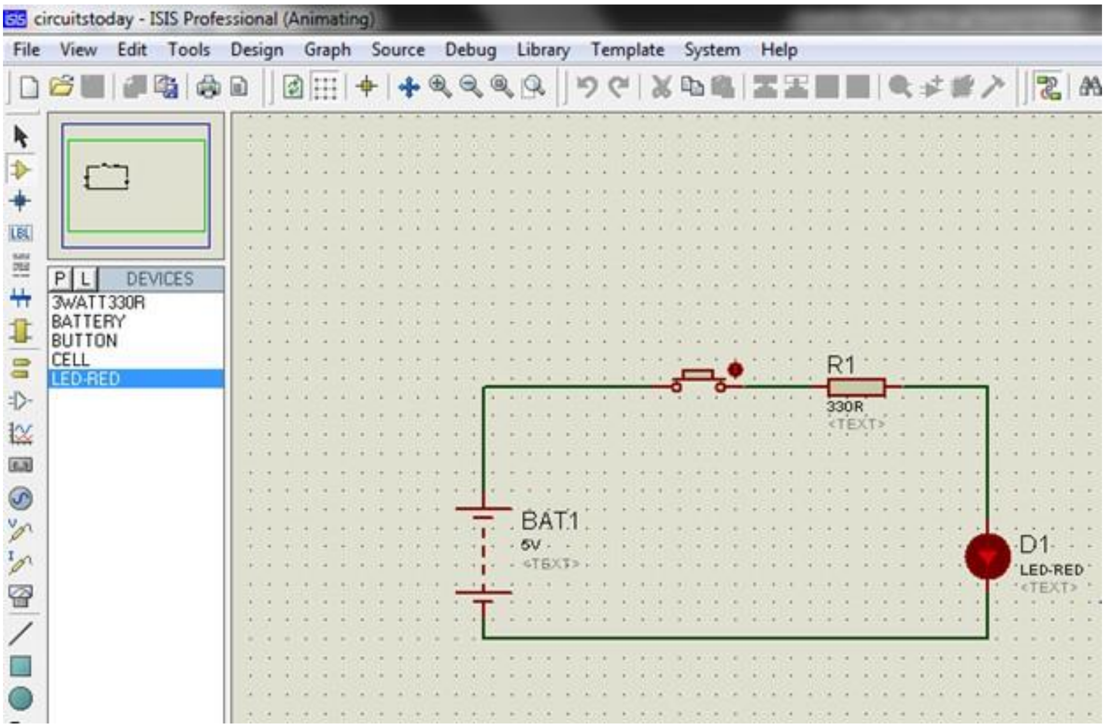


Figure 4.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

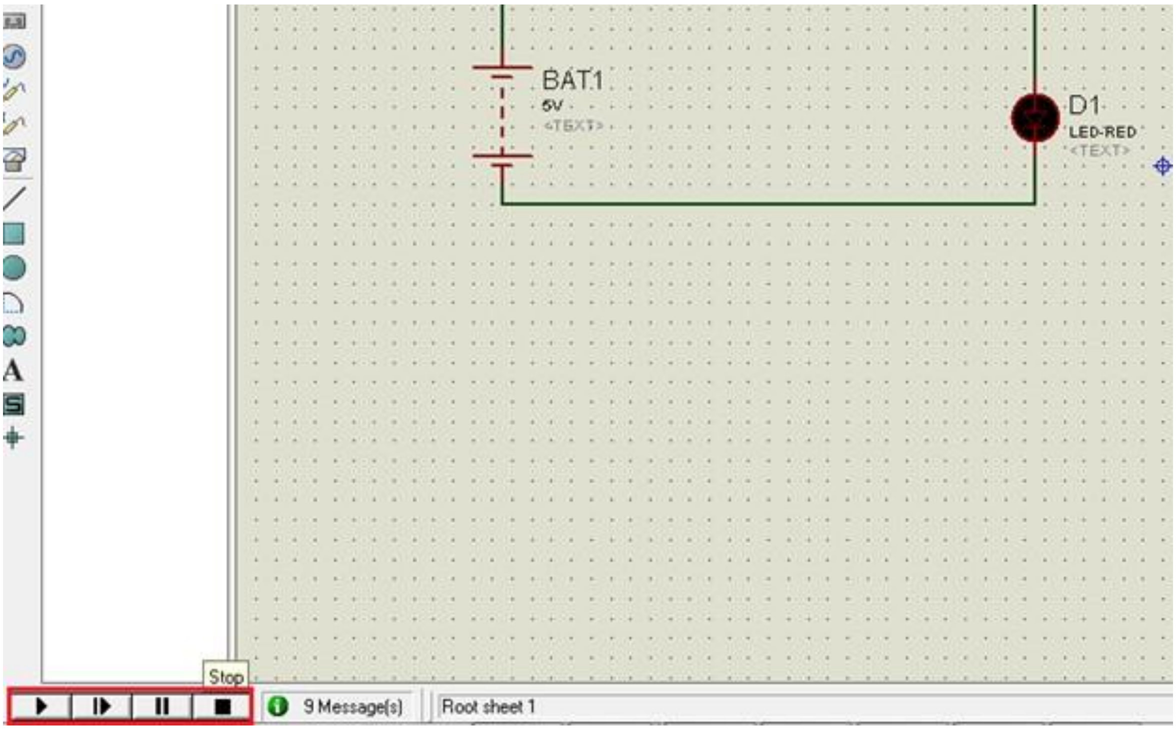


Figure 4.13 Simulation Step-Pause-Stop Buttons

4.3 Arduino IDE

The **Arduino Integrated Development Environment (IDE)** is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Arduino Pro IDE	
Developer(s)	Arduino Software
Preview release	v0.0.2 / 28 October 2019; 3 months ago[8]

Repository	github.com/arduino/Arduino
Written in	C, C++
Operating system	Windows, macOS, Linux
Platform	IA-32, x86-64, ARM
Type	Integrated development environment
License	LGPL or GPL license
Website	blog.arduino.cc/2019/10/18/arduino-pro-ide-alpha-preview-with-advanced-features/

In October 2019 the Arduino organization began providing early access to a new Arduino Pro IDE with debugging[9] and other advanced features.

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

4.3.1 Installation

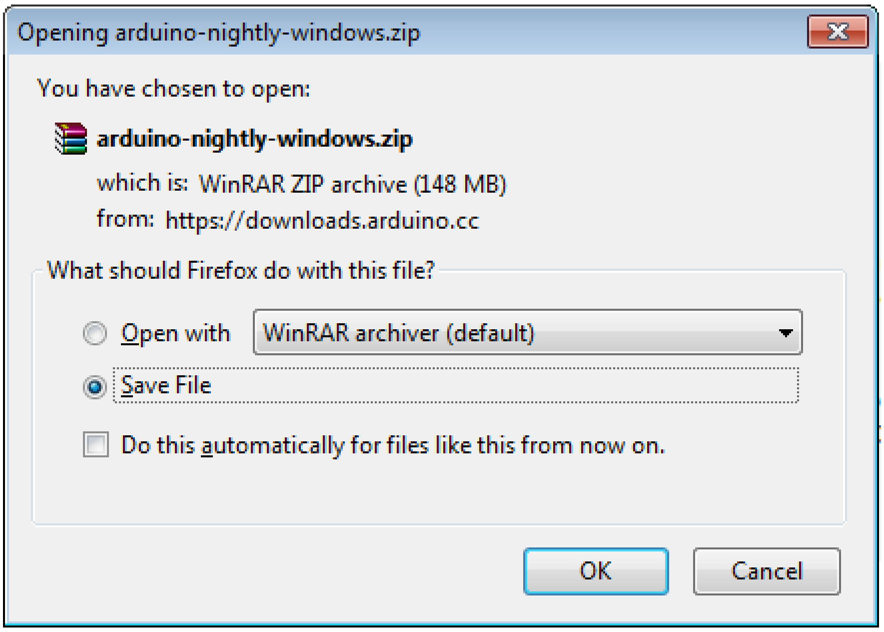
In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



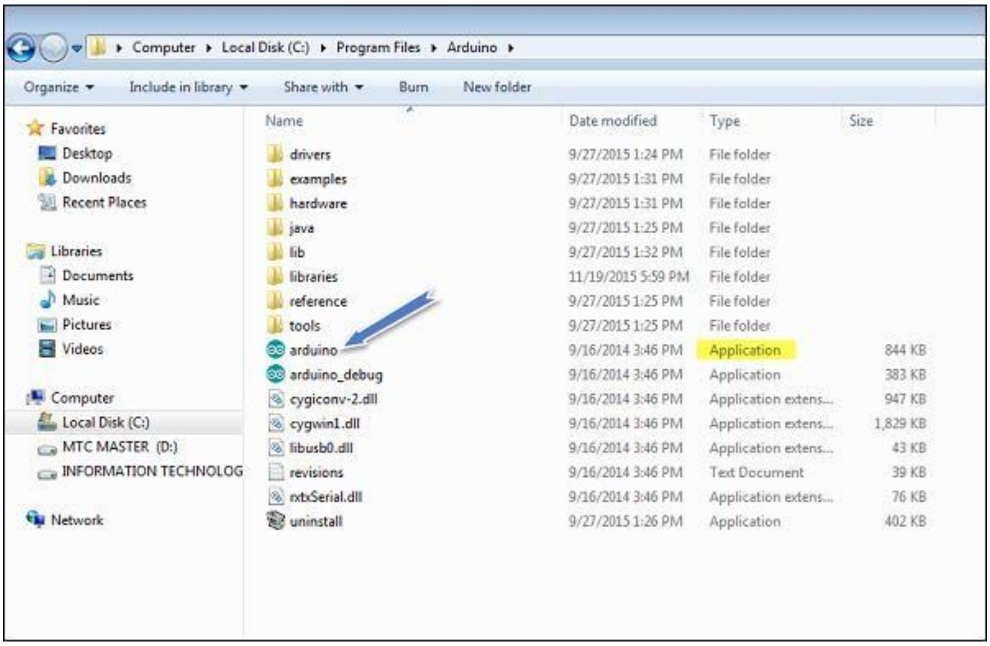
Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

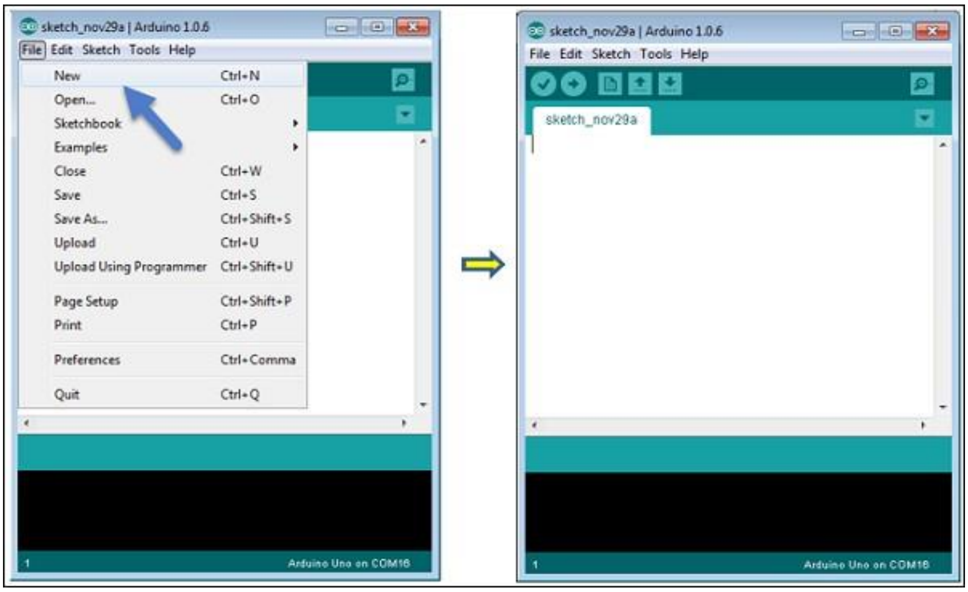


Step 5 − Open your first project.

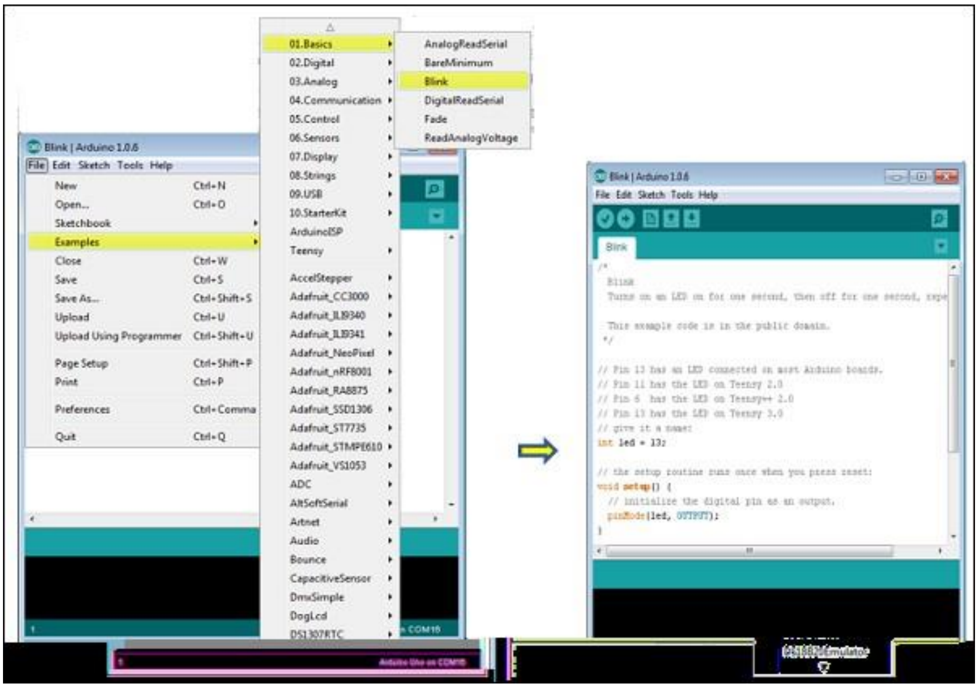
Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.



To open an existing project example, select File → Example → Basics → Blink.

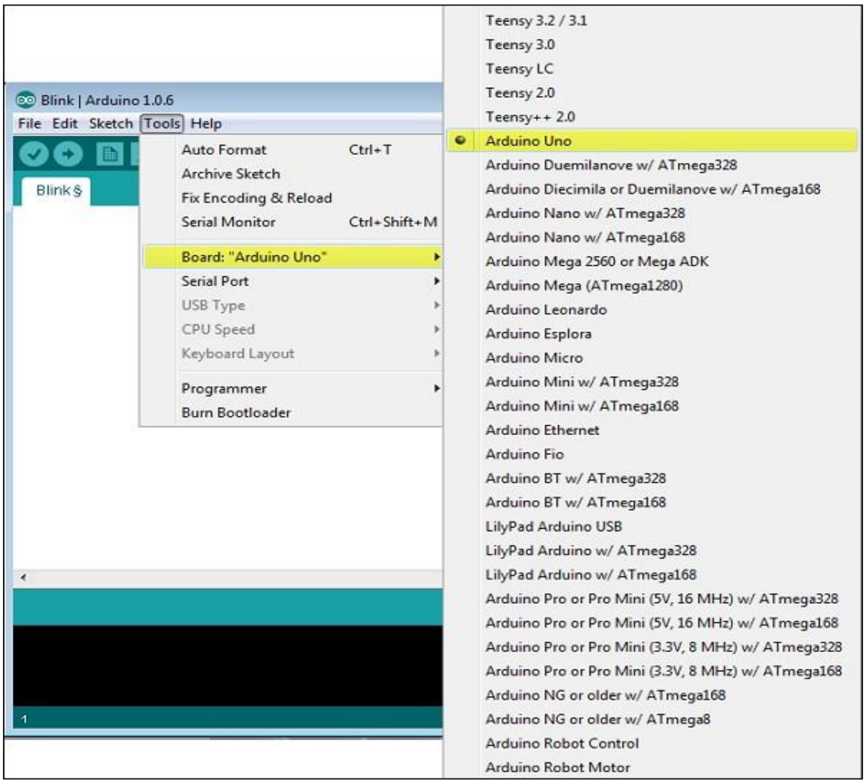


Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

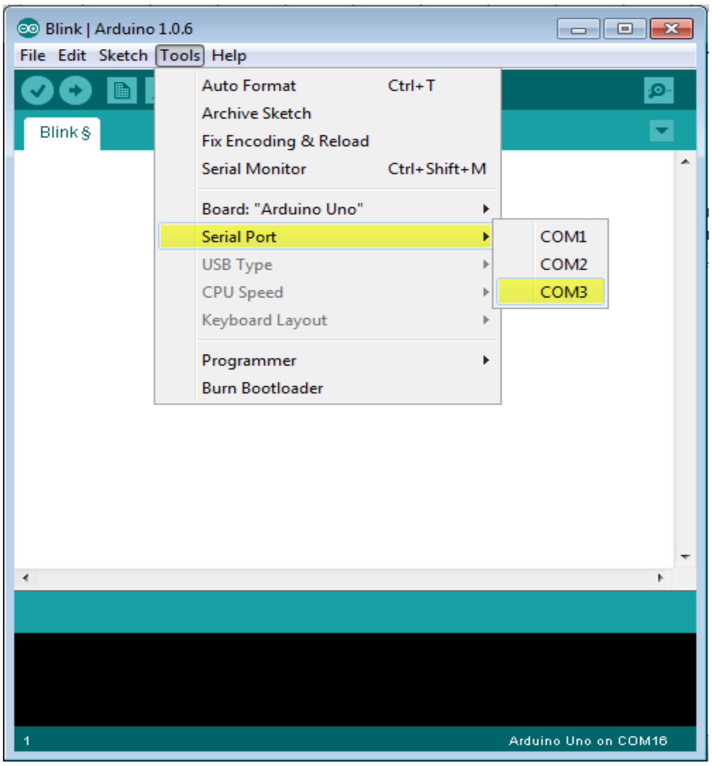
Go to Tools → Board and select your board.



Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

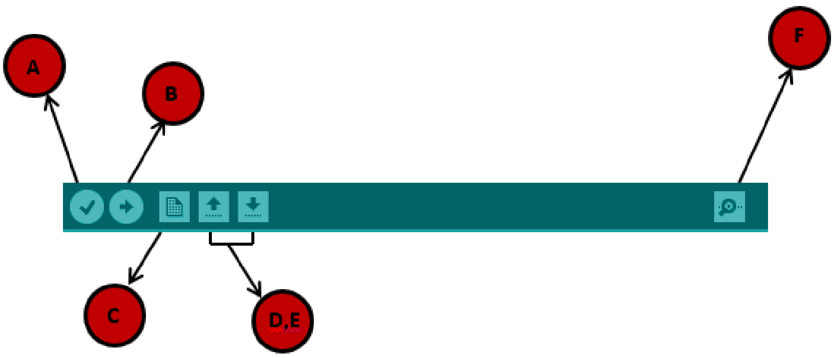
Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.



Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.



A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

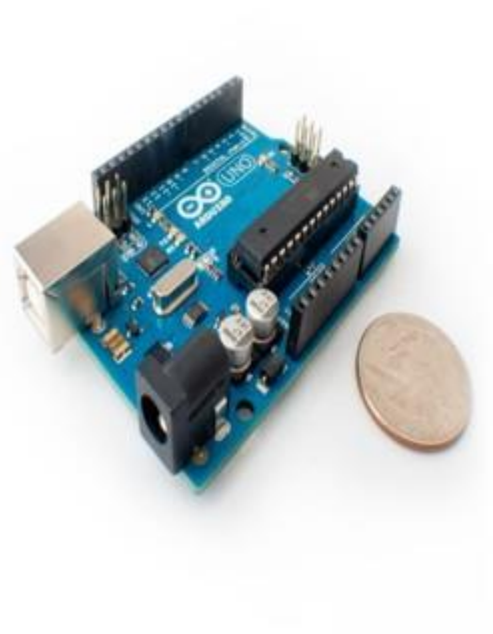
CHAPTER 5
HARDWARE

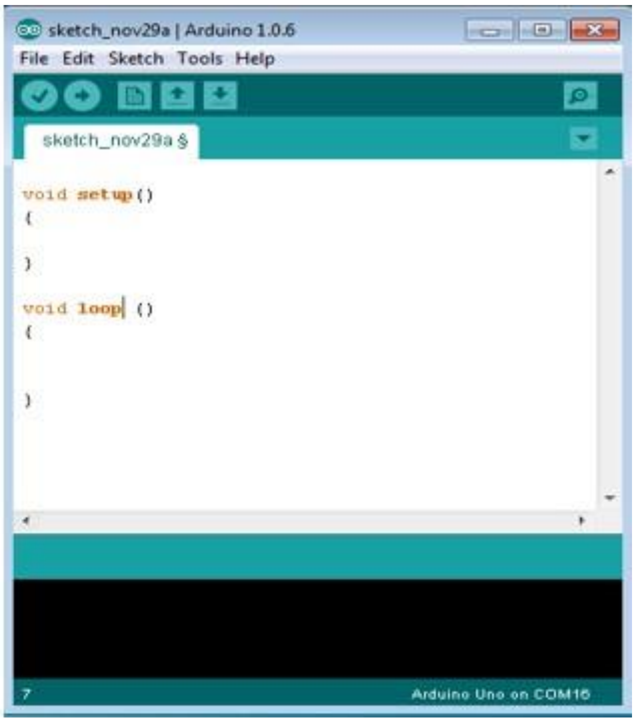
5.1 Arduino

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.





5.1.1 Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Table 5.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header

Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 5.2Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

5.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

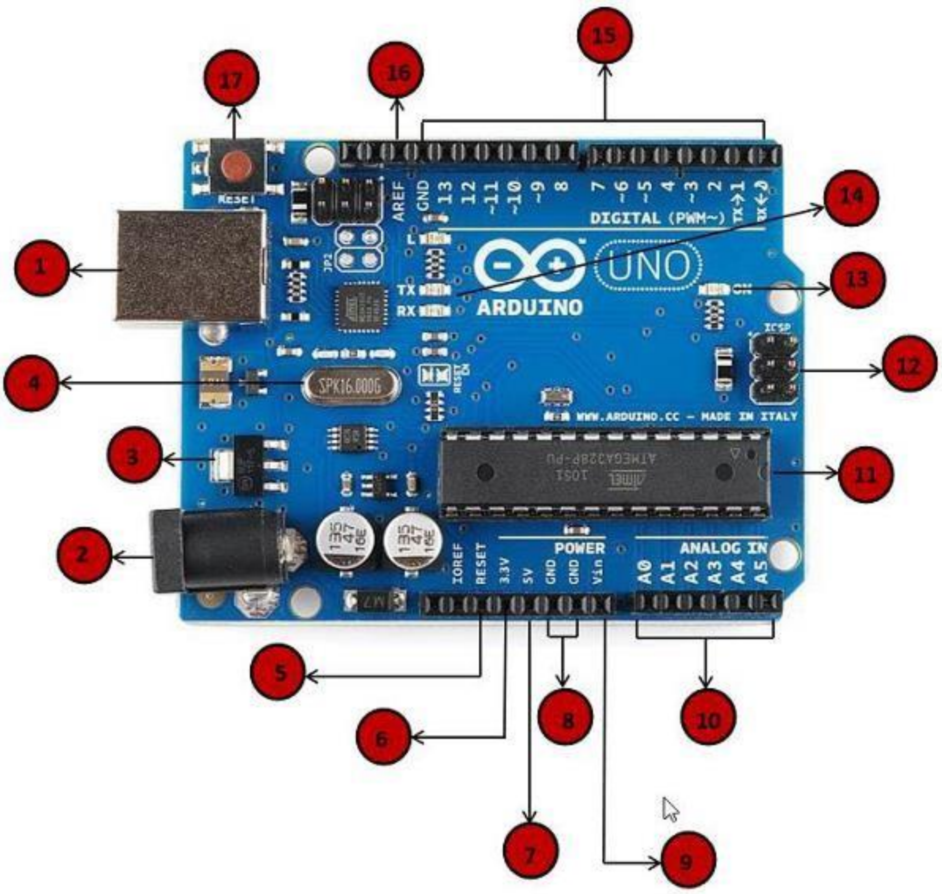
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

5.4 Arduino boards based on AT91SAM3X8E microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.



1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

5.2 Liquid Cristal Display

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

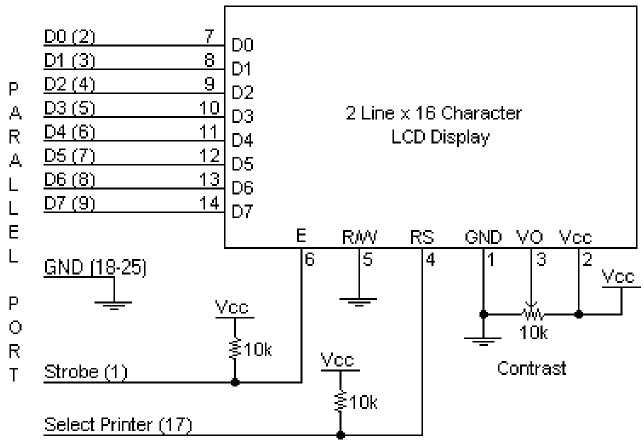
A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCDs connected to the contollers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines(RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

5.2.1 Description Of 16x2:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all-Parallel Ports. It however doesn't show the use of the Status Port as an input. So what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

Schematic Diagram



- Above is the quite simple schematic. The LCD panel's *Enable* and *Register Select* is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.
- We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.
- The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.

16 x 2 Alphanumeric LCD Module Features:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

16 x 2 Alphanumeric LCD Module Specifications:

Pin	Symbol	Level	Function
1	V_{SS}	-	Power, GND
2	V_{DD}	-	Power, 5V
3	Vo	-	Power, for LCD Drive
4	RS	H/L	Register Select Signal H: Data Input L: Instruction Input
5	R/W	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)
6	E	H,H->L	Enable
7-14	DB0-DB7	H/L	Data Bus; Software selectable 4- or 8-bit mode
15	NC	-	**NOT CONNECTED**
16	NC	-	**NOT CONNECTED**

FEATURES:

- 5 x 8 dots with cursor
- Built-in controller (KS 0066 or Equivalent)
- \+ 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)
- N.V. optional for + 3V power supply

Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are:

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

Figure Address locations for a 1x16 line LCD

Even limited to character based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8,16,20,24,32 and 40 characters are all standard, in one, two and four line versions.

Several different LC technologies exists. “supertwist” types, for example, offer Improved contrast and viewing angle over the older “twisted nematic” types. Some modules are available with back lighting, so that they can be viewed in dimly-lit conditions. The back lighting may be either “electro-luminescent”, requiring a high voltage inverter circuit, or simple LED illumination.

PIN DESCRIPTION:

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections).

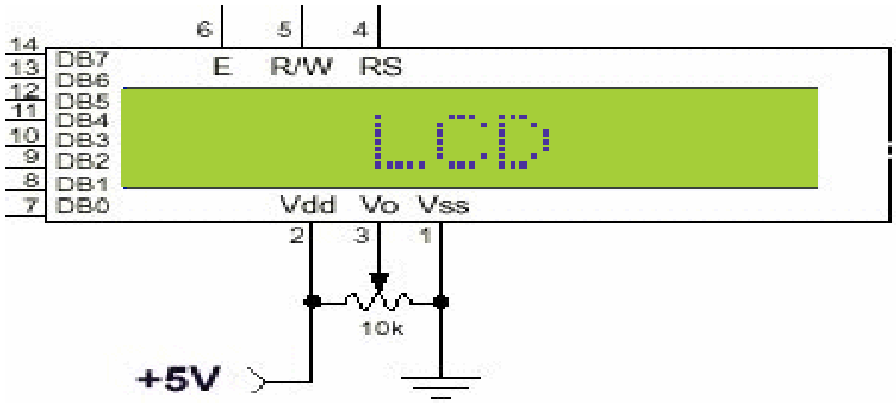


Figure 5.1 Pin diagram of 1x16 lines LCD

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Figure 5.2 Pin specifications

CONTROL LINES:

EN: Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS: Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW: Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands, so RW will almost always be low. Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Logic status on control lines:

• E - 0 Access to LCD disabled

- 1 Access to LCD enabled

• R/W - 0 Writing data to LCD

- 1 Reading data from LCD

• RS - 0 Instructions

- 1 Character

Writing data to the LCD:

1) Set R/W bit to low

2) Set RS bit to logic 0 or 1 (instruction or character)

3) Set data to data lines (if it is writing)

4) Set E line to high

5) Set E line to low

Read data from data lines (if it is reading) on LCD:

1) Set R/W bit to high

2) Set RS bit to logic 0 or 1 (instruction or character)

3) Set data to data lines (if it is writing)

4) Set E line to high

5) Set E line to low

Entering Text:

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the d.i.l. switch pack with a of re-wiring is necessary.

LCD Commands:

There are some present commands instructions in LCD, which we need to send to LCD through some microcontroller. Some important command instructions are given below:

Table 5.5 Commands

Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left
0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

Description

This is the circuit diagram of a simple **corrosion free water level indicator** for home and industries. In fact the level of any conductive non corrosive liquids can be measured using this circuit. The circuit is based on 5 transistor switches. Each transistor is switched on to drive the corresponding LED , when its base is supplied with current through the water through the electrode probes.

One electrode probe is (F) with 6V AC is placed at the bottom of tank. Next probes are placed step by step above the bottom probe. When water is rising the base of each transistor gets electrical connection to 6V AC through water and the corresponding probe. Which in turn makes the transistors conduct to glow LED and indicate the level of water. The ends of probes are connected to corresponding points in the circuit as shown in circuit diagram. Insulated Alluminium wires with end insulation removed will do for the probe. Arrange the probes in order on a PVC pipe according to the depth and immerse it in the tank.AC voltage is use to prevent electrolysis at the probes. So this setup will last really long. I guarantee at least a 2 years of maintenance free operation. That’s what I got and is still going.

Components

T1 – T5 BC 548 or 2N2222 Transistors

R1-R5 2.2K 1/4 W Resistors

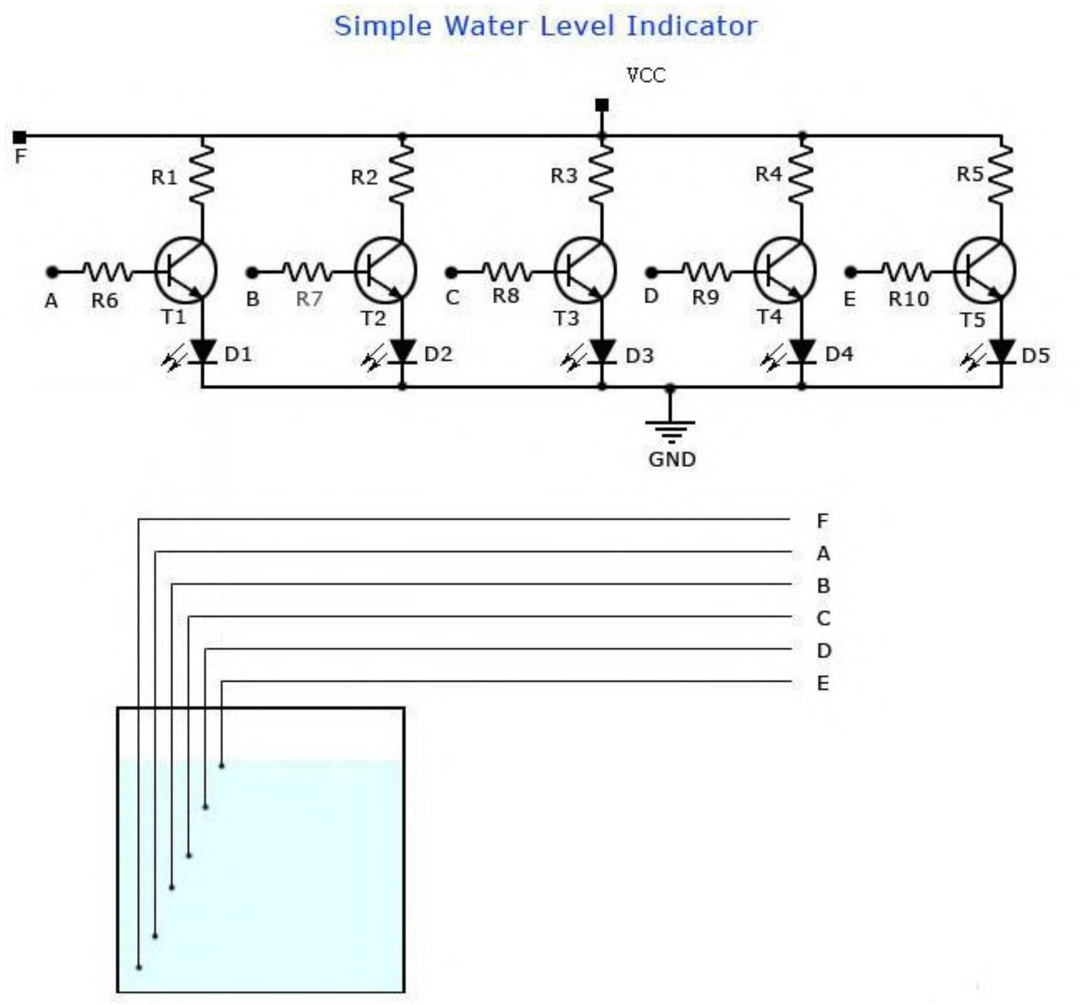
R6-R10 22K 1/4 W Resistors

D1 – D5 LED’s (color your choice)

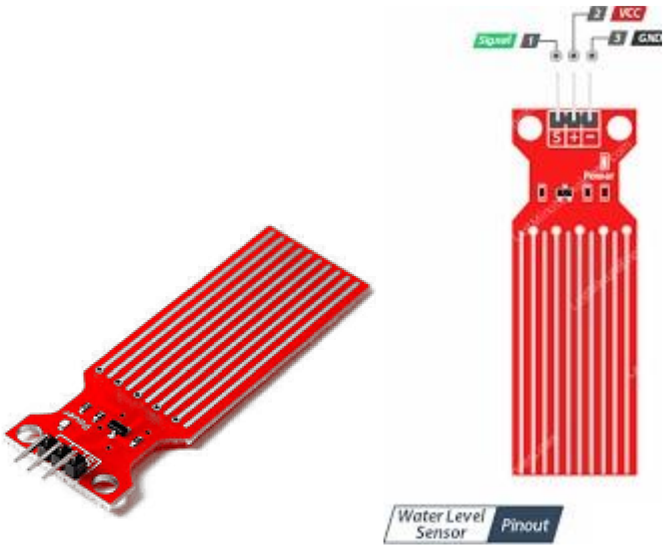
Notes:

Use a transformer with 6V 500 mA output for power supply.Do not use a rectifier! we need pure AC. Use good quality insulated Aluminum wire for probes.If Aluminum wires are not available try Steel or Tin.Copper is the worst.Try the circuit first on a bread board and if not working properly, make adjustments with the resistance values .This is often needed because conductivity of water changes slightly from place to place.The type number of the transistors used here are not critical and any small signal NPN transistor will do the job. Few other suitable type numbers are BC546, BC107, PN2222, BC337, BF494, ZTX300, BEL187 etc. The circuit can be enclosed in a plastic box with holes for revealing the LEDs .

Water Level Indicator Circuit Diagram and Sensor Arrangement.



Water Level Indicator



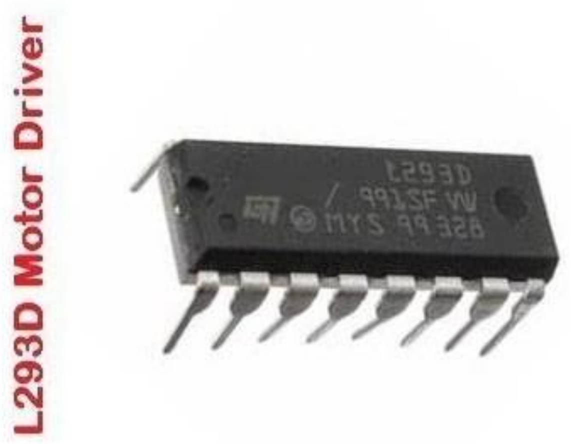
L293D

L293D is basically a high current dual motor driver/controller Integrated Circuit (IC). It is able to drive load having current up to 1A at the voltage ranging from 4.5V to 36V. Motor driver usually act as current amplifier because they receive a low current signal as an input and provides high current signal at the output.

Motors usually operates on this higher current. L-293D has to builtin H-Bridge driver circuits and is able to control two DC motors at a time in both clockwise and counter clockwise direction. It has two enable pins and they should be kept high in order to control the motor. By changing the polarity of applied signal motor can be rotated in either clockwise or counter clockwise direction. If L 293D enable pin is high, its corresponding driver will provide the desired out. If the enable pin is low, there will be no output. L-293D has different features including internal ESD protection, large voltage supply range, large output current per channel, high noise immunity input etc. L 293D plays a vital role in electronics era and has several different applications e.g relay drivers, DC motor drivers, stepping motor drivers etc. The further detail about L 293D motor driver/controller will be given later in this tutorial.

Introduction to L293D

L293D is basically a motor driver or controller. It has two builtin H-bridge circuits which are able to control two DC motors simultaneously in both clockwise and counter clockwise direction. It acts as an current high amplifier because it take low current signal at its input and provides higher current signal at the output in order to drives different load e.g stepper motor & DC motors. Its features include large inpu voltage supply range, large output current high noise immunity input signals etc. Its common real life applications include stepping motor drivers, relay drivers, DC motor drivers etc. L-293D motor driver/controller is shown in the figure given below.



1. L293D Pins

- L-293D has sixteen (16) pins, having different individual functions.
- All of these sixteen pins along with their serial no, are given in the table shown below.

2. L293D Pin Functions

- L 293D each pin has different tasks to perform when they are in working condition.
- The tasks assigned to each pin are given in the table shown below.

L293D Pins	
Pin. No	Pin Name
1	Enable 1,2
2	Input 1
3	Output 1
4	GND
5	GND
6	Output 2
7	Input 2
8	Vcc2
9	Enable 3,4
10	Input 3
11	Output 3
12	GND
13	GND
14	Output put 4
15	Input 4
16	Vcc1

3. L293D Pinout

- Pinout diagram of any device presents the pins configuration through a completely labelled diagram.
- L293D pinout diagram is shown in the figure given below.

DC Motors

The brushed DC motor is one of the earliest motor designs. Today, it is the motor of choice in the majority of variable speed and torque control applications.

Advantages

- Easy to understand design
- Easy to control speed
- Easy to control torque
- Simple, cheap drive design

Easy to understand design

The design of the brushed DC motor is quite simple. A permanent magnetic field is created in the stator by either of two means:

- Permanent magnets
- Electro-magnetic windings

If the field is created by permanent magnets, the motor is said to be a "permanent magnet DC motor" (PMDC). If created by electromagnetic windings, the motor is often said to be a "shunt wound DC motor" (SWDC). Today, because of cost-effectiveness and reliability, the PMDC motor is the motor of choice for applications involving fractional horsepower DC motors, as well as most applications up to about three horsepower.

At five horsepower and greater, various forms of the shunt wound DC motor are most commonly used. This is because the electromagnetic windings are more cost effective than permanent magnets in this power range.

Caution: If a DC motor suffers a loss of field (if for example, the field power connections are broken), the DC motor will immediately begin to accelerate to the top speed which the loading will allow. This can result in the motor flying apart if the motor is lightly loaded. The possible loss of field must be accounted for, particularly with shunt wound DC motors.

Opposing the stator field is the armature field, which is generated by a changing electromagnetic flux coming from windings located on the rotor. The magnetic poles of the armature field will attempt to line up with the opposite magnetic poles generated by the stator field. If we stopped the design at this point, the motor would spin until the poles were opposite one another, settle into place, and then stop -- which would make a pretty useless motor!

However, we are smarter than that. The section of the rotor where the electricity enters the rotor windings is called the commutator. The electricity is carried between the rotor and the stator by conductive graphite-copper brushes (mounted on the rotor) which contact rings on stator. Imagine power is supplied:

The motor rotates toward the pole alignment point. Just as the motor would get to this point, the brushes jump across a gap in the stator rings. Momentum carries the motor forward over this gap. When the brushes get to the other side of the gap, they contact the stator rings again and -- the polarity of the voltage is reversed in this set of rings! The motor begins accelerating again, this time trying to get to the opposite set of poles. (The momentum has carried the motor past the original pole alignment point.) This continues as the motor rotates.

In most DC motors, several sets of windings or permanent magnets are present to smooth out the motion.

Easy to control speed

Controlling the speed of a brushed DC motor is simple. The higher the armature voltage, the faster the rotation. This relationship is linear to the motor's maximum speed.

The maximum armature voltage which corresponds to a motor's rated speed (these motors are usually given a rated speed and a maximum speed, such as 1750/2000 rpm) are available in certain standard voltages, which roughly increase in conjuntion with horsepower. Thus, the smallest industrial motors are rated 90 VDC and 180 VDC. Larger units are rated at 250 VDC and sometimes higher. Specialty motors for use in mobile applications are rated 12, 24, or 48 VDC. Other tiny motors may be rated 5 VDC.

Most industrial DC motors will operate reliably over a speed range of about 20:1 -- down to about 5-7% of base speed. This is much better performance than the comparible AC motor. This is partly due to the simplicity of control, but is also partly due to the fact that most industrial DC motors are designed with variable speed operation in mind, and have added heat dissipation features which allow lower operating speeds.

Easy to control torque

In a brushed DC motor, torque control is also simple, since output torque is proportional to current. If you limit the current, you have just limited the torque which the motor can achieve. This makes this motor ideal for delicate applications such as textile manufacturing.

Simple, cheap drive design

The result of this design is that variable speed or variable torque electronics are easy to design and manufacture. Varying the speed of a brushed DC motor requires little more than a large enough potentiometer. In practice, these have been replaced for all but sub-fractional horsepower applications by the SCR and PWM drives, which offer relatively precisely control voltage and current. Common DC drives are available at the low end (up to 2 horsepower) for under US$100 -- and sometimes under US$50 if precision is not important.

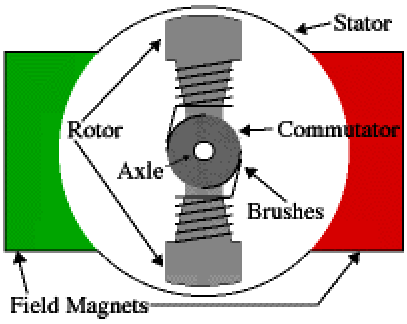
Large DC drives are available up to hundreds of horsepower. However, over about 10 horsepower careful consideration should be given to the price/performance tradeoffs with AC inverter systems, since the AC systems show a price advantage in the larger systems. (But they may not be capable of the application's performance requirments).

Disadvantages

- Expensive to produce
- Can't reliably control at lowest speeds
- Physically larger
- High maintenance
- Dust

WORKING OF DC MOTOR

In any electric motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. As you are well aware of from playing with magnets as a kid, opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion.



Principle

When a rectangular coil carrying current is placed in a magnetic field, a torque acts on the coil which rotates it continuously.

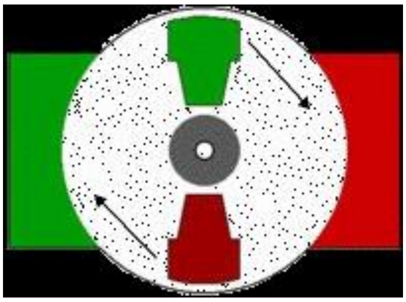
When the coil rotates, the shaft attached to it also rotates and thus it is able to do mechanical work.

Every DC motor has six basic parts -- axle, rotor (a.k.a., armature), stator, commutator, field magnet(s), and brushes. In most common DC motors (and all that BEAMers will see), the

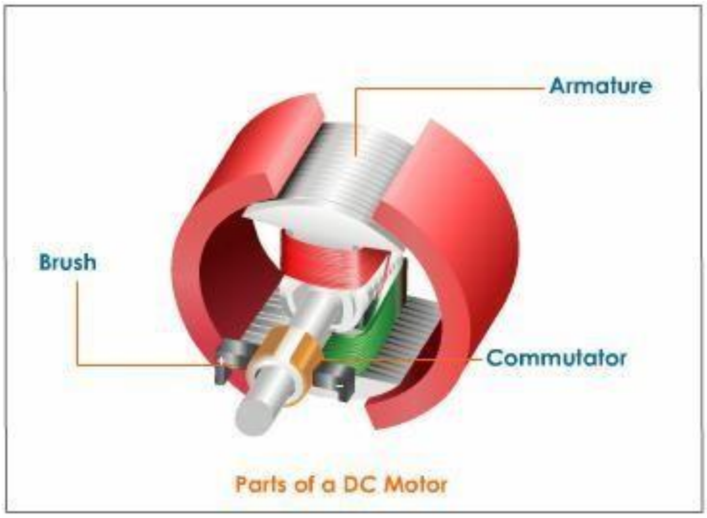
external magnetic field is produced by high-strength permanent magnets[1]. The stator is the stationary part of the motor -- this includes the motor casing, as well as two or more permanent magnet pole pieces. The rotor (together with the axle and attached commutator) rotate with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a common motor layout -- with the rotor inside the stator (field) magnets.

The geometry of the brushes, commentator contacts, and rotor windings are such that when power is applied, the polarities of the energized winding and the stator magnet(s) are misaligned, and the rotor will rotate until it is almost aligned with the stator's field magnets. As the rotor reaches alignment, the brushes move to the next commentator contacts, and energize the next winding. Given our example two-pole motor, the rotation reverses the direction of current through the rotor winding, leading to a "flip" of the rotor's magnetic field, driving it to continue rotating.

In real life, though, DC motors will always have more than two poles (three is a very common number). In particular, this avoids "dead spots" in the commutator. You can imagine how with our example two-pole motor, if the rotor is exactly at the middle of its rotation (perfectly aligned with the field magnets), it will get "stuck" there. Meanwhile, with a two-pole motor, there is a moment where the commutator shorts out the power supply (i.e., both brushes touch both commutator contacts simultaneously). This would be bad for the power supply, waste energy, and damage motor components as well. Yet another disadvantage of such a simple motor is that it would exhibit a high amount of torque "ripple" (the amount of torque it could produce is cyclic with the position of the rotor).



Construction and Working



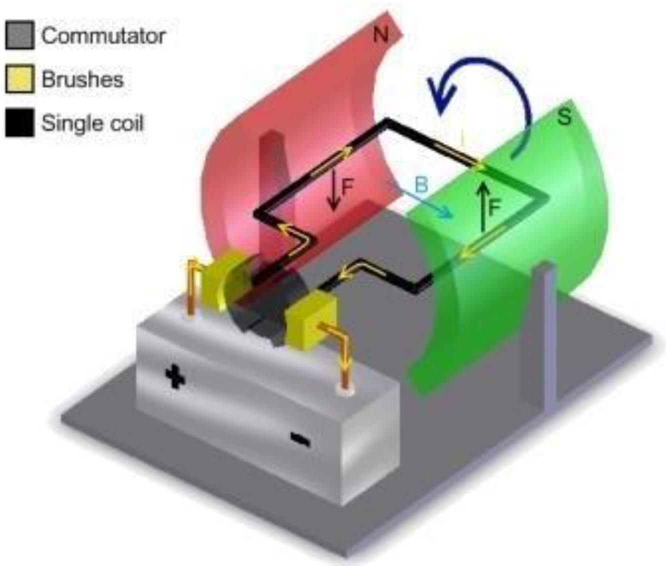
Parts of a DC Motor

Armature

A D.C. motor consists of a rectangular coil made of insulated copper wire wound on a soft iron core. This coil wound on the soft iron core forms the armature. The coil is mounted on an axle and is placed between the cylindrical concave poles of a magnet.

Commutator

A commutator is used to reverse the direction of flow of current. Commutator is a copper ring split into two parts C_1 and C_2. The split rings are insulated form each other and mounted on the axle of the motor. The two ends of the coil are soldered to these rings. They rotate along with the coil. Commutator rings are connected to a battery. The wires from the battery are not connected to the rings but to the brushes which are in contact with the rings.



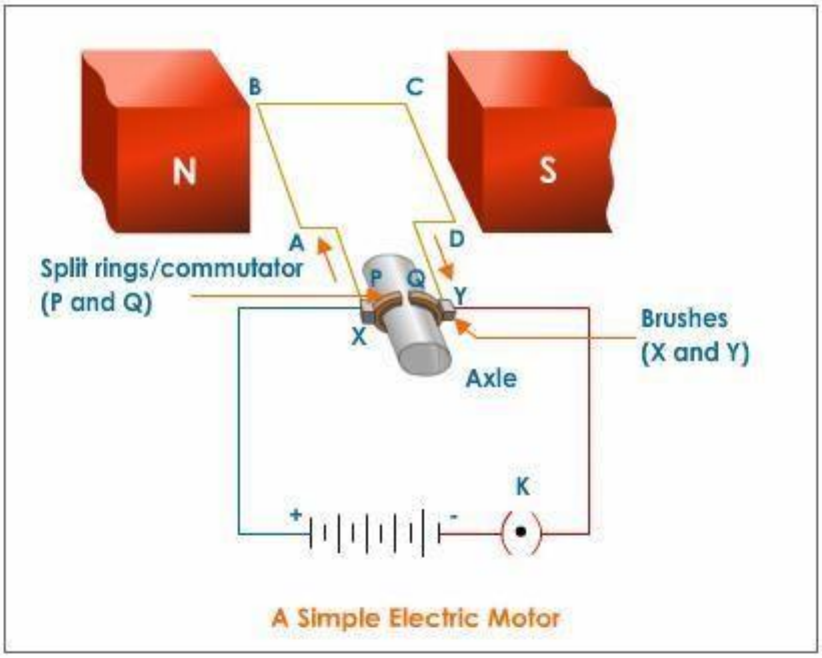
Brushes

Two small strips of carbon, known as brushes press slightly against the two split rings, and the split rings rotate between the brushes.

The carbon brushes are connected to a D.C. source.

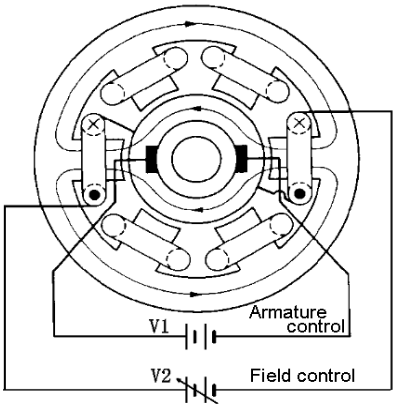
Working of a DC Motor

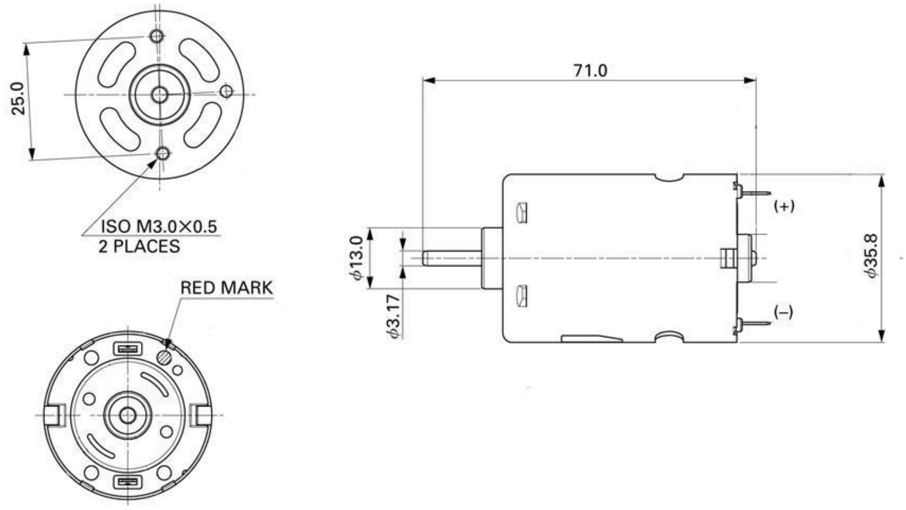
When the coil is powered, a magnetic field is generated around the armature. The left side of the armature is pushed away from the left magnet and drawn towards the right, causing rotation.



When the coil turns through 90^0, the brushes lose contact with the commutator and the current stops flowing through the coil.

Now when the coil turns through 180^0, the sides get interchanged. As a result the commutator ring C_1 is now in contact with brush B_2 and commutator ring C_2 is in contact with brush B_1. Therefore, the current continues to flow in the same direction.



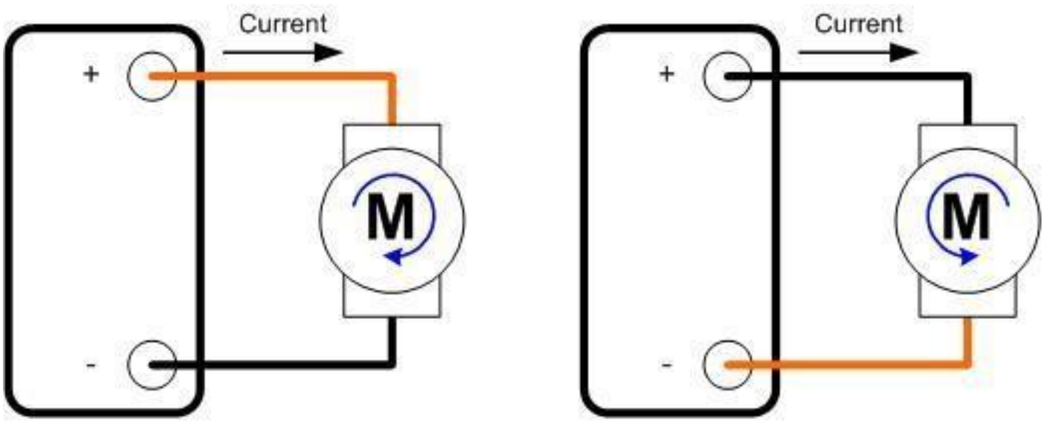


PARAMETRS OF THE DC MOTRS

1. Direction of rotation
2. Motor Speed
3. Motor Torque
4. Motor Start and Stop

Direction of Rotation

A DC Motor has two wires. We can call them the positive terminal and the negative terminal, although these are pretty much arbitrary names (unlike a battery where these polarities are vital and not to be mixed!). On a motor, we say that when the + wire is connected to + terminal on a power source, and the - wire is connected to the - terminal source on the same power source, the motor rotates clockwise (if you are looking towards the motor shaft). If you reverse the wire polarities so that each wire is connected to the opposing power supply terminal, then the motor rotates counter clockwise. Notice this is just an arbitrary selection and that some motor manufacturers could easily choose the opposing convention. As long as you know what rotation you get with one polarity, you can always connect in such a fashion that you get the direction that you want on a per polarity basis.



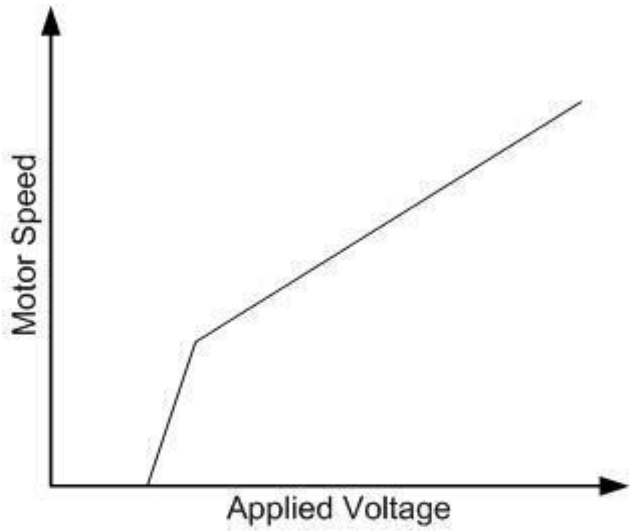
DC Motor Rotation vs Polarity

Facts:

- DC Motor rotation has nothing to do with the voltage magnitude or the current magnitude flowing through the motor.
- DC Motor rotation does have to do with the voltage polarity and the direction of the current flow.

DC Motor Speed

Whereas the voltage polarity controls DC motor rotation, voltage magnitude controls motor speed. Think of the voltage applied as a facilitator for the strengthening of the magnetic field. In other words, the higher the voltage, the quicker will the magnetic field become strong. Remember that a DC motor has an electromagnet and a series of permanent magnets. The applied voltage generates a magnetic field on the electromagnet portion. This electromagnet field is made to oppose the permanent magnet field. If the electromagnet field is very strong, then both magnetic entities will try to repel each other from one side, as well as atract each other from the other side. The stronger the induced magnetic field, the quicker will this separation/attaction will try to take place. As a result, motor speed is directly proportional to applied voltage.



Motor Speed Curve

One aspect to have in mind is that the motor speed is not entirely lineal. Each motor will have their own voltage/speed curve. One thing I can guarantee from each motor is that at very low voltages, the motor will simply not move. This is because the magnetic field strength is not enough to overcome friction. Once friction is overcome, motor speed will start to increase as voltage increase.

The following video shows the concept of speed control and offers some ideas on how this can be achieved.

Motor Torque

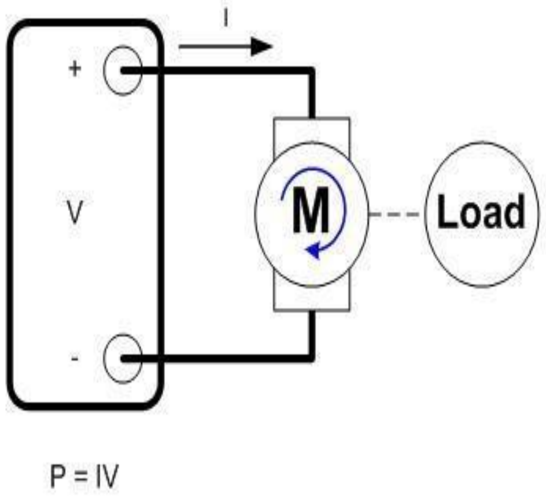
In the previous segment I kind of described speed as having to do with the strength of the magnetic field, but this is in reality misleading. Speed has to do with how fast the magnetic field is built and the attraction/repel forces are installed into the two magnetic structures. Motor strength, on the other hand, has to do with magnetic field strength. The stronger the electromagnet attracts the permanent magnet, the more force is exerted on the motor load.

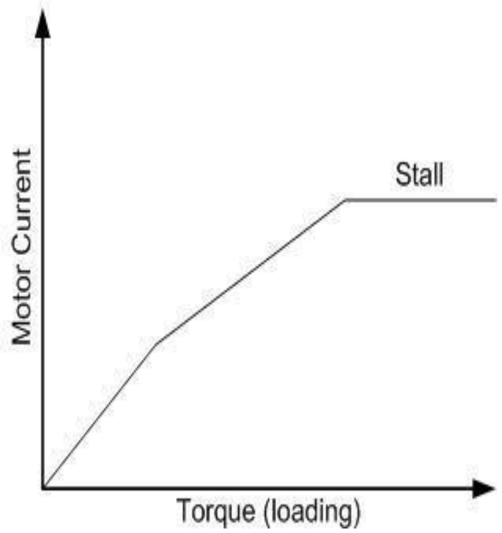
Per example, imagine a motor trying to lift 10 pounds of weight. This is a force that when multiplied by a distance (how much from the ground we are lifting the load) results in WORK. This WORK when exerted through a predetermined amount of time (for how long we are lifting the weight) gives us power. But whatever power came in, must come out as energy can not be created or destroyed. So that you know, the power that we are supplying to the motor is computed by

$$P = IV$$

Where P is power, I is motor current and V is motor voltage

Hence, if the voltage (motor speed) is maintained constant, how much load we are moving must come from the current. As you increase load (or torque requirements) current must also increase.





Motor Loading

One aspect about DC motors which we must not forget is that loading or increase of torque can not be infinite as there is a point in which the motor simply can not move. When this happens, we call this loading "Stalling Torque". At the same time this is the maximum amount of current the motor will see, and it is refer to Stalling Current. Stalling deserves a full chapter as this is a very important scenario that will define a great deal of the controller to be used. I promise I will later write a post on stalling and its intricacies.

Motor Start and Stop

You are already well versed on how to control the motor speed, the motor torque and the motor direction of rotation. But this is all fine and dandy as long as the motor is actually moving. How about starting it and stopping it? Are these trivial matters? Can we just ignore them or should we be careful about these aspects as well? You bet we should!

Starting a motor is a very hazardous moment for the system. Since you have an inductance whose energy storage capacity is basically empty, the motor will first act as an inductor. In a sense, it should not worry us too much because current can not change abruptly in an inductor, but the truth of the matter is that this is one of the instances in which you will see the highest currents flowing into the motor. The start is not necessarily bad for the motor itself as in fact the motor can easily take this Inrush Current. The power stage, on the other hand and if not properly designed for, may take a beating.

Once the motor has started, the motor current will go down from inrush levels to whatever load the motor is at. Per example, if the motor is moving a few gears, current will be proportional to that load and according to torque/current curves.

Stopping the motor is not as harsh as starting. In fact, stopping is pretty much a breeze. What we do need to concern ourselves is with how we want the motor to stop. Do we want it to coast down as energy is spent in the loop, or do we want the rotor to stop as fast as possible? If the later is the option, then we need braking. Braking is easily accomplished by shorting the motor outputs. The reason why the motor stops so fast is because as a short is applied to the motor terminals, the Back EMF is shorted. Because Back EMF is directly proportional to speed, making Back EMF = 0, also means making speed = 0.

Truth Table

High Left	High Right	Low Left	Low Right	Description
On	Off	Off	On	Motor runs clockwise
Off	On	On	Off	Motor runs anti-clockwise
On	On	Off	Off	Motor stops or decelerates
Off	Off	On	On	Motor stops or decelerates

As already said, H-bridge can be made with the help of transistors as well as MOSFETs, the only thing is the power handling capacity of the circuit. If motors are needed to run with high current then lot of dissipation is there. So, head sinks are needed to cool the circuit. Now you might be thinking in why i did not discuss the cases like High side left on and Low side left on or high side right on and low side right on. Clearly seen in the diagram, you don't want to burn your power supply by shorting them. So that is why those combinations are not discussed in the truth table.

CHAPTER 6
RESULTS

6.1 Circuit Diagram:

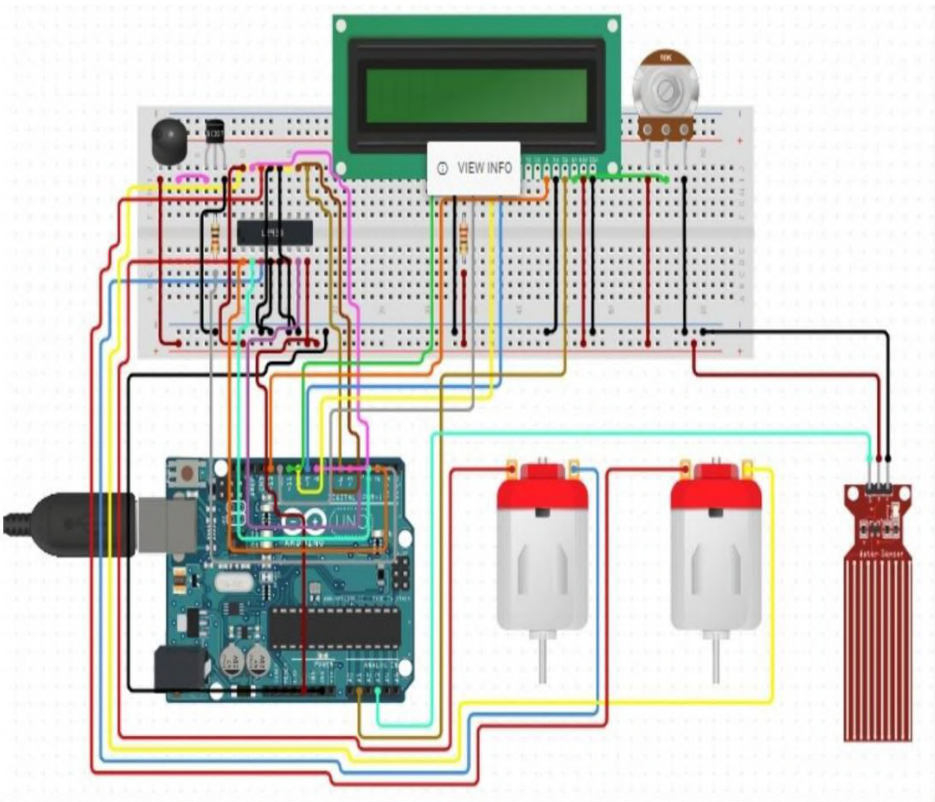


Fig no 34

Fig no 35

Fig no 36

Fig no 37

Fig no 38

6.2 RESULTS AND OUTPUT

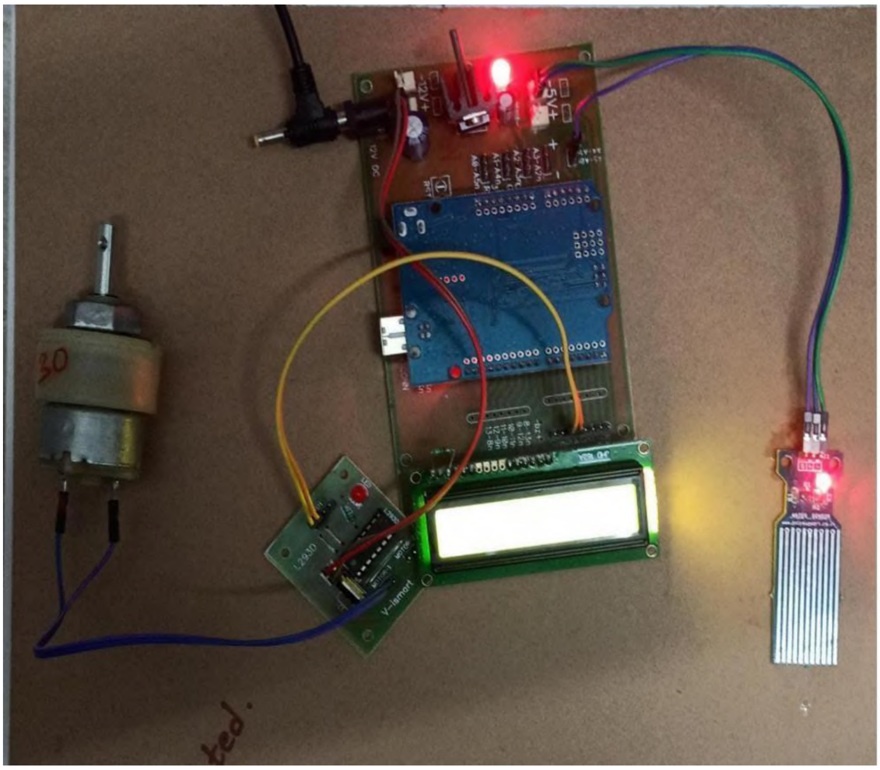


Fig no 39

CONCLUSION

In order to complete the project, the hardware is initially tested on bread board. Subsequently, the hardware is wired on general purpose PCB. This software is written in C language and simulation is tested on pc. Afterwards the code is dumped into a micro controller. The integrated hardware and software are tested successfully. The principle proved thorough this project can be utilized in many real time applications.

REFERENCE

[1] Rafiquzzaman, M (2011) 'Microcontroller Theory and Applications with the PIC 18F"

[2] Di Jasio, L (2008) "Programming 32-Bit Microcontrollers in C: Exploring the PIC 32 (Embedded Technology)"

[3] Wilmshurst,T (2009) "DESIGNING Embedded Systems with PIC Microcontrollers: Principles and Applications"

[4] Reese, R.B., Bruce, Bruce, J. W. And Jone, B. A. (2008) "Microcontrollers: From Assembly Language to PC using the PIC Family"

[5] Morton, J (2005) "The PIC Microcontroller: Your Personal Introductory Course"

[6] Huang, H and Chartrand, L (2004) "PIC Microcontroller" An Introductory to Software and Hardware Interfacing".

A

PROJECT REPORT

On

GREENHOUSE MONITORING AND CONTROL SYSTEM USING IOT

Submitted by

1)Ms. Dugyala Krupalini (17K81A0413)

2)Ms. Hima Umesh (17K81A0420)

3)Ms. Nukala Varshita (17K81A0435)

in partial fulfillment for the award of the degreeof

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Mr.K.Nishakar

Assosciate Professor

M.Tech,Ph.D

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST.MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

NBA &NAAC A+ ACCREDITED

Dhulapally, Secunderabad – 500 100

JUNE 2021

St.MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

NBA &NAAC A+ Accredited
Dhulapally, Secunderabad-500 100

Dept of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the project entitled **GREENHOUSE MONITORING AND CONTROL SYSTEM USING IOT** is being submitted by **Ms. Dugyala Krupalini (17K81A0413), Ms.Hima Umesh (17K81A0420) , Ms. Nukala Varshita (17K81A0435)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

INTERNAL GUIDE
Mr.K.Nishakar
Assosciate Professor

HEAD OF THE DEPARTMENT
Dr.B.HARI KRISHNA
Professor

EXTERNAL EXAMINER

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **DUGYALA KRUPALINI** WITH ROLL NO.**17K81A0413, HIMA UMESH** WITH ROLL NO.**17K81A0420**, **N. VARSHITA** WITH ROLL NO.**17K81A0435**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**GREENHOUSE MONITORING AND CONTROL SYSTEM USING IOT EMBEDDED**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. ★ Kompally-500014 ★

ORUGANTI VENKAT

DIRECTOR
TRAININGS & PLACEMENTS LASYA IT SOLUTIONS
PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of **Bachelor of Technology** in Department of Electronics and Communication Engineering', session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled **'Greenhouse Monitoring and Control System using IOT'** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

Dugyala Krupalini 17K81A0413
Hima Umesh 17K81A0420
Nukala Varshita 17K81A0435

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would b incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna**, **Head &Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our Dept . project coordinator Dr.A.Anand,Professor** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide, **Mr.K.Nishakar** , **Assosciate Professor**, Department of Electronics and Communication Engineering and our class project coordinator **Ms.P.Pushpa, Assistant Professor** ,Department of ECE for guidance and encouragement in carrying out the project work.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1.Dugyala Krupalini
2. Hima Umesh
3.Nukala Varshita

ABSTRACT

A green house is where plants such as flowers and vegetables are grown. During the day the greenhouses warm up when the sun-rays penetrates through it , which then heats the plant, soil and structure. Green houses help to protect the crops from many diseases, particularly those that are soil borne and splash onto plants in the rain. Greenhouse effect is a natural phenomenon and beneficial to human being. Numerous farmers fail to get good profits from the greenhouse crops for the reason that they can't manage two essential factors, which determines plant growth as well as productivity. Green house temperature should not go below a certain degree, High humidity can result to crop transpiration, condensation of water vapor on various greenhouse surfaces, and water evaporation from the humid soil. To overcome such challenges, this greenhouse monitoring and control system comes to rescue. This project demonstrates the design and implementation of a various sensors for greenhouse environment monitoring and controlling. This greenhouse control system is powered by Atmega328 microcontroller it consists of temperature sensor, light sensor, soil moisture sensor, LDR sensor, LCD display module, 12v DC fan, Bulb and pump. Temperature sensor, senses the level of temperature. if it goes high DC fans gets on and when the temperature goes low the fan gets off. Soil moisture sensor, senses the water level as the level decreases the pumps gets on. In the absence of light, the LDR sensor senses and the bulb starts glowing. By this way it will become easy to monitor and control the system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY OF COMMONLY USED TERMS

AODTPC: Adaptive on demand transmission control protocol is a key technique to save the energy of a sensor node in wireless sensor network.Power level regulation of a sensor node in time varying propagation environment still needs deep investigation due to uncertain behaviour of wireless fading channel**.**

GREENHOUSE: A greenhouse is a structure with walls and roof made chiefly of transparent material, such as glass, in which plants requiring regulated climatic conditions are grown. These structures range in size from small sheds to industrial-sized buildings.

NEXT GENERATION NETWORK: It is a packet based network able to provide services including telecommunication services and able to make use of multiple broadband.

TCP/IP PROTOCOL: The **Internet protocol suite** is the conceptual model and set of communications protocols used in the Internet and similar computer networks. It is commonly known as **TCP/IP** because the current foundational protocols in the suite are the Transmission Control Protocol (TCP) and the Internet Protocol (IP). During its development, versions of it were known as the **Department of Defense** (**DoD**) **model** because the development of the networking method was funded by the United States Department of Defense. Its implementation is a protocol stack.

CHAPTER-1
INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

We live in the world where everything can be controlled and operated automatically, but there are still few sectors in our country where automation hasnot been adopted or not being put to a full-fledged use, perhaps because of several reason one such reason is cost and one such field is ‘agriculture’. Agriculture has been one of the primary occupations of man since early civilization and even today manual interventions in farming are inevitable. Greenhouse form an important part of the agriculture and horticulture sectors in the country as they can be used to grow plants under controlled climatic conditions for optimal growth. Greenhouse technology is the technique of providing favorable environmental conditions for plants . It replaces the direct supervision. Now a day, due to urbanization and lack of land availability there is a great need to construct the greenhouse, which will be revered mainly for growing crops. Greenhouse monitoring and controlling projects is used to measure the various parameters like temperature , light, moisture content and to display them on LED. Continuous monitoring of these environmental factors gives relevant information pertaining to the individual effects of the various factors towards obtaining effects of the various factors towards obtaining maximum crop production. Unlike open farming where natures control takes the upper hand, green house prevents a closed environment that can be strictly controlled by humans in order to provide optimal conditions for the growth of plants . The effectiveness in greenhouse crop production depends significantly on the modification of optimal growth conditions to achieve high yield at lower costs, good quality and low environmental load.

1.2 OBJECTIVE

The main objective of the this project is to design and implement an efficient monitoring system through which the required parameters are monitored remotely using internet and the data gathered from the sensors are stored in the cloud and to project the estimated trend on the web browser. A solution for monitoring the temperature, moisture content in soil and light intensity value crossing its threshold value ranges, for example temperature levels in the air, moisture content in soil and light intensity in the greenhouse exceeding the normal levels etc., in the environment using wireless embedded computing system is proposed in this paper. The solution also provides an intelligent remote monitoring for a particular area of interest. The embedded system is an integration of sensor devices, wireless communication which enables the user to remotely access the various parameters and store the data in cloud.

1.3 SCOPE

The automated greenhouse is applicable for domestic usage. If the scale of usage is to be increased ,there would be a need of more requirement of components to measure the desired parameters. In this project the temperature, light intensity and soil moisture content are measured. It does not cover all the factors that can be taken into consideration to create a favorable environment .Other factor includes ph level of the soil or the humidity content in the greenhouse. The values for when the soil is considered to be too wet or too dry is dependent on the type of soil used. In those cases the values have to be set individually with the help of tests.

1.4 MATERIAL REQUIREMENT

1.4.1 HARDWARE REQUIREMENT:

1. **ARDUINO UNO**: It is a microprocessor which is used to transfer data from Temperature sensor to LCD and Wifi module. Ardunio takes the input from Temperature Sensor in Analog form and converts it into digital form. This converted data is sent to the Wifi module. The Ardunio sents the power required by the Temperature Sensors. So Arduino gets power from Wifi module to work and arduino works according to the code dump.

2. **TEMPERATURE SENSOR**: The temperature sensor senses the temperature around it. The temperature sensor senses the temperature in mechanical form. The temperature sensor changes this mechanical form into Analog form as the Arduino does not takes the input in mechanical form.

3. **LIGHT SENSOR:** Photo resistors, also known as light dependent resistors (LDR), are light sensitive devices most often used to indicate the presence or absence of light, or to measure the light intensity. In the dark, their resistance is very high, sometimes up to 1MΩ, but when the LDR sensor is exposed to light, the resistance drops dramatically, even down to a few ohms, depending on the light intensity. LDRs have a sensitivity that varies with the wavelength of the light applied and are nonlinear devices. They are used in many applications but are sometimes made obsolete by other devices such as photodiodes.

4.**SOIL MOISTURE SENSOR:** The soil moisture sensor is used to measure the values of the volumetric water content in soil. Since the direct gravimetric measurement of free soil and weighing of a sample, soil moisture sensors measure the volumetric water content indirectly by using some other property of the soil, such as electrical resistance, dielectric constant, or interaction with neutrons, as a proxy for the moisture content.

5.**ESP8266 WI-FI MODULE:** The wifi module in the circuit helps in connecting the microcontroller with the wifi network. The data received by the Wifi module is sent to the TELNET app where we can see the variations in the graph due to changes in the temperature.The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network.The ESP8266 is capable highly of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

6. **LCD** : The LCD we used here is 16x2 display LCD. The LCD helps us to know the data is getting transferred or uploaded to TELNET app or not.

7. **DC FAN** : The direct current fans, or DC fans, are powered with a potential of fixed value such as the voltage of a battery. Typical voltage values for DC fans are 5V, 12V, 24V and 48V. In contrast, the alternating current fans, or AC fans, are most powered with a changing voltage of positive

8.**BULB**:A light-bulb produces light from electricity.In addition to lighting a dark space, they can be used to show an electronic device is on, to direct traffic, for heat, and for many other purposes

9.**WATER PUMP:** DC 5V operated submersible water pump is ideal for making automatic plant watering system using Arduino and esp8266 with the wifi module.

10.**POWER SUPPLY UNIT:** The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as “Regulated D.C Power Supply”.

1.4.2 SOFTWARE REQUIREMENT :

1.**ARDUINO IDE**: Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs.

2.**PROTEUS SOFTWARE:** Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline "From concept to completion".It is a proprietry software tool suite used primarily for electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

1.5 PROCUMENT OF EQUIPMENT

All the required hardware components were procured from an electronic manufacturing shop and before assembling and interfacing hardware components to Arduino UNO micro-controller, individual sensors were connected to Arduino in Proteus Software and understood the working of every individual sensor in Proteus Software. Then assembled all the sensors and interfaced to Arduino UNO micro-controller using connecting wires through bread board and wrote the program to make work all the hardware sensors in Arduino IDE and understood the working of all sensors practically. Later fixed the threshold limit for the temperature sensor and all the information was sent to the IOT Platform.

1.5.0 INTRODUCTION TO EMBEDDED SYSTEMS

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.

The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes.

The applications software on such processors is sometimes referred to as firmware.

The simplest devices consist of a single microprocessor (often called a "chip"), which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine.

As the embedded system is the combination of both software and hardware. Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

- ### MICRO PROCESSOR (µp)

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel- injection systems for automobiles

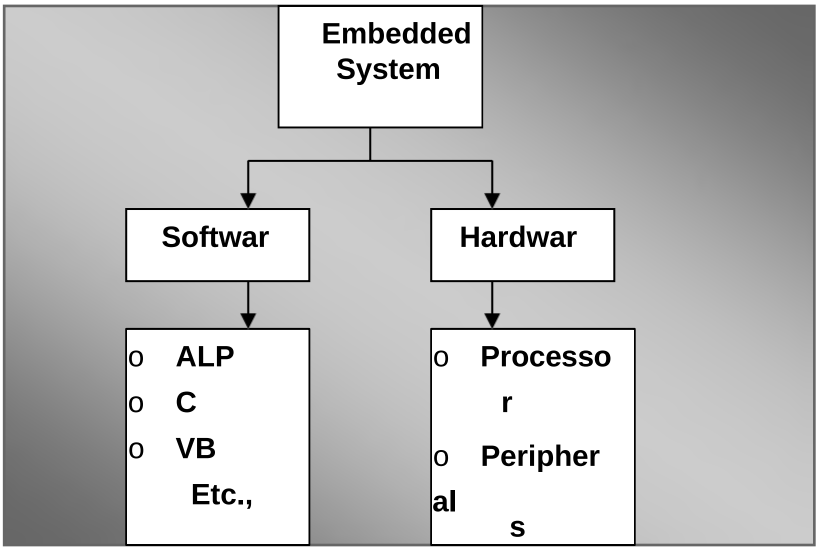


Fig.1.1 Block Diagram of Embedded System

➢ CLASSIFICATION OF PROCESSORS

- Micro Processor (µp)
- Micro controller (µc)
- Digital Signal Processo1r (DSP)
- Application Specific Integrated Circuits (ASIC)

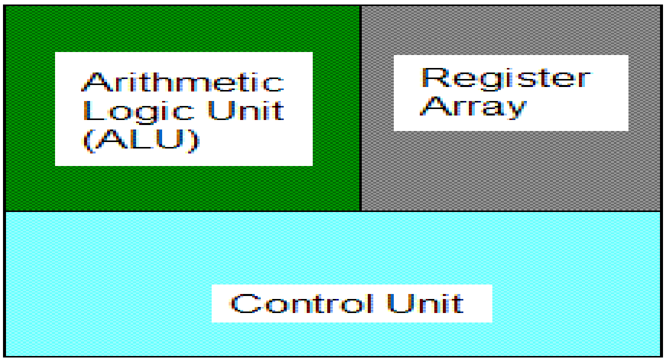


Fig.1.2 Three Basic Elements of a Microprocessor

1.5.1 HARVARD ARCHITECTURE

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

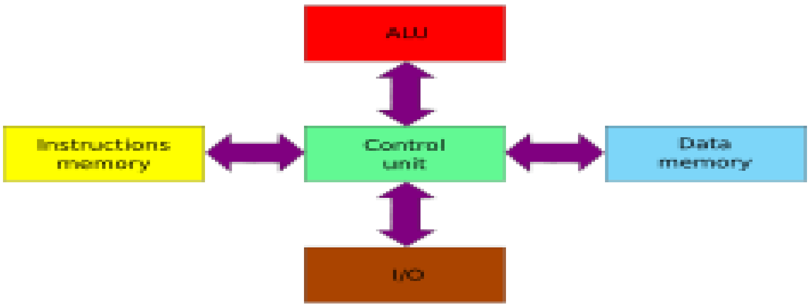


Fig.1.3 Harvard Architecture

1.5.2 VON-NEUMANN ARCHITECTURE

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. They cannot be performed at the same time. The von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A stored-program digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

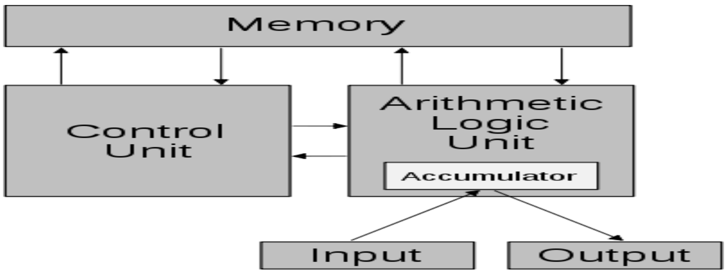


Fig.1.4 Schematic of the Von-Neumann Architecture

• USES OF THE HARVARD ARCHITECTURE

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behaviour must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.

Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc.

1.5.3 BASIC DIFFERENCES BETWEEN HARVARD AND VON-NEUMANN ARCHITECTURE

- The primary difference between Harvard architecture and the Von Neumann architecture is Von Neumann architecture data and programs are stored in the same memory and managed by the same memory and managed by the same information and handling system. Whereas the Harvard architecture stores data and programs in separate memory devices and they are handled by different subsystems.
- In a computer using the Von-Neumann architecture without cache; the central processing unit (CPU) can either be reading and instruction or writing/reading data to/from the memory. Both of these operations cannot occur simultaneously as the data and instructions use the same system bus.
- In a computer using the Harvard architecture the CPU can both read an instruction and access data memory at the same time without cache. This means that a computer with Harvard architecture can potentially be faster for a given circuit complexity because data access and instruction fetches do not contend for use of a single memory pathway.
- Today, the vast majority of computers are designed and built using the Von Neumann architecture template primarily because of the dynamic capabilities and efficiencies gained in designing, implementing, operating one memory system as opposed to two. Von Neumann architecture may be somewhat slower than the contrasting Harvard Architecture for certain specific tasks, but it is much more flexible and allows for many concepts unavailable to Harvard architecture such as self-programming, word processing and so on.

1.6 INTRODUCTION TO IOT

The Internet of things (IoT) describes the network of physical objects—a.k.a. "things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet.Things have evolved due to the convergence of multiple technologies, real-time analytics, machine learning, ubiquitous computing, commodity sensors, and embedded systems.[1] Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", including devices and appliances (such as lighting fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers. The IoT can also be used in healthcare systems. Internet of Things (IOT) is a network of devices that connect directly with each other to capture and share data through a secure service layer that connects to a central command and control server in the cloud. The closure\look suggest that the way people collect, record and analyze data—not just in health care but in every industry today. The idea of devices connecting directly with each other is basically called Internet of Things. The Internet of Things also a called the Internet of Objects, refers to a wireless network between objects, usually the network will be wireless and self-configuring. Internet of Things (IOT) is one of the major component advances in present time that links the internet with everyday sensor and working devices. Smart objects play an important role in the Internet of Things vision, since embedded communication and information technology would have the potential to change the utility of these objects. Using sensors they are able to recognize their condition, and via built-in connecting power they would be able to interact with each other.

➢ BENEFITS OF IOT

UBIQUITOUS NETWORKS -: Personal Wi-Fi on your smart phones and on many of the other devices. Everyone (and everything) wants as well as needs to be connected.

Connected computing -:We want all of the devices, smart phones, televisions(colored or black and white), dvd players, vehicles etc. to keep record of what we are doing, seeing, reading, and/or listening to as we sway through the day, from one place to another – the handoffs from device to device is happening already.

ANALYTICS-AS-A-SERVICE-: The API and App economies are already wide and growing which enables to “do something interesting” as long as it can be connected to an API or can invoke an App that carries out a network-based service. The thing is a data generator as well as collector that learns from, makes forecast, and maybe even takes data-driven actions in response to the data that are collected too.

MARKETING AUTOMATION-: Smart phone customer engagement, geological-location, Apple’s iBeacon etc. are all developing a network of knowledge and information regarding customers’ locations, intentions, preferences, as well as buying patterns. Obviously, the degree of geological location-based knowledge needs to maintain the right balance between user privacy as well as the timely delivery of important and significant products and services to the particular user.

SUPPLY CHAIN ANALYTICS-: Delivering the just-in-time products at the time of need (inclusive of the use of RFID-based tracking). Significantly, everything is a customer (inclusive of machines, automobiles, manufacturing plants, ATM machines, etc.), as well as the IOT is monitoring, watching, as well as waiting for the product needs to arise.

1.7 LITERATURE SURVEY

Since 1990‟s, for greenhouse and environment monitoring various kinds of systems have been developed. But due to lack of awareness, cost and implementation factors, these systems were left behind. Later a DSP based prototype Greenhouse Environment Monitoring system developed in 2010. The newer scenario of decreasing water tables, drying up of rivers and tanks, unpredictable environment presents urgent need for proper utilization of water. To cope up with this use of temperature and moisture sensor at suitable locations for monitoring of crops is implemented in. . We also visited few greenhouses and observed and recorded the working methods of the framers, which provided me a very clear idea how the maintenance and monitoring activities. Proposed the techniques for selection hardware ,provided basics and reference models on which an IoT system can be based and developed. Comparative study of some existing systems provided insights provides the first node to start, already available systems in IoT-powered gardening and agriculture, like Bitponics, and Harvest Geek are either not available in India or are very costly which add up a considerable production cost overhead on the crops or Agro-based products.

CHAPTER-2

LITERATRE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH AREA:

1) "Greenhouse Monitoring System Using IoT," proposed by M. Krishna Mohan, JakkulaLikhitha, TejaswyYamarthi, KagithaSravani on 02, March 2017 : This system is mainly applicable for greenhouses .Various sensors are deployed for sensing illumination,temparature and moisture content .The system uses IOT and is hosted on server .The technology used is ARM7 and WiFi.This can work as both web based and also based on mobile application.Future research proposed to this is to find which part of plant is having dead cells and which part is healthy.Also to calculate the amount of chemicals required for plant growth.

2) "IoT Based Smart Greenhouse Automation Using Arduino, proposed by D.O.Shirsath, Punam Kamble, Rohini Mane, Ashwini Kolap, R. S. More " IJIRCST, Volume-5, Issue-2, March 2017 : This system is mainly applicable for greenhouses .Various sensors are deployed for sensing humidity, temparature and moisture content .The system uses IOT and is hosted on server .The technology used is arduino controlling and remote monitoring using wireless communication. This can work as both web based and also based on mobile application. Future research proposed to this is to provide more security to the data stored in the IOT.

3)"An IoT-based wireless imaging and sensor node system for remote greenhouse pest monitor- ing," Chemical Engineering Transactions, 58, 2017.pp 601-606 was proposed by Dan JericArcegaRustia , Ta-Te Lin. This system is implemented using various sensors that are used to detect the crucial parameters of the plant such as atmospheric pressure and light intensity. The technology used is image processing and k-means algorithm. This system is purely web based. Future research proposed is to deploy more nodes in different location for pest monitoring.

4)A decision support system based on multi sensor data fusion for sustainable greenhouse management ," Jour- nal of Cleaner Production, Feb 2017, pp. 1-9. proposed by Giuseppe Aiello , Irene Giovino , MariangelaVallone , Pietro Cata- nia ,AntonellaArgento,". The system is mainly applicable for agricultural sector. This system helps in detecting humidity and temparature of the greenhouse. Alogorithm used here is data fusion algorithm.

2.2 REVIEW ON RELATED LITERATURE:

Idea of growing plants in environmentally controlled areas has existed since roman times. In the 13thcentury, greenhouses were built in Italy. They were originally called giardini botanici(botanical gardens).Then, in 1450, sanga yorok written description of greenhouses which was designed to regulate the temperature and humidity requirements of plants & crops. As the time passed, every country adopts the greenhouse system with some enhancement. Due to development in agriculture field, there are many ways to monitor & control the various parameters like temperature, humidity, soil moisture, light intensity. System can be wired or wireless. In the wired system, power source is wire. This system more reliable but complex to handle. When in wireless sensor network, power source of node is battery but it has also a drawback relative to battery which is of limited capacity. To overcome this drawback AODTPC protocol with kalman estimator is used. [5] IoT Technology has been used to reduce the distance between the staff in the article “Things” and its digital impersonation in data frameworks. It’s seen as the NGN of the internet [1]. The IoT is driven by an extension of the Internet through the incorporation of physical articles joined with a capacity to provide more quick-witted administrations to the earth as more information ends up noticeably accessible. Several application areas going from Green-IT and vitality effectiveness to coordination’s are now beginning to profit by Internet of Things ideas [2]. A Greenhouse provides basic methods for employment to its proprietor and must be financially pragmatic for the specific atmosphere in which it stands. Also, Greenhouse could be defined as advanced Innovation for Protected Horticulture addresses the major natural elements of light, temperature, and irrigation [3]. This paper investigates the usefulness of using the IoT based on the greenhouse to utilize low-cost tools and decrease the effort of the Pleasants. This could be implemented by including the automation in irrigation process, conserving temperature, and the degree of brightness inside the greenhouse structure .More recent attention has been focused on the provision of improvement on the Internet of things (IoT) and how to utilize it with the various applications. The IoT is interesting subject that recently whiteness very large number of papers that aimed to develop this technology. in next paragraphs, several papers about the greenhouse with IoT should be presented .In 2013, C. wenshun et al. published a paper in which they described the structure of services platform, even operating layer, collecting, and control layer, arrange transport layer and etc. then described the functions of Zhuangluo sunlight greenhouse. Finally, how to run interface and working condition of the system[4]. T. Quynh et al. (2015) suggest an article that analyzes various models of greenhouse in Vietnam and monitoring parameters required and proposes to apply single-path and multiple-path routing protocol in order to improve the IoT monitoring performance[6].

J. Zhao et al. (2010) proposes a paper that studies the integration of control networks and information networks in IoT in practical agriculture production. They also suggest a method that combines the remote monitoring system and wireless communications [7].

This paper has been written to show that IoT save peasants efforts and make the work much easier when the IoT used in Greenhouse applications. Also, this paper tries to provide a solution for some network problems such as dynamic Internet Protocol (IP) and global IP.

The proposed greenhouse has been implemented with three intelligent control systems. These system could be controlled by an intelligent Fuzzy controllers that providing the best performance. The Fuzzy controllers must be resided completely inside Microcontroller chip without require any extra processing. This development feature leads to the more sophisticated improvements of fuzzy systems in practical environments.

2.3 CONCLUSION ON REVIEWS:

Hence as the technology is advancing day by day at a faster pace and helps make the lives easier ,hence in the system internet of things is used for makings things smarter. Here in the proposed system it consists of two parts– in the first part consists of controlling.Controlling is achieved by connecting the sensors to the Arduino uno and the sensors senses the level of the parameters being measured and sends the data collected to Arduino uno. In the second part ,the sytem is monitored remotely. The recieved data is then sent to the cloud and stored and could be retrieved anytime from anywhere.

CHAPTER-3
PROJECT DESIGN

3.1 OVERVIEW OF THE DESIGN:

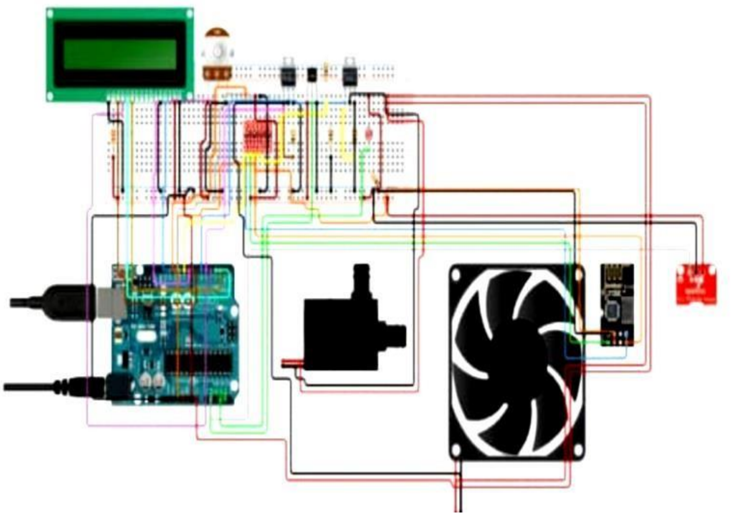


Fig 3.1 Overview of the Design

The temperature sensors output pin is connected to A0 and A1 of arduino analog pins. The Vcc and ground of Temperature Sensors is connected to 5V and ground of Arduino. The transmitter and receiver of Arduino are connected to the receiver and transmitter of Wifi module respectively. The VSS, VEE, RW, D0- D3 pins of LCD are grounded. The RS, E, D4-D7 pins of LCD are connected to 13-8 digital pins of Arduino.

3.2 EQUIPMENT ANALYSIS:

3.2.1 ARDUINO:

3.2.1.1 INTRODUCTION

It is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A

Power barrel jack, an ICSP header and a reset button.

- The ATmega8 microcontroller contains 32 general purpose working registers. As shown in the below figure these registers are directly connected to ALU. Two registers can carry one single instruction consequently in one clock cycle.
- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

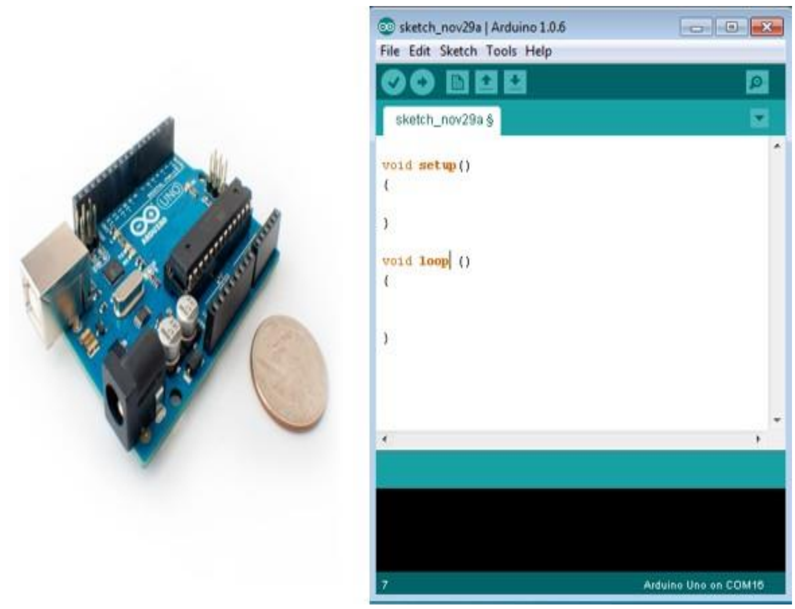


Fig.3.2 Arduino UNO

3.2.1.2 BOARD TYPES

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE. The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V. Here is a list of different Arduino boards available.

Table 3.1 Arduino Boards based on ATMEGA328 Microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header

Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.2 Arduino Boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 3.3 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header

Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 3.4 Arduino Boards based on ATMEGA2560 Microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table 3.5 Arduino Boards based on AT91SAM3X8E microcontroller

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.

5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that

	your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

3.2.1.3 Board Description

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most of them remain the same.

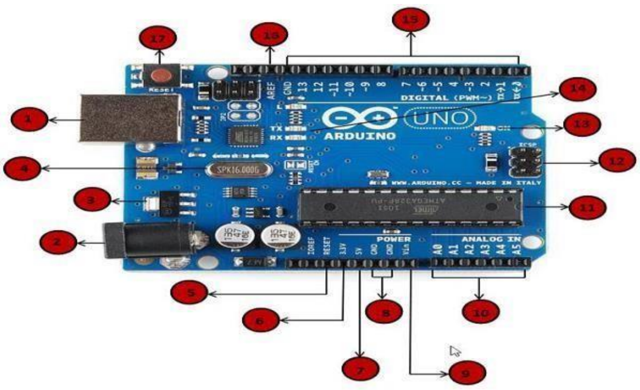


Fig.3.3 Arduino UNO Pin Description

Table 3.6 Arduino UNO Pin Description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

3.2.1.4 Arduino Family

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality.

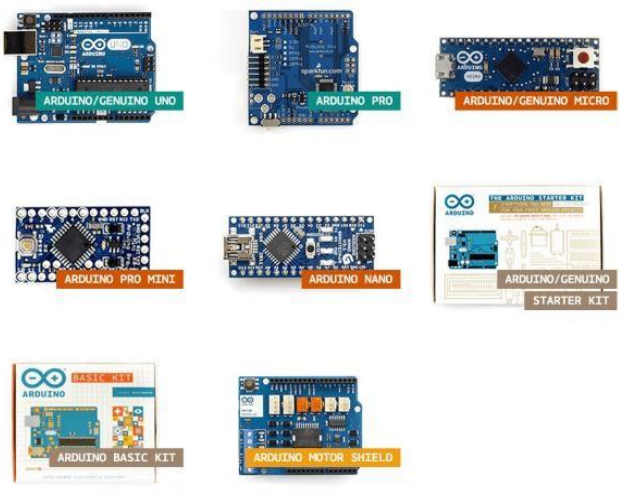


Fig.3.4 Arduino Family

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality.

3.2.1.5 Shields

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

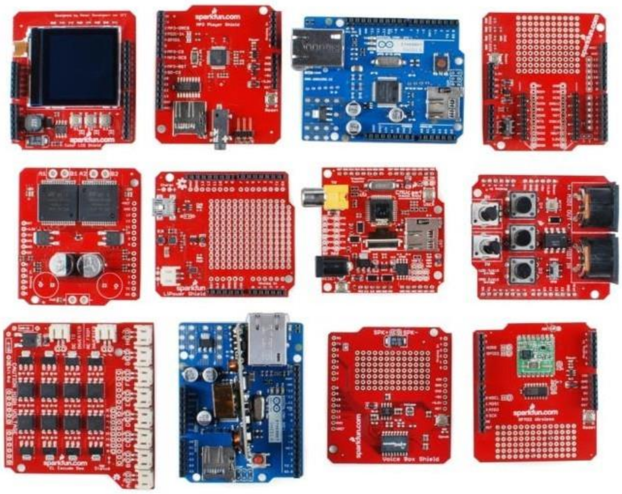


Fig.3.5 Arduino Shields

3.2.1.6Pin description of atmega328

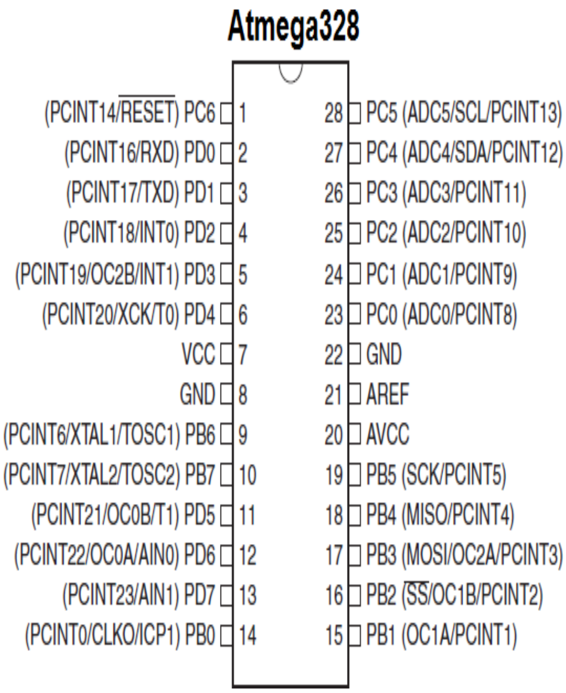


Fig.3.6 Pin Diagram of ATMEGA32

The 14 digital input/output pins can be used as input or output pins by using pinMode(), digitalRead() and digitalWrite() functions in arduino programming. Each pin operate at 5V and can provide or receive a maximum of 40mA current, and has an internal pull-up resistor of 20-50 KOhms which are disconnected by default. Out of these 14 pins, some pins have specific functions as listed below:

Serial Pins 0 (Rx) and 1 (Tx): Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.

External Interrupt Pins 2 and 3: These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

PWM Pins 3, 5, 6, 9 and 11: These pins provide an 8-bit PWM output by using analogWrite() function.

SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK): These pins are used for SPI communication.

In-built LED Pin 13: This pin is connected with an built-in LED, when pin 13 is HIGH – LED is on and when pin 13 is LOW, its off.

Along with 14 Digital pins, there are 6 analog input pins, each of which provide 10 bits of resolution, i.e. 1024 different values. They measure from 0 to 5 volts but this limit can be increased by using AREF pin with analog Reference() function.

Analog pin 4 (SDA) and pin 5 (SCA) also used for TWI communication using Wire library. Arduino Uno has a couple of other pins as explained below:

AREF: Used to provide reference voltage for analog inputs with analogReference() function.

Reset Pin: Making this pin LOW, resets the microcontroller.

➢ Advantages of arduino

- It is cheap
- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
- For beginners, it is very simple to use.

➢ Applications

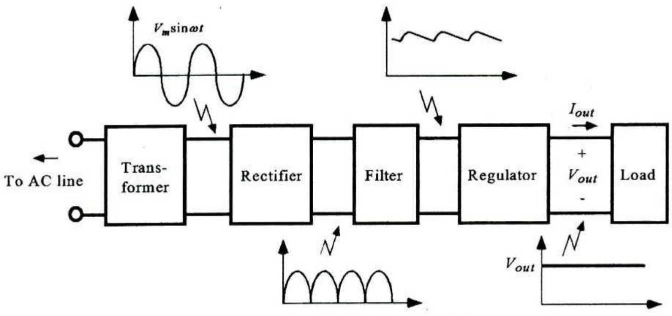
ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low-powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model.

3.2.2 REGULATED POWER SUPPLY*:*

3.2.2.1 INTRODUCTION

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as “Regulated D.C Power Supply”.

3.2.2.2 Block diagram of power supply:



Components of a typical linear power supply

Fig.3.7 Block Diagram of Power Supply

3.2.2.3 Description of power supply:

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a a different form of energy – such as solar, mechanical, or chemical - into electrical energy. A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered. For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

3.2.2.4 Transformer

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or primary winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the secondary winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

➢ Basic Principle of Transformer

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa. It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.For beginners, it is very simple to use.

➢ Working of Transformer:

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

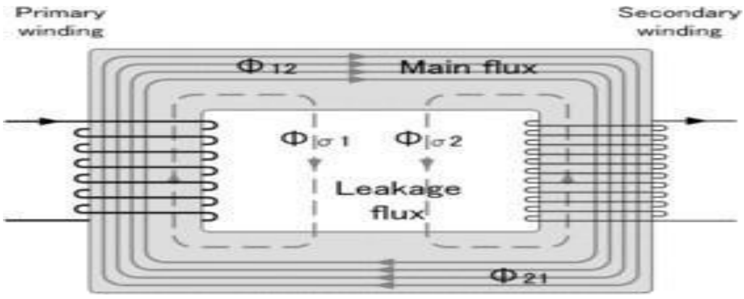


Fig.3.8 Basic Transformer

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f. The iron core is laminated to prevent 'eddy currents' flowing in the core.These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

3.2.2.5 CLASSIFICATION OF TRANSFORMER

1. Step-Up Transformer
2. Step-Down Transformer

(I) Step-Down Transformer*:*

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply. Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn's ratio of the two sets of windings determines the amount of voltage transformation.

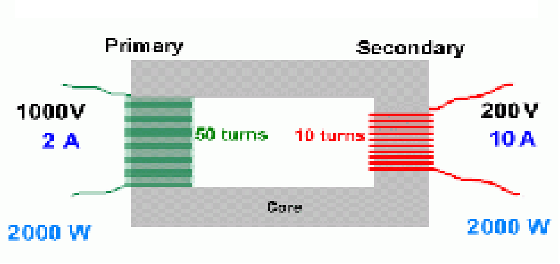


Fig.3.9 Step-Down Transformer

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of thisit is possible to do some non-standard applications using standard transformers.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single phase steps down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase steps up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

(II) Step-up transformer

A step up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input. Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply. A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage. The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply. A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

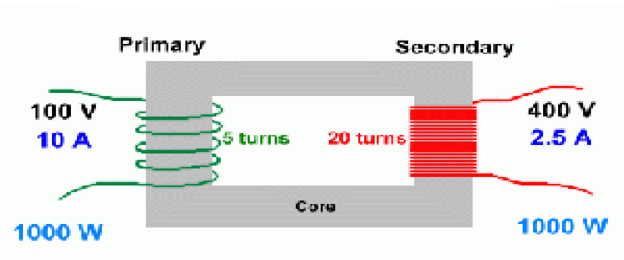


Fig.3.10 Step-Up Transformer

3.2.2.6 RECTIFIER:

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as 'half-wave' and 'full-wave' rectifiers. Both use components called diodes to convert **AC** into DC.

- **The Half-Wave Rectifier*:***

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure.

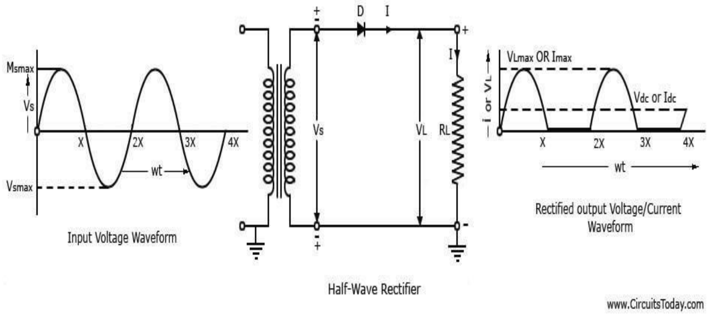


Fig 3.11 Half Wave Rectifier

Figure shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward- biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs. While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all. The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC.

- **The Full-wave Rectifier:**

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC.

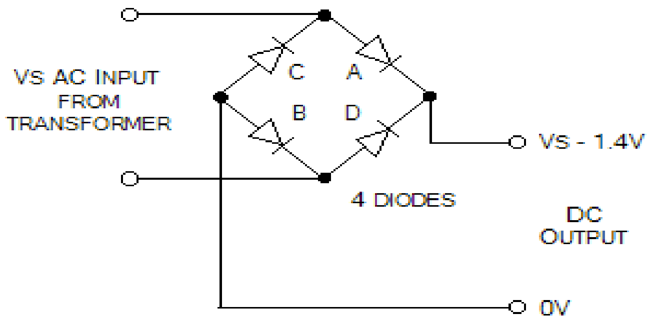


Fig.3.12 Full-Wave Rectifier

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased. While the full-wave rectifier is an improvement on the half- wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V.

3.2.3 LIQUID CRISTAL DISPLAY

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other. A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCDs connected to the contollers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively. Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines(RS RW D7 D6 D5 D4).

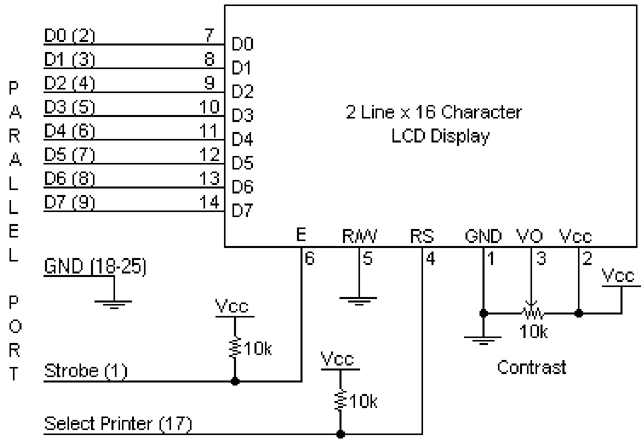


Fig.3.13 Schematic Diagram Of LCD

➢ DESCRIPTION OF 16X2

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all-Parallel Ports. It however doesn't show the use of the Status Port as an input. So what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

- Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.
- We make no effort to place the Data bus into reverse direction. Therefore we hard wire the R/W line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

3.2.3.1 PIN DESCRIPTION

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections).

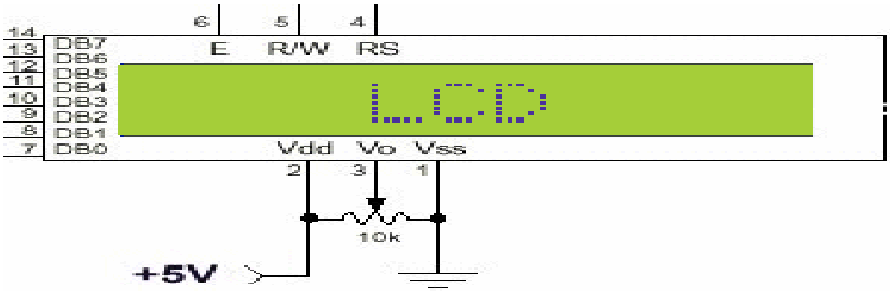


Fig4.20 Pin Diagram Of 1x16 Lines LCD

Table 3.7 16 x 2 Alphanumeric LCD Module Specifications:

Pin	Symbol	Level	Function
1	V_{SS}	-	Power, GND
2	V_{DD}	-	Power, 5V
3	Vo	-	Power, for LCD Drive
4	RS	H/L	Register Select Signal H: Data Input L: Instruction Input
5	R/W	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)
6	E	H,H->L	Enable
7-14	DB0-DB7	H/L	Data Bus; Software selectable 4- or 8-bit mode
15	NC	-	**NOT CONNECTED**
16	NC	-	**NOT CONNECTED**
5	R/W	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)
6	E	H,H->L	Enable
7-14	DB0-DB7	H/L	Data Bus; Software selectable 4- or 8-bit mode
15	NC	-	**NOT CONNECTED**
16	NC	-	**NOT CONNECTED**

Table 3.8 Pin Specifications

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

3.2.4 LIGHT DEPENDENT RESISTOR

A photoresistor or light dependent resistor or cadmium sulfide (CdS) cell is a resistor whose resistance decreases with increasing incident light intensity. It can also be referred to as a photoconductor. A photo resistor is made of a high resistance semiconductor. If light falling on the device is of high enough frequency, photons absorbed by the semiconductor give bound electrons enough energy to jump into the conduction band. The resulting free electron (and its hole partner) conduct electricity, there by lowering resistance. A photoelectric device can be either intrinsic or extrinsic. An intrinsic semiconductor has its own charge carriers and is not an efficient semiconductor, e.g., silicon. In intrinsic devices the only available electrons are in the valence band, and hence the photon must have enough energy to excite the electron across the entire bandgap. Extrinsic devices have impurities, also called dopants, added whose ground state energy is closer to the conduction band; since the electrons do not have as far to jump, lower energy photons (i.e., longer wavelengths and lower frequencies) are sufficient to trigger the device. If a sample of silicon has some of its atoms replaced by phosphorus atoms (impurities), there will be extra electrons available for conduction. This is an example of an extrinsic semiconductor.

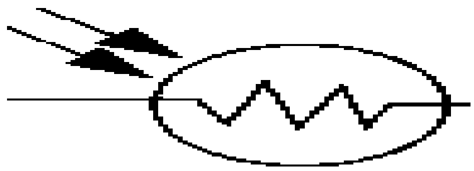


Fig.3.15 The Symbol For A Photoresistor

LDRs or Light Dependent Resistors are very useful especially in light/dark sensor circuits. Normally the resistance of an LDR is very high, sometimes as high as 1000 000 ohms, but when they are illuminated with light resistance drops dramatically. The animation opposite shows that when the torch is turned on, the resistance of the LDR falls, allowing current to pass through it.

- **Basic Structure**

Although there are many ways in which LDR's or photo resistors can be manufactured, ther are naturally a few more common methods that are seen. Essentially the LDR or photo resistor consists of a resistive material sensitive to light that is exposed to light. The photo resistive element comprises section of material with contacts at either end. Although many of the material used for light dependent resistors are semiconductors, when used as photo resistors, they are used only as a resistive element and there are no p-n junctions. Accordingly, the devices purely passive. A typical structure for a Light Dependent Resistor uses an active semiconductor layer that is deposited on an insulating substrate. The semiconductor is normally lightly doped to enable it to have the required level of conductivity. Contacts then placed either side of the exposed area. In many instances the area between the contacts is in the form of zig zag, or inter digital pattern. This maximizes the exposed area and by keeping the distance between the contacts small it enhances the gain. It also possible to use a poly crystalline semiconductor that is deposited onto a substrate such as ceramic. This makes for a very low cost light dependent resistor.

- **Operation**

Light Dependent Resistor made of a high resistance semiconductor, if light falling on the is of high enough efficiently, photon absorbed by the semiconductor give bound electrons enough energy to jump into the conduction band. The resulting free electron (and its hole partner) conduct electricity, thereby lowering resistance. In intrinsic devices, the only available electrons are in the valence band, and hence the photon must have enough energy to excite the electrons across the entire band gap. Extrinsic devices have impurities added, which have a ground state energy closer to the conduction band, since the electrons don't have so far to jump, lower energy photons (i.e., longer wavelengths and lower frequencies) will suffice to trigger the device.

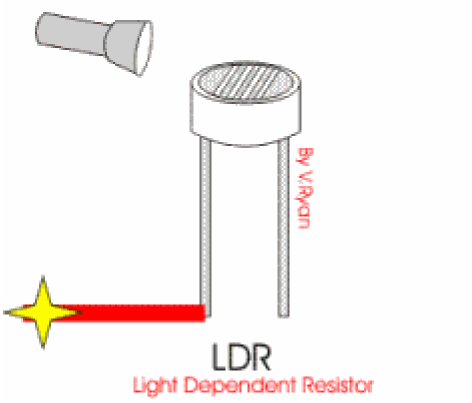


Fig.3.16 LDR Sensor

3.2.5 TEMPERATURE SENSOR (LM35)

3.2.5.1 GENERAL DESCRIPTION:

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in °Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°Cover a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package (Fig 3.2.1).

Fig.3.17 Temperature Sensor

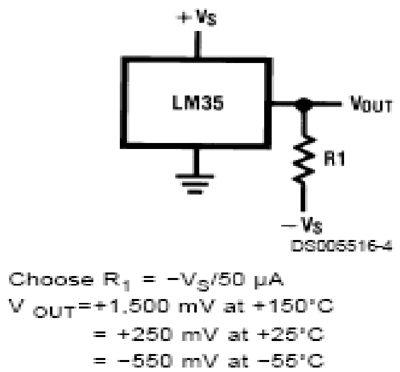


Fig.3.18 LMS

➢ Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full −55° to +150°C range

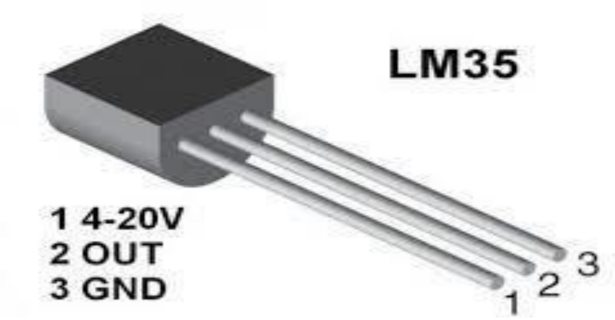


Fig.3.19 LM35

3.2.6 SOIL SENSOR

The moisture of the soil plays an essential role in the irrigation field as well as in gardens for plants. As nutrients in the soil provide the food to the plants for their growth. Supplying water to the plants is also essential to change the temperature of the plants. The temperature of the plant can be changed with water using the method like transpiration. And plant root systems are also developed better when rising within moist soil. Extreme soil moisture levels can guide to anaerobic situations that can encourage the plant's growth as well as soil pathogens. This article discusses an overview of the soil moisture sensor, working and it's applications.

3.2.6.1 What is a Soil Moisture Sensor?

The soil moisture sensor is one kind of sensor used to gauge the volumetric content of water within the soil. As the straight gravimetric dimension of soil moisture needs eliminating, drying, as well as sample weighting. These sensors measure the volumetric water content not directly with the help of some other rules of soil like dielectric constant, electrical resistance, otherwise interaction with neutrons, and replacement of the moisture content. The relation among the calculated property as well as moisture of soil should be adjusted & may change based on ecological factors like temperature, type of soil, otherwise electric conductivity. The microwave emission which is reflected can be influenced by the moisture of soil as well as mainly used in agriculture and remote sensing within hydrology.

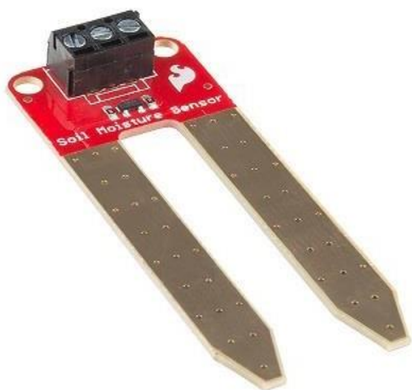


Fig.3.20 Soil Moisture Sensor Device

Fig.3.21 Soil Moisture Sensor FC-28

3.2.6.2 SOIL MOISTURE SENSOR PIN CONFIGURATION

The FC-28 soil moisture sensor includes 4-pins

VCC pin is used for power

- A0 pin is an analog output
- D0 pin is a digital output
- GND pin is a Ground

This module also includes a potentiometer that will fix the threshold value, & the value can be evaluated by the comparator-LM393. The LED will turn on/off based on the threshold value.

3.2.6.3 WORKING PRINCIPLE

This sensor mainly utilizes capacitance to gauge the water content of the soil (dielectric permittivity). The working of this sensor can be done by inserting this sensor into the earth and the status of the water content in the soil can be reported in the form of a percent. This sensor makes it perfect to execute experiments within science courses like environmental science, agricultural science, biology, soil science, botany, and horticulture.

➢ **Specifications**

The specification of this sensor includes the following.

- The required voltage for working is 5V
- The required current for working is <20mA
- Type of interface is analog
- The required working temperature of this sensor is 10°C~30°C

3.3 SOFTWARE COMPONENTS

- Proteus simulation
- Arduino software
- Programming language

3.3.1 Arduino Software

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs.

3.3.1.2 Arduino

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

1) Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50

2) Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to windows.

3) Simple, clear programming environment - The Arduino Software (IDE) is easy- to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

4) Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

5) Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers c an make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money

6) Getting Started with Arduino and Genuino products:- Install the Arduino Software (IDE) on Windows PCs This document explains how to install the Arduino Software (IDE) on Windows machines.

7) Download the Arduino Software(IDE).

➢ How to Download the Arduino Software (IDE)

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

3.3.1.3 INSTALLATION

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.3.22 cable

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig.3.23 Mini-B Cable

Step 2 − Download Arduino IDE Software

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

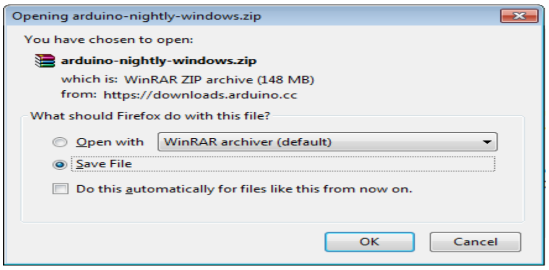


Fig.3.24 Arduino IDE Software

Step 3 − Power Up Your Board

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port. Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE

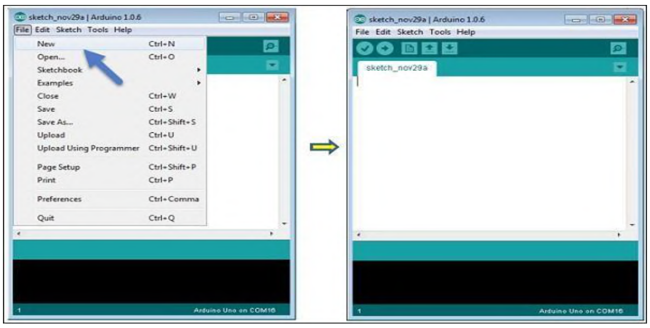
After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

Step 5 − Open Your First Project

Once the software starts, you have two options −

Create a new project.

Open an existing project example.



.

Fig.3.25 Creating New File

Step 6 – Select Your Arduino Board

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

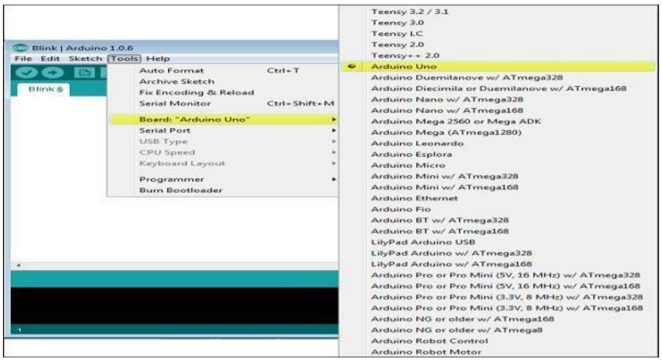


Fig.3.26 Tools Board

Step 7 – Select Your Serial Port

Select the serial device of the Arduino board. Go to Tools → Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

Step 8 – Upload The Program To Your Board

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

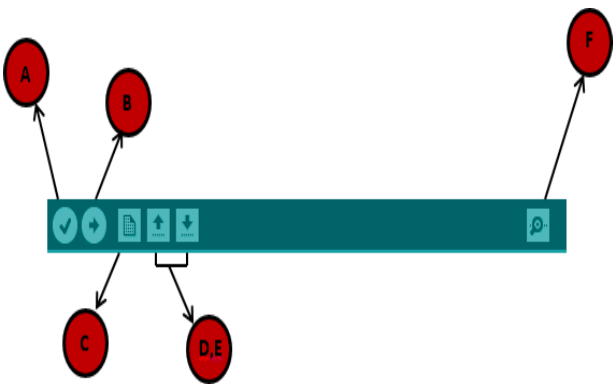


Fig.3.27 Uploading Program

A − Used to check if there is any compilation error. **B** − Used to upload a program to the Arduino board. **C** − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar. If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

- **Connecting a Battery**

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24 V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs.

- **Moving On**

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run. Start the Arduino development environment. In Arduino-speak, programs are called “sketches”, but here we will just call them programs. In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
 void setup()
{
Serial.begin(9600); Serial.println("Hello World");
            } void loop() {}
```

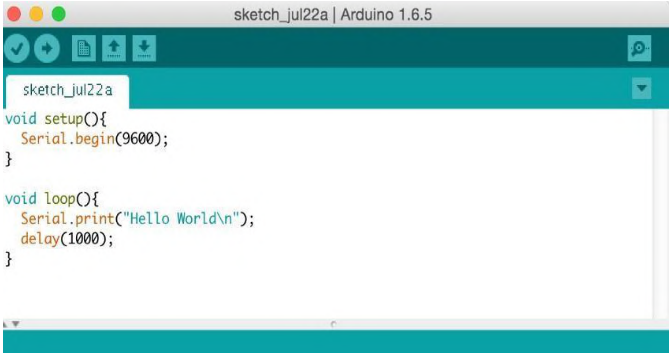


Fig.3.28 Arduino

3.3.2 PROTEUS*:*

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

3.3.2.1 About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

3.3.2.2 Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc. ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

3.3.2.3 Starting New Design

Step 1: Open ISIS software and select New design in File menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

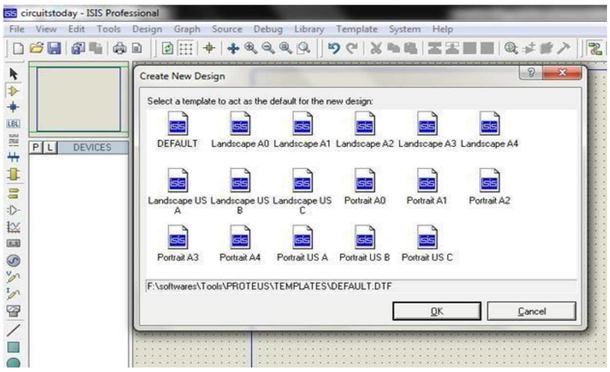


Fig.3.29 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

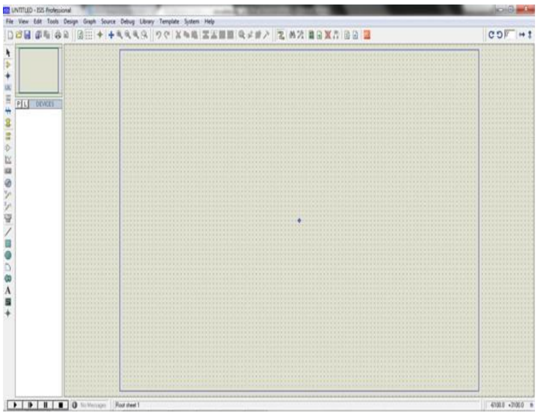


Fig.3.30 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

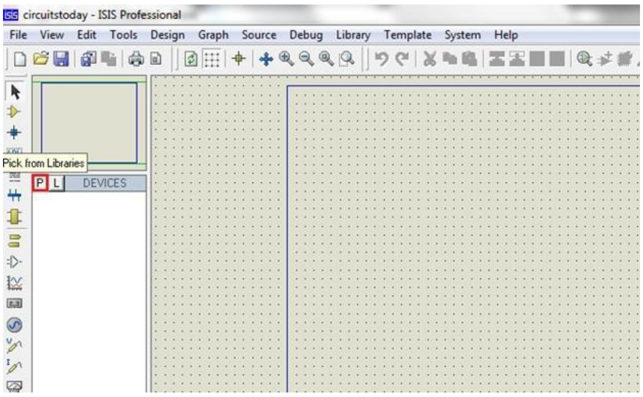


Fig.3.31 Pick From Libraries

Step 6: Select the components from categories or type the part name keyword.

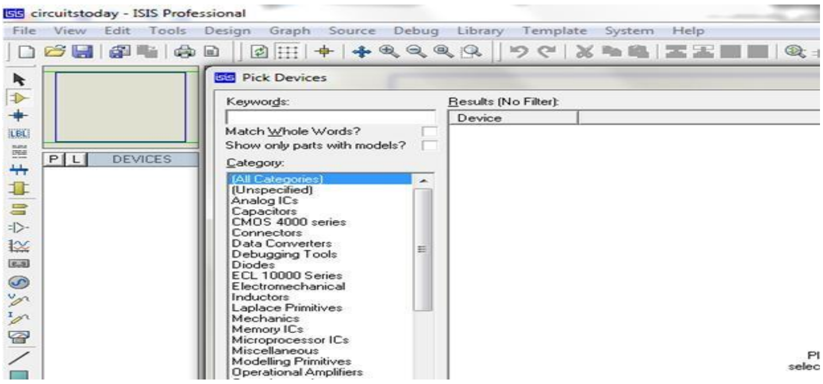


Fig.3.32 Key Word Textbox

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click. Place all the required components and route the wires i.e., make connections. Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively. Double click on the component to edit the properties of the components and click on Ok.

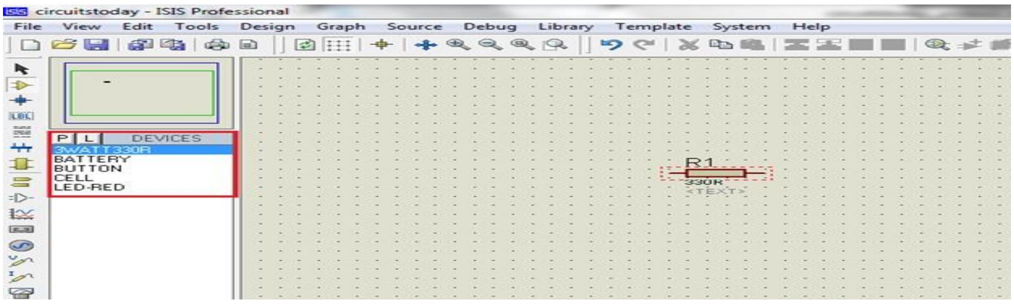


Fig.3.33 Component Selection

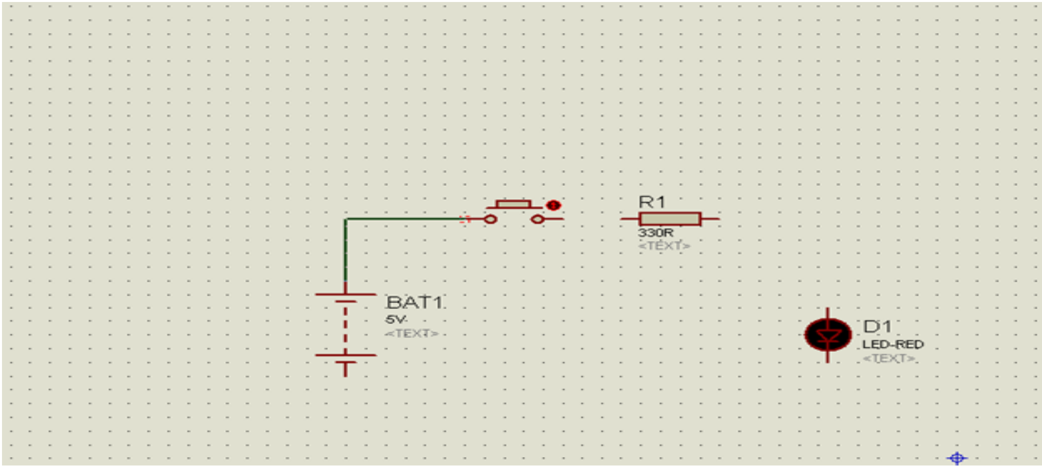


Fig.3.34 Simulation

Step 8: After connecting the circuit, click on the play button to run the simulation. Simulation can be stepped, paused or stopped at any time. In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

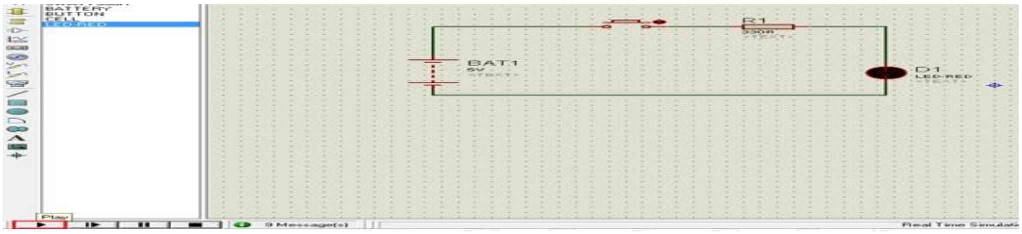


Fig.3.35 Simulation Run

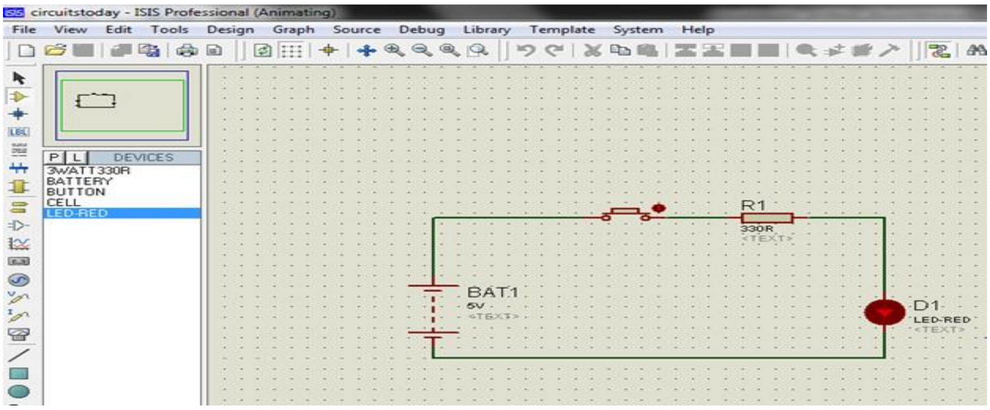


Fig .3.36 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

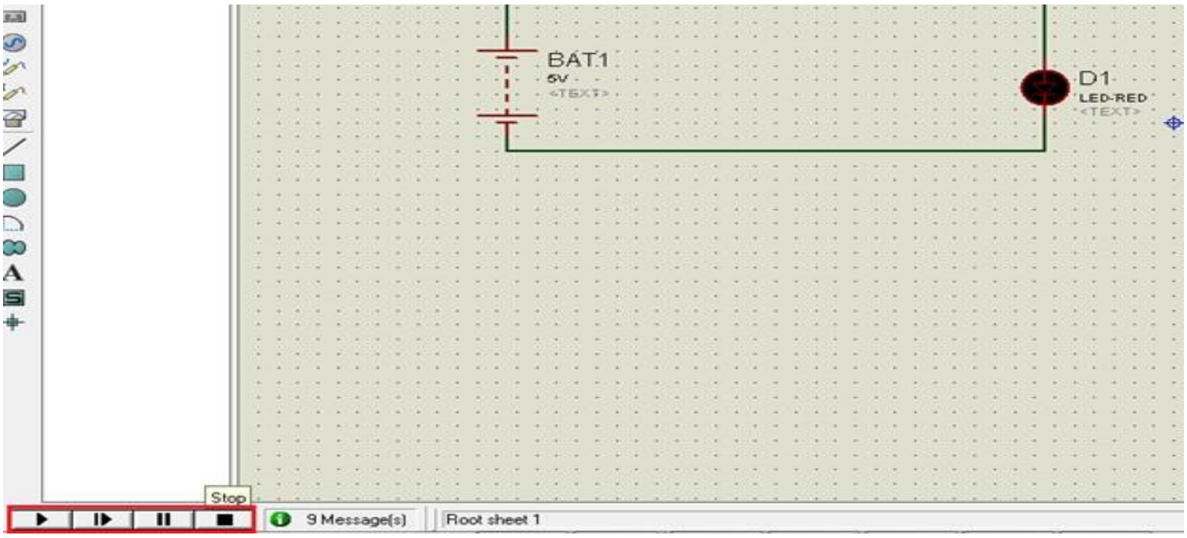


Fig. 3.37 Simulation Step-Pause-Stop Buttons

3.3.3 TELNET MOBILE APP:

Telnet is a client-server protocol, based on a reliable connection-oriented transport. Typically, this protocol is used to establish a connection to TCP port number 23, where a Telnet server application (telnetd) is listening. Telnet, however, predates TCP/IP and was originally NCP protocols.

Even though Telnet was an ad hoc protocol with no official definition until March 5, 1973,[3] the name actually referred to Teletype Over Network Protocol as the RFC 206 (NIC 7176) on Telnet makes the connection clear:[4]

The TELNET protocol is based upon the notion of a virtual teletype, employing a 7-bit ASCII character set. The primary function of a User TELNET, then, is to provide the means by which its users can 'hit' all the keys on that virtual teletype. Essentially, it used an 8-bit channel to exchange 7-bit ASCII data. Any byte with the high bit set was a special Telnet character. On March 5, 1973, a Telnet protocol standard was defined at UCLA with the publication of two NIC documents: Telnet Protocol Specification, NIC 15372, and Telnet Option Specifications, NIC 15373. Many extensions were made for Telnet because of its negotiable options protocol architecture. Some of these extensions have been adopted as Internet standards, IETF documents STD 27 through STD 32. Some extensions have been widely implemented and others are proposed standards on the IETF standards track. Telnet is best understood in the context of a user with a simple terminal using the local Telnet program (known as the client program) to run a logon session on a remote computer where the user's communications needs are handled by a Telnet server program. When Telnet was initially developed in 1969, most users of networked computers were in the computer departments of academic institutions, or at large private and government research facilities. In this environment, security was not nearly as much a concern as it became after the bandwidth explosion of the 1990s. The rise in the number of people with access to the Internet, and by extension the number of people attempting to hack other people's servers, made encrypted alternatives necessary.

3.4 DEFINE THE MODULES:

3.4.1 WIFI MODULE

The ESP8266 is a low-cost WiFi module that can be integrated easily into IoT devices. The ESP8266 is a small WiFi module built around the ESP8266 chip that can connect your microcontroller to the internet wirelessly for a very small cost. It can be a great option for Internet of Things (IoT) projects, but can be difficult to work with for beginner hobbyists who do not have prior experience with the module.

Fig.3.38 ESP8266 ESP-01 Module

3.3.2 ESP-01 Features

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 1MB Flash Memory
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)

The first feature to notice about the ESP8266 is its awkwardly spaced header pins. The module has 8 pins that serve different functions, but they are packed in a 4×2 arrangement that makes plugging the module into a breadboard impossible. This means that to prototype projects on a breadboard, you'll need male-female jumper wires to connect the pins on the ESP8266 to rows on the breadboard. If you'd like to make your prototyping more compact, you can also purchase breadboard breakouts for the ESP8266 such as this one. For prototyping, I chose to just use jumper wires.

The pinout for the ESP8266's pins are according to the following diagram:

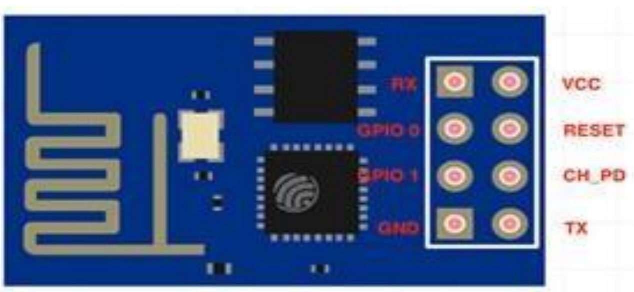


Fig.3.39 ESP8266 Pinout

RX: UART serial communication receive pin

GPIO 0: GPIO pin (unused in this project)

GPIO 1: GPIO pin (unused in this project)

GND: Connection to Ground

VCC: Connection to 3.3V Vcc (Vcc cannot exceed 3.3V!)

RESET: Reset pin (pull down to reset)

CH_PD: Chip enable and power down pin

TX: UART serial communication transmit pin

Note that the maximum voltage input for the ESP8266 is 3.3V. Any input voltage greater than 3.3V will damage the module! To program settings on the ESP8266, we'll first need to connect it to a serial terminal on a computer through which we can send it special commands. Settings that we'll have to program include, for example, the SSID and password for the wifi network the module will be connected to. To connect the ESP8266 to a computer and configure its settings, we'll need a USB to serial adapter with 3.3V logic, along with a serial terminal program. Fortunately for us, we have the Arduino and the Arduino IDE's serial monitor! This means that we'll just have to connect the ESP8266 module to the Arduino and upload a custom sketch to the Arduino. Use the following diagram to connect the ESP8266 module to the Arduino:

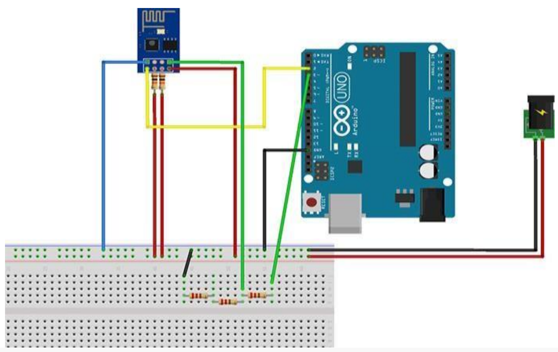


Fig.3.40 Connection Between ESP8266 and Arduino Uno

- Arduino Uno or similar module
- ESP8266 ESP-01 module
- 2 x 1kΩ resistors for CH_PD and RESET pull-up
- 3 x 220Ω resistors for serial line voltage divider
- 3.3V regulated power supply for ESP8266 (you cannot plug the power line into the Arduino's 3.3V out pin because the ESP8266 draws a lot of current!)
- Breadboard and jumper wires

First, the ESP8266 module is powered by a 3.3V regulated power supply. As described earlier, the ESP8266 has a maximum voltage input rating of 3.3 volts. Using a power supply rated higher than this recommended 3.3 volts could fry the module! When connecting the ESP8266 to this power supply, you should not only connect the Vcc and Ground pins to the power supply lines, but also connect the power supply's ground line to the Arduino's ground pin. Since the Arduino will be powered by the USB connection to the laptop, creating a common ground essentially creates a common reference (you can think of it as a baseline) to compare voltages and thereby interpret digital high and low signals.

Second, the receive (RX) line for the ESP8266 module is connected to the output of a resistor voltage-divider circuit. We do this to shift the serial communication logic level (the highs and lows of the digital signals that make up the serial communications) from a logic high of 5 volts on the Arduino to a logic high of 3.3 volts on the ESP8266 module. Again, the ESP8266 is specified with 3.3V logic, so connecting the module's receive line directly to the Arduino's transmit line could damage the device. We do not need to shift levels on the ESP8266's transmit line because the module's 3.3V logic high is a high enough voltage to also register as a logic high on the Arduino.

Finally, there are two pull-up resistors on the ESP8266's CH_PD and RESET pins. These are specified by the module's creators to ensure that the device functions properly while it is being used. To reset the ESP8266 or disable it, these pins must be pulled down, but since we do not want those processes to occur, we will instead pull these pins high through some 10kΩ pull-up resistors.

3.4.3 STRUCTURE AND CONFIGURATION:

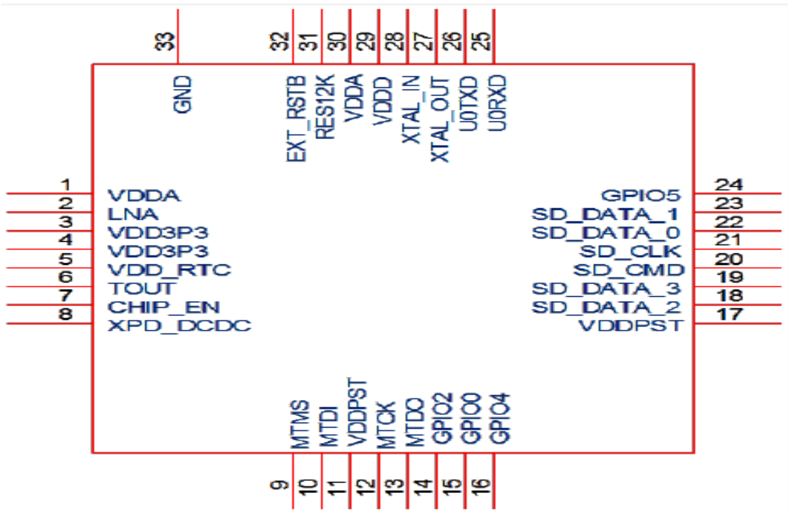


Fig.3.41 Structure and configuration

3.5 MODULE FUNCTIONALITIES:

3.5.1 ESP8266 WIFI MODULE

The ESP8266 WiFi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all WiFi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community. This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existance interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts. There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support. In the Documents section below you will find many resources to aid you in using the ESP8266, even instructions on how to transform this module into an IoT (Internet of Things) solution.

CHAPTER-4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM

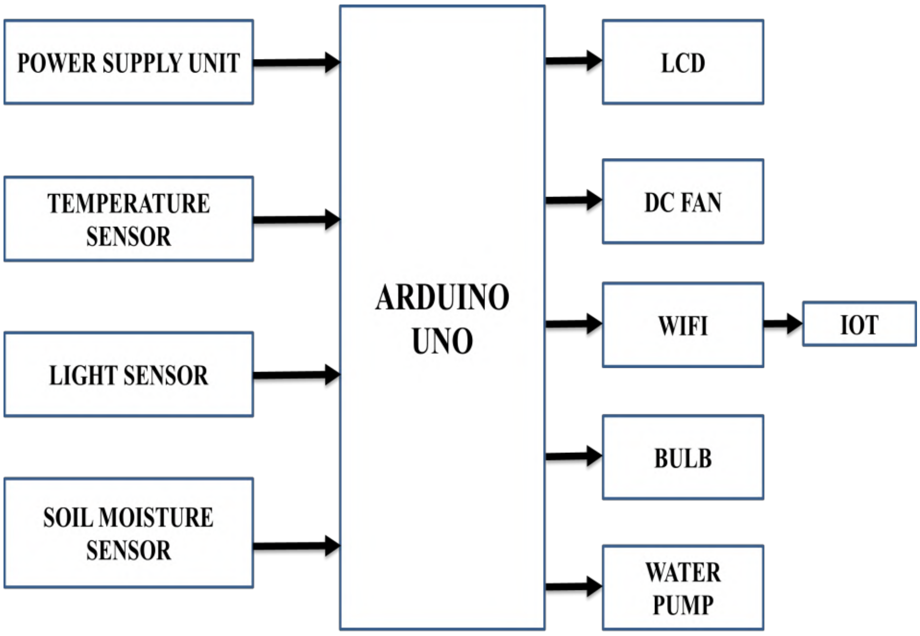


Fig.4.1 Block Diagram

The system consists of three input sensors to detect temperature,light intensity and soil moisture sensor.Corresponding load is given to the arduino uno as DC fan,bulb and water pump respectively.The data sensed is sent to the microcontroller and it controls the load according to the varying parameter.At the same time ,data regarding these parameters are sent to IOT module .WiFi module is used to establish connection between microcontroller and the WiFi network and make TCP/IP connections and sends the data to the IOT.

4.2 FLOW CHART

The system is turned on. The parameters such as temperature, light intensity and soil moisture is read using the temperature sensor, LDR and soil moisture sensor respectively. The read parameters are then sent to arduino .If the temperature exceeds the threshold value then the fan is turned on to cool the system. When the temperature becomes normal the fan gets turned off automatically. Initially the light sensor is kept low. Hence when the sensor detects no light the bulb is turned off. The data collected from the soil moisture sensor is sent to the microcontroller and it performs the operation of turning the pump on if the soil is detected as dry,else turned off if the soil retains the moisture content. The sensed data is constantly sent to the IOT where the WiFi module establishes a connection and the user retrieves the data.

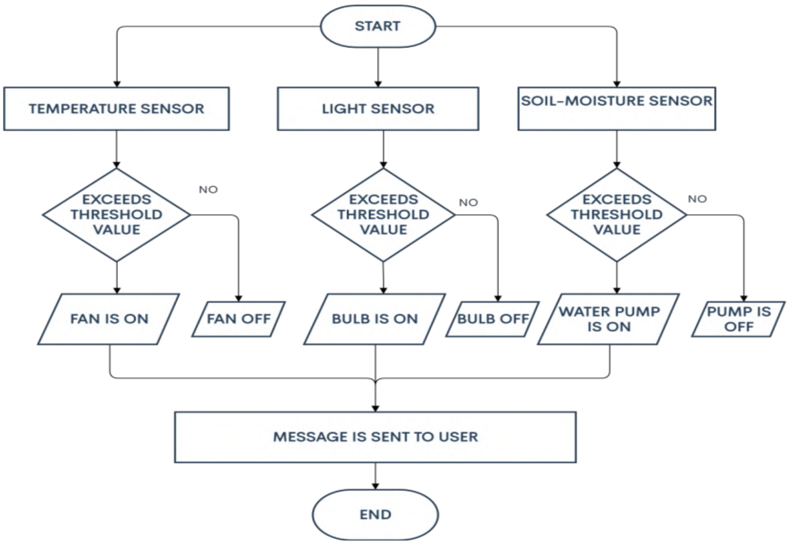


Fig 4.2 Flow Chart

4.3 IMPLEMENTATION STAGES

Stage-1: Firstly on lcd the message is displayed showing IOT Based Greenhouse indicating the succesful turning on of the system.

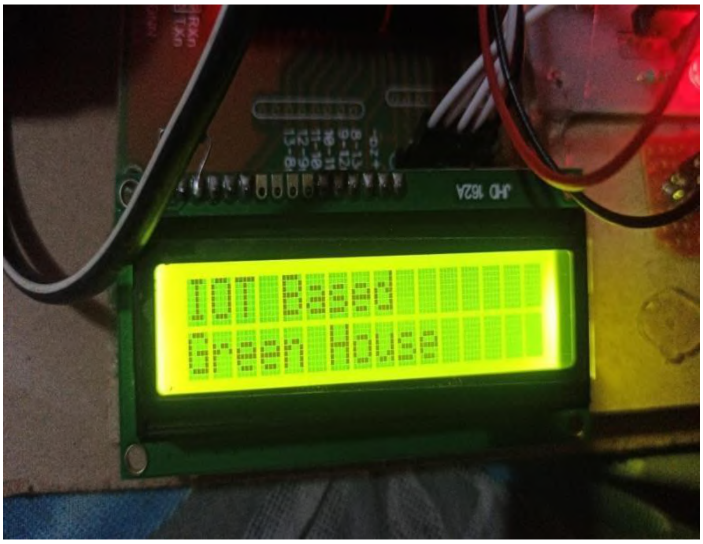


Fig.4.3 LCD Display When The System Is Turned On

Stage-2: Secondly on lcd receive a message displaying waiting for link that indicates the wifi is not yet connected.

Fig.4.4 Waiting For Link

Stage-3: In Mobile Telnet App, message such as welcome is shown when the wifi is connected to ESP8266 WIFI Module.

Fig.4.5 Welcome Message Displayed

Stage-4: After the system is turned on,the detected parameters are shown on the LCD

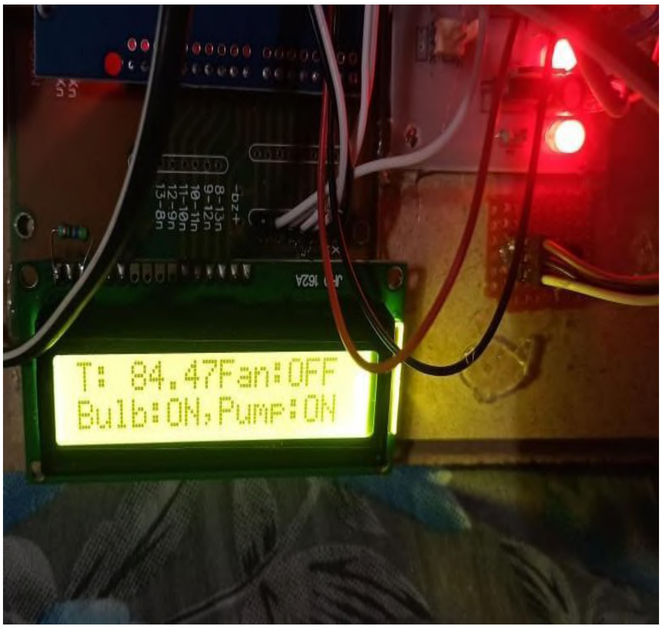


Fig.4.6 System Working During Night With Normal Temperature And Dry Soil

Stage-5: The soil is made wet and then parameter gets displayed on the LCD.

Fig.4.7 Result When Soil Is Wet

Stage-6: When the temperature inside the greenhouse structure rises,the message displayed on LCD is noted indicating the fan is turned on to cool the system.

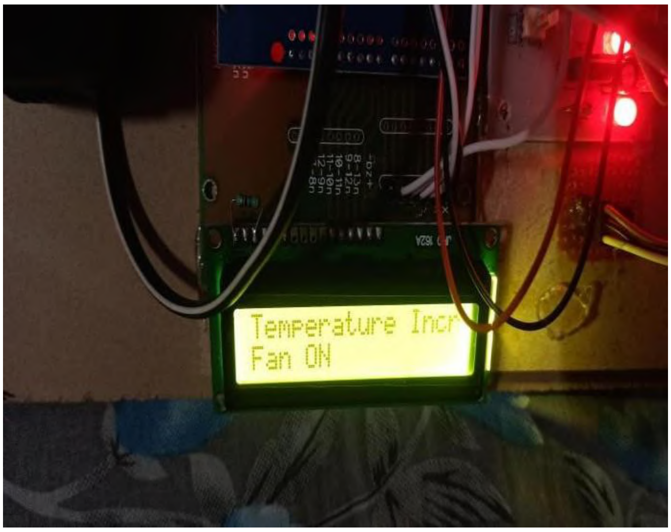


Fig.4.8 When The Temperature Exceeds The Set Threshold Value

Stage-7-Finally,the system is tested during the day indicating the presence of light which in turn puts on the bulb.

Fig.4.9 Result during Day Time

4.5 RESULTS

This hardware setup gives a brief idea as to how the project “Greenhouse Monitoring And Control System Using IOT” works. In the hardware implementation, all the sensors used are connected to the microcontroller. The system has been wired using jumper cables. A DC power supply of 5V is given to the microcontroller and to the sensors. Besides, 9V DC power supply is provided for light, fan and the water pump. An LCD is mounted on the board. The connections are made between the microcontroller and the sensors. The WiFi module establishes a path between the microcontroller and the user's device to get the sensor value in order to monitor and control the system. If the values of the parameters exceeds the load connected to it gets automatically turned on and when the value is in normal condition the fan, light and the motor remains off. The system is capable of sending the measured values to the user's device such as a mobile via wireless connection. Thus controlling and monitoring occurs simultaneously.

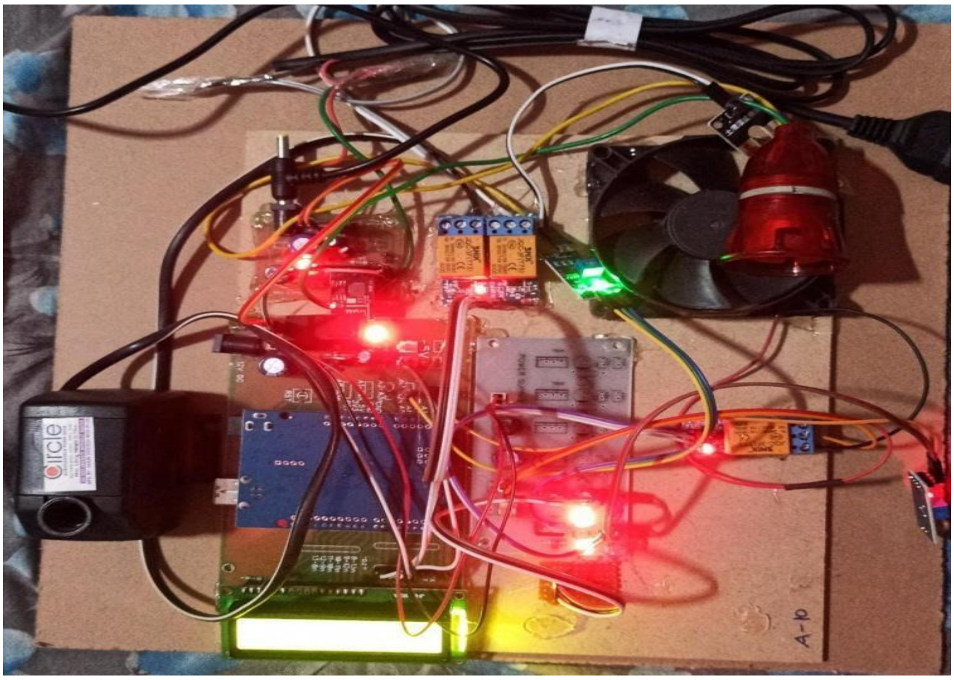


Fig.4.10 Final Output

CHAPTER-5

PROJECT TESTING

5.1 OVERVIEW OF TESTING METHODS

Today, government wants to take the greenhouse monitoring and control system one step further, with more sophisticated systems included in it to increase the cultivation and the production rate.Department of Agriculture and Cooperation has been promoting capital investment and has been extending financial assistances for the building of greenhouse structures.It is thus obvious that for ensuring rapid growth of quality infrastructure for protected cultivation.This has necessitated prescribing technical standards specifications and linking the same with normative costs in larger cost in large interest of the industry and user farmers.

Security: IoT is a combination of physical objects (“things”), sensors, embedded software and keeps a goal of effective connectivity and faithful data exchange.
The current graphs imply that IoT market growing at remarkable rate. However, some of these devices suffer from limited memory, power consumption and processing power.

These issues may cause IoT to become penetrable and hence security is at utmost priority to IoT domain.

As stated in a 2015 report by Hewlett-Packard on IoT research [3]:
1. Per 100, 70 devices still use un-encrypted network services
2. Per 100, 90 devices collected at least one piece of personally identifiable information (via device, cloud or mobile app)
3. Per 100, 70 devices (with cloud and mobile app components) enabled an attacker to identify valid user accounts through enumeration.
4. 80% of devices (with cloud and mobile app components) failed to require passwords of sufficient complexity
5.Behind every 10 device every 6 devices user interfaces (UI) were vulnerable

Following are the vulnerabilities of the IoT as per The Open Web Application Security Project (OWASP)

1) Insufficient protection compositions.
2) Dubiously secured Software/Firmware
3) Doubtful Network Services
4) Doubtful Web Interface
5) Worst Physical Security
6) Doubtful Cloud Interface
7) Doubtful Mobile Interface
8) Privacy Concerns
9) Doubtful Authentication/Authorization
10) Inadequate Transport Encryption Ongoing approaches to secure IoT have attempted to put a grip on communication protocol-based mechanisms, such as encryption for data-at-rest or in-transit.

But this itself is doubtful if the respective endpoints themselves are capable of being modified either by local access or remote connections.

Gartner claims that by year 2020 more than 25% of identified attacks in a particular company will be on IoT devices or systems, even though IoT will only contribute to less than 10% of IT security budgets.

The testing methods of the project which is Greenhouse Monitoring and Controlling System using IoT is it requires a mobile app, wifi connection . Firstly on lcd we recieve a message displaying waiting for link that indicates the wifi is not yet connected.In Mobile Telnet App, we will get a message like welcome when the wifi is connected to ESP8266 WIFI Module.After the wifi is connected,we will get a message on LCD displaying Please Scan UR Card.When we scan the card, if sufficient amount is available, we will get a message access granted for person and remaining balance will be displayed on LCD.When we scan the card, if sufficient amount is not available, we will get a message access denied for person and balance will be displayed on LCD.When card is recharged, amount will be added to card and display’s person recharge done on LCD Screen.

CHAPTER-6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION:

Currently the existing system have limitation of not being able to simultaneously measure the parameters. This system enables people to monitor and manage growing conditions of their greenhouses. The use of modern sensors with integration of internet will deliver real-time updates about the crops and help people grow their plants more effectively and increase the yields. With all the tests results it can be concluded that the project will provide a solution for automating greenhouse activities and irrigation activities. Implementation of such systems can definitely help in overall production .The IOT used in the system enables the conveyance of the information gathered about greenhouse to the user..The project depicts how technology reduces human efforts.

6.2 FUTURE ENHANCEMENT:

- The system performance can be further expanded by increasing operating speed, memory capacity and instruction cycle period of microcontroller.
- To operate multiple greenhouses concurrently ,a multi-controller system can be developed. Connection establishment of WiFi with other devices like MODEM or cellular phones can be introduced apart from android phones.
- Greenhouses with high value crops are vulnerable targets for thieves. So security systems can be implemented by adding on surveillance networks with CCTV.
- More sensors can be added in addition to the ones already used like rain sensors.
- Time bound administration of fertilizers, insecticides and pesticides can be introduced.
- A multi controller system can be developed that will enable a master controller along with its slave controllers to automate multiple greenhouses simultaneously.

PUBLICATION

Submitted Paper in the Conference ICSMEC - 21 with Paper ID (ICISEC21-0053) and got Acceptance for the Paper.

REFERENCES

[1]. M.K Gayathri and J. Jayasakthi , “Providing Smart Agricultural Solutions to Farmers for Better Yielding Using IOT,” 2015 IEEE International Conference on Technological Innovations for Agriculture and Rural Development.

[2]. Viswanath Naik , “IOT based Green House Monitoring System”, International Journal for Electronics & Communication Engineering and Technology, vol 6,issue 6,June 2015,pp 45-47

[3]. P.S Asolkar, “An Effective Method of Controlling Green House and Crop Monitoring Using GSM”,2015 International Conference on Computing Communication Control and Automation.

[4]. Orazio Mirabella, “Hybrid Wireless Networking Infrastructure for Green House Management”, IEEE transactions on Instrumentations and Measurement ,Vol 60,February 2011.

[5]. M.Manscuo and F.Busttaffa , “A wireless Sensor Network for Monitoring Environmental Variables in Green House”, 6th IEEE International Workshop on Factory Communication System, June 2006.

BIBILOGRAPHY

[AEMC Instruments, 2008] AEMC Instruments (2008). Why Mea- sure Soil Resistivity? Available from: http://www.pema.ie/PDFs/ App-Ground-SoilResistivity.pdf [cited2015-03-07].

[D-Robotics UK, 2010]D-Robotics UK (2010). DHT11 Humidity and Temperature Sensor. Available from: http://www.micro4you.com/files/sensor/DHT11.pdf [cited 2015-03-08].

[Electronics Tutorials, 2014b]Electronics Tutorials (2014b). Temperature Sensors. Available from: http://www.electronics-tutorials.ws/io/io_3.html[cited 2015-05-10].

[Giorgos Lazaridis, 2010] Giorgos Lazaridis (2010) . How DC Fans Work. Available from http://pcbheaven.com/wikipages/How_DC_Fans_Work/ [cited 2015-05- 06].

[MathWorks, 2014]MathWorks (2014). Arduino Support from MATLAB. Avail- able from: http://se.mathworks.com/hardware-support/arduino-matlab. html [cited2015-05-07].

APPENDIX

4.4 PROJECT CODE

```
#include "LiquidCrystal.h"

LiquidCrystal lcd(13,12,11,10, 9,8); // RS, E, D4, D5, D6, D7

#include <SoftwareSerial.h>

SoftwareSerial mySerial (2, 3); //RX, TX respetively

#define Temp Pin A4

int Idr=A5;

int soil-A3;

int TempValue;

int pump=4;

int bulb=5;

int fan=6;

char res [130];

char buff[130];

void sendwifi (String chr, unsigned int len)

{

mySerial.print("AT+CIPSEND=0,");
```

```
mySerial.println(len-1);
delay(2000);
mySerial.print(chr);
delay(2000);
chart;
void setup()
{
pinMode (ldr, INPUT);
pinMode (soil, INPUT);
pinMode (fan, OUTPUT);
pinMode(pump, OUTPUT);
pinMode (bulb, OUTPUT);
digitalWrite(fan, LOW);
digitalWrite (pump, LOW);
digitalWrite (bulb, LOW);
Serial.begin(9600);
mySerial.begin(115200);
lcd.begin(16, 2);
lcd.clear(); lcd.setCursor(0, 0); lcd.print("IOT Based");
delay(30);
lcd.setCursor(0,1);
lcd.print("Green House");
delay(3000);
mySerial.print ("AT\r\n");
delay(1000);
mySerial.print("ATEO\r\n");
delay(1000);
```

```
mySerial.print ("AT+CWMODE=3\r\n");
delay(1000);
mySerial.print("AT+CWSAP=\"Project\",\"project1235\",5,0\r\n\r\n");
delay(1000);
mySerial.print ("AT+CIPMUX=1\r\n");
delay(1000);
mySerial.print ("AT+CIPSERVER=1,23\r\n");
delay(1000);
lcd.clear(); lcd.setCursor(0, 0); lcd.print("WATING FOR LINK");
while (1)
if (mySerial.available())
{
//if (Esp.find("O, LINK"))
if (mySerial.find("O, CONNECT"))
lcd.clear(); lcd.setCursor(0, 0); lcd.print("LINK FOUND");
break;
delay(1000);
sendwifi ("WELCOME \r\n", 11);
delay(1000);
}
void loop()
{
TempValue = analogRead (TempPin); // Getting LM35 value and saving it in
variable
float TempCel = ( TempValue/1024.0) +500; // Getting the celsius value
from 10 bit analog value
float Temp Farh = (TempCel+9)/5 + 32; // Converting Celsius into
```

Fahrenhiet

lcd.clear(); lcd.setCursor(0,0);lcd.print("T:

"); lcd.print (TempCel); lcd.setCursor(0,0); lcd.print("Fan: OFF");

lcd.setCursor(0,1);lcd.print("Bulb:ON, Pump:ON");

delay(100);

sendwifi ("Temp: Normal \r\n",16);

delay(100);

sendwifi ("Bulb, Pump: ON, Fan: OFF \r\n",28);

delay(100);

digitalWrite (pump, HIGH);

digitalWrite (bulb, HIGH);

digitalWrite (fan, LOW);

delay(2000);

if (TempCel>=40)

lcd.clear(); lcd.setCursor(0,0);lcd.print("Temperature Increased");

lcd.setCursor(0,1);lcd.print("Fan ON");

delay(100);

sendwifi ("High Temperature \r\n", 20);

delay(10);

digitalWrite(fan, HIGH);

delay(3000);

}

if (digitalRead(dr) ==LOW)

{

1

lcd.clear(); lcd.setCursor(0,0);lcd.print("Day Time");

lcd.setCursor(0,1);lcd.print("Bulb OFF");

```
delay(100);
sendwifi ("Bulb OFF \r\n", 12);
delay(10);
digitalWrite (bulb, LOW);
delay(3000);
if (digitalRead (soil) ==LOW)
{
lcd.clear(); lcd.setCursor(0,0); lcd.print("Soil is Wet");
lcd.setCursor(0,1); lcd.print("Pump OFF");
delay (100);
sendwifi ("Pump OFF \r\n", 12);
delay(10);
digitalWrite (pump, LOW);
delay(3000);
}
}
```

A

PROJECT REPORT

On

REAL TIME VEHICLE THEFT CONTROL SYSTEM USING ENGINE TEMPERATURE MEASUREMENT

Submitted by

1) Ms.G.Jeevani (17K81A0418)

2) Ms.P.Keerthi (17K81A0436)

3) Ms.K.Supriya (18K85A0406)

in partial fulfillment for the award of the degreeof

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Mr.G.Ramesh

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST.MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE

An UGC Autonomous Institute

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

Dhulapally, Secunderabad-500 100.

NBA &NAAC A+ Accredited

2020-2021

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**Real Time Vehicle Theft Control System Using Engine Temperature Measurement**", is being submitted by **G.Jeevani (17K81A0418), P.Keerthi (17K81A0436), K.Supriya (18K85A0406)** in partial fulfillment of the requirement for the award of the degree of **Bachelor Of Technology in Electronics and Communication Engineering** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Internal Guide	**Head of the Department**
Mr.G.Ramesh	Dr.B.Hari Krishna
Department of ECE	Department of ECE

Internal Examiner	External Examiner

Place:

Date:

DECLARATION

We, the students of **Bachelor of Technology** in Department of **Electronics and Communication Engineering**, session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled **'Real Time Vehicle Theft Control System Using Engine Temperature Measurement'** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

G.Jeevani (17K81A0418)
P.Keerthi (17K81A0436)
K.Supriya (18K85A0406)

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna**, **Head & Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept. project coordinator Dr.A.Anand, Professor** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide **Mr.G.Ramesh, Assistant Professor**, Department of ECE and our class project coordinator **Ms.A.Sowmya, Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. G.Jeevani
2. P.Keerthi
3. K.Supriya

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND DEFINITIONS

REAL TIME VEHICLE THEFT CONTROL SYSTEM USING ENGINE TEMPERATURE MEASUREMENT

ABSTRACT

This project deals with the design and development of a theft control system for an automobile, which is being used to prevent or control the theft of a vehicle. The developed system makes use of an embedded system based on GSM technology. The designed and developed system is installed in the vehicle. An interfacing mobile is also connected to the microcontroller, which is in turn, connected to the engine. Once, the vehicle is being stolen, the information is being used by the vehicle owner for further processing. The information is passed onto the central processing system. By reading the signals received by the mobile, one can control the ignition of the engine; say to lock it or to stop the engine immediately.

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

This project is used when the vehicle is stolen i.e., when the temperature in the engine varies the information is passed onto the central processing system, whereby sitting at a remote place, a particular number is dialed by them to the interfacing mobile that is with the hardware kit which is installed in the vehicle.The owner can say to lock it or to stop the engine immediately. Again, it will come to the normal condition only after entering a secured password. The owner of the vehicle & the central processing system will know this secured password. The main concept in this design is introducing the mobile communications into the embedded system. The designed unit is very simple & low cost. The entire designed unit is on a single chip. When the vehicle is stolen, owner of vehicle may inform to the central processing system, then they will stop the vehicle by just giving a text to that secret number and with the help of SIM tracking knows the location of vehicle and informs to the local police or stops it from further movement. Here temperature sensor is used to know the engine position with help of heat dissipated in it.

1.2 OBJECTIVES OF THE STUDY

The main objective of this project is to develop a automatic ignition control system of vehicle for detecting theft. Here are some of the methods to provide safety measures at low cost. Theft prevention and safety measures can be provided by using Digital Ignition lock with highly efficient sensors like MEMS. This project designs an automatic ignition control system of vehicle which monitors abnormal situations. Many ideas were either proposed or formulated to provide safety to the vehicles by using the combination of pattern ignition with the MEMS to protect the vehicles from theft.

1.3 SCOPE OF THE STUDY

The vehicle theft and tracking machine is an essential kit to reduce the no. of increased theft of vehicles which is commonly known as antitheft kit. Most of the theft cases are occurred due to the Hot wiring, Hijacking and opportunistic theft. Certain safety measures are only available at high cost. These GSM and GPS tracking systems for cars project significantly reduce the time, manpower and operates without interference of humanoid. In the modern world, there are various new technologies like GPS, GSM, RFID, Biometric Recognition. Mobile communication has been integrated into vehicles for security purposes. In this project GPS technology is used to find the exact location of the vehicle and GSM is used to send the message to the owner of the vehicle.

1.4 MATERIAL REQUIREMENT

HARDWARE REQUIREMENT :

- Arduino Uno
- Power Supply Unit
- Temperature Sensor
- GPS Module
- LCD
- L293D
- DC Motor
- GSM Module

SOFTWARE REQUIREMENT :

- Arduino IDE
- Proteus

1.5 PROCUMENT OF EQUIPMENT

Arduino:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- We can control board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. We can simply use a USB cable.
- The Arduino IDE uses a simplified version of C++, making it easier to learn to program.

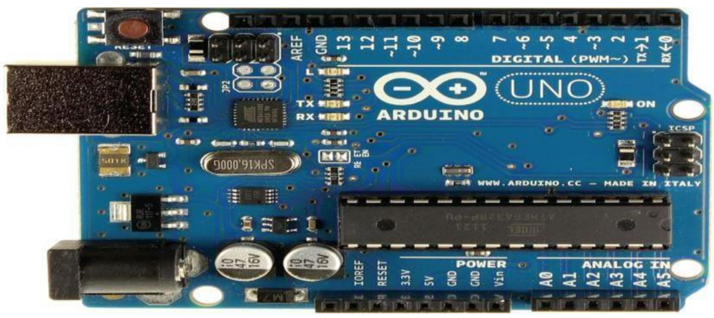


Fig 1.1: Arduino Uno

GPS:

The Global Positioning System (GPS), is one of the global navigation satellite systems (GNSS) that provides geolocation and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. The GPS does not require the user to transmit any data, and it operates independently of any telephonic or internet reception, though these technologies can enhance the usefulness of the GPS positioning information. The GPS provides critical positioning capabilities to military, civil, and commercial users around the world.

Fig 1.2: GPS Module

GSM Module:

GSM (Global System for Mobile communications) is an open, digital cellular technology used for transmitting mobile voice and data services.GSM (Global System for Mobile communication) is a digital mobile telephone system that is widely used in Europe and other parts of the world. GSM uses a variation of Time Division Multiple Access (TDMA) and is the most widely used of the three digital wireless telephone technologies (TDMA, GSM, and CDMA). GSM digitizes and compresses data, then sends it down a channel with two other streams of user data, each in its own time slot.

Fig 1.3: GSM Module

TEMPERATURE SENSOR (LM35):

LM35 is a temperature sensor that outputs an analog signal which is proportional to the instantaneous temperature. The output voltage can easily be interpreted to obtain a temperature reading in Celsius. The advantage of lm35 over thermistor is it does not require any external calibration. The coating also protects it from self-heating. Low cost and greater accuracy made it popular . LM35 can measure from -55 degrees centigrade to 150-degree centigrade. The accuracy level is very high if operated at optimal temperature and humidity levels. The conversion of the output voltage to centigrade is also easy and straight forward.

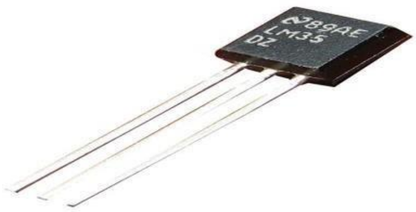


Fig 1.4: Temperature Sensor

Proteus:

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It is a software suite containing schematic,simulation as well as PCB designing.

ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.
The designer can also develop 2D drawings for the product.

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components.

CHAPTER 2

LITERATURE SURVEY

2.0 LITERATURE REVIEW ON RESEARCH AREA

In India according to vehicle theft, burglary census from 2013 the vehicle thefts are increasing nearly 8.47 % on an average, for this situation the technology to avoid the theft of the vehicle must also be increased, Microcontroller based real time vehicle theft detection and prevention system provide solution for this problem. The Global system of mobile (GSM) communication is a globally accepted standard for cellular communication. The vehicle owner uses Subscriber Identity Module (SIM) inserted with in his mobile phone to send messages to GSM modem and GPS system which is a part of the vehicle theft prevention system that is attached to vehicle. This system is used for any vehicle like bus, bikes or cars and is cost effective. Some people uses the GPS system only to the vehicle to trace the vehicle location like the latitude, longitude and speed of the vehicle but not useful for controlling the vehicle. Some people uses only GSM for controlling the vehicle but not useful to trace the vehicle, some researchers uses GSM, GPS system to control the vehicle as well as to trace its location. The literature review of the work is as follows.

A researcher developed an anti-burglary vehicle security system, which uses thumb impression to start the vehicle. The authorized persons thumb impressions are stored in the database of the system. The vehicle is started if the finger print of the database is matched. If anyone accessed the vehicle by chance then the fuel tank will be emptied through the relay bolt fitted to the tank at the same time it gives alarm that the vehicle is theft so that the unauthorized person cannot refill the emptied fuel tank.

Some researchers uses GSM, GPS & RFID security system for taxi like vehicles. For starting the vehicle the worker must use the RFID card in which the identification number is provided such that the identification numbers already preloaded in to the database of the system, If the number is matched, GPS and GSM comes in to play and sends SMS to the vehicle owner the location like latitude and longitude of the vehicle. If the owner detected theft

by chance then he sends the SMS to the GSM such that it will lock the doors of the vehicle.

A researcher used GSM system, Microcontroller, and relay switch for the ignition system. It theft is detected the Microcontroller activates the GSM system to send SMS to the owner, If the owner gives reply to the SMS then the relay switch is activated and it deactivate the ignition system.

Another researcher used GPS-GSM system that uses Google earth application. The system contains GPS module provided in the vehicle, this GPS module exchanges information with the GSM system to send SMS to the owner. After getting SMS to the owner, he can trace the latitude, longitude and speed of the vehicle using Google earth application.

2.1 REVIEW ON RELATED LITERATURE

Vehicle tracking has become necessary in this modern age due to the fact that several vehicles have been stolen and are still being stolen. The owners of these vehicles often find it very challenging to locate and recover their vehicles. But with the advent of vehicle tracking technologies, it has become quite easier to locate these vehicles. It is not just enough to recover the stolen vehicle but there is a need to ensure that the identity of the thief is also captured by a camera so that he can be arrested and penalized. This act will serve as a deterrent to many vehicle thieves. Some of the anti-vehicle theft control systems that have been designed include:

A GPS–GSM Based Tracking System: The system uses the global positioning system to determine the precise location of an object, person or other asset to which it is attached and using GSM modem this information could then be transmitted to a remote user. It provides tele-monitoring system for inter-cities transportation vehicles such as taxis and buses. This system contains single-board embedded system that is equipped with GPS and GSM modems along with ARM processor that is installed in the vehicle. During object motion, its location can be reported by SMS message. This system finds its application in real time traffic surveillance. The result achieved by this project is the fact that positional data (in terms of latitude and longitude) of objects carrying the system can easily be retrieved. Also a stolen vehicle could be immobilised remotely by sending an SMS to the SIM number in the GSM module. However this is not without limitations. There is no way the thief can be identified so that he could be arrested after he had stolen the vehicle.

In a GSM and GPS based vehicle location and tracking system using AT89S52 microcontroller was developed.

In a Remote Vehicle Tracking and Driver Health Monitoring System Using GSM Modem and Google Maps was developed. The GSM modem at the control centre receives the coordinates through Short Message Service (SMS) and updates the main database. The information then is accessed through the website and the position of the vehicle is displayed through the Google Maps application.

In 'Vehicle tracking system with GPS GSM Interface and Self-Created Map' was carried out basically to provide remote monitoring capabilities by the owner or company manager. The Geographical Information system (GIS) was incorporated into the design to give exact or nearby location of vehicle on a map that was self-created. In order to get the position on Google map, Google API was used.

In a GSM and GPS based system for Vehicle Tracking and Employee Security System was proposed. It consisted of a car unit, emergency button and company unit. Car unit was placed inside the car. When the car picked up the employee; he/she had to swap an RFID card. The micro controller then matched the RFID card number with its database records and sends the employee's identity, cab identity & the cab position co-ordinates to the company unit via GSM module. Emergency button is a part of the car unit. There are three to four emergency buttons in the car. These buttons were placed at such position so that an employee can access them easily i.e. near the door unlocking handle. If employee found himself/herself in a problem, he/she would press the button. Microcontroller then detects the action & sends a signal to the GSM which will coordinate with the company unit and police. Microcontroller also sent a signal to the relay which will turn off the car ignition & stop the car.

2.2 CONCLUSIONS ON REVIEWS

Chen, H., Chiang "Real Time Vehicle Ceasing and Tracking Using GSM and GPS Technology" described to track the theft vehicle by using GPS and GSM technology. This system puts into the sleeping mode after the vehicle gets handled by the owner or authorized persons through the reset button over it.

Ramani, S.Valarmathy "Vehicle Tracking and Locking System Based on GSM and GPS" described When the theft identified, the responsible people send SMS to the micro controller, then issue the control signals to stop the engine motor. This design will continuously watch a moving Vehicle and report the status of the Vehicle on demand

Ch. Bhanu Prakash "Design and Implementation of a Vehicle Theft Control Unit using GSM and CAN Technology" described the present security system that will warn the owner of the vehicle by sending SMS when there has been an intrusion into the vehicle. SMS is a good choice of the communication to replace the conventional alarm.

CHAPTER 3

PROJECT DESIGN

3.0 OVERVIEW OF THE DESIGN

The main scope of this project is to send an alert message to the owner of the vehicle when the vehicle is stolen. This project includes a GSM modem, GPS system, microcontroller, buzzer and solenoid valve for fuel cut-off and a sensor to detect vehicle theft. When someone tries to steal the vehicle then microcontroller gets interrupted and orders GSM Modem to send the SMS, the owner receives a SMS that his car is being stolen and also the exact location of the vehicle through GPS system. With a message from owner mobile, the ignition system and fuel supply to the vehicle is cut off and the buzzer gets off.

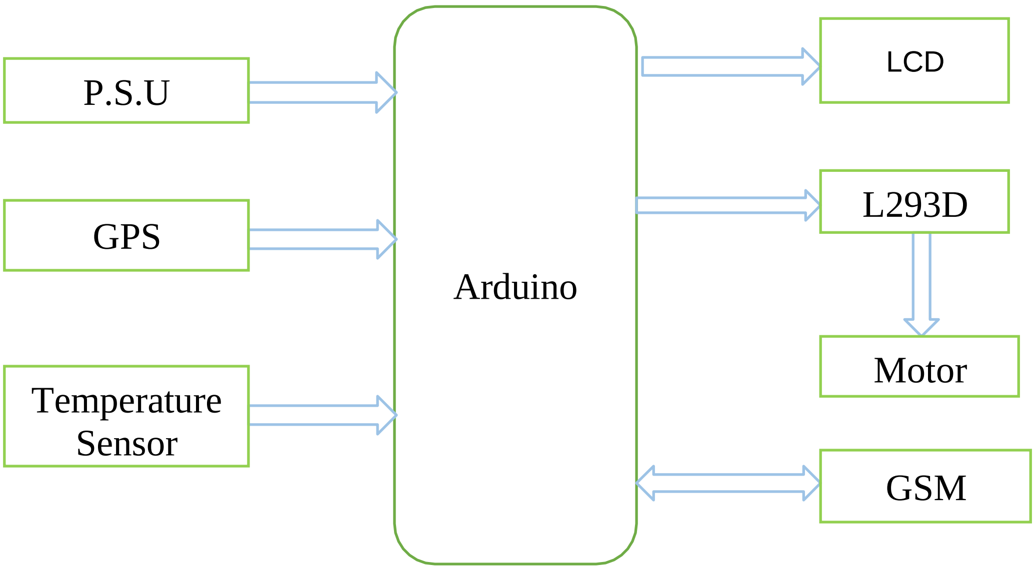


Fig 3.1: Block Diagram

While we go for work after riding the vehicle, the antitheft system must be kept in the active mode with the help of a switch present in the system. If anyone by chance starts the vehicle which is already in active mode, the voltage in the circuit becomes high which gives the signal to the microcontroller. This microcontroller again sends signal to the GSM, GPS to send SMSto the owner, and also its location like Longitude and Latitude of the vehicle through GPS to GSM and GSM sends the SMS to the owner. If the owner finds any threat to the vehicle then the owner with the help of SMS he will cut off the ignition.

3.1 EQUIPMENT ANALYSIS

Every lab running some kind of behavioral research makes use of several types of equipment and software for experimental control. The goal is typically to record events (behavioral or physiological) and generate signals (i.e., to control or synchronize different machines). Such tasks, in most cases, require millisecond-to-millisecond accuracy and thus require particular attention, since modern operating systems (OS) are not designed to operate in real-time and with such accuracy therefore, several different approaches are used, such as programming the experimental I/O task via dedicated and optimized software packages (i.e., E-Prime, Presentation, Psychophysics Toolbox for MATLAB, etc.) or devolving critical tasks to dedicated hardware with internal high precision clocks (i.e., external I/O boards from National Instruments, Measurement Computing, Cambridge Electronics Devices, etc.). In both cases, the solution tends to be expensive (especially for external boards), suboptimal insofar as it relies heavily on the OS's accuracy (this holds for all experimental control software), or unsuited for specific experiments.

In some cases, however, many low-level I/O tasks do not need specific software packages or expensive boards. For example, if the experimenter wants an event to be triggered when an event is detected via some sensors (i.e., touch sensor, force sensor, etc.), it is not necessary to use expensive hardware or software. In fact, simple and cheap microcontroller boards may solve many of these laboratory I/O tasks. Such boards are physical computing platforms based on a simple microcontroller and a development environment for writing software. These devices can be used to develop interactive objects, taking inputs from a variety of switches or sensors and controlling a variety of lights, motors, and other physical outputs. Projects of this kind can usually be stand-alone, or they can communicate with software running on a computer. Such boards have been around for several years and typically offer similar characteristics differing only in processor architecture (ARM, ATMEL, etc.), programming language (C/C++, BASIC, etc.), or other features (i.e., number of I/O channels, presence of analog channels, etc.). Several manufacturers have proposed quite popular solutions such as Parallax Inc., Coridium Corporation, FTDI, Picaxe, Arduino, as well as many others. All of these boards cost is very high . However, programming these boards can be quite complicated, and the user requires at least some basic electronics knowledge. Thus, the main obstacle to widespread use of these boards in psychological and neurophysiological labs is the steep learning curve.

However, Arduino boards offer one critical advantage: the open source philosophy (both hardware and software), which capitalizes on the massive nonexpert community that has flourished around the Arduino concept. A very rough estimate of the size of the community can be gleaned from a Google search reporting more than 12 million hits for "arduino." In fact, a large user base and the growing market have shown increasing interest around the Arduino concept. There are hundreds of open-source projects one can use or modify according to specific (experimental) needs. Many web tutorials cover basic programming and electronics issues, and there are active forums for help. Thus, learning to use Arduino boards may be a lot easier than learning to use similar products from other manufacturers. Given the available support from the Arduino community, even researchers with little programming and electronics background should consider using Arduino rather than other similar boards.

Arduino hardware consists of an open hardware design with an Atmel AVR processor. Arduino boards can be purchased preassembled, but hardware design information is also available for those willing to build or modify them . Several third-party makers have produced Shields (add-on boards) that are able to extend the basic capabilities of an Arduino . Among these shields, it is worth mentioning that the Motor Control Shield allows the control of DC motors and read encoders, the Xbee shield allows multiple Arduino boards to communicate wirelessly, and the Critical Velocity Accelerometer Shield integrates a 3-axis accelerometer. Additionally, third parties (+30) have released several variations based on the Arduino concept. These are companies building boards (typically with better specs or lower price) using the Arduino software.

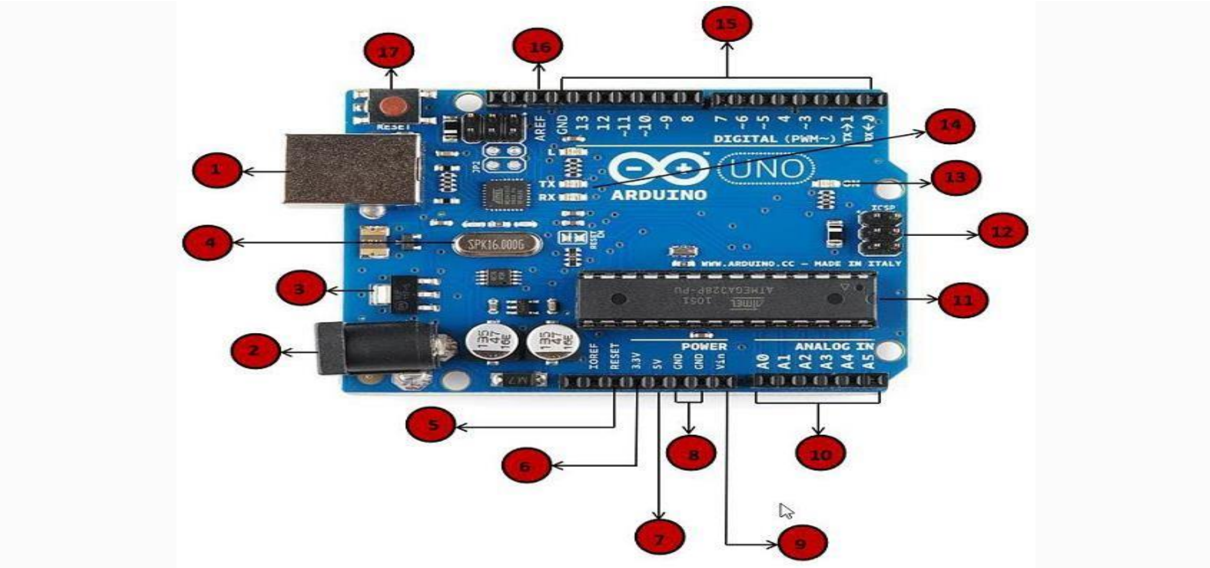


Fig 3.2: Arduino Board Pins

Table 3.1: Pin Description of Arduino

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).

6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

15	**Digital I/O** The ArduinoUNO board has14 digital I/O pins(15)(of which 6 provide PWM (Pulse Width Modulation)output.These pins can be configured to work as input digital pins to read logic values(0 or 1)or as digital output pins to drive different modules like LEDs, relays, etc.The pins labeled “~”can be used to generatePWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

The software consists of a standard programming language and a firmware that runs on the board. Arduino hardware is programmed using a language that is simplified C++, in a processing-based IDE. The software is then compiled and loaded on board. Arduino boards are compatible also with Flash, Processing, MaxMSP, and MATLAB, and a few lines of code often suffice to enable quite powerful behaviors . The basic programming structure of an Arduino is composed of at least two parts. These are the setup and the loop components. In the set-up, which runs at the beginning and only once to set pin mode or serial communication, the variables are declared. The second part runs in a loop that enables the script to change, to respond, and to control the Arduino board. After declaring variables, controlling the Arduino involves classic control structures (IF, IF…ELSE, FOR, etc.), arithmetic operators (+, -, /, *, etc.), and comparison operators (>, <, etc.) or boolean (AND, OR, etc.). There is also a set of commands for analog and digital read and write such as digitalwrite() or digitalread(). Furthermore, other commands can set temporal delays in milliseconds, perform basic mathematical and trigonometry operations (min/max, absolute value, square root, sine, cosine, etc.), or generate random numbers.

Design of power supply:

Power supply is the major concern for every electronic device .Since the controller and other devices used are low power devices there is a need to step down the voltage and as well as rectify the output to convert the output to a constant dc. Power supply unit is the basic requirement for electronic devices.

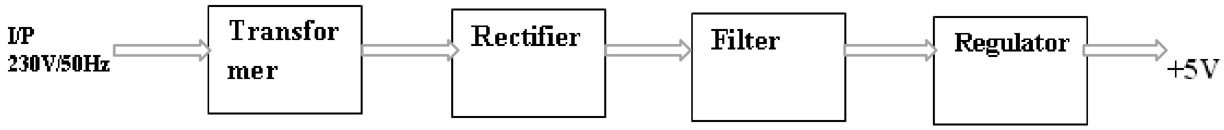


Fig 3.3: Power Supply

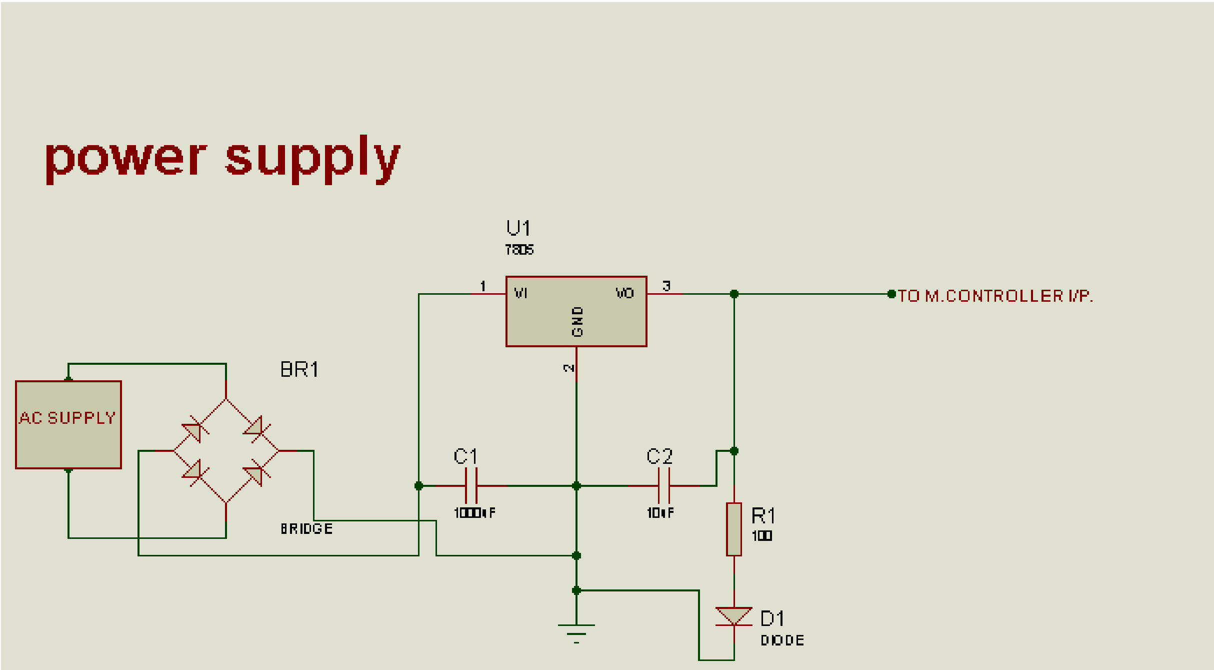


Fig 3.4: Circiut Diagram of power supply

Arduino IDE

The Arduino **Integrated Development Environment (IDE)** is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Liquid Cristal Display

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCDs connected to the contollers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Description Of 16x2:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all-Parallel Ports. It however doesn't show the use of the Status Port as an input. So we are interfacing a 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

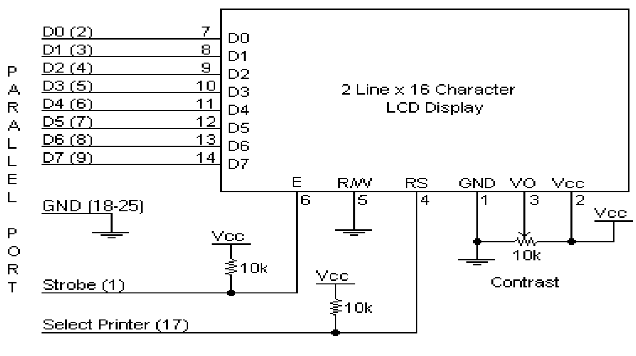


Fig 3.5: Schematic of LCD

Above is the quite simple schematic. The LCD panel's *Enable* and *Register Select* is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

L293D

L293D is basically a high current dual motor driver/controller Integrated Circuit (IC). It is able to drive load having current up to 1A at the voltage ranging from 4.5V to 36V. Motor driver usually act as current amplifier because they receive a low current signal as an input and provides high current signal at the output

L293D is basically a motor driver or controller. It has two builtin H-bridge circuits which are able to control two DC motors simultaneously in both clockwise and counter clockwise direction. It acts as an current high amplifier because it take low current signal at its input and provides higher current signal at the output in order to drives different load e.g stepper motor & DC motors. Its features include large inpu voltage supply range, large output current high noise immunity input signals etc. Its common real life applications include stepping motor drivers, relay drivers, DC motor drivers etc. L-293D motor driver/controller is shown in the figure given below.

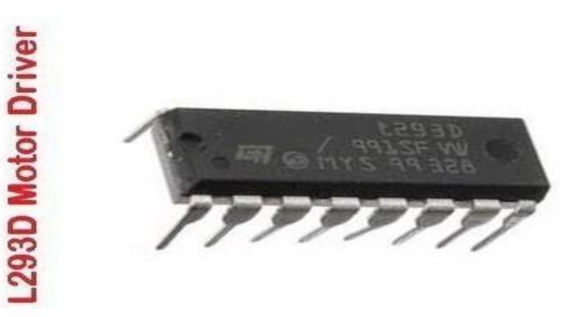


Fig 3.6: L293D Motor Driver

L293D Pins

- L-293D has sixteen (16) pins, having different individual functions.
- All of these sixteen pins along with their serial no, are given in the table shown below.

L293D Pin Functions

- L 293D each pin has different tasks to perform when they are in working condition.
- The tasks assigned to each pin are given in the table shown below.

Table 3.2: Pin Description of L293D

L293D Pins	
Pin. No	Pin Name
1	Enable 1,2
2	Input 1
3	Output 1
4	GND
5	GND
6	Output 2
7	Input 2
8	V_{CC2}
9	Enable 3,4
10	Input 3
11	Output 3
12	GND
13	GND
14	Output put 4
15	Input 4
16	V_{CC1}

DC Motors

The brushed DC motor is one of the earliest motor designs. Today, it is the motor of choice in the majority of variable speed and torque control applications.

Advantages

- Easy to understand design
- Easy to control speed
- Easy to control torque
- Simple, cheap drive design

WORKING OF DC MOTOR

In any electric motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. As you are well aware of from playing with magnets as a kid, opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion

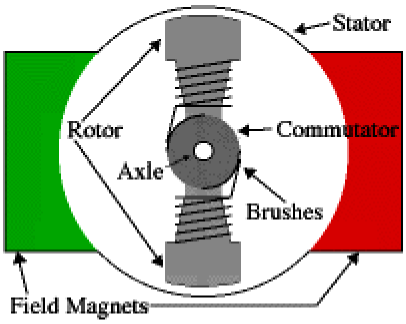


Fig 3.7: DC Motor Internal View

3.2 DEFINE THE MODULES

An efficient automotive security system is implemented for anti-theft using arduino with Global System for Mobile Communication (GSM). As far as vehicle security is concerned many options are available depending upon the technology being adopted. With the development and applications of lots of embedded techniques, car security system design and analyses are continually improving. Many new techniques, such as biometric recognition technique, image processing technique, communication technique and so on, have been included into car security systems. Statistics show that the number of cars is growing rapidly and so is the number of car theft attempts, locally and internationally. Although there are a batch of car security systems that had been produced lately, but the result is still unsatisfactory as the number of car theft cases still increases. The thieves are inventing cleverer and stronger stealing techniques that need extra powerful security systems.

This proposed project presents an anti-theft-control system for automobiles that tries to stop a vehicle from being stolen. In present days, vehicle theft is increasing rapidly and people have started using anti- theft-control systems in different automobiles systems. These anti-theft control systems are very expensive, but this project is designed cost-effectively by using arduino along with the GSM. GSM system is also installed in the vehicle for sending the information to the owner.The system contains

- GSM Module
- GPS Module
- Temperature Sensor
- DC Motor
- LCD Display

3.3 MODULE FUNCTIONALITIES

FUNCTIONALITY OF GSM MODULE:

A GSM modem or GSM module is a hardware device that uses GSM mobile telephone technology to provide a data link to a remote network. From the view of the mobile phone network, they are essentially identical to an ordinary mobile phone, including the need for a SIM to identify themselves to the network. GSM modems typically provide TTL-level serial interfaces to their host. They are usually used as part of an embedded system

GSM technology was developed as a digital system using the time division multiple access (TDMA) technique for communication purposes. A GSM digitizes and reduces the data, then sends it down through a channel with two different streams of client data, each in its own particular time slot. The digital system has the ability to carry 64 kbps to 120 Mbps of data rates.

There are various cell sizes in a GSM system such as macro, micro, pico, and umbrella cells. Each cell varies as per the implementation domain. There are five different cell sizes in a GSM network macro, micro, pico, and umbrella cells. The coverage area of each cell varies according to the implementation environment.

The time division multiple access (TDMA) technique relies on assigning different time slots to each user on the same frequency. It can easily adapt to data transmission and voice communication and can carry 64kbps to 120Mbps of data rate.

GSM Technology Architecture:

The main elements in the GSM architecture include the following.

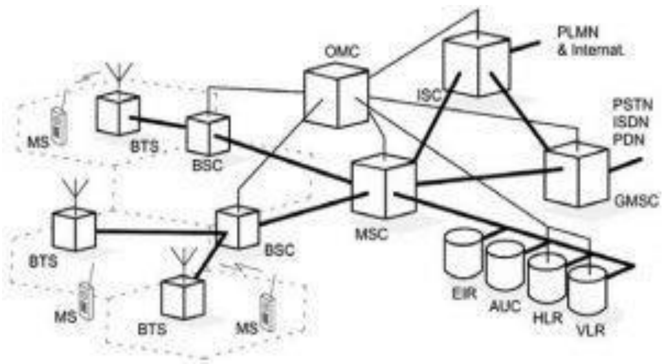


Fig 3.8: Architecture of GSM

- Network and Switching Subsystem (NSS)
- Base-Station Subsystem (BSS)
- The mobile station (MS)
- Operation and Support Subsystem (OSS)

Network Switching Subsystem (NSS)

In GSM system architecture, it includes different elements, which are frequently known as the core system/network. Here, it is basically a data network including a variety of units to provide the major control as well as interfacing of the entire mobile network system. The core network includes the major elements which are discussed below.

Base Station Subsystem (BSS)

It acts as an interface between the mobile station and the network subsystem. It consists of the Base Transceiver Station which contains the radio transceivers and handles the protocols for communication with mobiles. It also consists of the Base Station Controller which controls the Base Transceiver station and acts as an interface between the mobile station and mobile switching center.

Mobile Station

It is the mobile phone which consists of the transceiver, the display, and the processor and is controlled by a SIM card operating over the network.The MS (Mobile stations) or ME (mobile equipment) are most generally identified through cell otherwise mobile phones which are the part of a GSM mobile communications n/w that the operator observes & operates. At present, their dimension has reduced radically whereas the functionality level has very much increased. And one more benefit is that the time among charges has drastically enlarged. There are different elements to the mobile phone, though the two essential elements are the hardware & the SIM.

Operation and Support Subsystem (OSS)

The operation support subsystem (OSS) is a part of the complete GSM network architecture. This is connected to the NSS & the BSC components. This OSS is mainly used to control the GSM network & the BSS traffic load. It should be noted down that when the number of BS enhances through the subscriber population scaling then some of the preservation tasks are moved to the base transceiver stations so that the ownership cost of the system can be reduced.

Features of GSM Module:

- The features of the GSM module include the following.
- Improved spectrum efficiency
- International roaming
- Compatibility with integrated services digital network (ISDN)
- Support for new services.
- SIM phonebook management
- Fixed dialing number (FDN)
- Real-time clock with alarm management
- High-quality speech
- Uses encryption to make phone calls more secure
- Short message service (SMS)

Working of GSM Module

From the below circuit, a GSM modem duly interfaced to the MC through the level shifter IC Max232. The SIM card mounted GSM modem upon receiving digit command by SMS from any cell phone sends that data to the MC through serial communication. While the program is executed, the GSM modem receives the command ‘STOP’ to develop an output at the MC, the contact point of which are used to disable the ignition switch.

The command so sent by the user is based on an intimation received by him through the GSM modem ‘ALERT’ a programmed message only if the input is driven low. The complete operation is displayed over a 16×2 LCD display.

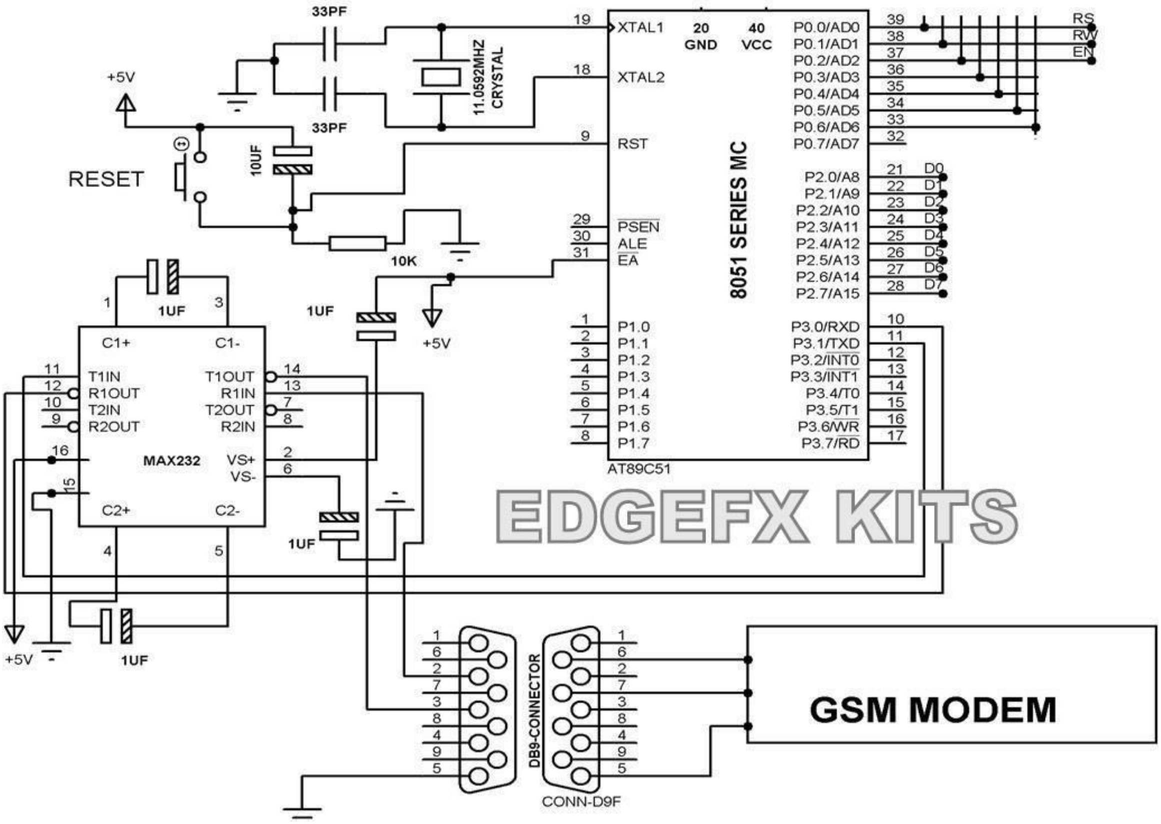


Fig 3.9: GSM Modem Circuit

FUNCTIONALITY OF GPS MODULE

GPS stands for Global Positioning System by which anyone can always obtain the position information anywhere in the world.GPS consists of the following three segments.

Space segment (GPS satellites):

A number of GPS satellites are deployed on six orbits around the earth at the altitude of approximately 20,000 km (four GPS satellites per one orbit), and move around the earth at 12-hour-intervals.

Control segment (Ground control stations):

Ground control stations play roles of monitoring, controlling and maintaining satellite orbit to make sure that the deviation of the satellites from the orbit as well as GPS timing are within the tolerance level.

User segment (GPS receivers): For receiving the signals

GPS positioning:

Firstly, the signal of time is sent from a GPS satellite at a given point. Subsequently, the time difference between GPS time and the point of time clock which GPS receiver receives the time signal will be calculated to generate the distance from the receiver to the satellite. The same process will be done with three other available satellites. It is possible to calculate the position of the GPS receiver from distance from the GPS receiver to three satellites. However, the position generated by means of this method is not accurate, for there is an error in calculated distance between satellites and a GPS receiver, which arises from a time error on the clock incorporated into a GPS receiver. For a satellite, an atomic clock is incorporated to generate on-the-spot time information, but the time generated by clocks incorporated into GPS receivers is not as precise as the time generated by atomic clocks on satellites. Here, the fourth satellite comes to play its role: the distance from the fourth satellite to the receiver can be used to compute the position in relations to the position data generated by distance between three satellites and the receiver, hence reducing the margin of error in position accuracy.

Factors that trigger GPS position errors:

Ionosphere :

The ionosphere is a portion of the upper atmosphere, between the thermosphere and the exosphere. When GPS signals pass through this layer, the propagation velocity of the GPS signal goes slower, hence causing propagation error.

Troposphere:

The troposphere is the lowest portion of Earth's atmosphere. Radio reflections caused by dry atmosphere and water vapor within provoke GPS position error.

Multipath propagation:

GPS signal is not immune to reflection when it hits on the ground, structures and many others. This phenomenon is called multipath propagation, one of the causes of GPS position errors.

State of reception of GPS depends upon the number of satellites tracked for positioning.If the number of the tracked satellites is great, GPS positioning becomes greater, but if there were a fewer satellites tracked for positioning, it would be difficult to generate GPS position.

3.4 RELATED DESIGNS

This project is developed to provide security for the vehicles and also it is designed with less complexity.The following figure shows the schematic diagram of the project.

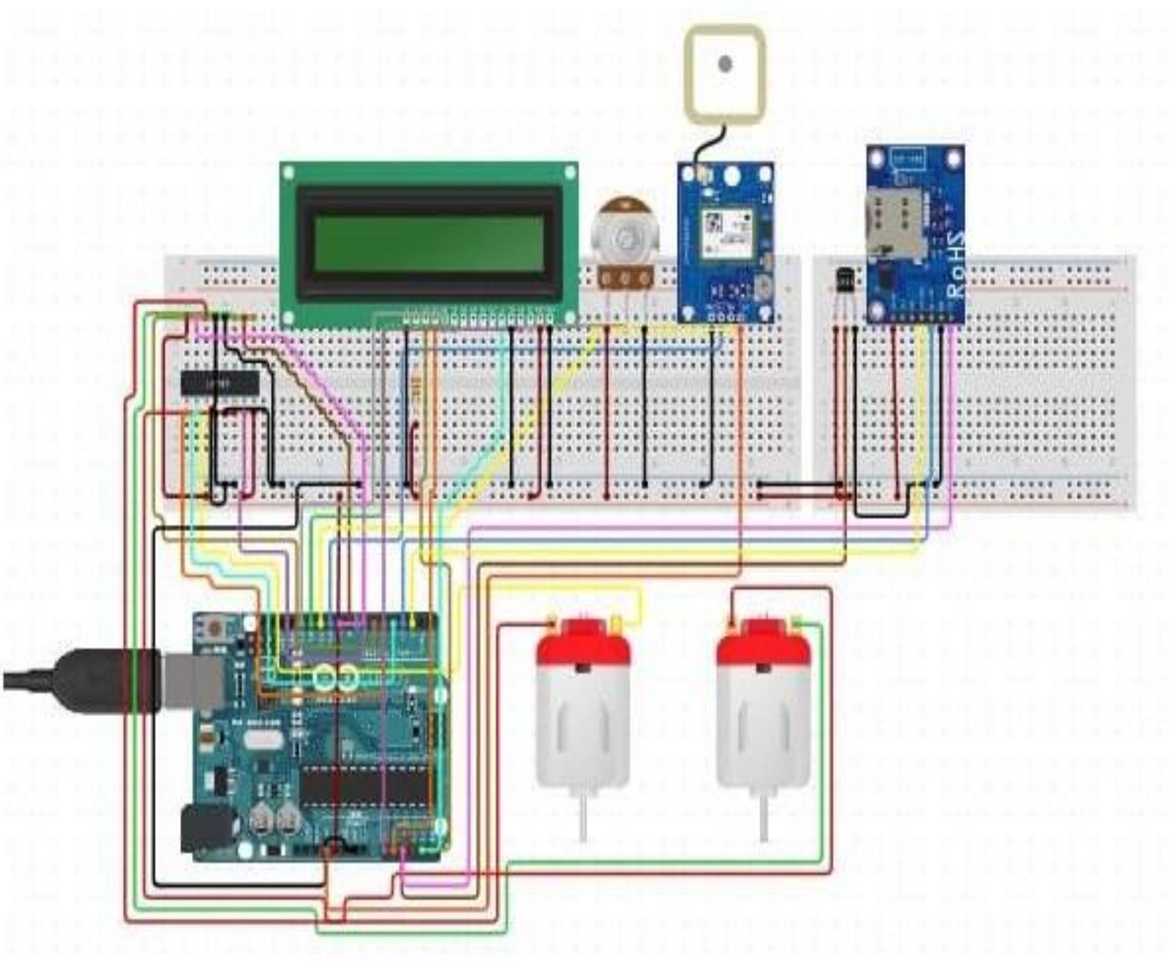


Fig 3.10: Schematic Diagram

CHAPTER 4

PROJECT IMPLEMENTATION

4.0 IMPLEMENTATION STAGES

The steps to implement this project are

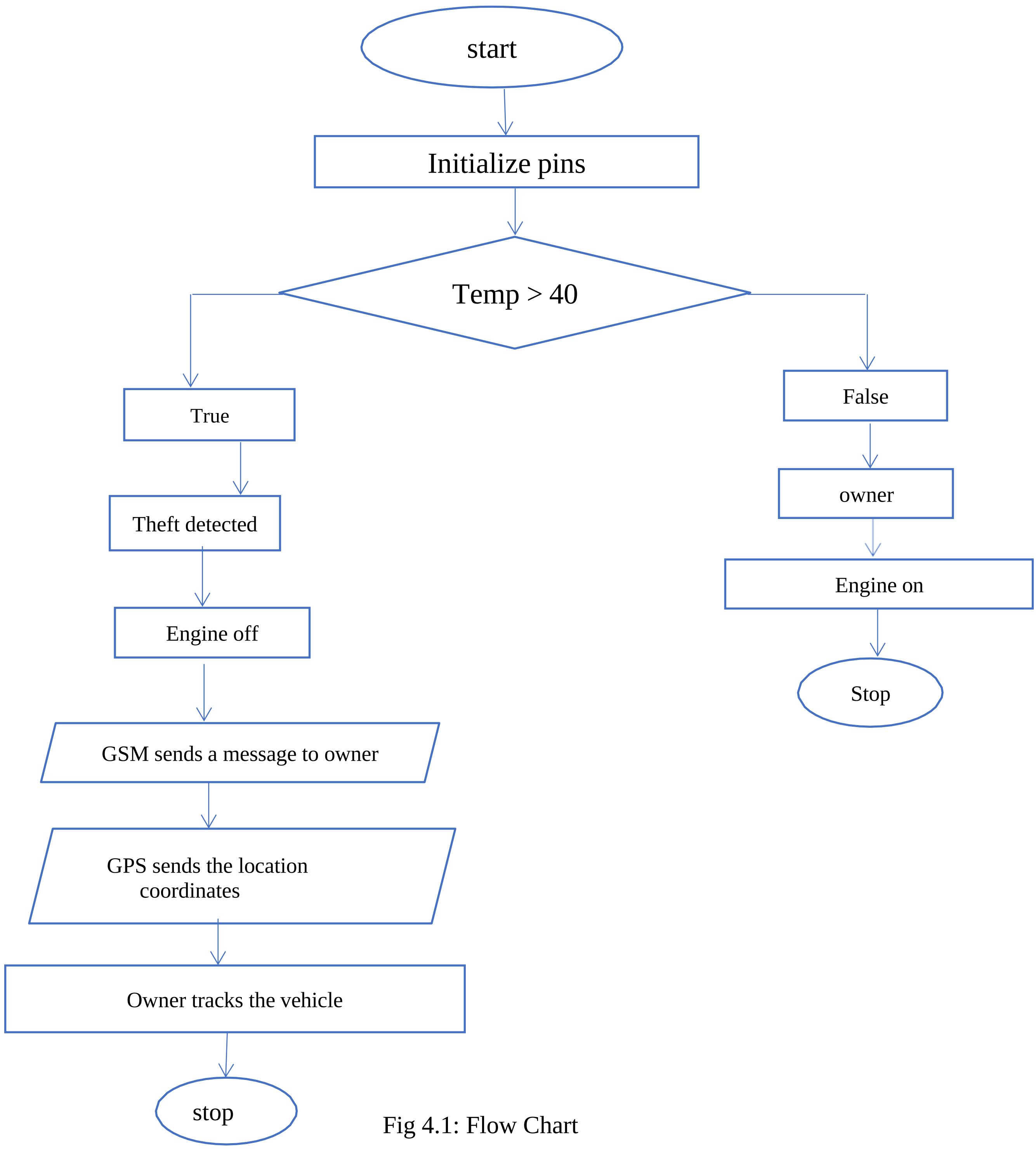


Fig 4.1: Flow Chart

The basic idea for this project is such that when the engine temperature increases beyond certain value then it will go into off position. The vehicle can be restarted by entering the secured password which is only known to the owner of the vehicle.The above mentioned stages in the flow chart are the different implementation stages to perform the project.

4.1 RESULTS

The proposed system is described that integrates new technologies offering a tracking of stolen vehicle by using GPS and GSM technology and provides vehicle security.The trainer kit of the proprosed system is designed as follows.

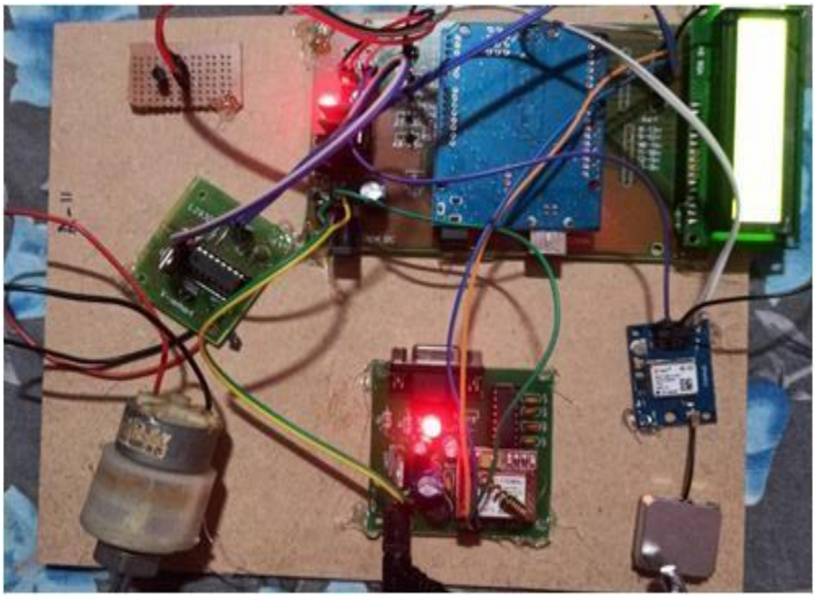


Fig 4.2: Final Outcome

The above figure depicts that the system is in tracking mode indicated by the red LED lighting up.

The system has an advantage of real time tracking of stolen vehicle and also it is easy to access.

The following are the figures on the LCD display that shows the engine position whether it is on or off and also it displays the location coordinates in the form of latitude and longitude by using the GPS module.

Fig 4.3: LCD Display

The messages that are sent to the owner mobile about the location coordinates are shown in the following figure.

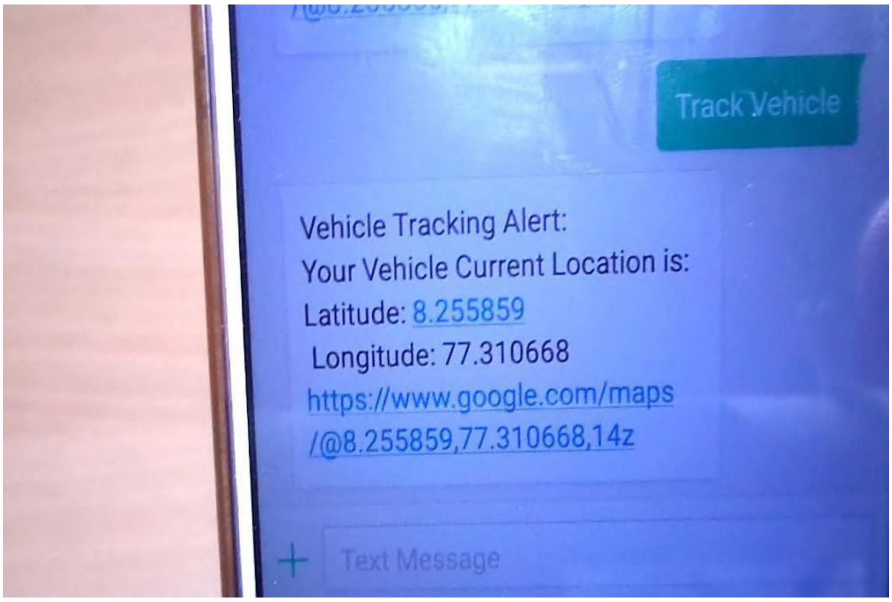


Fig 4.4: Messages sent to owner

CHAPTER 5

PROJECT TESTING

5.0 OVERVIEW OF TESTING METHODS

In this project, we are interfacing GSM Module to Arduino. There are different kinds of GSM modules available. We are using the most popular module based on Simcom SIM300 and Arduino Uno for this project. Interfacing a GSM module to Arduino is pretty simple. We should only make 3 connections between the gsm module and arduino.

A GSM Module is basically a GSM Modem (like SIM 300) connected to a PCB with different types of output taken from the board – say TTL Output (for Arduino, 8051 and other microcontrollers) and RS232 Output to interface directly with a PC (personal computer). The board will also have pins or provisions to attach mic and speaker, to take out +5V or other values of power and ground connections. These type of provisions vary with different modules.

For our project of connecting a gsm modem or module to arduino and hence send and receive sms using arduino – its always good to choose an arduino compatible GSM Module – that is a GSM module with TTL Output provisions.

We use SIM300 GSM Module – This means the module supports communication in 300MHz band.

GSM modules are manufactured by different companies. They all have different input power supply specs. We need to double check our GSM modules power requirements.

Note: GSM Modules are manufactured by connecting a particular GSM modem to a PCB and then giving provisions for RS232 outputs, TTL outputs, Mic and Speaker interfacing provisions etc. The most popular modem under use is SIM 900 gsm modem from manufacturer SIMCom. They also manufacture GSM Modems in bands 850, 300 and other frequency bands.

We can feed the data from gsm module directly to Arduino only if the module is enabled with TTL output pins. Otherwise we have to convert the RS232 data to TTL using MAX232 IC and feed it to Arduino. Most of the gsm modules are equipped with TTL output pins.

Booting the GSM Module:

1. Insert the SIM card to GSM module and lock it.

2. Connect the adapter to GSM module and turn it ON!

3. Now wait for some time (say 1 minute) and see the blinking rate of ‘status LED or ‘network LED’ (GSM module will take some time to establish connection with mobile network)

4. Once the connection is established successfully, the status/network LED will blink continuously every 3 seconds. We may try making a call to the mobile number of the sim card inside GSM module. If we hear a ring back, the gsm module has successfully established network connection.

Now let’s see how to connect a gsm module to Arduino

Connecting GSM Module to Arduino:

There are two ways of connecting GSM module to arduino. In any case, the communication between Arduino and GSM module is serial. So we are supposed to use serial pins of Arduino (Rx and Tx). So if we are going with this method, we may connect the Tx pin of GSM module to Rx pin of Arduino and Rx pin of GSM module to Tx pin of Arduino. GSM Tx –> Arduino Rx and GSM Rx –> Arduino Tx. Now connect the ground pin of arduino to ground pin of gsm module. We made 3 connections and the wiring is over. Now we can load different programs to communicate with gsm module and make it work.

In the coding part,the program has two objectives as described below:

1 Send SMS using Arduino and GSM Module – to a specified mobile number inside the program

2. Receive SMS using Arduino and GSM Module – to the SIM card loaded in the GSM Module.

TESTING METHOD OF GPS:

As GPS technology becomes more common, GPS receiver manufacturers, OEM integrators, and contract manufacturers struggle to determine the appropriate standard tests to verify GPS receiver performance. Verification procedures require a controlled environment that facilitates precise repeatability. In most cases, using actual GPS satellite signals received through an antenna does not provide such an environment. This excerpt from Agilent's GPS Receiver Testing white paper introduces a real-time GPS signal simulation technique capable of generating the required GPS signals for a repeatable and flexible test environment.

GPS Test Requirements:

The GPS user experience for commercial applications is affected by several factors. GPS devices which provide an enhanced user experience will sell better, so manufacturers are looking for factors to differentiate their receivers. Typical factors which determine the outcome of the user experience include the following factors:

When a GPS device is turned on, how long is it until the position of the receiver is determined?

When a weak or poor signal area is encountered, can the receiver still determine its position?

If the signal is interrupted and then restored, how long does it take for the receiver to recover and resume calculating its position?

Accuracy of the calculated location.

There are of course other factors such as cost, user interface, turn-by- turn navigation, spoken directions, and so forth that are important to users, but these are not so dependent on the GPS receiver performance.For commercial or military applications, there may be many other kinds of GPS conditions that are important, such as:

How accurately can a position or time be determined?

How repeatable is the solution?

How sensitive is the receiver to interference or jamming?

How rapidly can the receiver report its position (if the receiver is moving rapidly – such as in an airplane)?

Here are five steps to testing GPS receivers.These steps need to be completed in the sequential order as listed. Not following the numbered sequence will often force us to repeat tests in the event changes are made to the system.

Step 1: Antenna Testing

The first step in receiver testing is to ensure our antennas work properly. This consists of ensuring the antennas are tuned on-frequency

Step 2: Conducted Receiver Sensitivity Testing

The next step in system testing is conducting receive sensitivity testing, and it is the first time we actually test the receiver itself. There are a number of problems that can impact receiver performance such as conducted power supply noise, power supply errors (such as voltage or current problems), firmware setup, and module management errors. Conducted receive sensitivity testing is the responsibility of the radio integrator.

Step 3: Residual Error Test

With residual error, a conducted RF test is done with signal levels turned up so a signal is strong. It should not be so strong that it might impact the linearity of the receiver. It should be sufficiently strong so there is no ambiguity that the signal the receiver is seeing is well within the sensitivity range. For most modern GPS modules a number of approximately – 100dBm seen at the RF input of the module would be a good starting point. Set the GPS signal generator up, let the GPS get a fix and then start examining the test cases.

Step 4: Radiated receiver sensitivity testing

With radiated receiver sensitivity testing the entire receive system is being tested, as it would work in the field, with the exception that it is being done in a controlled and repeatable RF environment. Testing of this kind is done in an anechoic RF chamber with a GPS signal generator. It cannot be done using the real satellite constellation because we cannot control the signal levels of the satellites. In order to know the path loss of the test setup, typically a network analyzer is used to measure this.

Step 5: Self-quieting

All wireless devices emit radio energy. The amplitude and frequency of these emissions are directly related to the circuitry chosen and how it is implemented. Because a GPS receiver is so sensitive, it takes deliberate attention to noise suppression to have a chance of keeping the rest of our electronics quiet enough, from a radio standpoint, so that they do not interfere with the GPS receiver. The further the antenna is away from our electronics the less trouble we will have.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION

The anti-theft control system is a breakthrough in today's world where automobile theft has increased rapidly. Today, majority of the communication and its applications are used by using GSM. The combination of GSM in anti-theft control system ensures efficiency and security as is required by it. The GSM also ensures that the effectiveness of the system is not just restricted to local use but can be covered over a wide range .Our proposed project can prevent the vehicle from hotwiring and fraudulent theft.

Our Vehicle theft prevention project acts as communication medium between the owner and the embedded system. This project with low cost is very useful to general public. It works for low range vehicles also.This automatic car theft detection system will alert the user about unauthorized access and then help to locate the vehicle correctly. The system is versatile, extendable and totally adjustable to user needs.

FUTURE ENHANCEMENT

We can develop this project in future by adding fingerprint sensor and face detection sensor. Presently, only the message feature is available. For ease of operation, developments can be made to include the Call Feature. Microphone can also be interfaced to the GMS module so that during theft activity, voice call could be established with the owner.

In future, this security system can be improvised to function as an integrated-data-security system for car communication systems. It would ensure that all the data exchanged within the vehicle and outside the vehicle is protected.

PUBLICATION

This project is published in the following conference

Title of the paper: Real Time Vehicle Theft Control System Using Engine Temperature Measurement

Conference : International Conference on Smart Modernistic in Electronics and Communication (ICSMEC – 21)

Status : Accepted

REFERENCES

- [1] M. A. Elmala et al., “Electronic interface system with 7.5/12.5V actuation for MEMS accelerometer,” 2017 Intl Conf on Advanced Control Circuits Systems (ACCS) Systems & 2017 Intl Conf on New Paradigms in Electronics & Information Technology (PEIT)
- [2] S. Sahabiswas et al., “Drunken driving detection and prevention models using Internet of Things,” 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)
- [3] K. Seelam and C. J. Lakshmi, “An Arduino based embedded system in passenger car for road safety,” 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)
- [4] M. Malathi, R. Sujitha and M. R. Revathy, “Alcohol detection and seat belt control system using Arduino,” 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, 2017, pp. 1-3. doi: 10.1109/ICIIECS.2017.8275841
- [5] Sunehra and K. Jhansi, "Implementation of microcontroller-based driver assistance and vehicle safety monitoring system," 2015 International Conference on Information Processing (ICIP), Pune, 2015, pp. 423- 428. doi: 10.1109/INFOP.2015.7489420

- [6] V. P. Kumar, K. Rajesh, M. Ganesh, I. R. P. Kumar and S. Dubey, "Overspeeding and Rash Driving Vehicle Detection System," 2014 Texas Instruments India Educators' Conference (TIIEC), Bangalore, 2014, pp. 25- 28. doi: 10.1109/TIIEC.2014.013
- [7] T. Shyam Ramanath, A. Sudharsan and U. Felix Udhayaraj, "Drunken driving and rash driving prevention system," 2010 International Conference on Mechanical and Electrical Technology, Singapore, 2010, pp. 599-603.

BIBLIOGRAPHY

- https://create.arduino.cc/projecthub/muchika/vehicle-tracking-system-based-on-gps-and-gsm-57b814
- https://link.springer.com/article/10.3758/s13428-011-0163-z#Fig1
- file:///C:/Users/LENOVO/Downloads/4935-Manuscript-16733-1-10-20170915%20(1).pdf
- https://www.arrow.com/en/research-and-events/articles/arduino-and-gps-introduction
- https://iarjset.com/upload/2017/si/NCETETE-2017/IARJSET-NCETETE%2020.pdf

APPENDICES

- https://www.google.com/search?q=functionality+of+gps+module&oq=&aqs=chrome.0.69i59i450l8.182931851j0j15&sourceid=chrome&ie=UTF-8
- https://www.google.com/search?q=module+functionality+of+gsm+module&oq=&aqs=chrome.4.69i59i450l8.210420817j0j15&sourceid=chrome&ie=UTF-8
- https://www.google.com/search?q=module+functionality+of+arduino&oq=&aqs=chrome.5.69i59i450l8.210505838j0j15&sourceid=chrome&ie=UTF-8

A

PROJECT REPORT

On

A SHARED REPRESENTATION FOR OBJECT TRACKING AND CLASSIFICATION USING SIAMESE NETWORK

Submitted by

1) P.SUSHMA(17K81A0437)

2) U.NIKITHA(17K81A0455)

3) V.JAYA SURYA (18K85A0401)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS & COMMUNICATION ENGINEERING

Under The Guidance of

Santhosh Singireddy

Assistant Professor

DEPARTMENT OF ELECTRONICS&COMMUNICATION EDUCATION

ST.MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

St.MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the project entitled **A Shared Representation For object Tracking And Classification Using Siamese Network** is being submitted by **P.Sushma(17K81A0437), U.Nikitha(17K81A0455), V.Jaya Surya(18K85A0401).** In partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS &COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verifiedand found satisfactory.

GUIDE	Head of the Department
Mr Santosh Singireddy	DR.B.Hari Krishna
Department of ECE	Department of ECE

Internal Examiner External Examiner

Place:

Date:

DECLARATION

We, the student of **Bachelor of Technology** in Department of Electronics And Communication, session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled **A Shared Representation For Object Tracking And Classification Using Siamese Network** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics.This result embodied in this project report has not been submitted in any university for award of any degree.

P.Sushma(17K81A0437)
U.Nikitha(17K81A0455)
V.Jaya Surya(18K85A0401)

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would b incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Aboveall, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **DR.B.HariKrishna**, **Head &Professor**, the Dept. Electronics And Communication for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our department project coordinator DR. A. Anand** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide, **Mr. S. Santosh Singireddy, Assistant Professor,** Department of ECE and our class project coordinator **Mrs. A. Sowmya, Assistant Professor,** Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri**, **Professor in IT** and **Dr. R. Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1.P.Sushma(17K81A0437)

2.U.Nikitha(17K81A0455)

3.V.Jaya Surya(18K85A0401)

ABSTRACT

Recently, Siamese neural networks have been employed to build several high performance object trackers capable of operating in real time. To further improve the tracking performance, one can train one network on the tracking task and another network on the task of object classification. One can then use the feature representations of both networks to obtain a tracker which performs better than each network on its own. This approach, however, has the downside that two networks have to be evaluated instead of one, resulting in runtime degradation. We demonstrate that it is feasible to train one Siamese network on the tracking and the classifications tasks simultaneously. Specifically, we achieve a tracking performance similar to the performance of two networks trained on tracking and classification separately. Since our approach does not depend on two separate networks though, it allows one to improve the performance of a Siamese network tracker without any runtime penalty.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Introduction To The Project:

Traffic data such as vehicle counts, speeds, and classification are important in traffic engineering applications, transportation planning, and Intelligent Transportation Systems (ITS). Collecting traffic data manually by direct observations of human observers has a number of drawbacks including high cost, extreme weather and difficulties imposed by staffing limitations While in-road technologies such as inductive loop detectors offer good accuracy for counts and presence detection, their installation and maintenance causes traffic disruption. Sensors that are placed on the pavements (magnetometers, road tubes) can be damaged by snow removal equipment or street sweepers. As mentioned in at times it is difficult to obtain accurate counts using intrusive technologies due to roadway geometry (e.g., geometry where there are significant lane changes or where vehicles do not follow a set path in making turns). Some of the non-intrusive roadside sensors might be prohibitive due to high cost (e.g., laser) or low precision (e.g., microwave).

Infrared sensors have an advantage of day/night operation and perform better than visible wavelength sensors in fog. However, in addition to the problem of unstable detection zones, for reliable operation at least one sen1 sor is required in each traffic lane (a notableexception is the TIRTL sensor Ultrasonic sensors exhibit difficulty in detecting snow- covered vehicles and are sensitive to changes in ambient temperature and humidity. In addition, the problem of detecting motorcycles remains elusive for the sensors described above.

The output of these sensors is a poor description of the traffic events. This is a serious limitation in case of a critical situation, where a human operator is required to make a decision based on the sensor data. In such cases, video sensors provide the information in the form of live video of the scene. In addition, a single video sensor placed at an appropriate position provides wide area coverage making it possible to detect incidents in multiple lanes simultaneously. The same is the case in calculating queue lengths. Another advantage of video is that it provides sufficient information for vehicle tracking to be feasible, which is useful for detecting events such as sudden lane changes, vehicles moving in the wrong direction, stalled vehicles etc.

There is a broad range of real world applications where it is necessary to track objects over several video frames. For example, in traffic surveillance scenarios it is crucial to track the movements of vehicles or pedestrians in order to estimate velocities or to detect traffic violations. When devising a tracking algorithm, one has to take into account the possibilities of out-of-plane rotations, illumination changes, occlusions and deformations of objects. While being flexible enough to handle these difficulties, the method should also have sufficient discriminant power in the sense that it should be capable of distinguishing one object from another in scenarios of many similar objects appearing close to each other. Furthermore, runtime aspects usually have to be taken into account. Since tracking algorithms are often used to process video streams, real-time performance is often a hard constraint**.**

With the rise of deep learning in various computer vision applications, it comes as no surprise that deep learning methods have also been employed in the domain of object tracking. One kind of deep learning tracking architecture are Siamese networks. These networks attempt to find a feature representation which is robust with respect to the before mentioned transformations. One can then use a similarity measure in feature space to locate an object in subsequent frames. Often, the performance of an artifical neural network can be improved by learning a feature representation which works well on another task simultaneously . For example, in the domain of object tracking proposed to train two separate networks: a classification network and a tracking network. The features of the classification network are used to improve the tracking performance. They statethat in their experiments jointly training one feature extractor on both the classification and the tracking task has a considerably worse tracking performance than separately training two networks. The separate training of two feature extractors has the downside that two networks have to be evaluated during tracking, leading to a worse runtime than when only one network has to be evaluated.

Our contributions can be summarized as follows: • We propose a Siamese tracking network architecture which has a single feature extractor. Unlike the feature extractor in other Siamese tracking networks, we train it such that the learned features are useful for the tracking as well as for the classification tasks. • We show experimentally that jointly training on both tasks can result in a tracker which performs similar to two separate networks, while only one feature extractor needs to be evaluated during inference.

1.2 INTRODUCTION TO IMAGE PROCESSING

IMAGE:

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Fig 1.2.1 General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

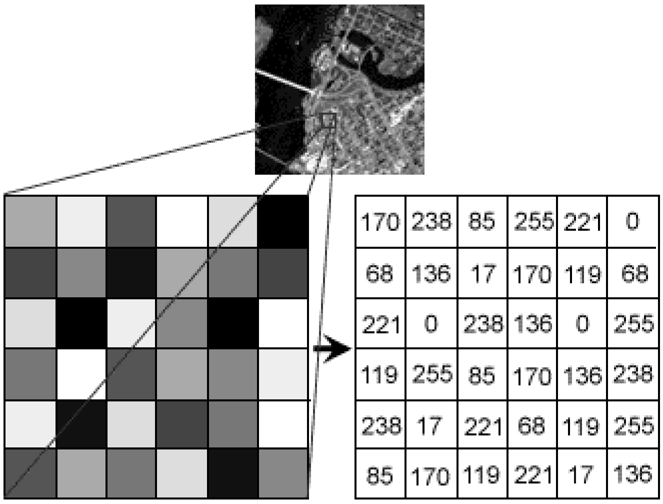


Fig 1.2.2 Image pixel

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

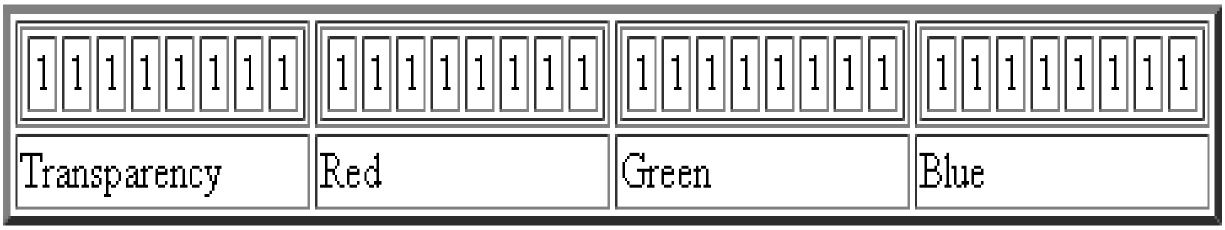


Fig 1.2.3 Transparency image

IMAGE FILE SIZES:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

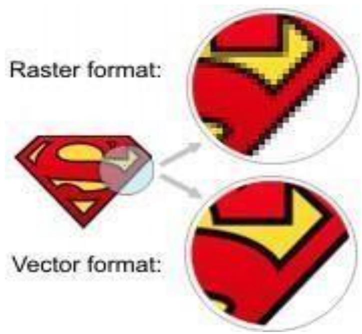


Fig 1.2.4 Resolution image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own nativeformat.

RASTER FORMATS:

These formats store images as bitmaps (also known as pixmaps)

JPEG/JFIF:

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored ina lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

EXIF:

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

TIFF:

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

PNG:

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

GIF:

GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

BMP:

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric descriptionwhich can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

CGM:

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical informationindependent from any particular application, system, platform, or device.

SVG:

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

IMAGE PROCESSING:

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

1.3 FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:

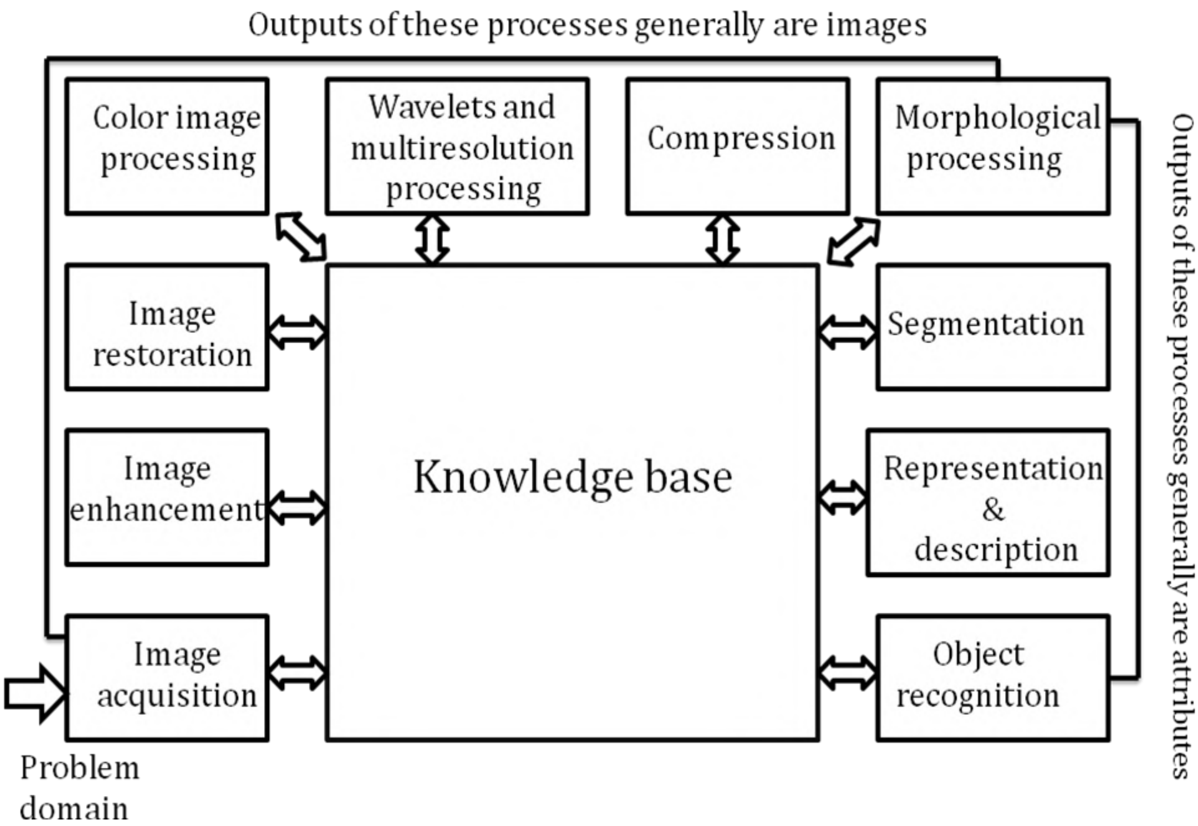


Fig 1.3.1 Image fundamental

Image Acquisition:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Fig 1.3.2 Digital camera image

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Fig 1.3.3 digital camera cell

Image Enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because "it looks better." It is important to keep in mind that enhancement is a very subjectivearea of image processing.

Fig 1.3.4 Image enhancement

Image restoration:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Fig 1.3.5 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, where as removal of image blur by applying a deblurring function is considered a restoration technique.

Color image processing:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Fig 1.3.6 Color & Gray scale image

Wavelets and multiresolution processing:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform based image processing since the late1950's, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

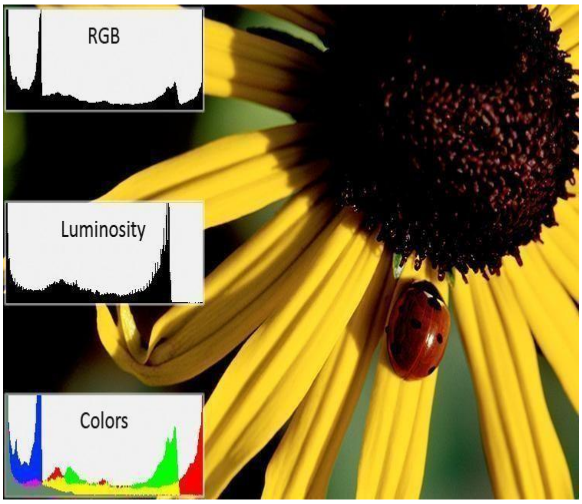


Fig 1.3.7 rgb histogram image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called Multiresolution theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

Compression:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

Morphological processing:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Fig 1.3.8 blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

Segmentation:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

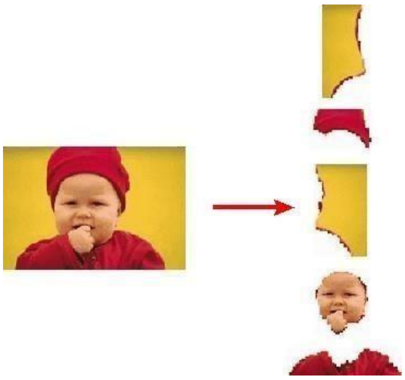


Fig 1.3.9 Image segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

Representation and description:

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

Object recognition:

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

Knowledgebase:

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an inter related to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

1.4 COMPONENTS OF AN IMAGE PROCESSING SYSTEM:

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached toequally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

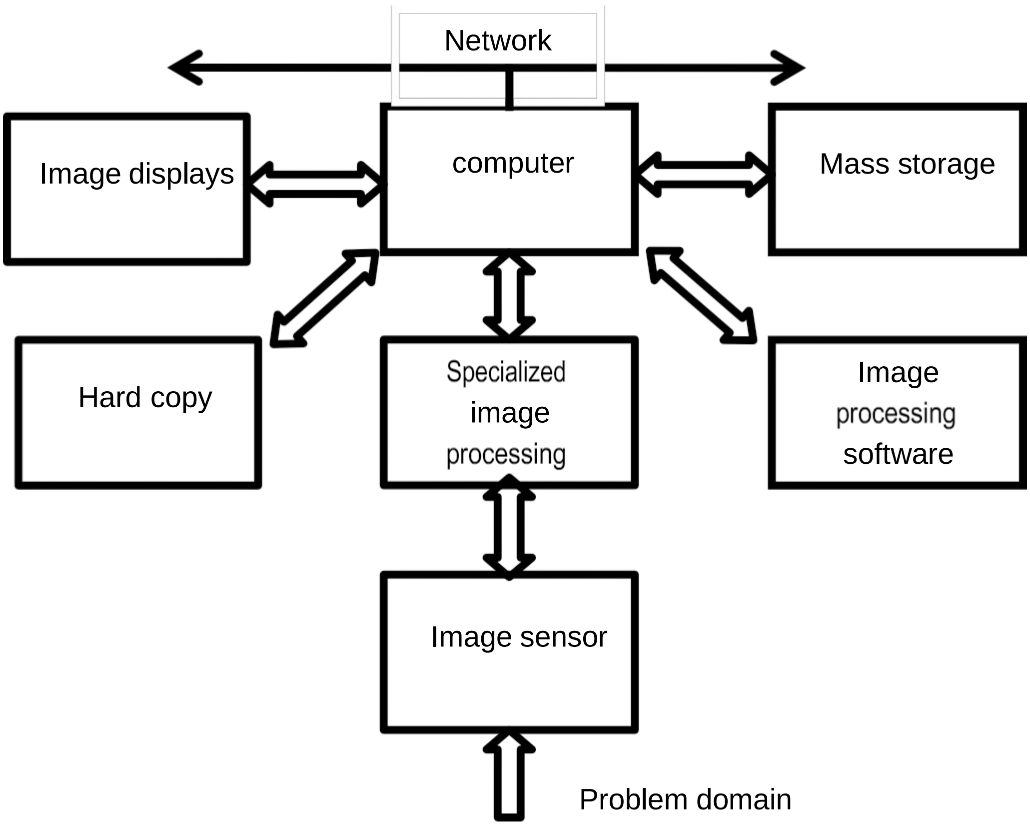


Fig 1.4.1 Component of image processing

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 1.24 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

Image sensors:

With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

Specialized image processing hardware:

Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

Computer:

The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

Image processing software:

Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commandsfrom at least one computer language.

Mass storage:

Mass storage capability is a must in image processing applications. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage forimage processing applications fall into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), Gbytes (meaning giga, or one billion, bytes), and Tbytes (meaning tera, or one trillion, bytes)

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig. 1.24. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in "jukeboxes" are the usual media for archival applications.

Image displays:

Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are there requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

Hardcopy:

Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

Network:

Networking is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of optical fiber and other broadband technologies.

Color and texture are two low-level features widely used for image classification, indexing and retrieval. Color is usually represented as a histogram, which is a first order statistical measure that captures global distribution of color in an image One of the main drawbacks of the histogram- based approaches is that the spatial distribution and local variations in color are ignored. Local spatial variation of pixel intensity is commonly usedto capture texture information in an image. Grayscale Co-occurrence Matrix (GCM) is a well-known method for texture extraction in the spatial domain. A GCM stores the number of pixel neighborhoods in an image that have a particular grayscale combination. Let I be an image and let p and Np respectively denote any arbitrary pixel and its neighbor in a given direction. If GL denotes the total number of quantized gray levels and gl denotes the individual gray levels, where, gl {0, . . .,GL _ 1}, then each component of GCM can be written as follows:

$$\mathrm{gcm}(i, j) = \Pr((\mathrm{gl}_{\mathrm{p}}, \mathrm{gl}_{N_{\mathrm{p}}}) = (i, j))$$

gcm(i, j) is the number of times the gray level of a pixel p denoted by glp equals i, and the gray level of its neighbor Np denoted by glNp equals j, as a fraction of the total number of pixels in the image. Thus, it estimates the probability that the gray level of an arbitrary pixel in an image is i, and that of its neighbor is j. One GCM matrix is generated for each possible neighborhood direction, namely, 0, 45, 90 and 135.Average and range of 14 features like Angular Second Moment, Contrast, Correlation, etc., are generated by combining all the four matrices to get a total of 28 features. In the GCM approach for texture extraction, color information is completely lost since only pixel gray levels are considered.

To incorporate spatial information along with the color of image pixels, a feature called color correlogram has recently been proposed. It is a three dimensional matrix that represents the probability of finding pixels of any two given colors at a distance 'd' apart Auto correlogram is a variation of correlogram, which represents the probability of finding two pixels with the same color at a distance 'd' apart. This atproach can effectively represent color distribution in an image. However, correlogram features do notcapture intensity variation Many image databases often contain both color as well as gray scale images. The color correlogram method does not constitute a good descriptor in such databases.

Another method called Color Co-occurrence Matrix (CCM) has been proposed to capture color variation in an image. CCM is represented as a three-dimensional matrix, where color pair of the pixels p and Np are captured in the first two dimensions of the matrixand the spatial distance 'd' between these two pixels is captured in the third dimension. This approach is a generalization of the color correlogram and reduces to the pure color correlogram for d = 1. CCM is generated using only the Hue plane of the HSV (Hue, Saturation and Intensity Value) color space. The Hue axis is quantized into HL number of levels. If individual hue values are denoted by hl, where $\mathrm{hl} \in \{0, \ldots, \mathrm{HL} - 1\}$, then each component of CCM can be written as follows:

$$\mathrm{ccm}(i,j) = \mathbf{Pr}((\mathrm{hl}_{\mathrm{p}}, \mathrm{hl}_{N_{\mathrm{p}}}) = (i,j))$$

Four matrices representing neighbors at angles 0, 90, 180 and 270 are considered. This approach was further extended by separating the diagonal and the non-diagonal components of CCM to generate a Modified Color Co-occurrence Matrix (MCCM).MCCM, thus, may be written as follows: MCCM = (CCMD;CCMND)

Here, CCMD and CCMND correspond to the diagonal and off-diagonal components of CCM. The main drawback of this approach is that, like correlogram, it also captures only color information and intensity information is completely ignored.

An alternative approach is to capture intensity variation as a texture feature from an image and combine it with color features like histograms using suitable weights . One of the challenges of this approach is to determine suitable weights since these are highly application-dependent. In certain applications like Content-based Image Retrieval (CBIR), weights are often estimated from relevance feedback given by users.

While relevance feedback is sometimes effective, it makes the process of image retrieval user-dependent and iterative. There is also no guarantee on the convergence of theweight-learning algorithms. In order to overcome these problems, researchers have triedto combine color and texture features together during extraction.

proposed two approaches for capturing color and intensity variations from an image using the LUV color space. In the Single-channel Co-occurrence Matrix (SCM), variations for each color channel, namely, L, U and V are considered independently. In the Multi channel Co-occurrence Matrix (MCM), variations are captured taking two channels at a time – UV, LU and LV. Since the LUV color space separates out chrominance (L and U) from luminance (V), SCM in effect, generates one GCM and two CCMs from each image independently. As a result, correlation between the color channels is lost

However, in MCM, the count of pair wise occurrences of the values of different channels of the color space is captured. Thus, each component of MCM can be written as follows:

mcmUV(i; j) = Pr((up; vNp) = (i; j))

mcmLU(i; j) = Pr((lp; uNp) = (i; j))

mcmLV(i; j) = Pr((lp; vNp) = (i; j))

Here, mcmUV(i, j) is the number of times the U chromaticity value of a pixel p denoted by up equals i, and the V chromaticity value of its neighbor Np denoted by vNp equals j, as a fraction of the total number of pixels in the image. Similarly, mcmLU(i, j) and mcmLV(i, j) are defined. One MCM matrix is generated for each of the four neighborhood directions, namely, 0, 45, 90 and 135.

Deng and Manjunath (2001) proposed a two-stage method called JSEG, which combines color and texture after image segmentation. In the first stage, colors are quantized to the required levels for differentiating between various regions of an image. Pixel values of the regions are then replaced by their quantized color levels to form a color map. Spatial variation of color levels between different regions in the map is viewed as a type of texture composition of the image.

Yu et al. (2002) suggested the use of color texture moments to represent both color and texture of an image. This approach is based on the calculation of

Local Fourier Transformation (LFT) coefficients. Eight templates equivalent to LFT are operated over an image to generate a characteristic map of the image. Each template is a

$3 \cdot 3$ filter that considers eight neighbors of the current pixel for LFT calculat ion. First and second order moments of the characteristic map are then used to generate a set of features.

In this paper, we propose an integrated approach for capturing spatial variation of both color and intensity levels in the neighborhood of each pixel using the HSV color space. In contrast to the other methods, for each pixel and its neighbor, the amount of color and intensity variation between them is estimated using a weight function. Suitable constraints are satisfied while choosing the weight function for effectively relating visual perception of color and the HSV color space properties. The color and intensity variations are represented in a single composite feature known as Integrated Color and Intensity Co-occurrence Matrix (ICICM). While the existing schemes generally treat color and intensity separately, the proposed method provides a composite view to both color and intensity variations in the same feature. The main advantage of using ICICM is that it avoids the use of weights to combine individual color and texture features. We use ICICM feature in an image retrieval application from large image databases.

Early result on this work was reported in (Vadivel et al., 2004a). In the next section, we describe the proposed feature extraction technique after introducing some of the properties of the HSV color space. Choice of quantization levels for colorand intensity axes, selection of parameter values and a brief overview of the image retrieval application

Integrated color and intensity co-occurrence matrix:

We propose to capture color and intensity variation around each pixel in a two-dimensional matrix called Integrated Color and Intensity Co-occurrence Matrix (ICICM).

This is a generalization of the Grayscale Co-occurrence Matrix and the Color Co-occurrence Matrix techniques. For each pair of neighboring pixels, we consider

their contribution to both color perception as well as gray level perception to the human eye. Some of the useful properties of the HSV color space and their relationship to human color perception are utilized for extracting this feature. In the next sub-section, we briefly explain relevant properties of the HSV color space. In the subsequent subsection, we describe how the properties can be effectively used for generating ICICM.

HSV color space:

HSV Color space: Basically there are three properties or three dimensions of color that being hue, saturation and value HSV means Hue, Saturation and Value. It is important to look at because it describes the color based on three properties. It can create the full spectrum of colors by editing the HSV values. The first dimension is the Hue. Hue is the other name for the color or the complicated variation in the color. The quality of color as determined by its dominant wavelength. This Hue is broadly classified into three categories. They are primary Hue, Secondary Hue and Teritiary Hue. The first and the foremost is the primary Hue it consists of three colors they are red, yellow and blue. The secondary Hue is formed by the combination of the equal amount of colors of the primaryHue and the colors of the secondary Hue which was formed by the primary Hue are Orange, Green and violet. The remaining one is the teritiary Hue is formed by the combination of the primary Hue and the secondary Hue. The limitless number of colors are produced by mixing the colors of the primary Hue in different amounts. Saturation is the degree or the purity of color. Then the second dimension is the saturation. Saturation just gives the intensity to the colors. The saturation and intensity drops just by mixing the colors or by adding black to the color. By adding the white to the color in spite of more intense the color becomes lighter. Then finally the third dimension is the Value. The value is the brightness of the color. When the value is zero the color space is totally black with the increase in the color there is also increase in the brightness and shows the various colors. The value describes the contrast of the color. That means it describes the lightness and darkness of the color. As similar to the saturation this value consists of the tints and shades. Tints are the colors with the added white and shades are the colors with the added black.

Properties of the HSV color space:

Sensing of light from an image in the layers of human retina is a complex process with rod cells contributing to scotopic or dim-light vision and cone cells to photopic or bright- light vision (Gonzalez and Woods, 2002). At low levels of illumination, only the rod cells are excited so that only gray shades are perceived. As the illumination level increases, more and more cone cells are excited, resulting in increased color perception. Various color spaces have been introduced to represent and specify colors in a way suitable for storage, processing or transmission of color information in images. Out of these, HSV is one of the models that separate out the luminance component (Intensity) of a pixel color from its chrominance components (Hue and Saturation). Hue represents pure color, which is perceived when incident light is of sufficient illumination and contains a single wavelength. Saturation gives a measure of the degree by which a pure color is diluted by white light. For light with low illumination, corresponding intensity value in the HSV color space is also low.

The HSV color space can be represented as a Hexa cone, with the central vertical axis denoting the luminance component, I (often denoted by V for Intensity Value). Hue, is a chrominance component defined as an angle in the range [0,2p] relative to the red axis with red at angle 0, green at 2p/3, blue at 4p/3 and red again at 2p. Saturation, S, is the other chrominance component, measured as a radial distance from the central axis of the hexacone with value between 0 at the center to 1 at the outer surface. For zero saturation, as the intensity is increased, we move from black to white through various shades of gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the color represented by its hue. When saturation is near 0, all the pixels in an image look alike even though their hue values are different.

As we increase saturation towards 1, the colors get separated out and are visually perceived as the true colors represented by their hues. Low saturation implies presence of a large number of spectral components in the incident light, causing loss of color information even though the illumination level is sufficiently high. Thus, for low values of saturation or intensity, we can approximate a pixel color by a gray level whilefor higher saturation and intensity, the pixel color can be approximated by its hue. Forlow intensities, even for a high saturation, a pixel color is close to its gray value. Similarly, for low saturation even for a high value of intensity, a pixel is perceived as gray. We use these properties to estimate the degree by which a pixel contributes to color perception and gray level perception.

One possible way of capturing color perception of a pixel is to choose suitable thresholds on the intensity and saturation. If the saturation and the intensity are above their respective thresholds, we may consider the pixel to have color dominance; else, it has gray level dominance. However, such a hard thresholding does not properly capture color perception near the threshold values. This is due to the fact that there is no fixed level of illumination above which the cone cells get excited. Instead, there is a gradual transition from scotopic to photopic vision. Similarly, there is no fixed threshold for the saturation of cone cells that leads to loss of chromatic information at higher levels of illumination caused by color dilution. We, therefore, use suitable weights that vary smoothly with saturation and intensity to represent both color and gray scale perception for eachpixel.

NON INTERVAL QUANTIZATION:

Due to the large range for each component by directly calculating the characteristics for the retrieval then the computation will be very difficult to ensure rapid retrieval. It is essential to quantify HSV space component to reduce computation and improve efficiency. At the same time, because the human eye to distinguish colors is limited, do not need to calculate all segments. Unequal interval quantization according the human color perception has been applied on H , S ,V components.

Based on the color model of substantial analysis, we divide color into eight parts.Saturation and intensity is divided into three parts separately in accordance with the human eyes to distinguish. In accordance with the different colors and subjective color perception quantification, quantified hue(H), saturation(S) and value(V)

In accordance with the quantization level above, the H, S, V three-dimensional feature vector for different values of with different weights to form one dimensional feature vector and is given by the following equation:

$$G = Qs*Qv*H+Qv*s+V$$

Where Qs is the quantized series of S and Qv is the quantized series of V. And now by setting Qs = Qv = 3, Then G = 9H+3S+V

$$H = \begin{cases} 0 \ \ if \ \ h \in [316,20] \\ 1 \ \ if \ \ h \in [21,40] \\ 2 \ \ if \ \ h \in [41,75] \\ 3 \ \ if \ \ h \in [76,155] \\ 4 \ \ if \ \ h \in [156,190] \\ 5 \ \ if \ \ h \in [191,270] \\ 6 \ \ if \ \ h \in [271,295] \\ 7 \ \ if \ \ h \in [296,315] \end{cases} \qquad S = \begin{cases} 0 \ \ if \ \ s \in [0,0.2) \\ 1 \ \ if \ \ s \in [0.2,0.7) \\ 2 \ \ if \ \ s \in [0.7,1) \end{cases} \qquad V = \begin{cases} 0 \ \ if \ \ v \in [0,0.2) \\ 1 \ \ if \ \ v \in [0.2,0.7) \\ 2 \ \ if \ \ v \in [0.7,1) \end{cases}$$

In this way three component vector of the HSV from one dimensional vector, Which quantize the whole color space for the 72 kinds of the main colors. So we can handle 72 bins of one dimensional histogram. This qualification is effective in reducing the images by the effect of the light intensity, but also reducing the computational time and complexity.

IMAGE RETRIEVAL:

Image retrieval is nothing but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords or the descriptions to the images so that the retrieval can be performed. Manual image annotation is time consuming, expensive and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc..,

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others. one can today generate volumes of images with content as diverse as family get-togethers and national parkvisits. Low-cost storage and easy Web hosting has fueled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchable image data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real-world image search system designers.

As far as technological advances are concerned, growth in Content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real world implications, applications, and constraints of the technology. Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future.

An image retrieval system designed to serve a personal collection should focus onfeatures such as personalization, flexibility of browsing, and display methodology. For example, Google's Picasa system [Picasa 2004] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods. The notable few applications include an FPGA implementation ofa color-histogram-based image retrieval system [Kotoulas and Andreadis 2003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 2003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 2002].

Discussion. Regardless of the nature of the collection, as the expected user-base grows, factors such as concurrent query support, efficient caching, and parallel and distributed processing of requests become critical. For future real-world image retrieval systems, both software and hardware approaches to address these issues are essential. More realistically, dedicated specialized servers, optimized memory and storage support, and highly parallelizable image search algorithms to exploit cluster computing powers are where the future of large-scale image search hardware support lies.

OVERVIEW OF TEXTURE:

We all know about the term Texture but for defining it is a hard time. One can differentiate the two different Textures by recognizing the similarities and differences. Commonly there are three ways for the usage of the Textures:

Based on the Textures the images can be segmented To differentiate between already segmented regions or to classify them.We can reproduce Textures by producing the descriptions. The texture can be analyzed in three different ways. They are Spectral, Structural and Statistical

1.5 DIGITAL IMAGE PROCESSING

Background:

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An important characteristic underlying the design of image processing systems is the significant level of testing & experimentation that normally is required before arriving at an acceptable solution. This characteristic implies that the ability to formulate approaches &quickly prototype candidate solutions generally plays a major role in reducing the cost & time required to arrive at a viable system implementation.

What is DIP

An image may be defined as a two-dimensional function f(x, y), where x & y are spatial coordinates, & the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital image by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensor, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis& computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting

& somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low- level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low- level process is characterized by the fact that both its inputs & outputs are images.

Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images. Finally higher- level processing involves “Making sense” of an ensemble of recognized objects,as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

What is an image?

An image is represented as a two dimensional function f(x, y) where x and y are spatial co-ordinates and the amplitude of 'f' at any pair of coordinates (x, y) is called the intensity of the image at that point.

Gray scale image:

A grayscale image is a function I (xylem) of the two spatial coordinates of the image plane.

I(x, y) is the intensity of the image at the point (x, y) on the image plane.

I (xylem) takes non-negative values assume the image is bounded by a rectangle [0, a] $\times$[0, b]I: [0, a] $\times$ [0, b] $\rightarrow$ [0, info)

Color image:

It can be represented by three functions, R (xylem) for red, G (xylem) for green *and* B (xylem) for blue.

An image may be continuous with respect to the x and y coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate's values is called sampling. Digitizing the amplitude values is called quantization.

Coordinate convention:

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image f(x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size M X N. The values of the coordinates (xylem) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates.

In many image processing books, the image origin is defined to be at (xylem)=(0,0).The next coordinate values along the first row of the image are (xylem)=(0,1).It is important to keep in mind that the notation (0,1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Following figure shows the coordinate convention. Note that x ranges from 0 to M-1 and y from 0 to N-1 in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (xylem) the toolbox uses the notation (race) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate topples, (alb), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentation refers to the coordinates. Less frequently the toolbox also employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refers to rows. This is the opposite of our use of variables x and y.

Image as Matrices:

The preceding discussion leads to the following representation for a digitized image function:

f (0,0)   f(0,1) ....f(0,N-1)

f (1,0)   f(1,1) ....f(1,N-1)

f (xylem)= f (M-1,0)   f(M-1,1)..................f(M-1,N-1)

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements. A digital image can be represented naturally as a MATLAB matrix:

f (1,1)   f(1,2)............f(1,N)

f (2,1)   f(2,2) ....f (2,N)

f =   f (M,1)   f(M,2).......... f(M,N)

Where f (1,1) = f(0,0) (note the use of a monoscope font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation f(p ,q) denotes the element located in row p and the column q. For example f(6,2) is the element in the sixth row and second column of the matrix f. Typically we use the letters M and N respectively to denote the number of rows and columns in a matrix. A 1xN matrix is called a row vector whereas an Mx1 matrix is called a column vector. A 1x1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using mono-scope characters. We use conventional Roman, italic notation such as f(x ,y), for mathematical expressions

Reading Images:

Images are read into the MATLAB environment using function imread whose syntax is Imread ('filename')

Format name	**Description**	**Extensions**
TIFF	Tagged Image File Format	tif, .tiff
JPEG	Joint Photograph Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	.xwd

Table 1.1:File formats to read images.

Here filename is a spring containing the complete of the image file(including any applicable extension).For example the command line

>> f = imread ('8. jpg');

Reads the JPEG (above table) image chestxray into image array f. Note the use of single quotes (') to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output If a semicolon is not included. MATLAB displays the results of the operation(s) specified in that line. The prompt symbol (>>) designates the beginning of a command line, as it appears in the MATLAB command window.

Data Classes:

Although we work with integers coordinates the values of pixels themselves are not restricted to be integers in MATLAB. Table above list various data classes supported by MATLAB and IPT are representing pixels values. The first eight entries in the table are refers to as numeric data classes. The ninth entry is the char class and, as shown, the last entry is referred to as logical data class.

All numeric computations in MATLAB are done in double quantities, so this is also a frequent data class encounter in image processing applications. Class unit 8 also is encountered frequently, especially when reading data from storages devices, as 8 bit images are most common representations found in practice. These two data classes, classes logical, and, to a lesser degree, class unit 16 constitute the primary data classes on which we focus. Many ipt functions however support all the data classes listed in table. Data class double requires 8 bytes to represent a number uint8 and int 8 require one byte each, uint16 and int16 requires 2bytes and unit 32.

Name	Description
Double	Double _ precision, floating_ point numbers the Approximate.
Uint8	unsigned 8_bit integers in the range [0,255].
Uint16	unsigned 16_bit integers in the range [0, 65535].
Uint 32	unsigned 32_bit integers in therange [0, 4294967295]
Int8	signed 8_bit integers in the range [-128,127].
Int 16	signed 16_byte integers in the range [32768, 32767].
Int 32	Signed 32_byte integers in the range [-2147483648, 21474833647].
Single	single _precision floating _point numbers with values
Char	characters (2 bytes per elements).
Logical	values are 0 to 1 (1byte per element).

Table 1.2 Data Classes In Matlab

Int 32 and single required 4 bytes each. The char data class holds characters in Unicode representation. A character string is merely a 1*n array of characters logical array contains only the values 0 to 1,with each element being stored in memory using function logical or by using relational operators.

Image Types:

The toolbox supports four types of images:

1 .Intensity images;

2. Binary images;

3. Indexed images;

4. R G B images.

Most monochrome image processing operations are carried out using binary or intensity images, so our initial focus is on these two image types. Indexed and RGB colour images.

Intensity Images:

An intensity image is a data matrix whose values have been scaled to represent intentions. When the elements of an intensity image are of class unit8, or class unit 16, they have integer values in the range [0,255] and [0, 65535], respectively. If the image is of class double, the values are floating point numbers. Values of scaled, double intensity images are in the range [0, 1] by convention.

Binary Images:

Binary images have a very specific meaning in MATLAB.A binary image is a logical array 0s and1s.Thus, an array of 0s and 1s whose values are of data class, say unit8, is not considered as a binary image in MATLAB .A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, we create an array B using the statement.

B=logical (A)

If A contains elements other than 0s and 1s.Use of the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.

Using relational and logical operators also creates logical arrays.

To test if an array is logical we use the I logical function: islogical(c).

If c is a logical array, this function returns a 1.Otherwise returns a 0. Logical array can be converted to numeric arrays using the data class conversion functions.

Indexed Images:

An indexed image has two components:

A data matrix integer, x

A color map matrix, map

Matrix map is an m*3 arrays of class double containing floating point values in the range [0, 1].The length m of the map are equal to the number of colors it defines.Each row of map specifies the red, green and blue components of a single color. An indexed images uses "direct mapping" of pixel intensity values color map values. The color of each pixel is determined by using the corresponding value the integer matrix x as a pointer in to map. If x is of class double ,then all of its components with values less than or equal to 1 point to the first row in map, all components with value 2 point to the second row and so on. If x is of class units or unit 16, then all components value 0 pointto the first row in map, all components with value 1 point to the second and so on.

RGB Image:

An RGB color image is an M*N*3 array of color pixels where each color pixel is triplet corresponding to the red, green and blue components of an RGB image, at a specific spatial location. An RGB image may be viewed as "stack" of three gray scale images that when fed in to the red, green and blue inputs of a color monitor

Produce a color image on the screen. Convention the three images forming an RGB color image are referred to as the red, green and blue components images. The data class of the components images determines their range of values. If an RGB image is of class double the range of values is [0, 1].

Similarly the range of values is [0,255] or [0, 65535].For RGB images of class units or unit 16 respectively. The number of bits use to represents the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8bit image, the corresponding RGB image is said to be 24 bits deep.

Generally, the number of bits in all component images is the same. In this case the number of possible color in an RGB image is $(2^b)^3$, where b is a number of bits in each component image. For the 8bit case the number is 16,777,216 colors

CHAPTER 2

LITERATURE SURVEY

2.1 Literature survey:

To measure the similarity between two objects, it is often beneficial to transform the data into another feature space and then compute the distances between vectors in this space. This can be accomplished by training a Siamese network to learn an appropriate transformation. Early research in this direction was conducted by who devised a system based on Siamese networks where handwritten signatures are compared with each other to perform signature verification. The SiamFC tracker is the basis of a multitude of Siamese networks for object tracking. It uses a network architecture similar to AlexNet for feature extraction. showed that using deeper and wider network architectures yields better tracking performance, at the cost of an increase in inference time. introduced a triplet loss function for training Siamese tracking networks. They found that, compared to the more commonly used logistic loss, using this loss function results in a better performance in all network architectures they investigated. A significant increase in tracking performance was achieved with the introduction of SA-Siam. In the VOT2018 benchmark SA-Siam achieved the second place among all real-time trackers. The basic idea behind SA-Siam is to train one network on the object tracking task and another network on object classification. The first network learns a feature representation which models the appearance of an object while the latter network outputs sparse features with more semantic meaning. The outputs of both networks are used for tracking, leading to a performance superior toboth networks used on their own. The SiamBM tracker improves the tracking performance of SASiam by adding angle estimation, spatial masks and template updating.

2.2 Existing models:

ResNet

To train the network model in a more effective manner, we herein adopt the same strategy as that used for DSSD (the performance of the residual network is better than that of the VGG network). The goal is to improve accuracy. However, the first implemented for the modification was the replacement of the VGG network which is used in the original SSD with ResNet. We will also add a series of convolution feature layers at the end of the underlying network. These feature layers will gradually be reduced in size that allowed prediction of the detection results on multiple scales. When the input size is given as 300 and 320, although the ResNet–101 layer is deeper than the VGG–16 layer, it is experimentally known that it replaces the SSD's underlying convolution network with a residual network,and it does not improve its accuracy but rather decreases it.

R-CNN

To circumvent the problem of selecting a huge number of regions, Ross Girshick et al. proposed a method where we use the selective search for extract just 2000 regions from the image and he called them region proposals. Therefore, instead of trying to classify the huge number of regions, you can just work with 2000 regions. These 2000 region proposals are generated by using the selective search algorithm which is written below.

Selective Search:

1. Generate the initial sub-segmentation, we generate many candidate regions
2. Use the greedy algorithm to recursively combine similar regions into larger ones
3. Use generated regions to produce the final candidate region proposals

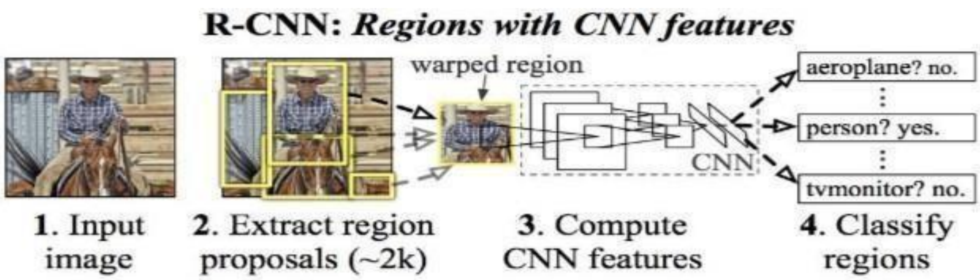


Fig. 2.1 R-CNN Architecture

These 2000 candidate regions which are proposals are warped into a square and fed into a convolutional neural network that produces a 4096-dimensional feature vector asoutput. The CNN plays a role of feature extractor and the output dense layer consists of the features extracted from the image and the extracted features are fed into an SVM for the classify the presence of the object within that candidate region proposal. In addition to predicting the presence of an object within the region proposals, the algorithm also predicts four values which are offset values for increasing the precision of the bounding box. For example, given the region proposal, the algorithm might have predicted the presence of a person but the face of that person within that region proposal could have been cut in half. Therefore, the offset values which is given help in adjusting the bounding box of the region proposal.

Fast R-CNN:

The same author of the previous paper(R-CNN) solved some of the drawbacks of R-CNN to build a faster object detection algorithm and it was called Fast R-CNN. The approach is similar to the R-CNN algorithm. But, instead of feeding the region proposals to the CNN, we feed the input image to the CNN to generate a convolutional feature map. From the convolutional feature map, we can identify the region of the proposals and warp them into the squares and by using an RoI pooling layer we reshape them into the fixed size so that it can be fed into a fully connected layer. From the RoI feature vector, we can use a soft max layerto predict the class of the proposed region and also the offset values for the bounding box.

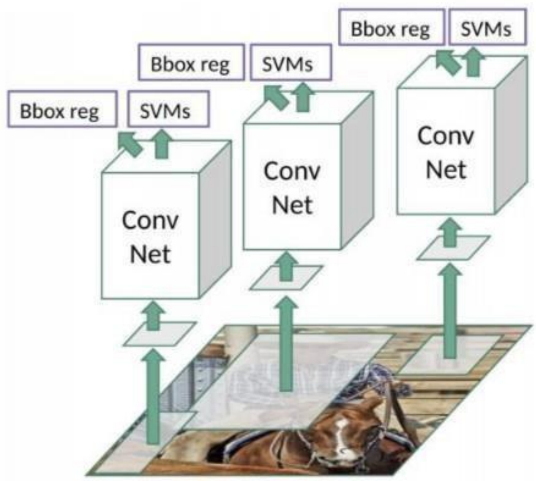


Fig. 2.2 Fast R-CNN Architecture

The reason “Fast R-CNN” is faster than R-CNN is because you don’t have to feed 2000 region proposals to the convolutional neural network every time. Instead, the convolution operation is always done only once per image and a feature map is generated from it.

Faster R-CNN:

Both of the above algorithms(R-CNN & Fast R-CNN) uses selective search to find out the region proposals. Selective search is the slow and time-consuming process which affect the performance of similar to Fast R-CNN, the image is provided as an input to a convolutional network which provides a convolutional feature map. Instead of using the selective search algorithm for the feature map to identify the region proposals, a separate network is used to predict the region proposals. The predicted he region which is proposals are then reshaped using an RoI pooling layer which is used to classify.

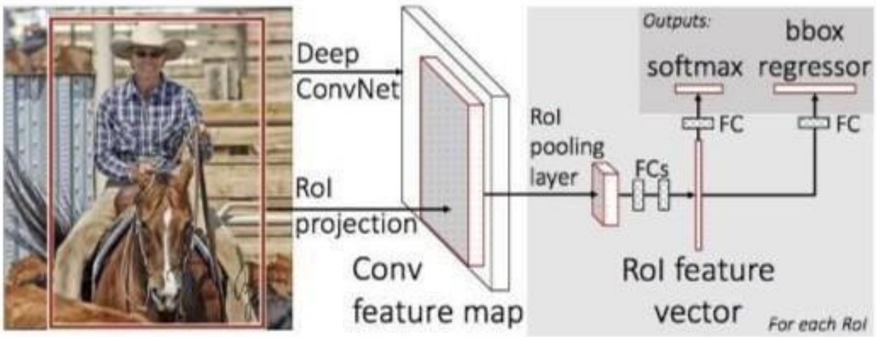


Fig. 2.3 Faster R-CNN Architecture

SDD:

The SSD object detection composes of 2 parts:

1. Extract feature maps, and

2.. Apply convolution filters to detect objects.

SSD uses VGG16 to extract feature maps. Then it detects objects using the Conv4_3 layer. For illustration, we draw the Conv4_3 to be 8 × 8 spatially (it should be 38 × 38). For each cell in the image(also called location), it makes 4 object predictions.

Each prediction composes of a boundary box and 21 scores for each class (one extra class for no object), and we pick the highest score as the class for the bounded object. Conv4_3 makes total of 38 × 38 × 4 predictions: four predictions per cell regardless of the depth of feature maps. A expected, many predictions contain no object. SSD reserves a class “0” to indicate

.

SSD does not use the delegated region proposal network. Instead, it resolves to a very simple method. It computes both the location and class scores using small convolution filters. After extraction the feature maps, SSD applies 3 × 3 convolution filters for each cell to make predictions. (These filters compute the results just like the regular CNN filters.) Each filter gives outputs as 25 channels: 21 scores for each class plus one boundary box.

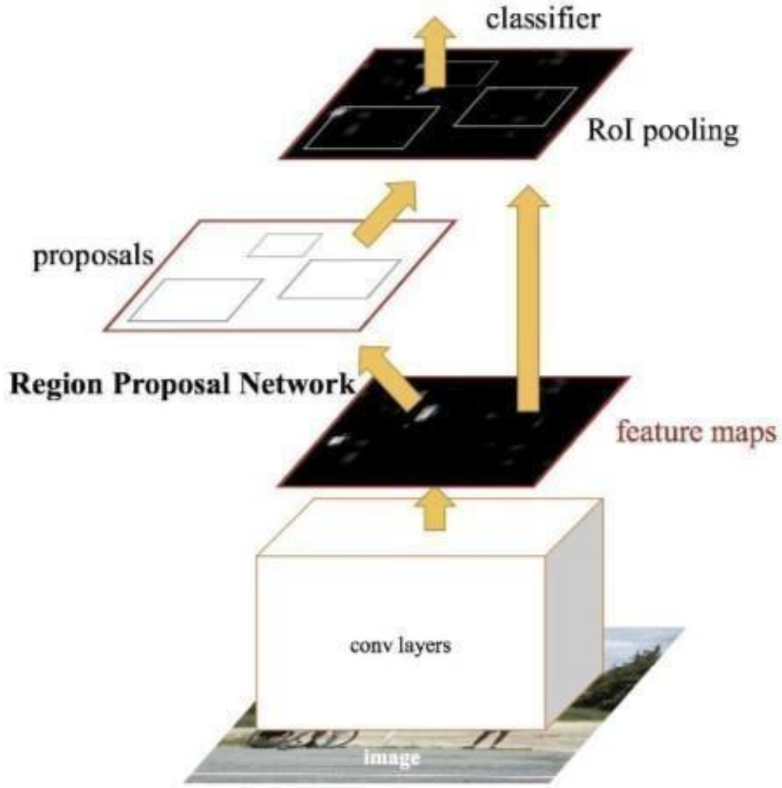


Fig. 2.4 SDD Architecture

Beginning, we describe the SSD detects objects from a single layer. Actually, it uses multiple layers (multi-scale feature maps) for the detecting objects independently. As CNN reduces the spatial dimension gradually, the resolution of the feature maps also decrease. SSD uses lower resolution layers for the detect larger-scale objects. For example, the 4× 4 feature maps are used for the larger-scale object.

MANet:

Target detection is fundamental challenging problem for long time and has been a hotspot in the area of computer vision for many years. The purpose and objective of target detection is, to determine if any instances of a specified category of objects exist in an image. If there is an object to be detected in a specific image, target detection return the spatial positions and the spatial extent of the instances of the objects (based on the use a bounding box, for example). As one of cornerstones of image understanding and computer visions, targets and object detection forms the basis for more complex and higher-level visual tasks, such as object tracking, image capture, instance segmentation, and others.Target detection is also widely used in areas such as artificial intelligence and information technology, including machine vision, automatic driving vehicles, and human–computer interaction. In recent times, the method automatic learning of represented features from data based on deep learning has effectively improved performance of target detection. Neural networks are foundation of deep learning. Therefore, design of better neural networks has become an key issue toward improvement of target detection algorithms and performance. Recently developed objectdetectors that has been based on convolutional neural networks (CNN) has been classified in two types:The first is two-stage detector type, such as Region-Based CNN (R–CNN), Region-Based Full Convolutional Networks (R–FCN), and Feature Pyramid Network (FPN), and the other is single-stage detector, such as the You Only Look Once (YOLO), Single-shot detector (SSD), and the RetinaNet. The former type generates an series of candidate frames as samples of data , and then classifies the samples based on a CNN; the latter type do not generate candidate frames but directly converts the object frame positioning problem into a regression processing problem.

To maintain realtime speeds without sacrificing precision in various object detectors described above, Liu et al proposed the SSD which is faster than YOLO and has a comparable accuracy to that of the most advanced region-based target detectors.SSD combines regression idea of YOLO with the anchor box mechanism of Faster R–CNN, predicts the object region based on the feature maps of the different convolution layers, and outputs discretised multi-scale and multi proportional default box coordinates. The convolution kernel predicts frame coordinates compensation of a series of candidate frames and the confidence of each category. The local feature maps of multi-scale area are used to obtain results for each position in the entire image. This maintains the fast characteristics of YOLO algorithm and also ensures that the frame positioning effect is similar to that is induced by the Faster R–CNN. However, SSD directly and independently uses two layers of the backbone VGG16 and four extra layers obtained by a convolution with stride 2 toconstruct feature pyramid but lacks strong contextual connections.

2.3 Proposed Model:

In recent years, the tracking model based on the Siamese Network has been widely used in the object tracking field to model the object tracking task as a similarity matching problem, which balances the tracking speed and accuracy. However, there are insufficient robustness, discriminative ability and generalisation ability for object deformation and complex background interference. In this paper, an improved Fully-convolutional Siamese Network is proposed. The Triplet Loss function is used as the model objective function instead of logistic loss, and the multi-channel attention mechanism is introduced to make the model pay more attention to the tracking related information and enhance the model discriminating ability. In the offline training phase, an effective data augmentation strategy is used to control the uneven distribution of sample categories and improve the generalisation ability of the model.

In the tracking phase, the Distractor-aware module is used to transfer the general feature representation domain to a specific object domain, thereby improving model discriminating ability. In experiments, the results on VOT2016 tracking benchmark shows that our model has a significant improvement over the SiamFC tracker in multiple evaluation indicators.

CHAPTER 3

BLOCK DIAGRAM

3.1 BLOCK DIAGRAM DESCRIPTION:

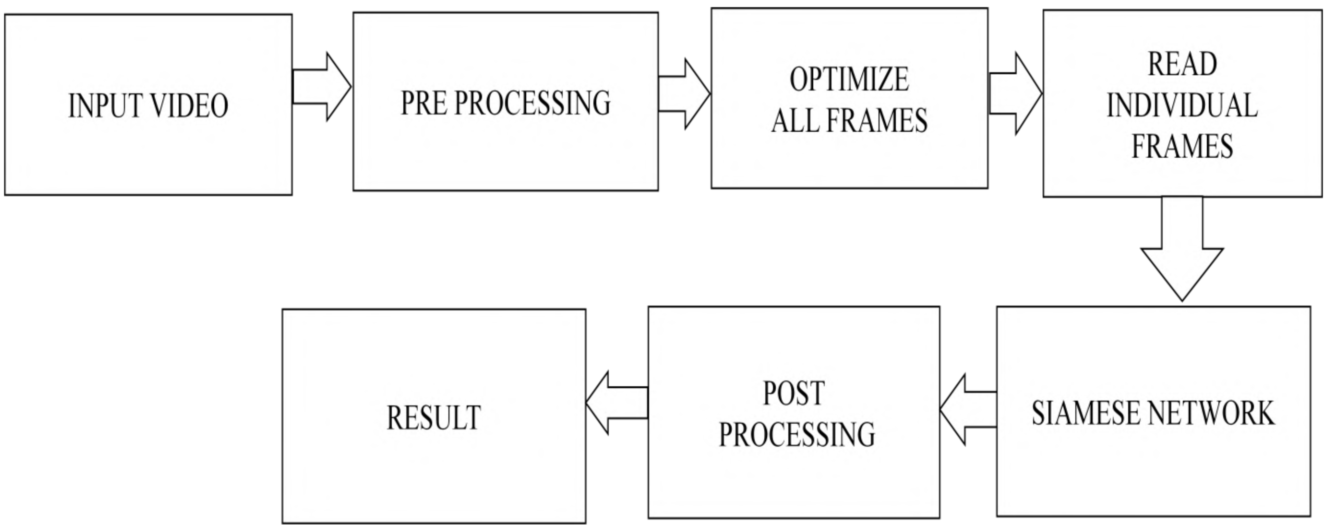


Fig 3.1 Block Diagram

Pre Processing:

We used a network architecture inspired by the SiamFC network. Specifically, as feature extractor we used a network similar to AlexNet. In order to support inputs of different sizes, our feature extractor is a fully-convolutional network. Let φ denote the function computed by the feature extractor. Then we train the network such that P i φ(z)iφ(z 0)ishould be large when z and z 0 are images of the same object and small otherwise. Given a templatez and a search image x, we can then compute φ(x) ? φ(z), where ? denotes the cross- correlation operator. The result should ideally be high

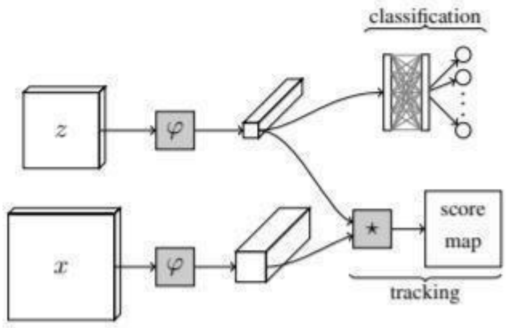


Fig.3.2 Architecture of our network

at the location where the object in z is located in image x and low everywhere else. Similar to the original AlexNet classification network, for the classification task we added fully-connected layers after the feature extractor. The output of this network head are classification labels. This network architecture forces the feature extractor to learn features which are helpful for the tracking task as well as for the classification task.

Siamese Network:

The cross-correlation of the output features should ideally yield a score map y with high values in the center and low values everywhere else. To formalize this requirement, let the score map be a function

$$y \ : \ \{0, \ldots, k\}^2 \to \mathbb{R}.$$

We define two thresholds a and b satisfying $a \leq b$. The grid of the domain of the score map can be partitioned into the sets

$$\begin{aligned} X &= (i,j)^T \in \{0, \ldots, k\}^2 : \|(i,j)^T - (\tfrac{k}{2}, \tfrac{k}{2})^T\| \leq a \\ Y &= (i,j)^T \in \{0, \ldots, k\}^2 : \|(i,j)^T - (\tfrac{k}{2}, \tfrac{k}{2})^T\| > b \\ Z &= \{0, \ldots, k\}^2 \setminus (X \cup Y). \end{aligned}$$

Then we want y(n) to be large for all $n \in X$ and to be small for all $n \in Y$. The scores y(n) for $n \in Z$ are disregarded. Usually, the logistic loss is used to train atracker:

$$l_L(y) = \frac{1}{|X|}\sum_{n\in X}\log(1+\exp(-y(n)))+$$
$$\frac{1}{|Y|}\sum_{m\in Y}\log(1+\exp(y(m)))$$

This loss function is small when the score map has a sharp peak in the center. During tracking, the maximum of the score map is used to update the position of the object. Thus, it is important that the differences between y(n), n ∈ X and y(n), n ∈ Y are large. To accomplish that, proposed to use the triplet loss

$$l_T(y) = \frac{1}{|X|\cdot|Y|}\sum_{\substack{n\in X\\ m\in Y}}\log(1+\exp(y(m)-y(n))).$$

The triplet loss has been used in several other computer vision applications, such as face recognition [11]. For the object classification task, we use the cross entropy loss lC . The loss function we try to minimize is then l(c, y) = lC (c) + μ · lT (y), (3) where c is the output of the classification head of our network.

Post Processing:

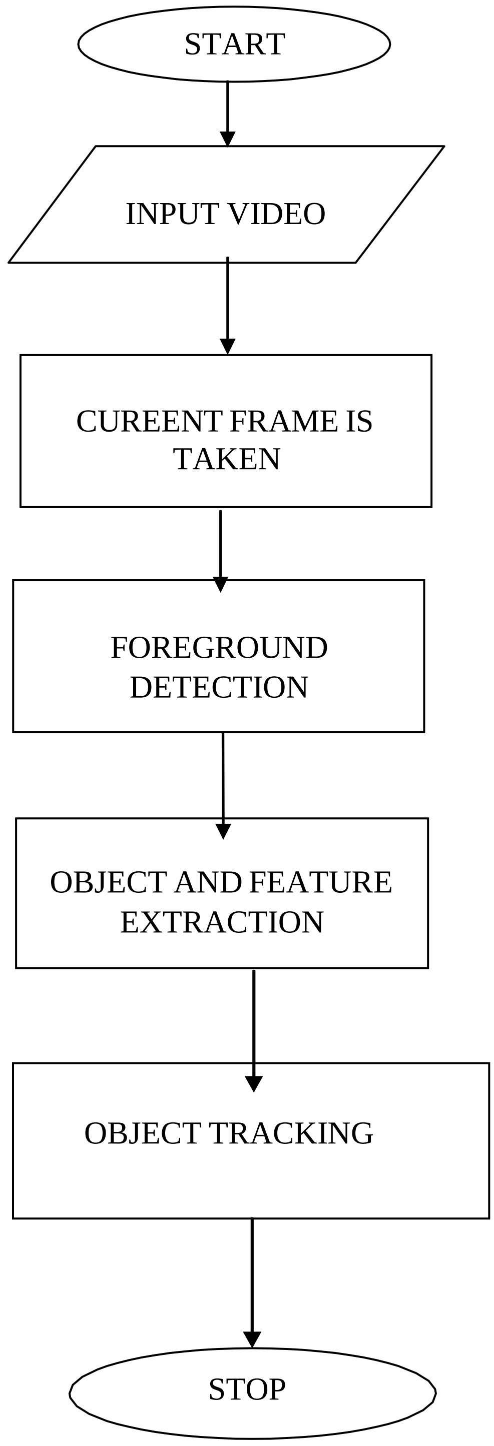
In the tracking problem, a tracker is provided with an initial image and a corresponding bounding box enclosing the object to be tracked. The task of the tracker is then to find the corresponding bounding boxes in the subsequent images. The na¨ıve way to solve this task with Siamese networks is to compute ϕ(z1) once, where z1 is a patch containing the target in the initial frame and use ϕ(z1) for all subsequent frames. showed experimentally that updating the template during tracking improves the performance of the tracker. Let zi be the target patch in frame i. Then the template ti for locating the target in frame i is defined as

$$t_2 = \varphi(z_1)$$
$$t_{i+1} = (1-\lambda)\cdot t_i + \lambda\cdot\varphi(z_i), \quad i>1$$

for some learning rate λ. We adopted this template updating scheme in our tracker.1 It is possible to devise other updating schemes. For example, proposed an updating scheme where $\phi(z1)$ is weighted higher than subsequent patches. Recently, proposed a more sophisticated updating scheme where the update function is learned by a neural network.

Result: Finally the tracking of the object will be shown during the run time of the code and tracking in the frames will be shown from one frame to another and classification of the objects in the each frame wil be shown parallel to the tracking of the object. This classification is shown in the top left of the screen.

3.2 FLOW CHART:



3.3 ALGORITHM:

Step 1: Start

Step 2: Upload the input video in the code and then the video will be splited into frames.

Step 3: The first frame is displayed and then the object which we want to track is to be bounded in a square box.

Step 4: The object to be tracked its feature are extracted and the background features are eliminated and then the centroid is located for the object.

Step 5: The centroid of the object is compared with frames there by the displacement is compared in all the frames of the object.

Step 6: The displacemet is shown according on the screen during thr run time from first frame to last frame.

Step 7: Stop.

Ki

CHAPTER 4

SOFTWARE IMPLEMENTATION

4.1. INTRODUCTION TO MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

4.2 THE MATLAB SYSTEM:

The MATLAB system consists of five main parts

Development Environment:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

The MATLAB Mathematical Function Library:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

The MATLAB Language:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

Graphics:

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

The MATLAB Application Program Interface (API):

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

4.3 GRAPHICAL USER INTERFACE (GUI):

MATLAB’s Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M- functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function’s GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.

A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type

guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted,

GUIDE opens a new (i.e., blank) window.

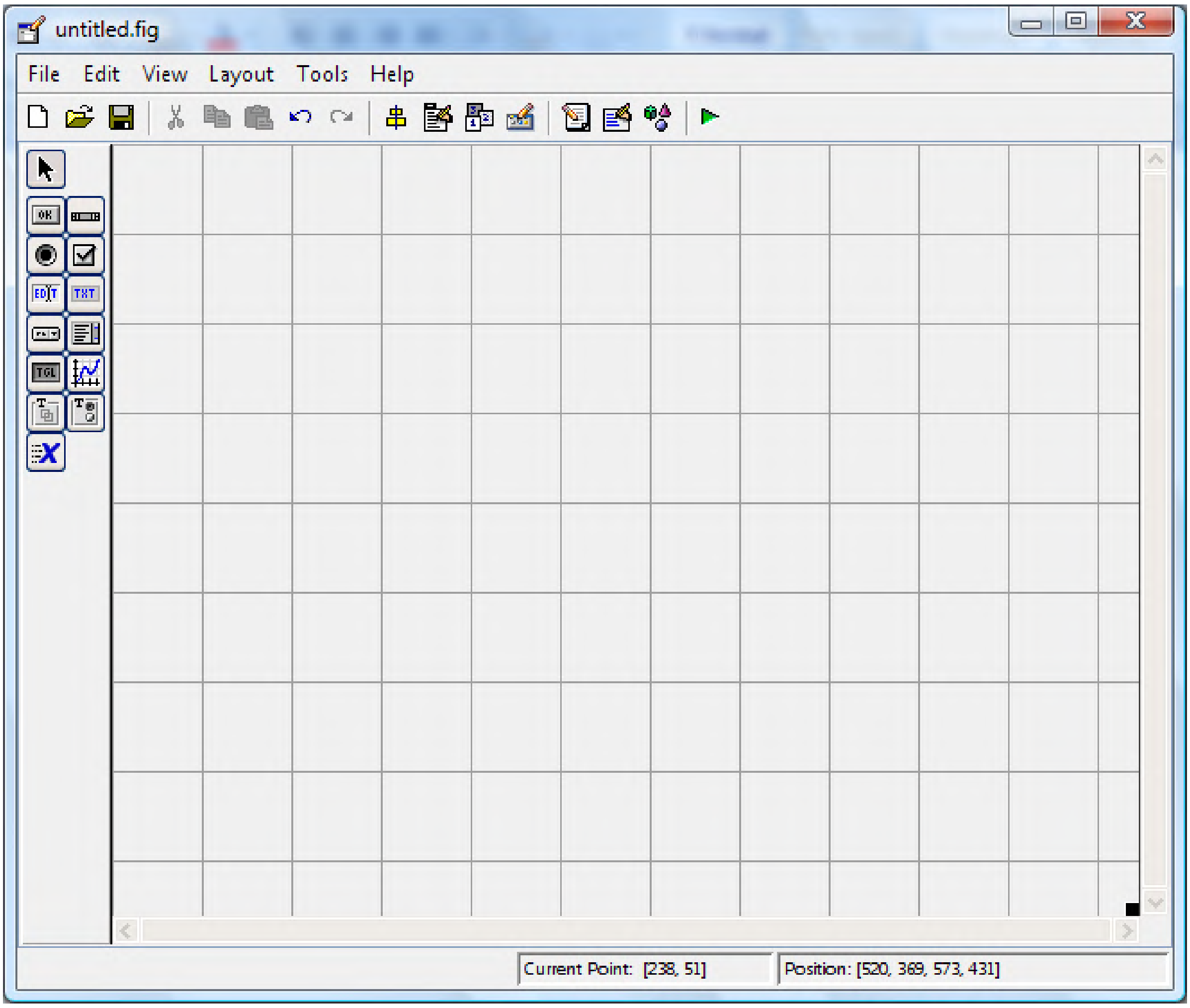


Fig 4.3.1 Blank Window

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

4.4 Getting Started

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

Introduction - describes the components of the MATLAB system.

Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

Manipulating Matrices - introduces how to use MATLAB to generate matrices andperform mathematical operations on matrices.

Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

4.5 DEVELOPMENT ENVIRONMENT

Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics underDevelopment Environment in the MATLAB documentation, which is available online as well as in print.

Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

Current Directory Browser
Workspace Browser
Array Editor

Editor/Debugger

Command Window

Command History

Launch Pad

Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type helpbrowser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace byusing functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace toa binary file called a MAT-file, which has a .mat extension. There are options for savingto different formats. To read in a MAT-file, select Import Data from the File menu, or usethe load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to runMATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

4.6 MANIPULATING MATRICES

Entering Matrices

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

Enter an explicit list of elements.
Load matrices from external data files.
Generate matrices using built-in functions.
Create matrices with your own functions in M-files.
Start by entering Dürer's matrix as a list of its elements. You have only to follow a few basic conventions:

Separate the elements of a row with blanks or commas.
Use a semicolon, ; , to indicate the end of each row.
Surround the entire list of elements with square brackets, [ ].
To enter Dürer's matrix, simply type in the Command Window

A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]

MATLAB displays the matrix you just entered.

A =

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

Operators

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

Table 4.1 List Of Arithmetic Operators

Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of theelementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of usefulconstants.

Pi	3.14159265...
I	Imaginary unit, $\sqrt{-1}$
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

Table 4.2 Functions And Its Values

4.7 GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

GUI Development Environment

The process of implementing a GUI involves two basic task.

Laying out the GUI components

Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of its children (uicontrols and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the callbacks, which are defined as subfunctions. This M-file is referred to as the application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

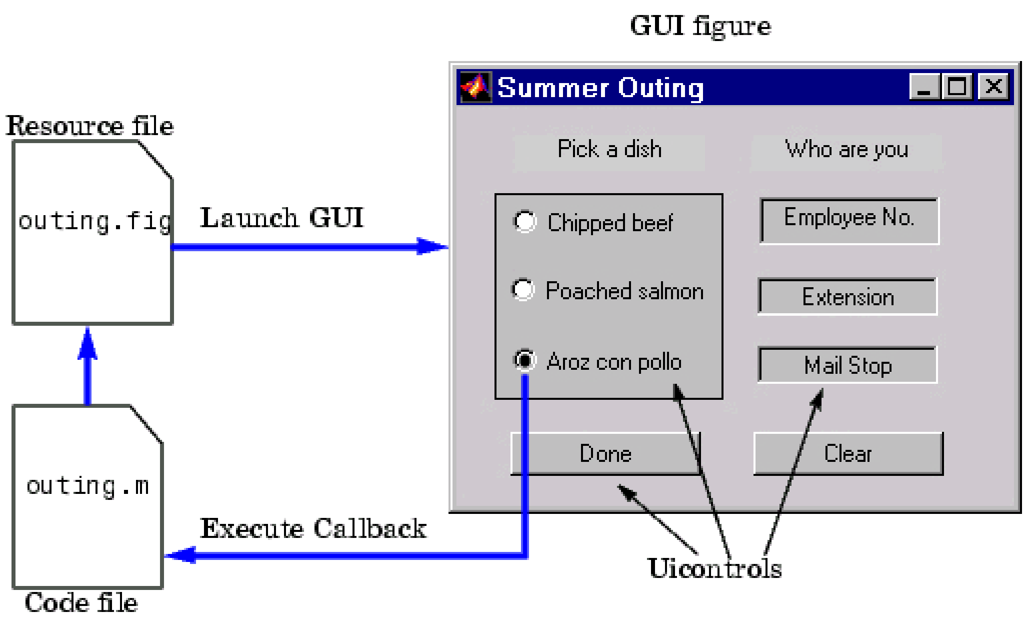


FIG 4.7.1 graphical user blocks

Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques

used in the application M-file.

Application Examples - a collection of examples that illustrate techniques

which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commandssuch as findobj, set, and get. If you issue another plotting command, the output is directedto the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and areprogrammable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- Listboxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed;

when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed and equal to the Min property when the toggle button is not depressed

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)

button_state = get(h,'Value');

if button_state == get(h,'Max')

   % toggle button is pressed

elseif button_state == get(h,'Min')
```

% toggle button is not pressed

End

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

a(:,:,1) = rand(16,128);

a(:,:,2) = rand(16,128);

a(:,:,3) = rand(16,128);

set(h,'CData',a)

Radio Buttons

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)

set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)

off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];

mutual_exclude(off)

% Continue with callback

  .

  .

  .
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)
if (get(h,'Value') == get(h,'Max'))
   % then checkbox is checked-take approriate action
else
   % checkbox is not checked-take approriate action
end
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)
user_string = get(h,'string');
% proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)

user_entry = str2double(get(h,'string'));

if isnan(user_entry)

    errordlg('You must enter a numeric value','Bad Input','modal')

end

% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menubar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menubar does not cause the editable text callback routine toexecute. This behavior is consistent with the respective platform conventions. Clicking onother components in the GUI execute the callback.

Static Text

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

Listboxes differentiate between single and double clicks on an item and set the figure SelectionType property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure SelectionType property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the indexcontained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

This callback checks the index of selected item and uses a switch statement to take action based on the value. If the contents of popup menu is fixed,you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

switch val
```

case 1

% The user selected the first item

case 2

% The user selected the second item

% etc.

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

string_list = get(h,'String');

selected_string = string_list{val}; % convert from cell array to string

% etc.
```

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is

grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's

ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes ButtonDownFcn property to define the callback.

Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,axes(handles.axes1)

makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axes. See GUI with Multiple Axes for and example that uses two axes.

Figure

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control.

CHAPTER 5

RESULT ANALYSIS

5.1 Results:

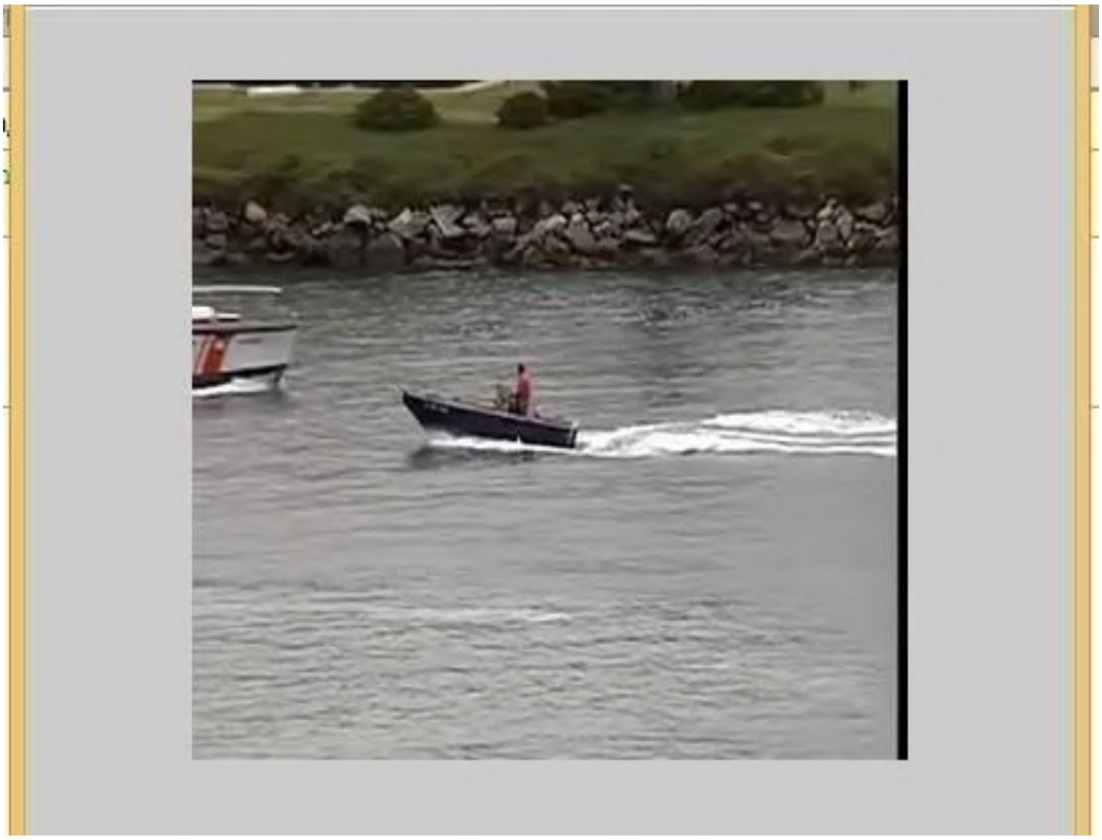


Fig 5.1 Output 1 of the system

In Fig 5.1, it shows the test frame display of the input video soon after we run the code.

Fig 5.2 Output 2 of the system

In the fig.2, we will select the object which has to be tracked and classified by dragging a box around the object

Fig 5.3 Output 3 of the system

In the fig.3, it shows the object is tracked up to 80 frames out of 191 frames, i.e till the point where the object is exactly visible.

CHAPTER 6

ADVANTAGES AND LIMITATIONS

6.1 Advantages:

- In object tracking you know to recognize same object in 2 different frames, helping you for example to count vehicles etc, because you know that exact objects showed and disappeared.
- In object tracking we use image measurements to estimate position of object, but also incorporate position predicted by dynamics.
- Object tracking is helpful in trying to detect articulated objects.
- An object tracking method will be more robust to changes in pose/viewpoint.
- Object tracking using Siamese network has High precision rate, High performance, Low misdetection rate and Less complexity.
- Object tracking with siamese networks require fewer data samples required and can work with highly imbalanced data.

6.2 Limitations:

- Difficulties in tracking objects can arise due to abrupt object motion, changing appearance patterns of both the object and the scene, nonrigid object structures, object-to-object and object-to-scene occlusions, and camera motion.
- Detecting of the object has following challenges: dual priorities, speed, multiple scales, limited data, and class imbalance.
- In this network, a large amount of training data because of so many pairs of classes.
- Not generalizable. A model trained for one task cannot be used for another task.
- Sensitive to some variations in the input.
- For object tracking single method cannot give good accuracy for different kind of videos with different situation like poor resolution, change in weather condition.
- It has non real time applications.
- It locates objects with horizontal bounding box.

6.3 Applications

Object tracking has an assortment of uses, some of which are

- Surveillance and security.
- Traffic checking.
- Video correspondence.
- Robot vision and activity.
- Defence systems.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion:

We have shown that training a feature extractor on both the classification and the tracking task can be beneficial for building a tracker. One key advantage of building a tracker like this instead of using separate feature extractors is that it is faster during inference. The runtime aspect is crucial, given that many tracking systems run in real-time mention that their feature extractor trained on classification learns very sparse features while the other feature extractor trained on tracking learns dense features. It might be interesting to investigate the feature representation learned by our network, which trains on both tasks simultaneously. It might also be worthwhile to investigate if the converse is also true, i.e. if a classifier can be improved by training its feature extractor on the task of tracking. Furthermore, it might be possible to train the feature extractor of a tracker on computer vision tasks other than classification and see similar improvements in tracking performance.

7.2 Future scope:

Our future work may include:

1. The results of object detection and classification in this experiment are good, but the performance of object tracking is not ideal. Tracking objects by using data association may not be the best solution. In the future, LSTM can be employed to predict the trajectory of objects, thereby improves the accuracy of object tracking.
2. The types of anomalies monitored by the network are limited. In future, more data can be trained to implement the detection of more abnormal categories.
3. The network proposed in this experiment cannot tracking objects across cameras due to the similarity of the same target in different backgrounds predicted by SiamRPN is not accurate enough. Meanwhile, for the actual application of video surveillance, anomaly detection is usually multiple camera-based, for example, tracking the same suspicious target in different cameras. In the future, the algorithm can be optimized to achieve multi-target tracking by using these cameras.

St. MARTIN'S ENGINEERING COLLEGE

An Autonomous Institute

A Non Minority College | Approved by AICTE | Affiliated to JNTUH, Hyderabad | NAAC-Accredited 'A+' Grade | 2(f) & 12(B) status (UGC) ISO 9001:2008 Certified | NBA Accredited | SIRO (DSIR) | UGC-Paramarsh | Recognized Remote Center of IIT, Bombay

Dhulapally, Secunderabad – 500100, Telangana State, India.

www.smec.ac.in

Acceptance Letter for online oral presentation & journal publication

Online Mega international Conference "Smart Modernistic in Electronics and Communication" *(ICSMEC–21) on 02nd & 03rd July 2021*

Dear P.Sushma, U.Nikitha, V.Jaya Surya:

Title of the paper: A SHARED REPRESENTATION FOR OBJECT TRACKING AND CLASSIFICATION USING SIAMESE NETWORKS

Paper ID: ICSMEC21-0075

With heartiest congratulations we are pleased to inform you that based on the recommendations of the expert reviewers, your paper has been accepted for journal publication and oral presentation in online mega International Conference on "Smart Modernistic in Electronics and Communication" ***(ICSMEC-21).***

ICSMEC-21 conference received over 400+ submissions from different parts of the world and regions so far, reviewed by international experts; the acceptance ratio is controlled below 20%. Your paper will be submitted for **Journal publications free of cost** in the appropriate given Journals after registration. **You will be given registration kit, International conference proceedings with ISBN (Soft copy), and international certificate of publication.**

You are now required to pay the conference registration fees (Faculty / Research Scholars: Rs.3500/-) as mentioned below A/C details on or before 5 working day after receiving this email else your paper will not be considered for presentation as well as publication in the Journal.

For NEFT/RTGS/IMPS

Account Details: (Current Account)

A/C Name: PRINCIPAL, St. Martin's Engineering College
Bank Name: SBI, Kukatpally, Hyderabad
Account No: 30667640268
IFSC code: SBIN0011664
Swift Code: SBININBB725 (Applicable for foreign author)

For PAYTM users

After payment, please send the registration form, payment proof and final paper in.doc/.docx format.

Your interest in *ICSMEC-21* is very much appreciated. Look forward to meeting you on the day of conference.

Yours sincerely,
ICSMEC- 21 Organizing Committees
icsmec21@smec.ac.in

(Organized by Department of ECE)

REFERENCES

[1] R. Caruana, “Multitask learning,” Machine Learning, vol. 28, no. 1, pp. 41–75, 1997.

[2] A. He, C. Luo, X. Tian, and W. Zeng, “A twofold siamese network for real-time object tracking,” in 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, 2018, pp. 4834–4843.

[3] J. Bromley, I. Guyon, Y. LeCun, E. Sackinger, and R. Shah, ¨ “Signature verification using a siamese time delay neural network,” in Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993], 1993, pp. 737–744.

[4] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, “Fully-convolutional siamese networks for object tracking,” in Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II, 2016, pp. 850–865.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States., 2012, pp. 1106– 1114.

[6] Z. Zhang and H. Peng, “Deeper and wider siamese networks for real-time visual tracking,” in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, 2019, pp. 4591–4600.

[7] X. Dong and J. Shen, “Triplet loss in siamese network for object tracking,” in Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII, 2018, pp. 472–488.

[8] M. Kristan et al., “The sixth visual object tracking VOT2018 challenge results,” in Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part I, 2018, pp. 3–53.

[9] A. He, C. Luo, X. Tian, and W. Zeng, “Towards a better match in siamese network based visual object tracker,” in Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part I, 2018, pp. 132–147.

[10]Y. Wu, J. Lim, and M. Yang, “Online object tracking: A benchmark,” in 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23- 28, 2013, 2013, pp. 2411–2418.

[11]F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, 2015, pp. 815–823.

[12]J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, “End-to-end representation learning for correlation filter based tracking,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, 2017, pp. 5000–5008.

[13]L. Zhang, A. Gonzalez-Garcia, J. v. d. Weijer, M. Danelljan, and F. S. Khan, “Learning the model update for siamese trackers,” in The IEEE International Conference on Computer Vision (ICCV), October 2019.

[14] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, 2015, pp. 448–456.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, “Imagenet large scale visual recognition challenge,” International Journal of Computer Vision, vol. 115, no. 3, pp. 211–252, 2015.

[16] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, 2018, pp. 7132–7141.

A

Project report on

AUTOMATIC RAIN WATER SENSING UMBRELLA

Submitted By

P. VINAY KUMAR (17K81A0439)

AKSHAY SAKRE (17K81A0443)

SHAIK ALTAF (17K81A0447)

in partial fulfilment for the award of

degree of

BACHELOR OF TECHNOLOGY

IN

Electronics & Communication Engineering

Under the Guidance of

Mrs.G.Laxmi Priyanka

M.Tech. Assistant Professor

St. MARTIN'S ENGINEERING COLLEGE

(Autonomous Institute)

Dhulapally, Secunderabad-500 100.

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE
UGC AUTONOMOUS
Dhulapally, Secunderabad-500 100
NBA &NAAC A+ Accredited

BONAFIDE CERTIFICATE

This is to certify that the project entitled "Automatic Rain Water Sensing Umbrella" is being submitted by **1. Mr. P. VINAY KUMAR (17K81A0439), 2. Mr. AKSHAY SAKRE (17K81A0443), 3. Mr. Shaik Altaf (17K81A0447)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Internal Guide	**Head of the Department**
G. Laxmi Priyanka Department of Electronics and Communication Engineering	Dr.G. Hari Krishna Department of Electronics and Communication Engineering
Internal Examiner	**External Examiner**

Place:

Date:

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **POLINATI VINAY KUMAR** ROLL NO. **17K81A0439, SAKRE AKSHAY** WITH ROLL NO. **17K81A0443**, **SHAIK ALTAF** WITH ROLL NO. **17K81A0447,** OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECTTITLED "**AUTOMATIC RAIN WATER SENSING UMBRELLA**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT

DIRECTOR
TRAININGS & PLACEMENTS LASYA IT
SOLUTIONS PVT LTD.

LasyaITSolutionsPvtLtd,BehindCinePlanet,Kompally,MedchalRoad,Secunderabad500014
Email :contact@lasyainfotech.com,cv@lasyainfotech.com
Website :www.lasyainfotech.com| contact: 7330666881/82/83/84/86

DECLARATION

We declare that this project report titled "AUTOMATIC RAIN WATER SENSING UMBRELLA" submitted in partial fulfillment of the degree of B. Tech in Electronics and Communication Engineering record of original work carried out by us under the guidance and supervision of Mrs. G. Laxmi Priyanka, and has not formed the name basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have beencited.

P.vinay kumar	**Akshay sakre**	**Shaik Altaf**
17K81A0439	**17K81A0443**	**17K81A0447**

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

We extended our deep sense of gratitude to Principal, **Dr. P. SANTOSH KUMAR PATRA**, St. Martin's Engineering College, Dhulapally, for permitting us to undertake this project.

We are also thankful to **Dr.B.Hari Krishna** Head of the Department, Electronics And Communications, St. Martin's Engineering College, Dhulapally, for his support and guidance throughout our project and as well as our project coordinator **Dr.A.Anand**, Professor in Electronics And Communication Engineering, for his valuable support.

We would like to express our sincere gratitude and indebtedness to our project supervisor **Mrs.G. Laxmi Priyanka**, Assistant Professor, Electronics And Communications St. Martin's Engineering College, Dhulapally, for her support and guidance throughout our project.

We sincerely thank our project internal guide , **Ms G. Laxmi priyanka** , Assistant professor , Department of ECE and our class project coordinator **Ms A.Sowmya** Assistant professor, Department of ECE for guidance and encouragement in carrying out this project work

Finally we are highly grateful and obliged to our college coordinator **Dr Malayadri** professor of IT and **Dr Ravi Kumar** associate professor of ECE for their advice supervision and vital contribution as and when required during this research work and for their support and cooperation That is difficult to express in words.

We express thanks to all those who have helped us in successfully completing the project.

1. P.vinay kumar
2. Akshay sakre
3. Shaik Altaf

Abstract

During monsoon, sellers with an open shop, neighbours drying clothes outside and many similar situations cause inconveniences in our life. In case of street vegetable sellers, they need to protect the vegetables, fruits and customers from rains in monsoon season. Even the tarpaulins don't prove to be of any use during torrential rains accompanied by a thunderstorm. To overcome such problems and help us live with the inconvenience raining system. this auto rain-sensing umbrella smart system comes up with a solution. This smart rain sensing system can detect the rain and opens up the umbrella. In this smart system, we have a raindrop sensing system, which gives a reading proportional to the amount of rain pouring on it. The smart system consisting of a rack and pinion system, the rack is fixed to umbrella such that when a sensor senses the exceeding value of raindrops, it gives a signal to the pinion attached to a motor. Then the motor starts rotating and the umbrella opens.

Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Introduction:

Natural resource elements are associated with day-to-day activities such as rainfall and sun light which is having both positive as well as negative impact on our lives and property. The sun emits ultraviolet energy which is one of the sources for vitamin D, staggering the growth of oncogenes and also used by plants during photosynthesis which is indirectly source of food for all living creature. Rainfall also having its very negative impact and such as destruction and damage of nursery bed herbs or plants and various ornamental flowers, skin disorder in humans, fever and sickness, and increase in mortality of living stocks, structural and materialistic deformation of properties such as fabrics [1]. A larger number and various types of structural and control system have been proposed in the past for particularly management, regulation, controlling and moderation of weather element to the life and property, but the retractable roof is the most suitable form for moderating these weather elements (Rain and Sunlight). Unlike permanent structures for the same purpose, the self-adjustable system only acts when a threshold is attained [2, 3]. The problem identified in previous model was for the continuous power supply so our designed model can be installed anywhere. It could be possibly access to renewable source of energy like sunlight which is cheaper and cleaner source of energy. The problem associated with the proper design of the umbrella which can be possibly be made portable and easy to use is also there and other components with that which are to be protected from the extreme weather conditions. The design proposed for many systems uses retractable roof of wooden which can possibly be damaged by rain so proper material selection has to be done for longer life [4, 5]. Sunlight and rainfall Activated Retractable Illustrates through his paper that natural climatic elements like sunlight and rainfall have a very huge importance in our day-to-day life. These two weather elements have both positive and negative impacts in our life. These two weather elements have some positive consequences so they cannot be ignored completely [6, 7]. To avoid the complete isolation of sunlight and rainfall a system is developed which will economical and give space reuse whenever the threshold value exceeds then roof automatically gets open up. An Algorithm is developed by the author to facilitate the automatic working of the roof whenever the intensity of these two weather elements exceeds the present value. In this system author has used an LDR sensor to detect to intensity of sunlight and ultrasonic and rain sensor for detection of rainfall intensity and pressure which will be helping in smart decision making. This system has a programmable device Arduino which will help in controlling of stepper motor for opening and closing of roof and two switches aiding for better control in all direction. ‘Smart Automation System Using Arduino and Rain Drop Sensor’ illustrated in his project that during rainy season the crops gets affected due to unexpected rainfall or even with hailstorm sometime in order to eliminate such things he developed a system to protect it from heavy rainfall [8]. The basic idea behind this research is to protect or to save the crops along with the rain water harvesting system. The rain sensing device along with the soil moisture sensing device is used in this system which will help in opening and closing of the automated roof. Automatic Rain Water and Crop Saving System discuses in his paper that agriculture is one of the most important sectors as it employs more than 45% of working population.

Agriculture sector contributes a larger section in our country’s GDP so its really important to look what are the factors affecting it in some way. Due to heavy rainfall farmers across globe face a lot of problem which affect the total income and productivity of the farmer and field respectively as unexpected rainfall washes off the grains and cereals. So, if they have to protect the crop a system can be developed which will ultimately look after the fields or the harvested crop and protect them from extreme condition of weather and also helps in rain water harvesting [9]. The problem stated above which is faced by people in day-to-day life by the extreme condition of weather leads to immense amount of problem in our lives. These problems are not only associated with money value but it also with emotional wellbeing of the person as loss of essential can lead to emotional breakdown of person. Various technological cannot be made to eliminate these problems completely but at certain extent to minimize the total cost. The equipment which is to be developed should be economical and easy to use. So that implementation of it can be done on larger areas. The Purpose this research is chosen to improve the design of the pre-existing umbrellas. Often times when rainfalls a person must constantly adjust the head of the umbrella to block the rain as it falls at different angles due to the wind. Our idea intends design a smart umbrella to be an extension to current umbrellas that will automatically adjust the head of the umbrella to block the maximum amount of rain according to the wind direction and speed. The designed system which is proposed should be light in weight, cost effective, and easy to attach to existing products in the market still existing adhesion problems. The modelled describes about all the important components which in turn will be used for making of system.

CHAPTER 2

EMBEDDED SYSTEMS

2.1 Embedded System

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.

The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes. The applications software on such processors is sometimes referred to as firmware

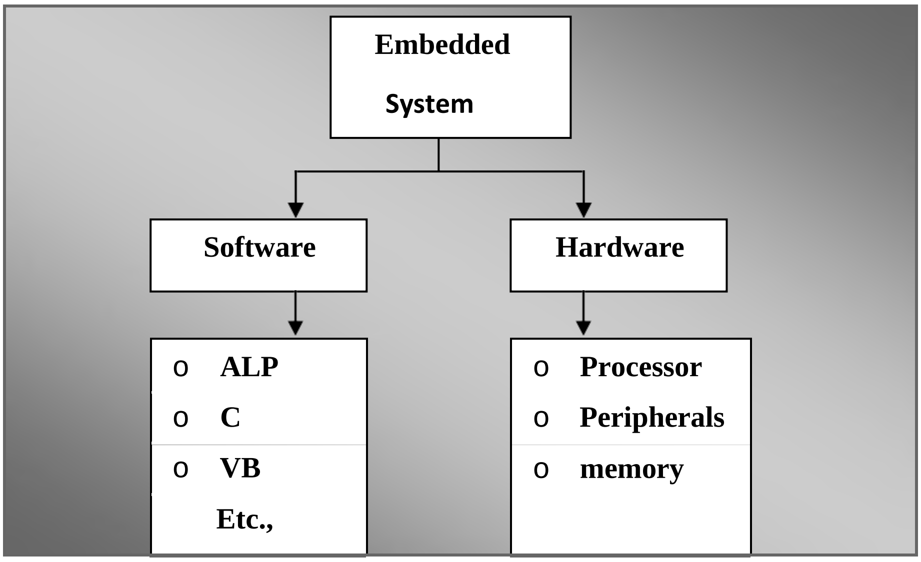


Figure 2.1 Block diagram of Embedded SystemNo table of figures entries found.

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

Memory: It is used to store data or address.

Peripherals: These are the external devices connected

Processor: It is an IC which is used to perform some task

Applications of embedded systems

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

Processors are classified into four types like:

- Micro Processor (µp)
- Micro controller (µc)
- Digital Signal Processor (DSP)
- Application Specific Integrated Circuits (ASIC)

Micro Processor (µp):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Three basic characteristics differentiate microprocessors:

- **Instruction set**: The set of instructions that the microprocessor can execute.
- **Bandwidth :** The number of bits processed in a single instruction.
- **Clock speed :** Given in megahertz (MHz), the clock speed determines how many instructions per second the processor can execute.

In both cases, the higher the value, the more powerful the CPU. For example, a 32-bit microprocessor that runs at 50MHz is more powerful than a 16-bit microprocessor that runs at 25MHz. In addition to bandwidth and clock speed, microprocessors are classified as being either RISC (reduced instruction set computer) or CISC (complex instruction set computer).

A microprocessor has three basic elements, as shown above. The ALU performs all arithmetic computations, such as addition, subtraction and logic operations (AND, OR, etc). It is controlled by the Control Unit and receives its data from the Register Array. The Register Array is a set of registers used for storing data. These registers can be accessed by the ALU very quickly. Some registers have specific functions - we will deal with these later. The Control Unit controls the entire process. It provides the timing and a control signal for getting data into and out of the registers and the ALU and it synchronizes the execution of instructions (we will deal with instruction execution at a later date).

Three Basic Elements of a Microprocessor

Micro Controller (µc):

A microcontroller is a small computer on a single **integrated circuit** containing a processor core, memory, and programmable **input/output** peripherals. Program memory in the form of **NOR flash** or **OTP ROM** is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the **microprocessors** used in **personal computers** or other general purpose applications.

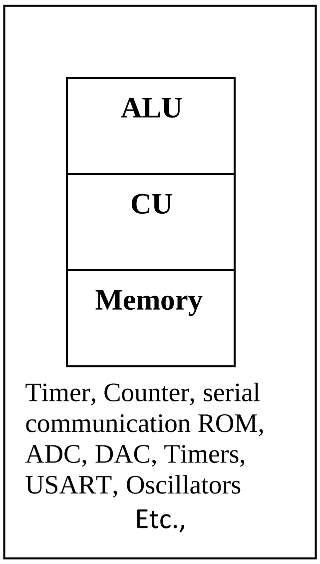


Figure 2.2 Block Diagram of Micro Controller (µc)

Digital Signal Processors (DSPs):

Digital Signal Processors is one which performs scientific and mathematical operation. Digital Signal Processor chips - specialized microprocessors with architectures designed specifically for the types of operations required in digital signal processing. Like a general-purpose microprocessor, a DSP is a programmable device, with its own native instruction code. DSP chips are capable of carrying out millions of floating point operations per second, and like their better-known general-purpose cousins, faster and more powerful versions are continually being introduced. DSPs can also be embedded within complex "system-on-chip" devices, often containing both analog and digital circuitry.

Application Specific Integrated Circuit (ASIC)

ASIC is a combination of digital and analog circuits packed into an IC to achieve the desired control/computation function

ASIC typically contains

- CPU cores for computation and control
- Peripherals to control timing critical functions
- Memories to store data and program
- Analog circuits to provide clocks and interface to the real world which is analog in nature
- I/Os to connect to external components like LEDs, memories, monitors etc.

2.2 Computer Instruction Set

There are two different types of computer instruction set there are:

1. RISC (Reduced Instruction Set Computer) and

2. CISC (Complex Instruction Set computer)

2.2.1 Reduced Instruction Set Computer (RISC)

A RISC (reduced instruction set computer) is a microprocessor that is designed to perform a smaller number of types of computer instruction so that it can operate at a higher speed (perform more million instructions per second, or millions of instructions per second). Since each instruction type that a computer must perform requires additional transistors and circuitry, a larger list or set of computer instructions tends to make the microprocessor more complicated and slower in operation.

Besides performance improvement, some advantages of RISC and related design improvements are:

- A new microprocessor can be developed and tested more quickly if one of its aims is to be less complicated.
- Operating system and application programmers who use the microprocessor's instructions will find it easier to develop code with a smaller instruction set.
- The simplicity of RISC allows more freedom to choose how to use the space on a microprocessor.

Higher-level language compilers produce more efficient code than formerly because they have always tended to use the smaller set of instructions to be found in a RISC computer.

2.2.2 RISC characteristics

- **Simple instruction set:**

In a RISC machine, the instruction set contains simple, basic instructions, from which more complex instructions can be composed.

- **Same length instructions.**

Each instruction is the same length, so that it may be fetched in a single operation.

- **1machine-cycleinstructions.**

Most instructions complete in one machine cycle, which allows the processor to handle several instructions at the same time. This pipelining is a key technique used to speed up RISC machines.

2.2.3 Complex Instruction Set Computer (CISC)

CISC, which stands for **Complex Instruction Set Computer,** is a philosophy for designing chips that are easy to program and which make efficient use of memory. Each instruction in a CISC instruction set might perform a series of operations inside the processor. This reduces the number of instructions required to implement a given program, and allows the programmer to learn a small but flexible set of instructions.

The advantages of CISC

At the time of their initial development, CISC machines used available technologies to optimize computer performance.

- Microprogramming is as easy as assembly language to implement, and much less expensive than hardwiring a control unit.
- The ease of micro-coding new instructions allowed designers to make CISC machines upwardly compatible: a new computer could run the same programs as earlier computers because the new computer would contain a superset of the instructions of the earlier computers.
- As each instruction became more capable, fewer instructions could be used to implement a given task. This made more efficient use of the relatively slow main memory.
- Because micro program instruction sets can be written to match the constructs of high-level languages, the compiler does not have to be as complicated.

The disadvantages of CISC

Still, designers soon realized that the CISC philosophy had its own problems, including:

- Earlier generations of a processor family generally were contained as a subset in every new version --- so instruction set & chip hardware become more complex with each generation of computers.
- So that as many instructions as possible could be stored in memory with the least possible wasted space, individual instructions could be of almost any length---this means that different instructions will take different amounts of clock time to execute, slowing down the overall performance of the machine.
- Many specialized instructions aren't used frequently enough to justify their existence --- approximately 20% of the available instructions are used in a typical program.
- CISC instructions typically set the condition codes as a side effect of the instruction. Not only does setting the condition codes take time, but programmers have to remember to examine the condition code bits before a subsequent instruction changes them.

2.3 Memory Architecture

There two different type's memory architectures there are:

- Harvard Architecture
- Von-Neumann Architecture

2.3.1 Harvard Architecture

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The **Harvard architecture** is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

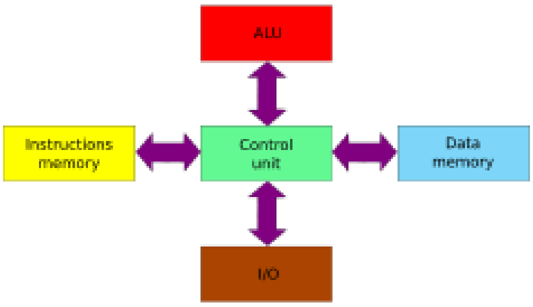


Figure 2.3 Harvard Architecture

Modern uses of the Harvard architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

- Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behavior must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.

- Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

2.3.2 Von-Neumann Architecture

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. The **von Neumann architecture** is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.A **stored-program** digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

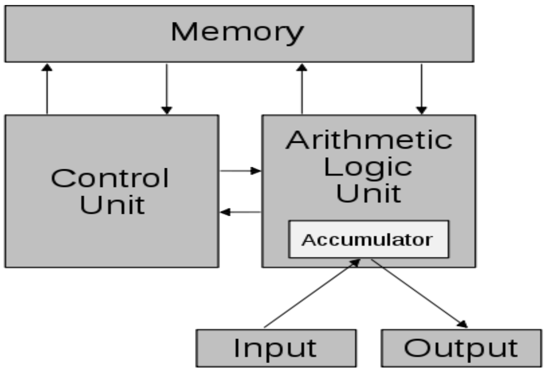


Figure 2.4 Schematic of the Von-Neumann Architecture.

Basic Difference between Harvard and Von-Neumann Architecture

- The primary difference between Harvard architecture and the Von Neumann architecture is in the Von Neumann architecture data and programs are stored in the same memory and managed by the same information handling system.
- Whereas the Harvard architecture stores data and programs in separate memory devices and they are handled by different subsystems.
- In a computer using the Von-Neumann architecture without cache; the central processing unit (CPU) can either be reading and instruction or writing/reading data to/from the memory. Both of these operations cannot occur simultaneously as the data and instructions use the same system bus.
- In a computer using the Harvard architecture the CPU can both read an instruction and access data memory at the same time without cache. This means that a computer with Harvard architecture can potentially be faster for a given circuit complexity because data access and instruction fetches do not contend for use of a single memory pathway.
- Today, the vast majority of computers are designed and built using the Von Neumann architecture template primarily because of the dynamic capabilities and efficiencies gained in designing, implementing, operating one memory system as opposed to two. Von Neumann architecture may be somewhat slower than the contrasting Harvard Architecture for certain specific tasks, but it is much more flexible and allows for many concepts unavailable to Harvard architecture such as self-programming, word processing and so on.

CHAPTER 3

POWER SUPPLY

3.1 Block Diagram

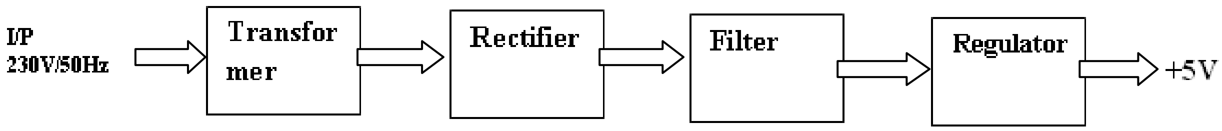


Figure 3.1 Power Supply

3.2 Circuit Diagram

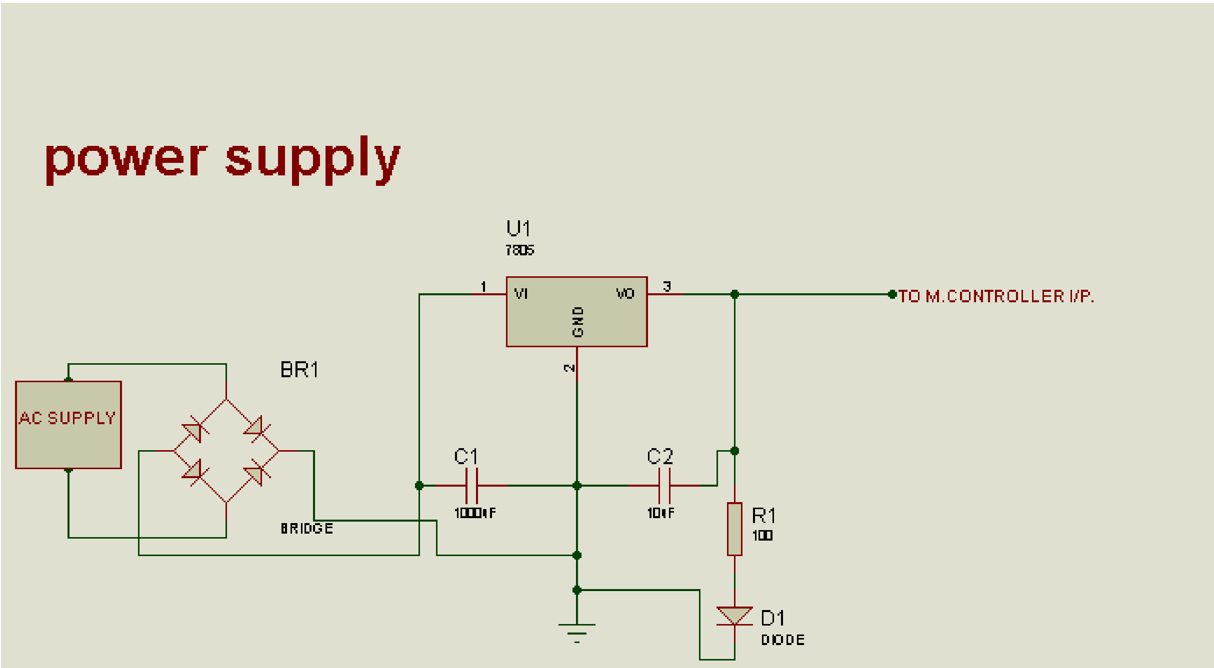


Figure 3.2 circuit diagram

Description

3.2.1 Transformer

A **transformer** is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

Transformer is a device that converts the one form energy to another form of energy like a transducer.

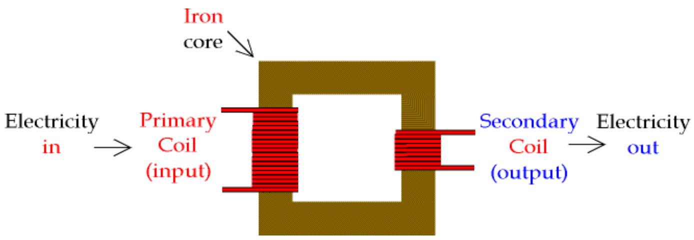


Figure 3.3 Transformer

Basic Principle:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

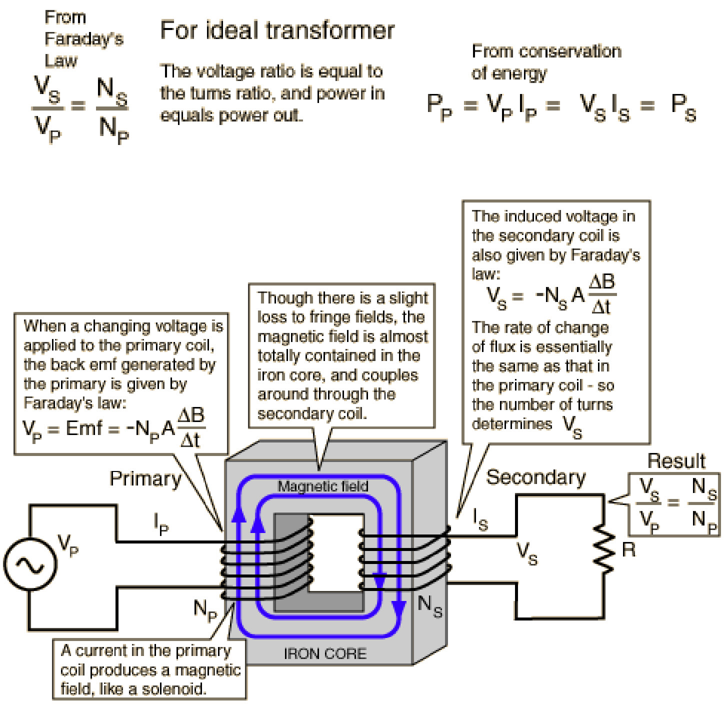


Figure: 3.4 Basic Principle

Transformer Working

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

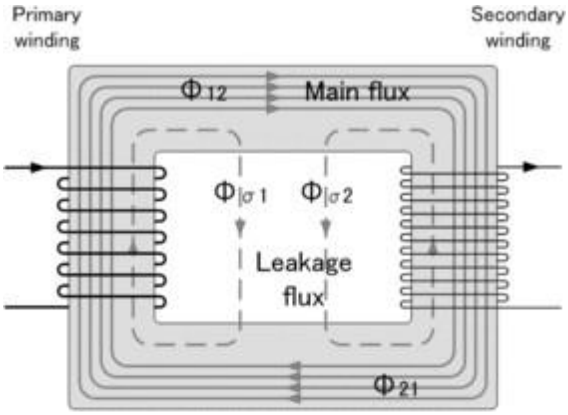


Figure: 3.5 Basic Transformers

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f.The iron core is laminated to prevent 'eddy currents' flowing in the core. These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

Classification of Transformer:

- Step-Up Transformer
- Step-Down Transformer

Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn's ratio of the two sets of windings determines the amount of voltage transformation.

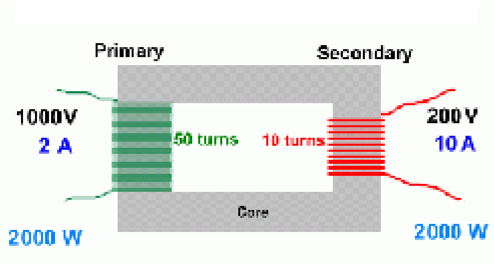


Figure: 3.6 Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single-phase step-down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase step up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

Step-Up Transformer:

A step-up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply.A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

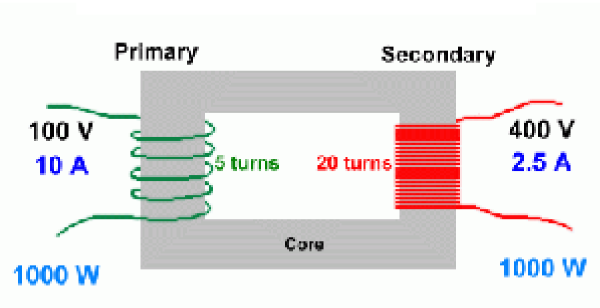


Figure: 3.7 Step-Up Transformer

Applications

Generally these **Step-Up Transformers** are used in industries applications only.

Turns Ratio and Voltage

The ratio of the number of turns on the primary and secondary coils determines the ratio of the voltages...

$$\frac{V_s}{V_p} = \frac{N_s}{N_p}$$

...where V_p is the primary (input) voltage, V_s is the secondary (output) voltage, N_p is the number of turns on the primary coil, and N_s is the number of turns on the secondary coil.

3.2.2 Diodes

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

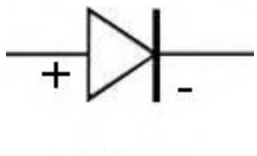


Figure 3.8 Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

3.2.3 Rectifier

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

The Half-wave Rectifier

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure .

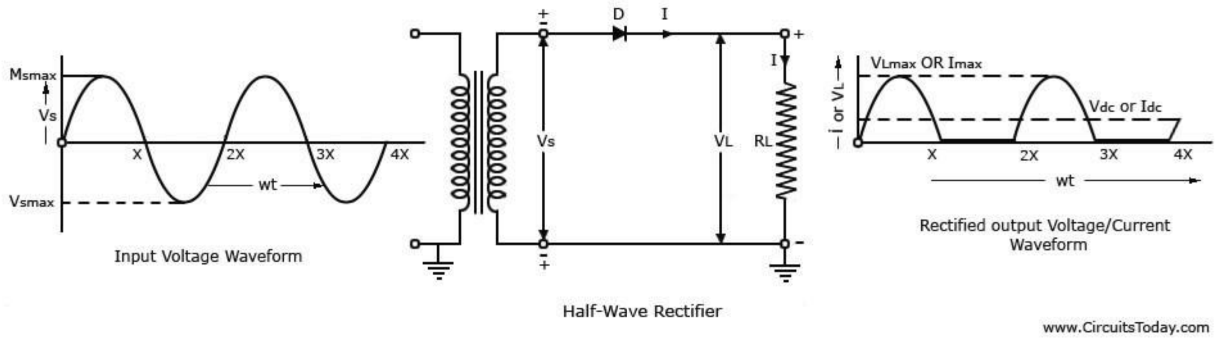


Figure 3.9 Half Wave Rectifier

Figure 2 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is 0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

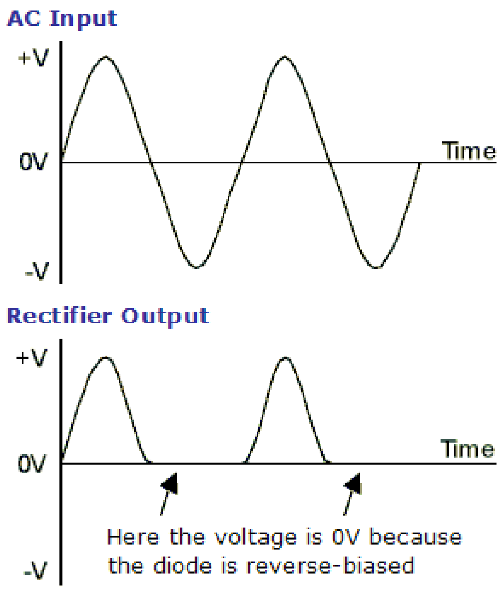


Figure 3.10 Half-Wave Rectification

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

The Full-wave Rectifier

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC. The resulting waveform is shown in figure 4.

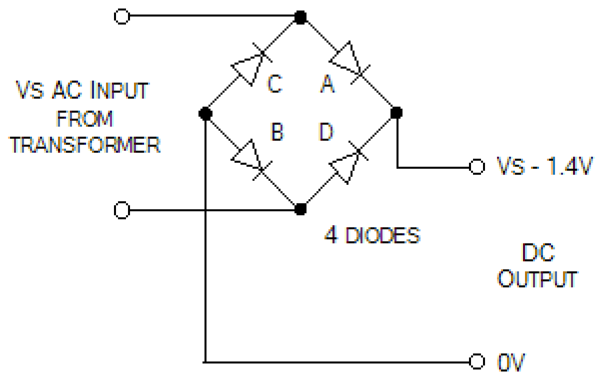


Figure 3.11 Full-Wave Rectifier

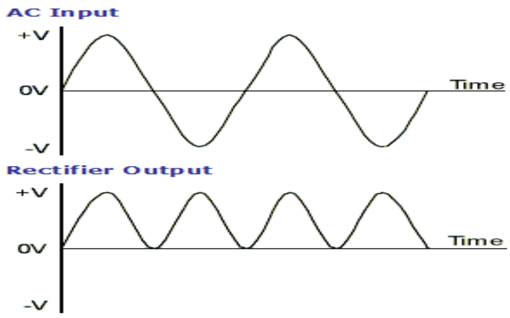


Figure 3.12 Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased.

While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

Capacitor Filter

The **capacitor-input filter**, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

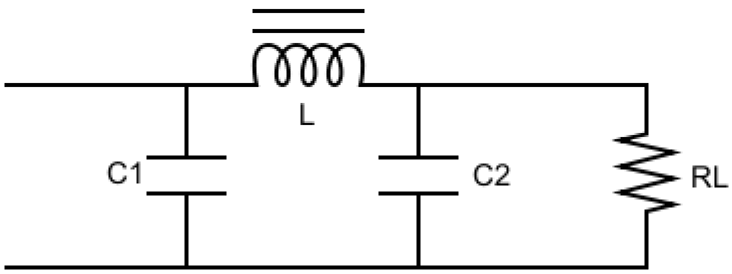


Figure:3.13 Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L

2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result only the DC component appears across the load RL.

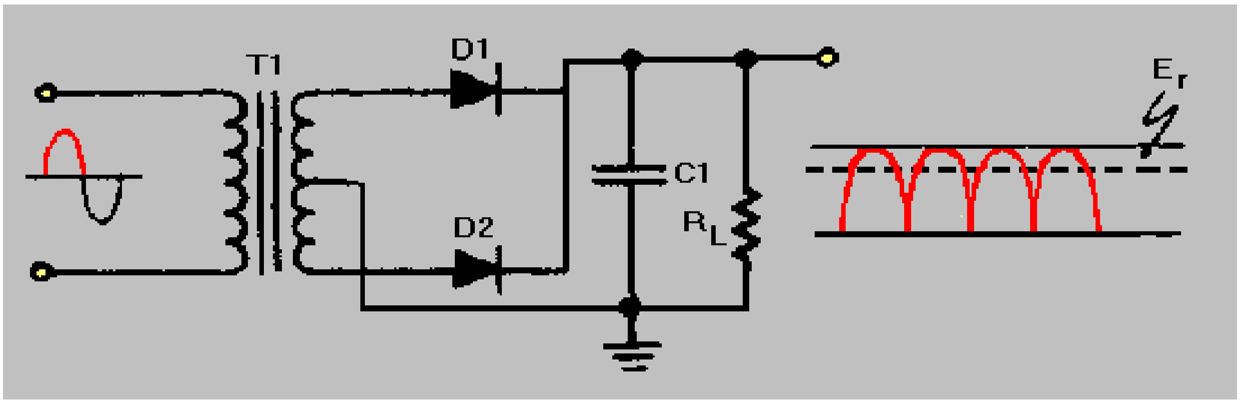


Figure: 3.14 Centered Tapped Full-Wave Rectifier with a Capacitor Filter

3.2.4 Voltage Regulator:

A **voltage regulator** is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx'indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx'indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consists the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

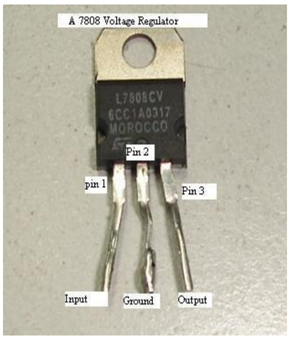


Figure 3.15 Regulator

CHAPTER 4

SOFTWARE

4.1 Proteus

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

4.1.1 About Proteus

It is a software suite containing schematic, simulation as well as PCB designing.

ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.

The designer can also develop 2D drawings for the product.

4.1.2 Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

4.2 Starting New Design

Step 1: Open ISIS software and select New design in File menu

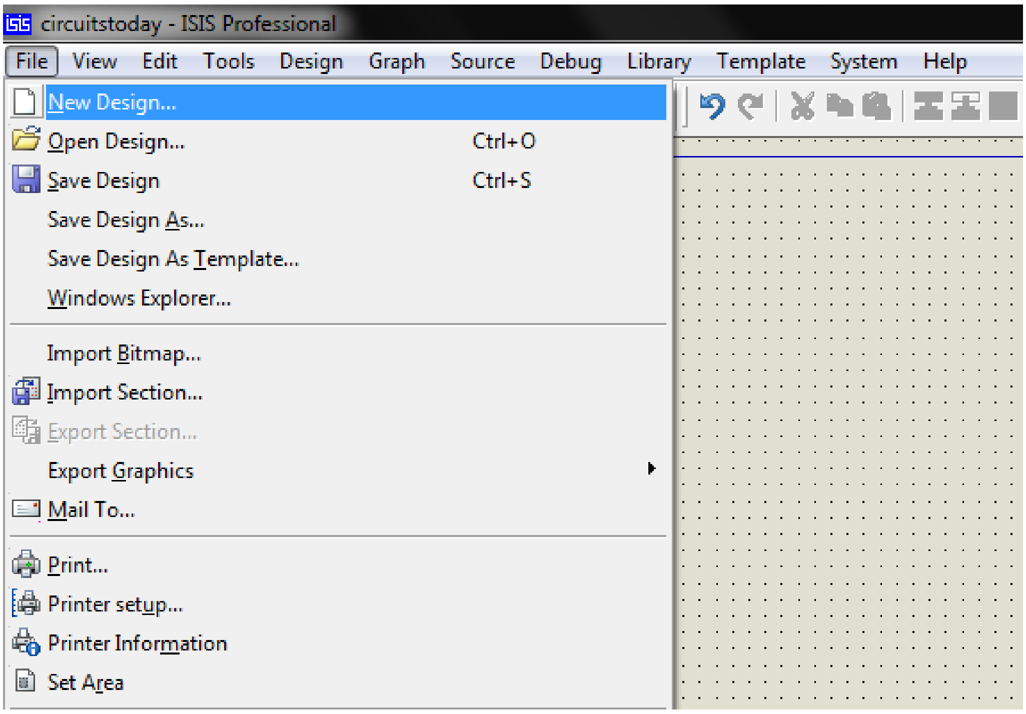


Figure 4.1 Proteus File Menus

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

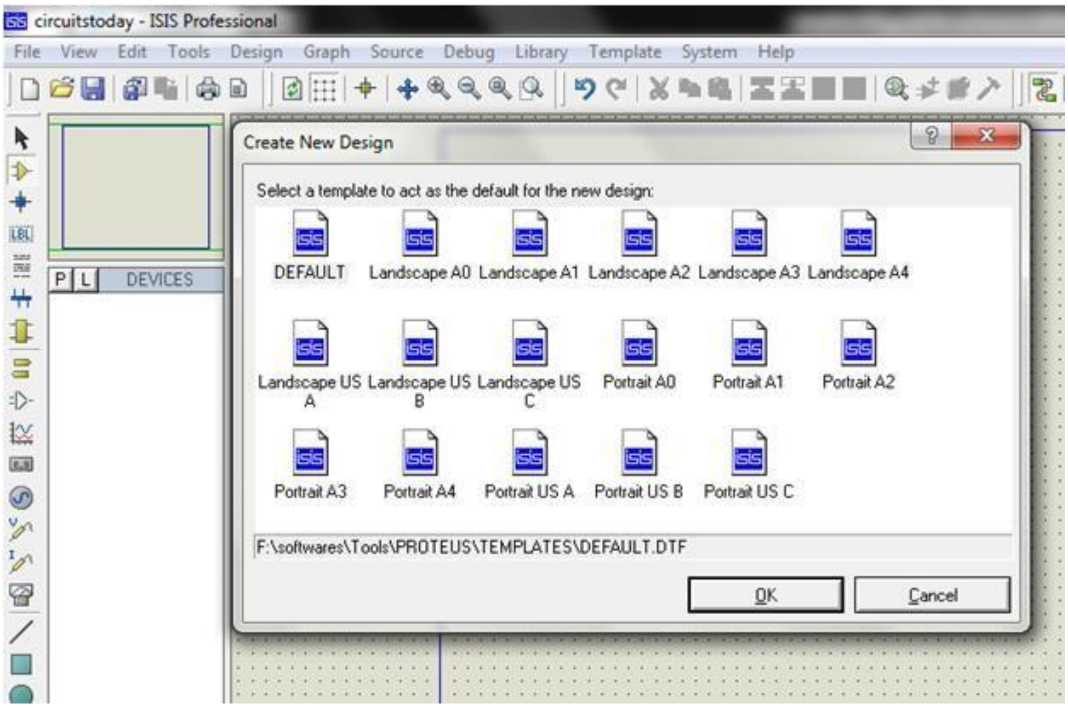


Figure 4.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

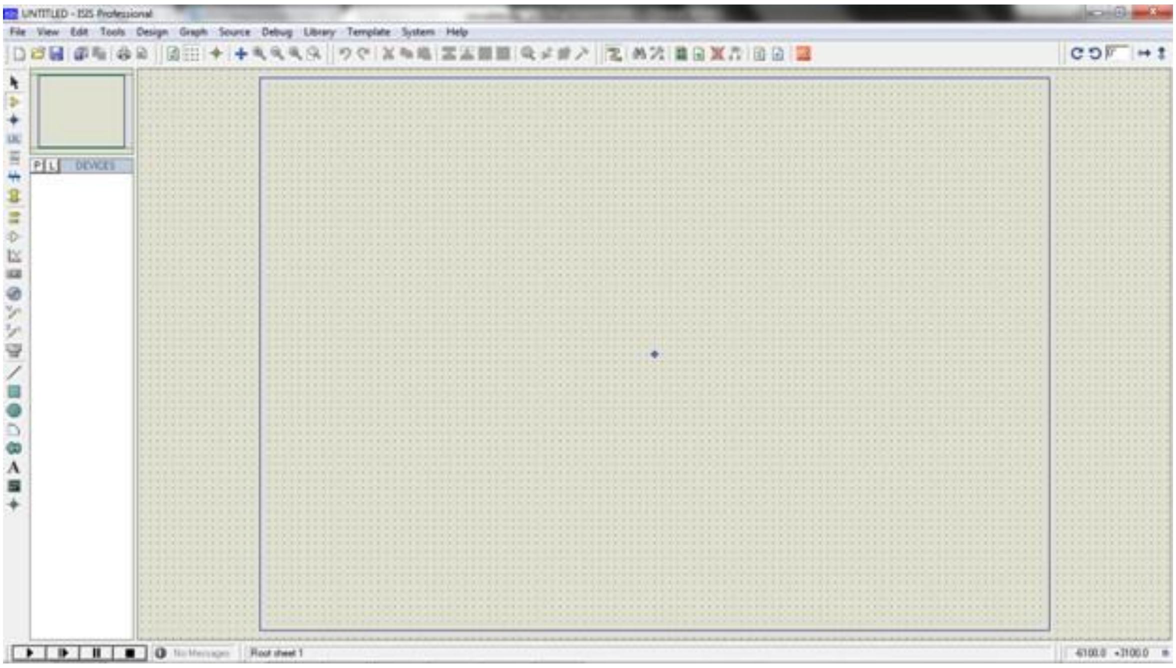


Figure 4.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

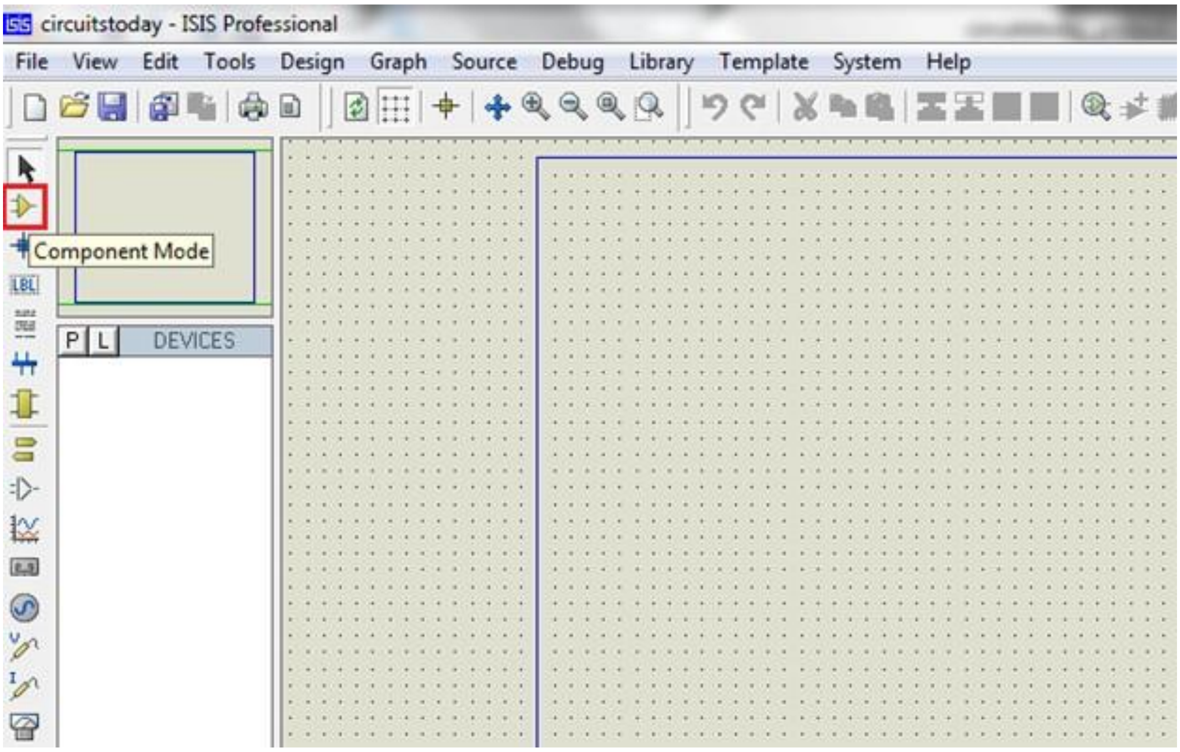


Figure 4.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

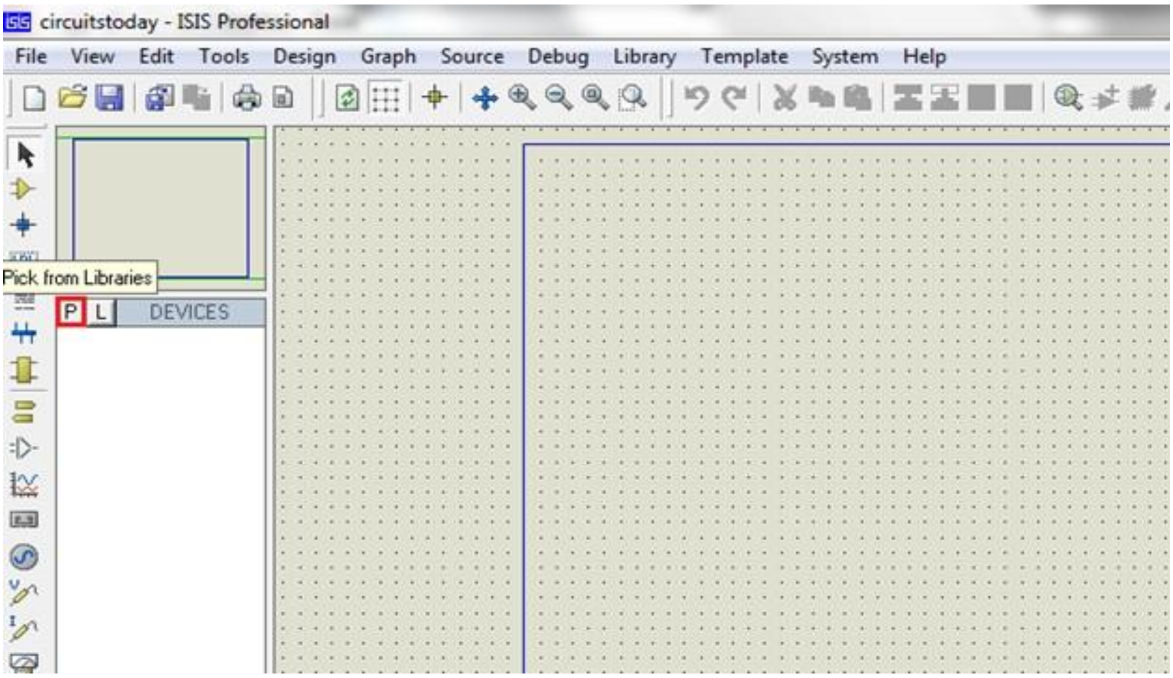


Figure 4.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

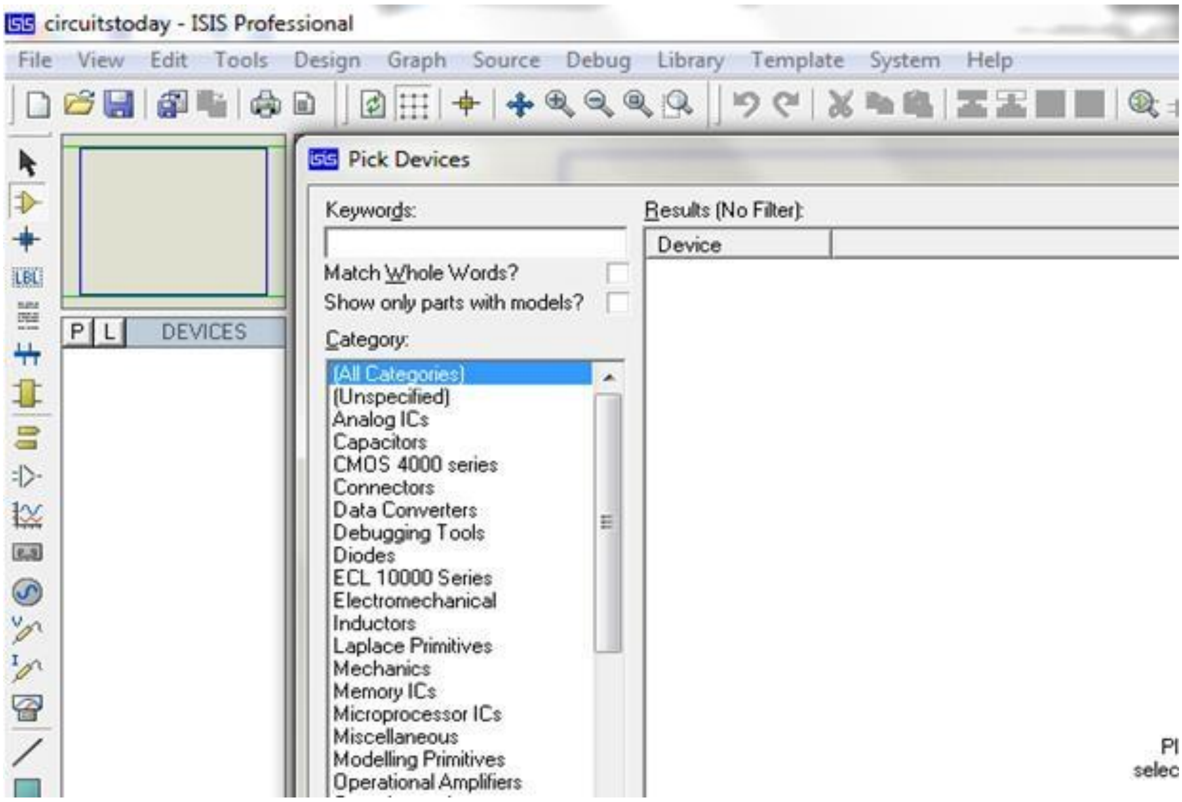


Figure 4.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

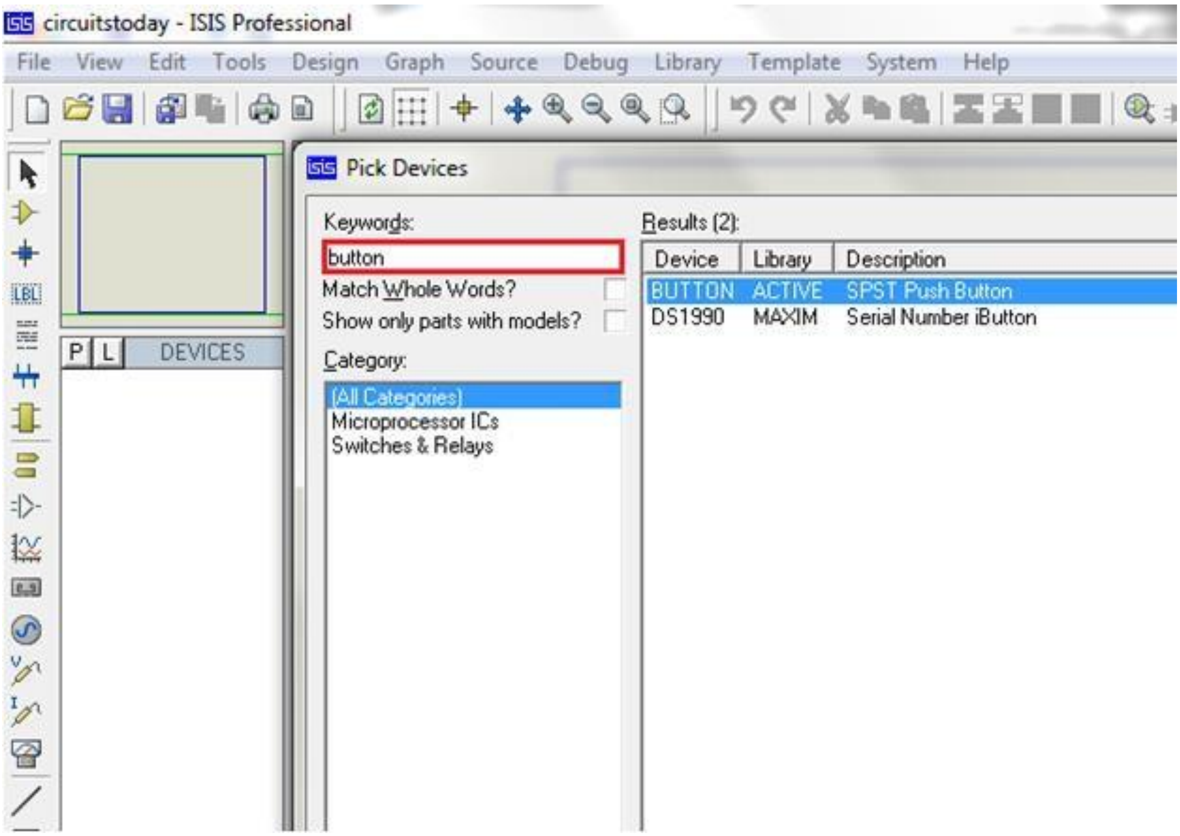


Figure 4.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

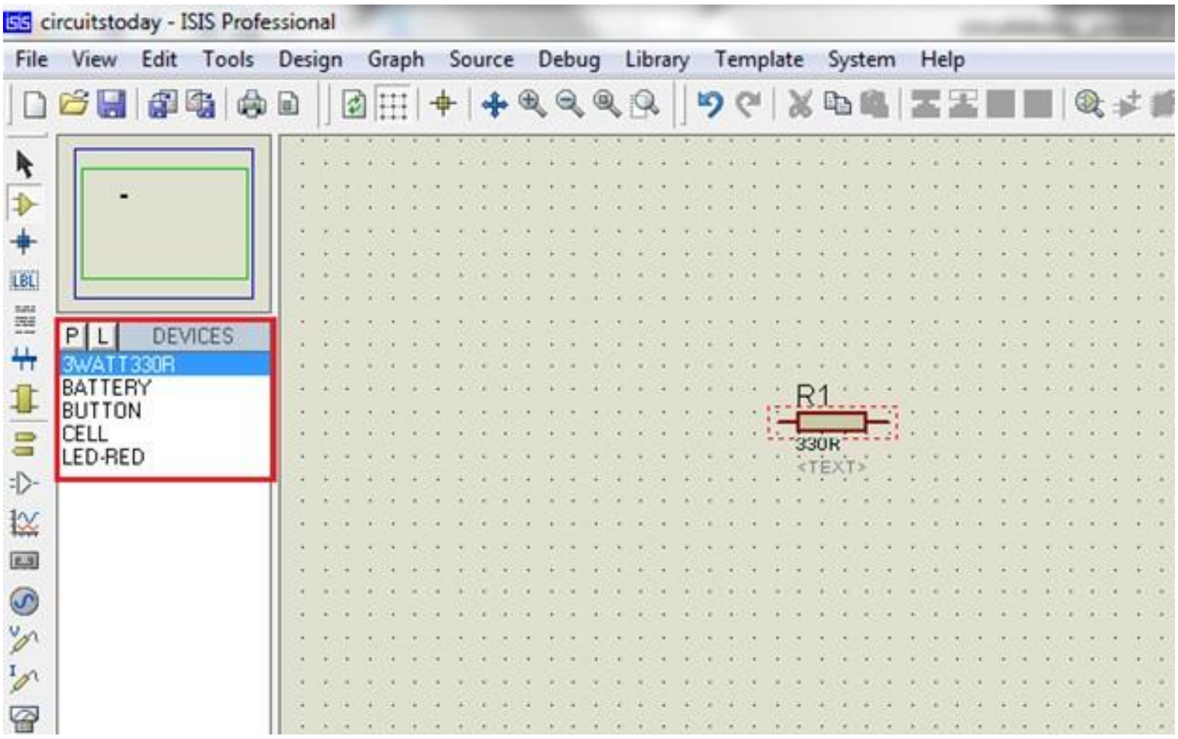


Figure 4.8 Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

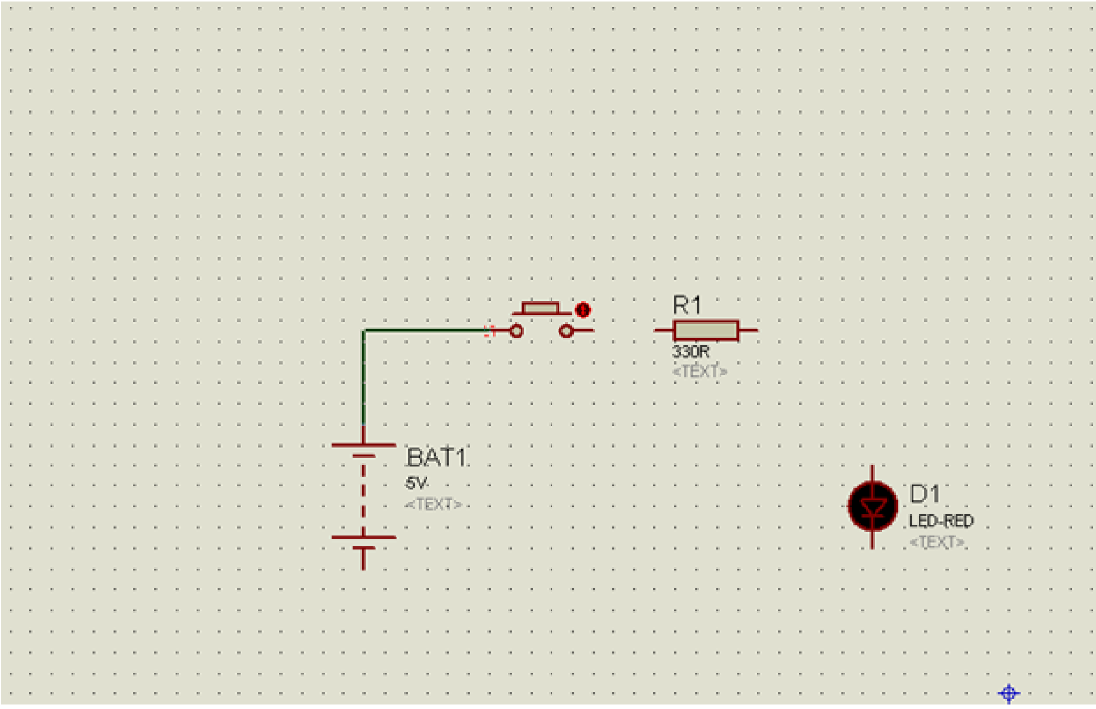


Figure 4.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

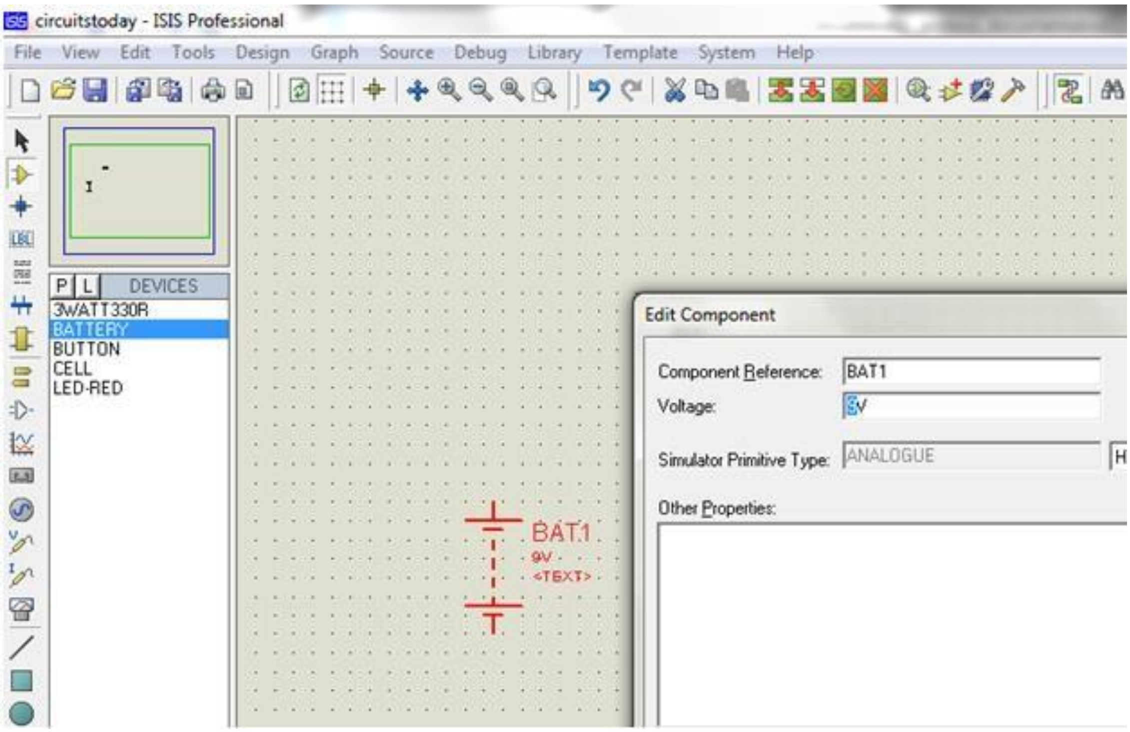


Figure 4.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

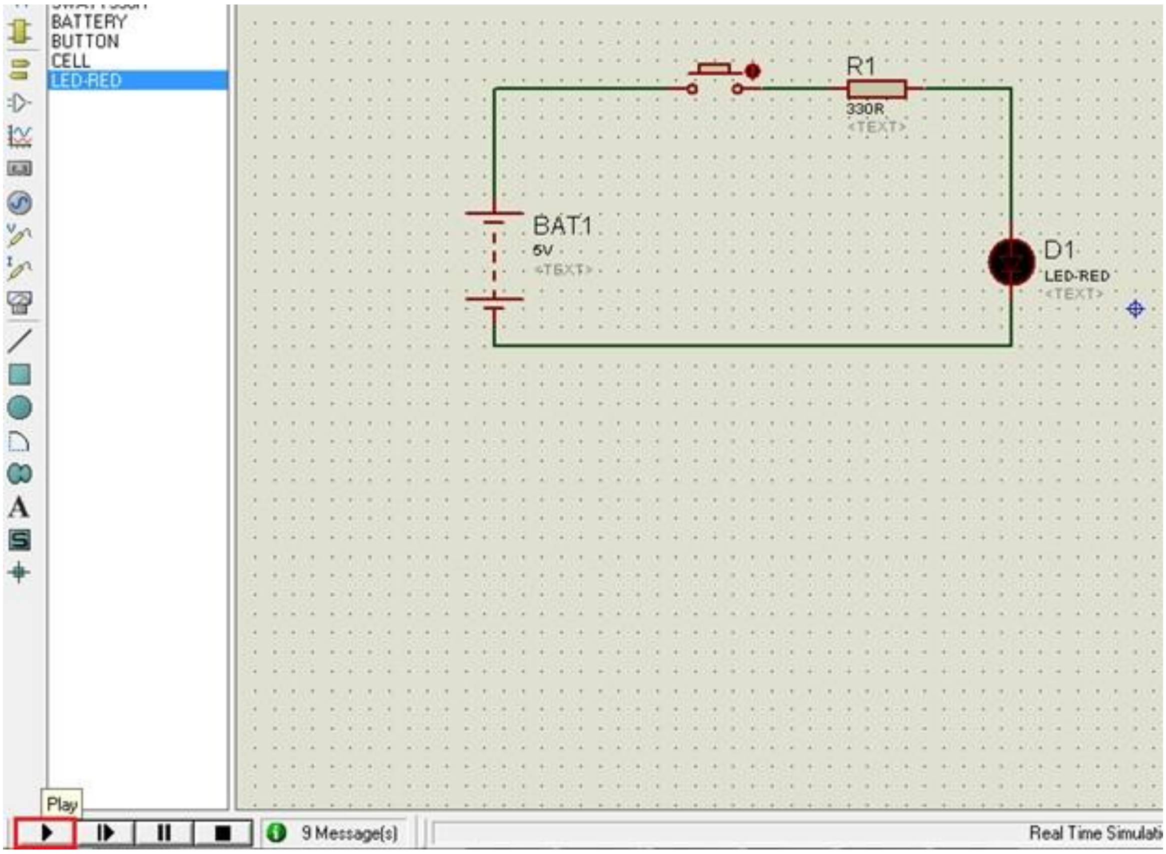


Figure 4.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

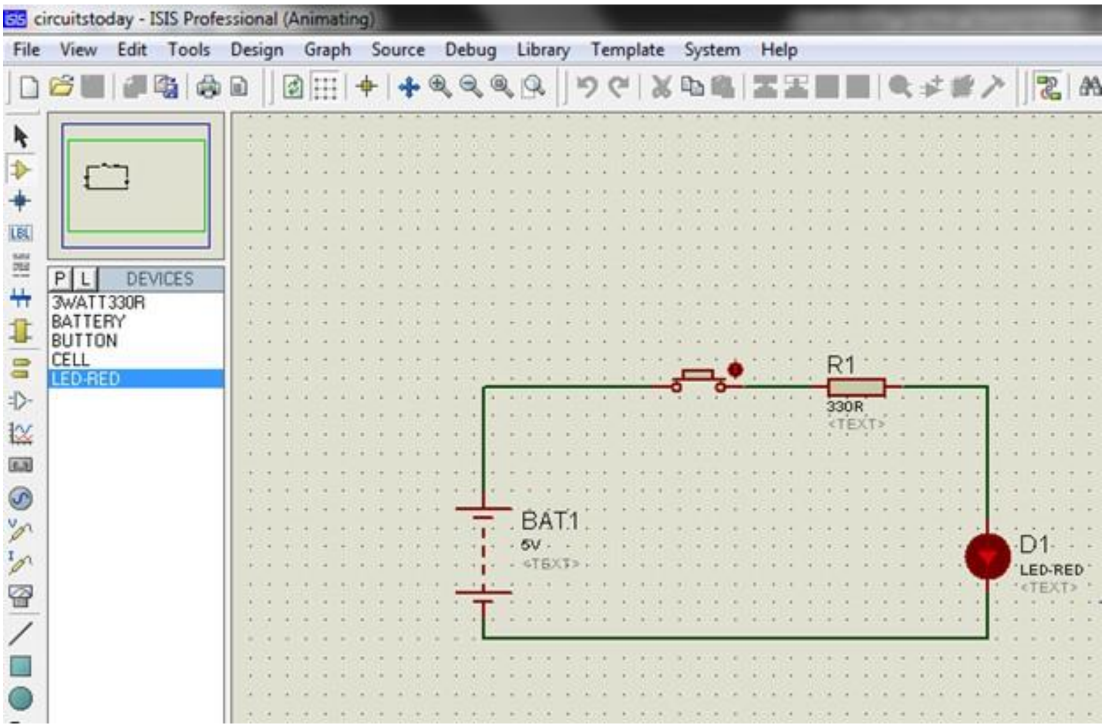


Figure 4.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

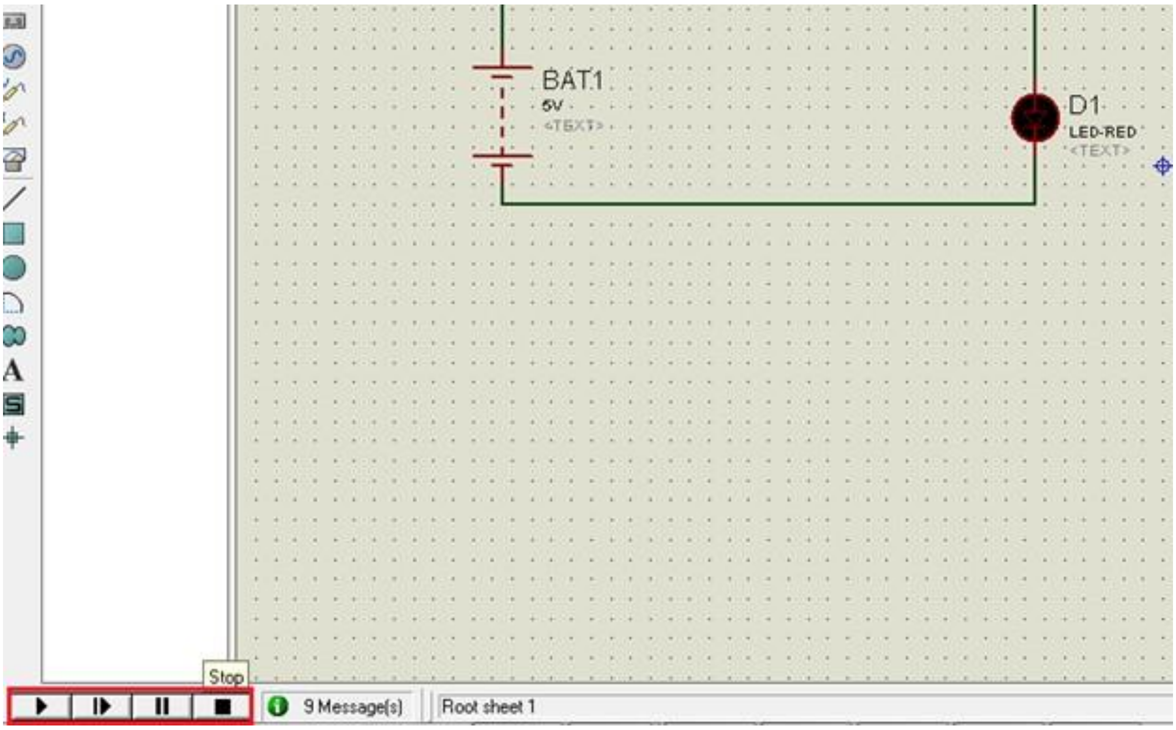


Figure 4.13 Simulation Step-Pause-Stop Buttons

4.3 Arduino IDE

The **Arduino Integrated Development Environment (IDE)** is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Arduino Pro IDE	
Developer(s)	Arduino Software
Preview release	v0.0.2 / 28 October 2019; 3 months ago[8]
Repository	github.com/arduino/Arduino
Written in	C, C++
Operating system	Windows, macOS, Linux
Platform	IA-32, x86-64, ARM
Type	Integrated development environment
License	LGPL or GPL license
Website	blog.arduino.cc/2019/10/18/arduino-pro-ide-alpha-preview-with-advanced-features/

Table 4.1 Arduino Pro Ide

In October 2019 the Arduino organization began providing early access to a new Arduino Pro IDE with debugging and other advanced features. After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

4.3.1 Installation

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Figure 4.14 USB Cable

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Figure 4.15 USB Mini B Cable

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

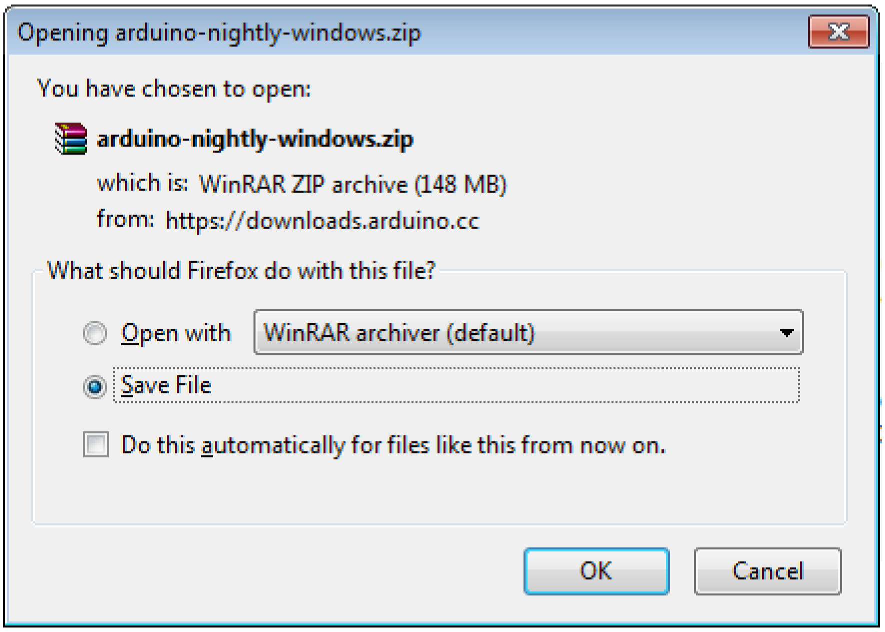


Figure 4.16 Arduino Window

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

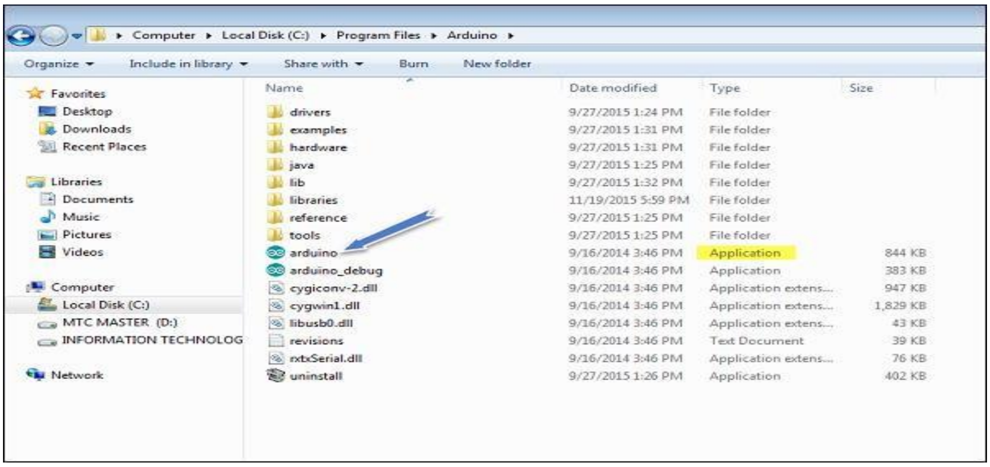


Figure 4.17 Local Disk

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.

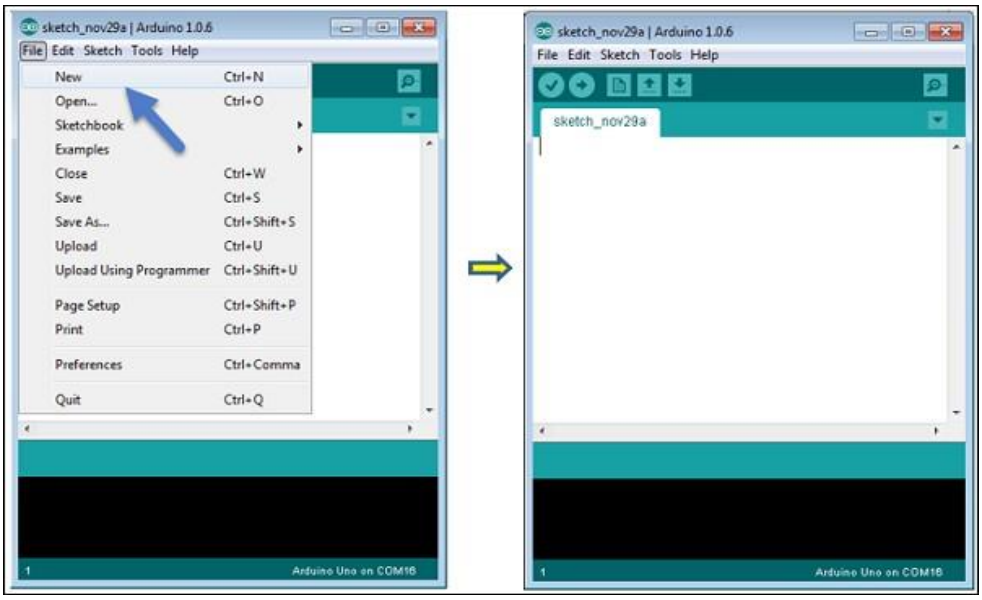


Figure 4.18 To create file

To open an existing project example, select File → Example → Basics → Blink.

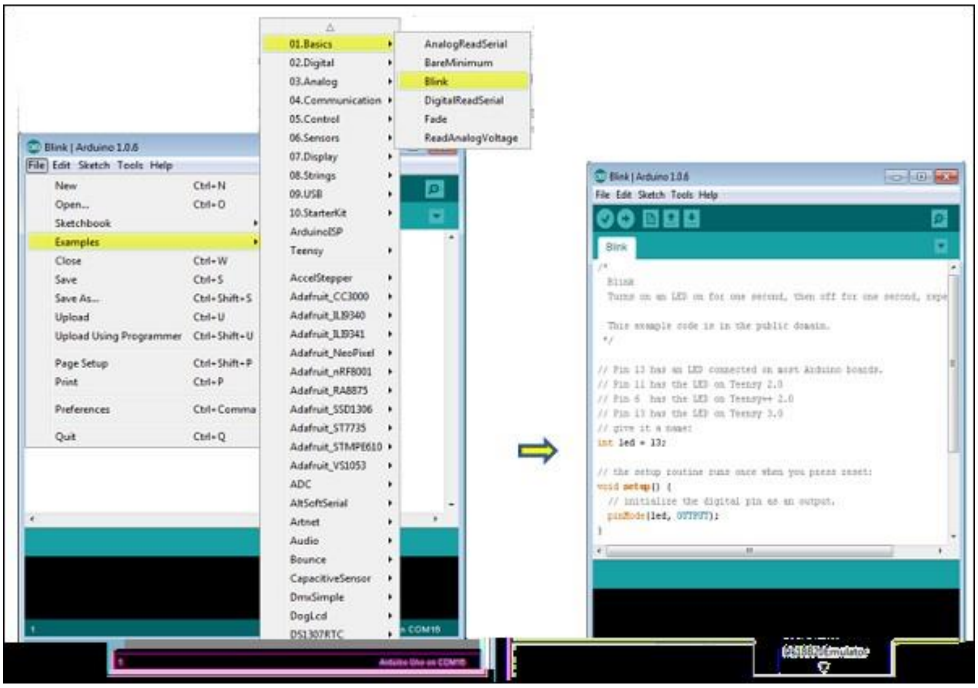


Figure 4.19 File Details

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

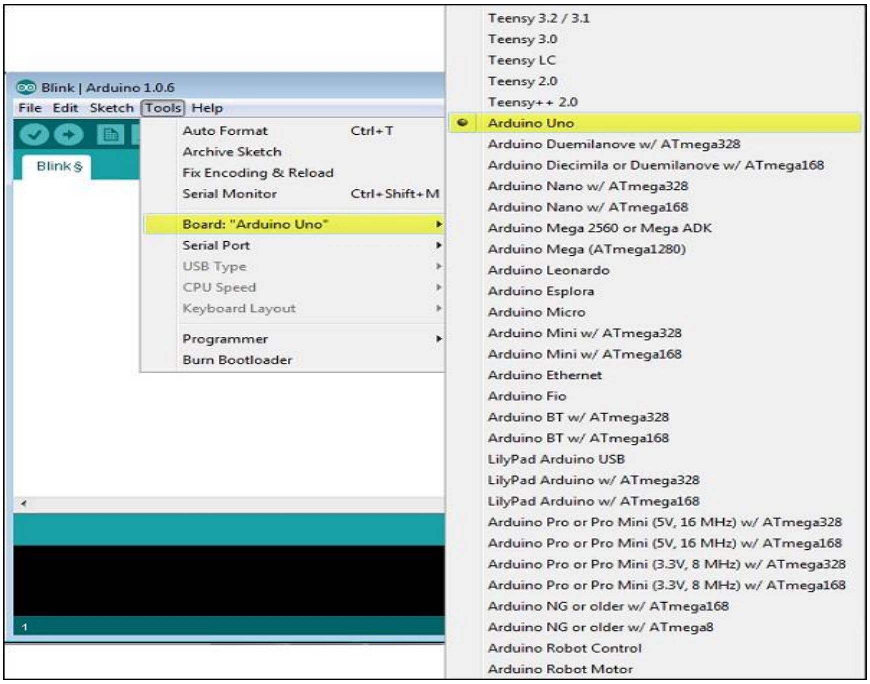


Figure 4.20 Select your arduino board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

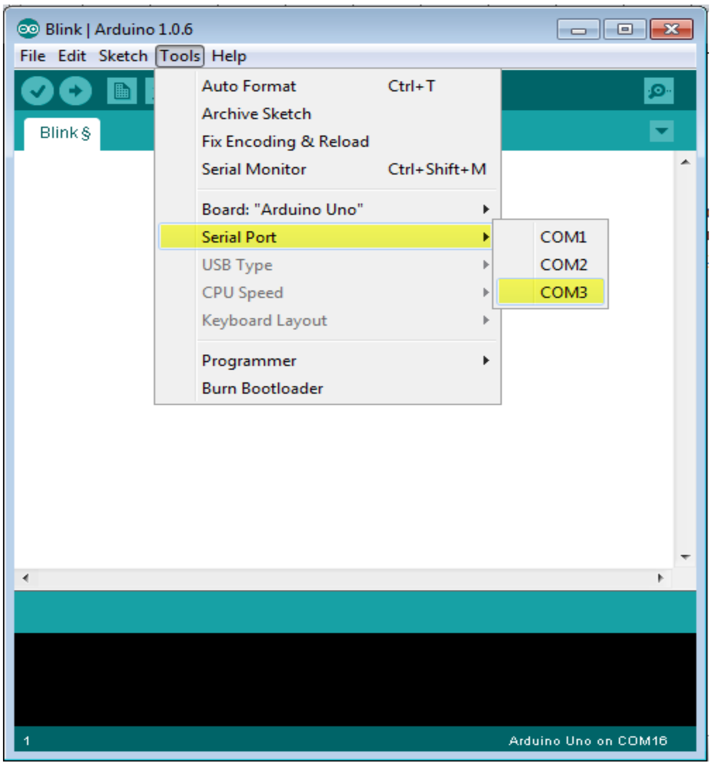


Figure 4.21 Select your serial port

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

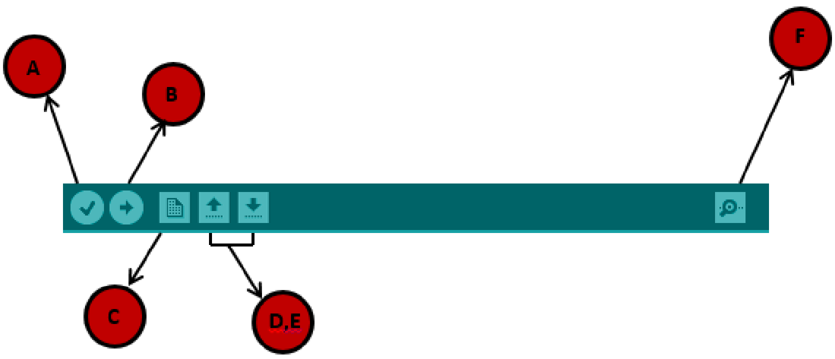


Figure 4.22 Upload the program

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

CHAPTER 5

HARDWARE

5.1 Arduino

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

5.1.1 Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Table 5.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header

LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 5.2Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 5.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 5.4 Arduino boards based on AT91SAM3X8E microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

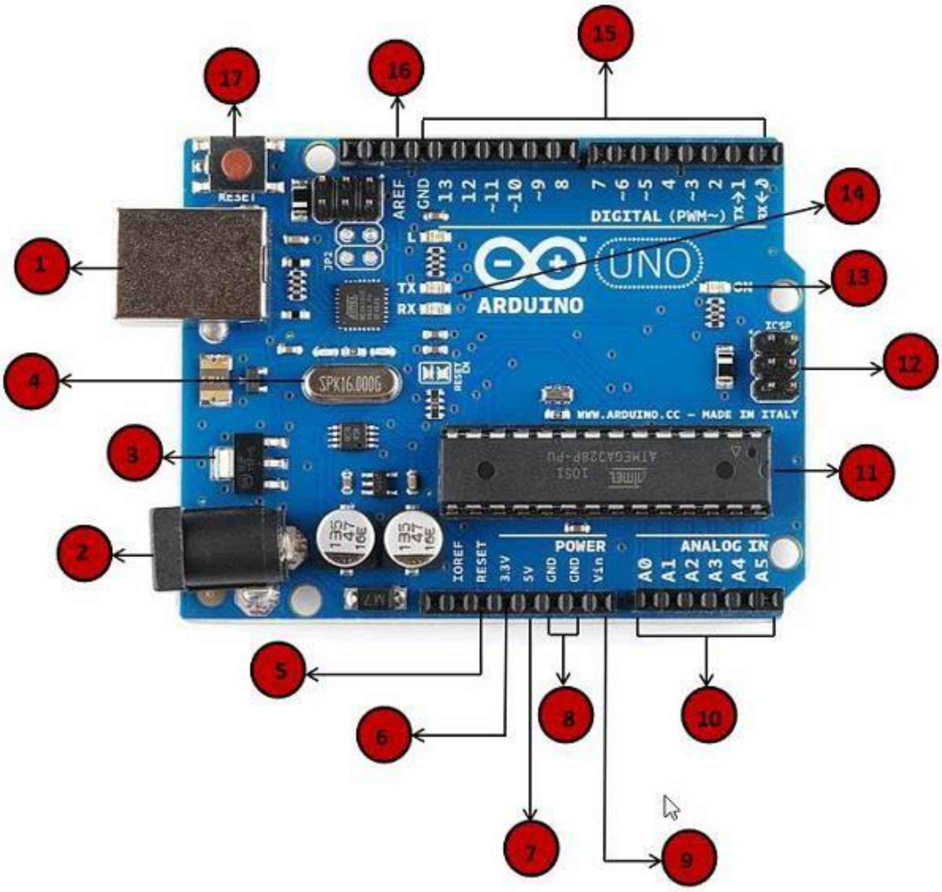


Figure 5.1 Arduino Board

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other

	elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.

12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

5.2 Liquid Cristal Display

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCDs connected to the contollers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines(RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

5.2.1 Description Of 16x2:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all-Parallel Ports. It however doesn't show the use of the Status Port as an input. So what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

16 x 2 Alphanumeric LCD Module Features:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

16 x 2 Alphanumeric LCD Module Specifications:

Pin	Symbol	Level	Function
1	V_{SS}	-	Power, GND
2	V_{DD}	-	Power, 5V
3	Vo	-	Power, for LCD Drive
4	RS	H/L	Register Select Signal H: Data Input L: Instruction Input
5	R/W	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)
6	E	H,H->L	Enable
7-14	DB0-DB7	H/L	Data Bus; Software selectable 4- or 8-bit mode
15	NC	-	**NOT CONNECTED**
16	NC	-	**NOT CONNECTED**

Table 5.5 LCD Display Pins

5.3 FEATURES:

• 5 x 8 dots with cursor

• Built-in controller (KS 0066 or Equivalent)

• + 5V power supply (Also available for + 3V)

• B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)

• N.V. optional for + 3V power supply

Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are:

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

Figure 5.2 Address locations for a 1x16 line LCD

Even limited to character based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8,16,20,24,32 and 40 characters are all standard, in one, two and four line versions.

Several different LC technologies exists. “supertwist” types, for example, offer Improved contrast and viewing angle over the older “twisted nematic” types. Some modules are available with back lighting, so that they can be viewed in dimly-lit conditions. The back lighting may be either “electro-luminescent”, requiring a high voltage inverter circuit, or simple LED illumination.

5.4 PIN DESCRIPTION:

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections).

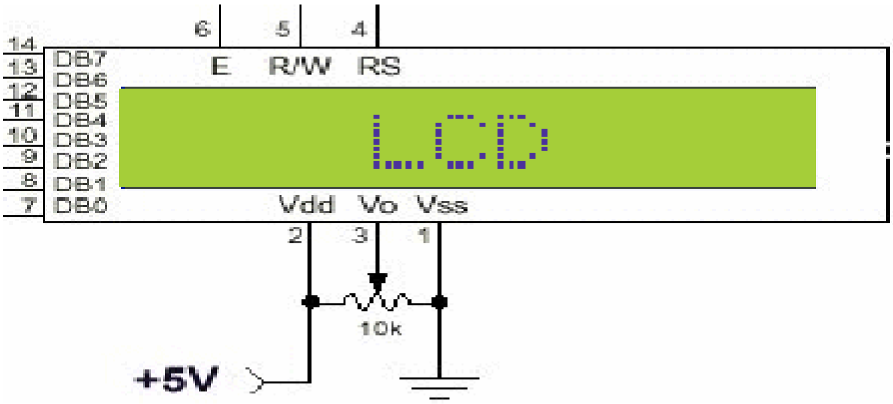


Figure 5.3 Pin diagram of 1x16 lines LCD

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Figure 5.4 Pin specifications

CONTROL LINES:

EN: Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS: Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW: Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands, so RW will almost always be low. Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Logic status on control lines:

• E - 0 Access to LCD disabled

- 1 Access to LCD enabled

• R/W - 0 Writing data to LCD

- 1 Reading data from LCD

• RS - 0 Instructions

- 1 Character

Writing data to the LCD:

1) Set R/W bit to low

2) Set RS bit to logic 0 or 1 (instruction or character)

3) Set data to data lines (if it is writing)

4) Set E line to high

5) Set E line to low

Read data from data lines (if it is reading) on LCD:

1) Set R/W bit to high

2) Set RS bit to logic 0 or 1 (instruction or character)

3) Set data to data lines (if it is writing)

4) Set E line to high

5) Set E line to low

Entering Text:

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the d.i.l. switch pack with a of re-wiring is necessary.

LCD Commands:

There are some present commands instructions in LCD, which we need to send to LCD through some microcontroller. Some important command instructions are given below:

Table 5.6 Commands

Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left
0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

Rain Drop Sensor

Raindrop Sensor is a tool used for sensing rain. It consists of two modules, a **rain board** that detects the rain and a **control module**, which compares the analog value, and converts it to a digital value. The raindrop sensors can be used in the automobile sector to control the windshield wipers automatically, in the agriculture sector to sense rain and it is also used in home automation systems.

Pin Configuration of Rain Sensor:

S.No:	Name	Function
1	VCC	Connects supply voltage- 5V
2	GND	Connected to ground
3	D0	Digital pin to get digital output
4	A0	Analog pin to get analog output

Raindrop Sensor Features:

- Working voltage 5V
- Output format: Digital switching output (0 and 1), and analog voltage output AO
- Potentiometer adjust the sensitivity
- Uses a wide voltage LM393 comparator
- Comparator output signal clean waveform is good, driving ability, over 15mA
- Anti-oxidation, anti-conductivity, with long use time
- With bolt holes for easy installation
- Small board PCB size: 3.2cm x 1.4cm

Note: The complete technical details can be found in the **Rain Sensor datasheet** given at the bottom of this page.

How to use Raindrop sensor:

Interfacing the raindrop sensor with a microcontroller like 8051, Arduino, or PIC is simple. The rain board module is connected with the control module of the raindrop sensor as shown in the below diagram.

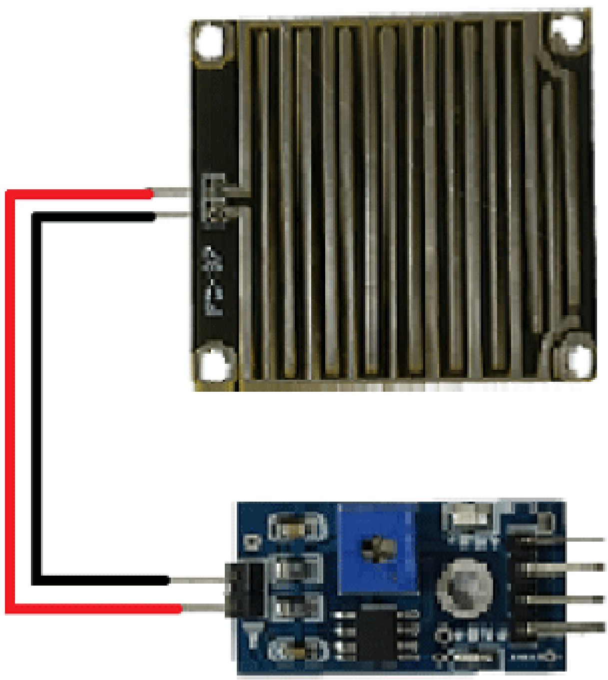


Figure 5.6 Raindrop Sensor

The control module of the raindrop sensor has 4 outputs. VCC is connected to a 5V supply. The GND pin of the module is connected to the ground. The D0 pin is connected to the digital pin of the microcontroller for digital output or the analog pin can be used. To use the analog output, the A0 pin can be connected to the ADC pin of a microcontroller. In the case of Arduino, it has 6 ADC pins, so we can use any of the 6 pins directly without using an ADC converter. The sensor module consists of a potentiometer, LN393 comparator, LEDs, capacitors and resistors. The pinout image above shows the components of the control module. The rain board module consists of copper tracks, which act as a **variable resistor**. Its resistance varies with respect to the wetness on the rain board. The below fig shows the rain board module.

Figure 5.7 raindrop sensor module

The circuit diagram of a raindrop sensor module is given below.

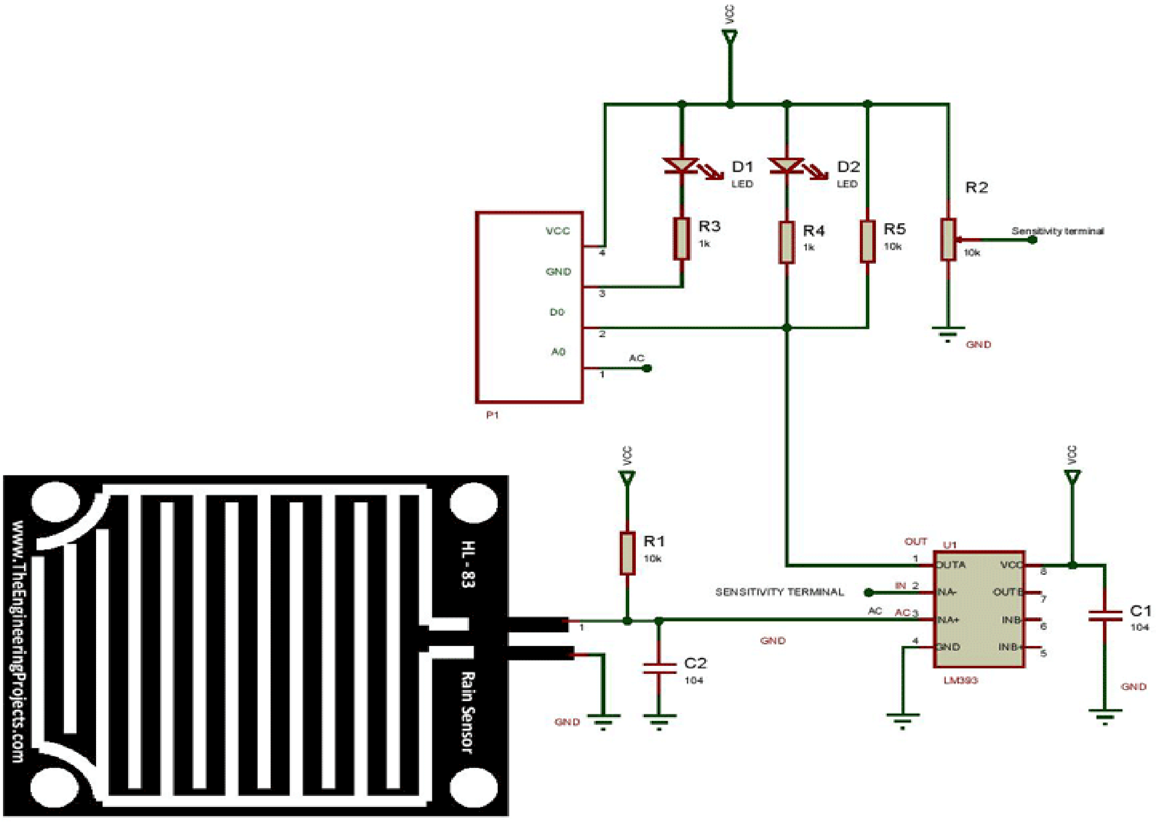


Figure 5.8 Raindrop sensor module circuit diagram

As shown in the above figure, the R1 resistor and the rain board module will act as a **voltage divider**. Capacitors C1 and C2 are used as a biasing element. The input for the Non-inverting terminal is taken from the connection point of the R1, and rain board module. Another point is taken from this connection and connected to the A0 terminal of the control module.The input to the inverting terminal of the LM393 is taken from the potentiometer (R2). The R2 resistor acts as a voltage divider, and by varying R2 we can vary the input voltage to the inverting terminal, which in turn affects the sensitivity of the control module. The connections are shown in the above fig. The resistors R3 and R4 will act as current limiting resistors, while resistor R5 will act as a pull-up resistor to keep the bus in a high state when not in use.

Working of Rain Sensor:

Case1: When the input of the inverting terminal is higher than the input of the non-inverting terminal.

Case2: If the input of the inverting terminal is lower than the input of the non-inverting terminal.

The input to the inverting terminal is set to a certain value by varying the potentiometer and the sensitivity is set. When the rain board module's surface is exposed to rainwater, the surface of the rain board module will be wet, and it offers minimum resistance to the supply voltage. Due to this, the minimum voltage will be appearing at the non-inverting terminal of LM393 Op-Amp. The comparator compares both inverting and non-inverting terminal voltages. If the condition falls under case (1), the output of the Op-Amp will be digital LOW. If the condition falls under case (2), the output of the Op-Amp will be digital HIGH. The below diagram shows the equivalent circuit of both the conditions.

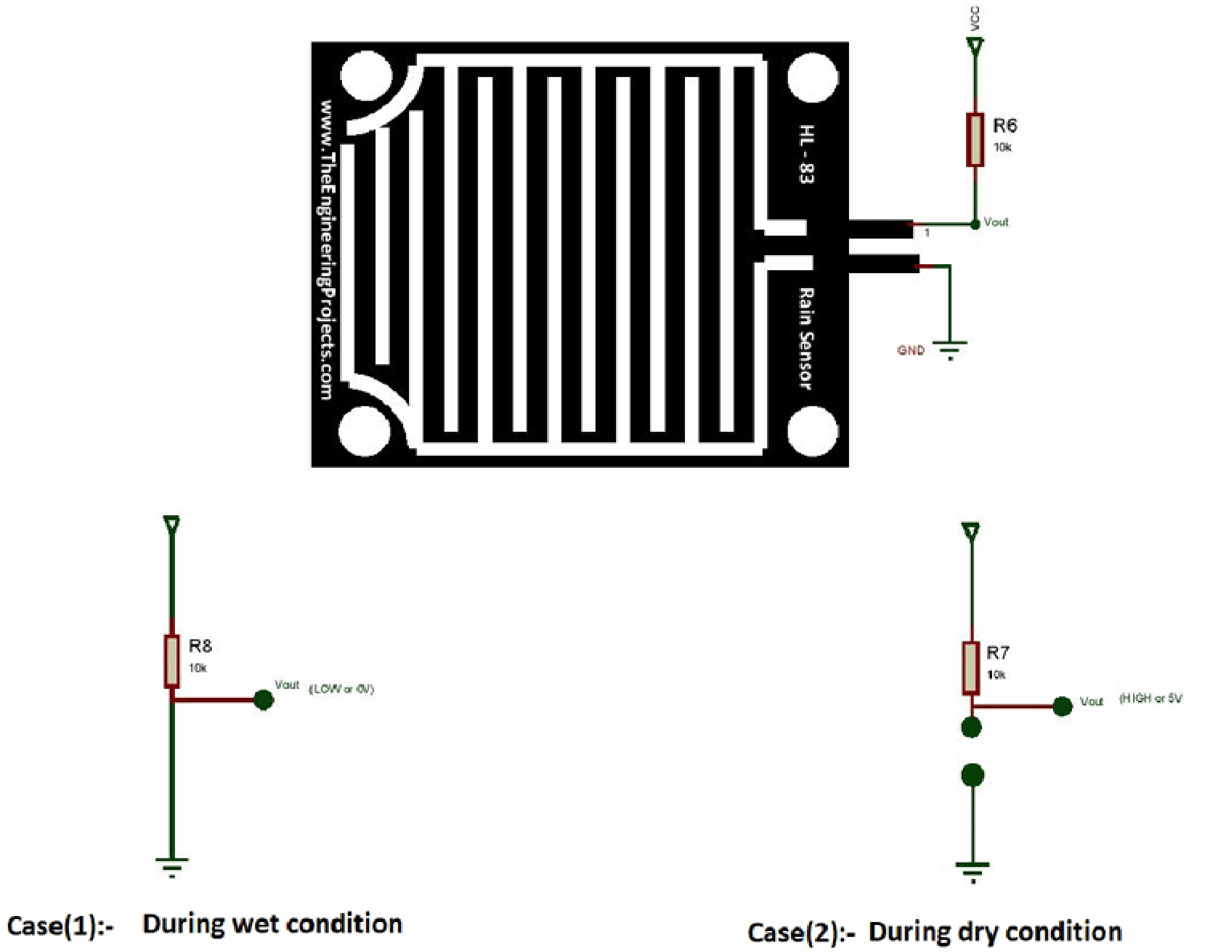


Figure 5.9 Sensor conditions

When the A0 pin is connected to the microcontroller, an additional analog to digital converter (ADC) circuit is used. In the case of Arduino, it consists of 6 ADC pins, which can be directly used for calculation purposes.

Applications of Rain sensor:

- Automatic windshield wipers
- Smart Agriculture

- Home-Automation

L293D

L293D is basically a high current dual motor driver/controller Integrated Circuit (IC). It is able to drive load having current up to 1A at the voltage ranging from 4.5V to 36V. Motor driver usually act as current amplifier because they receive a low current signal as an input and provides high current signal at the output.

Motors usually operates on this higher current. L-293D has to builtin H-Bridge driver circuits and is able to control two DC motors at a time in both clockwise and counter clockwise direction. It has two enable pins and they should be kept high in order to control the motor. By changing the polarity of applied signal motor can be rotated in either clockwise or counter clockwise direction. If L 293D enable pin is high, its corresponding driver will provide the desired out. If the enable pin is low, there will be no output. L-293D has different features including internal ESD protection, large voltage supply range, large output current per channel, high noise immunity input etc. L 293D plays a vital role in electronics era and has several different applications e.g relay drivers, DC motor drivers, stepping motor drivers etc. The further detail about L 293D motor driver/controller will be given later in this tutorial.

Introduction to L293D

L293D is basically a motor driver or controller. It has two builtin H-bridge circuits which are able to control two DC motors simultaneously in both clockwise and counter clockwise direction. It acts as an current high amplifier because it take low current signal at its input and provides higher current signal at the output in order to drives different load e.g stepper motor & DC motors. Its features include large inpu voltage supply range, large output current high noise immunity input signals etc. Its common real life applications include stepping motor drivers, relay drivers, DC motor drivers etc. L-293D motor driver/controller is shown in the figure given below.

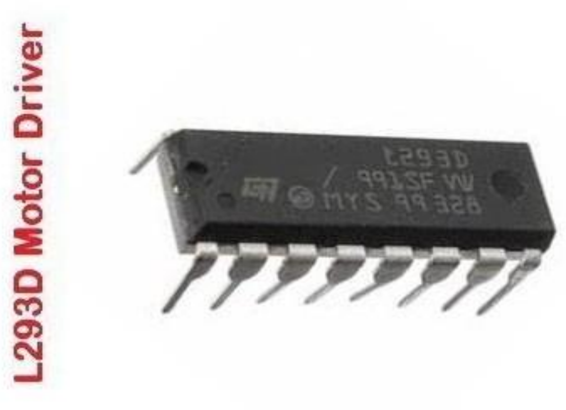


Figure 5.10 L293D Motor Driver

1. L293D Pins

- L-293D has sixteen (16) pins, having different individual functions.
- All of these sixteen pins along with their serial no, are given in the table shown below.

2. L293D Pin Functions

- L 293D each pin has different tasks to perform when they are in working condition.
- The tasks assigned to each pin are given in the table shown below.

L293D Pins	
Pin. No	Pin Name
1	Enable 1,2
2	Input 1
3	Output 1
4	GND
5	GND
6	Output 2
7	Input 2
8	Vcc2
9	Enable 3,4
10	Input 3
11	Output 3
12	GND
13	GND
14	Output put 4
15	Input 4
16	Vcc1

Table 5.7 L293D Pins

3. L293D Pinout

- Pinout diagram of any device presents the pins configuration through a completely labelled diagram.
- L293D pinout diagram is shown in the figure given below.

Figure 5.11 L293D Pinout

DC Motors

The brushed DC motor is one of the earliest motor designs. Today, it is the motor of choice in the majority of variable speed and torque control applications.

Advantages

- Easy to understand design
- Easy to control speed
- Easy to control torque
- Simple, cheap drive design

Easy to understand design

The design of the brushed DC motor is quite simple. A permanent magnetic field is created in the stator by either of two means:

- Permanent magnets
- Electro-magnetic windings

If the field is created by permanent magnets, the motor is said to be a "permanent magnet DC motor" (PMDC). If created by electromagnetic windings, the motor is often said to be a "shunt wound DC motor" (SWDC). Today, because of cost-effectiveness and reliability, the PMDC motor is the motor of choice for applications involving fractional horsepower DC motors, as well as most applications up to about three horsepower.At five horsepower and greater, various forms of the shunt wound DC motor are most commonly used. This is because the electromagnetic windings are more cost effective than permanent magnets in this power range.

Caution: If a DC motor suffers a loss of field (if for example, the field power connections are broken), the DC motor will immediately begin to accelerate to the top speed which the loading will allow. This can result in the motor flying apart if the motor is lightly loaded. The possible loss of field must be accounted for, particularly with shunt wound DC motors.Opposing the stator field is the armature field, which is generated by a changing electromagnetic flux coming from windings located on the rotor. The magnetic poles of the armature field will attempt to line up with the opposite magnetic poles generated by the stator field. If we stopped the design at this point, the motor would spin until the poles were opposite one another, settle into place, and then stop -- which would make a pretty useless motor!However, we are smarter than that. The section of the rotor where the electricity enters the rotor windings is called the commutator. The electricity is carried between the rotor and the stator by conductive graphite-copper brushes (mounted on the rotor) which contact rings on stator. Imagine power is supplied:The motor rotates toward the pole alignment point. Just as the motor would get to this point, the brushes jump across a gap in the stator rings. Momentum carries the motor forward over this gap. When the brushes get to the other side of the gap, they contact the stator rings again and -- the polarity of the voltage is reversed in this set of rings! The motor begins accelerating again, this time trying to get to the opposite set of poles. (The momentum has carried the motor past the original pole alignment point.) This continues as the motor rotates.In most DC motors, several sets of windings or permanent magnets are present to smooth out the motion.

Easy to control speed

Controlling the speed of a brushed DC motor is simple. The higher the armature voltage, the faster the rotation. This relationship is linear to the motor's maximum speed.

The maximum armature voltage which corresponds to a motor's rated speed (these motors are usually given a rated speed and a maximum speed, such as 1750/2000 rpm) are available in certain standard voltages, which roughly increase in conjuntion with horsepower. Thus, the smallest industrial motors are rated 90 VDC and 180 VDC. Larger units are rated at 250 VDC and sometimes higher. Specialty motors for use in mobile applications are rated 12, 24, or 48 VDC. Other tiny motors may be rated 5 VDC.Most industrial DC motors will operate reliably over a speed range of about 20:1 -- down to about 5-7% of base speed. This is much better performance than the comparible AC motor. This is partly due to the simplicity of control, but is also partly due to the fact that most industrial DC motors are designed with variable speed operation in mind, and have added heat dissipation features which allow lower operating speeds.

Easy to control torque

In a brushed DC motor, torque control is also simple, since output torque is proportional to current. If you limit the current, you have just limited the torque which the motor can achieve. This makes this motor ideal for delicate applications such as textile manufacturing.Simple, cheap drive designThe result of this design is that variable speed or variable torque electronics are easy to design and manufacture. Varying the speed of a brushed DC motor requires little more than a large enough potentiometer. In practice, these have been replaced for all but sub-fractional horsepower applications by the SCR and PWM drives, which offer relatively precisely control voltage and current. Common DC drives are available at the low end (up to 2 horsepower) for under US$100 -- and sometimes under US$50 if precision is not important.Large DC drives are available up to hundreds of horsepower. However, over about 10 horsepower careful consideration should be given to the price/performance tradeoffs with AC inverter systems, since the AC systems show a price advantage in the larger systems. (But they may not be capable of the application's performance requirments).

Disadvantages

- Expensive to produce
- Can't reliably control at lowest speeds
- Physically larger
- High maintenance
- Dust

WORKING OF DC MOTOR

In any electric motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. As you are well aware of from playing with magnets as a kid, opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion.

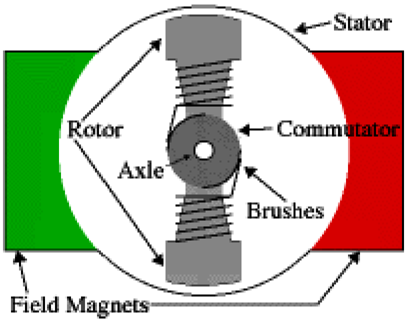


Figure 5.12 DC Motor

Principle

When a rectangular coil carrying current is placed in a magnetic field, a torque acts on the coil which rotates it continuously.When the coil rotates, the shaft attached to it also rotates and thus it is able to do mechanical work.Every DC motor has six basic parts -- axle, rotor (a.k.a., armature), stator, commutator, field magnet(s), and brushes. In most common DC motors (and all that BEAMers will see), the external magnetic field is produced by high-strength permanent magnets[1]. The stator is the stationary part of the motor -- this includes the motor casing, as well as two or more permanent magnet pole pieces. The rotor (together with the axle and attached commutator) rotate with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a common motor layout -- with the rotor inside the stator (field) .

The geometry of the brushes, commentator contacts, and rotor windings are such that when power is applied, the polarities of the energized winding and the stator magnet(s) are misaligned, and the rotor will rotate until it is almost aligned with the stator's field magnets. As the rotor reaches alignment, the brushes move to the next commentator contacts, and energize the next winding. Given our example two-pole motor, the rotation reverses the direction of current through the rotor winding, leading to a "flip" of the rotor's magnetic field, driving it to continue rotating.In real life, though, DC motors will always have more than two poles (three is a very common number). In particular, this avoids "dead spots" in the commutator. You can imagine how with our example two-pole motor, if the rotor is exactly at the middle of its rotation (perfectly aligned with the field magnets), it will get "stuck" there. Meanwhile, with a two-pole motor, there is a moment where the commutator shorts out the power supply (i.e., both brushes touch both commutator contacts simultaneously). This would be bad for the power supply, waste energy, and damage motor components as well. Yet another disadvantage of such a simple motor is that it would exhibit a high amount of torque "ripple" (the amount of torque it could produce is cyclic with the position of the rotor).

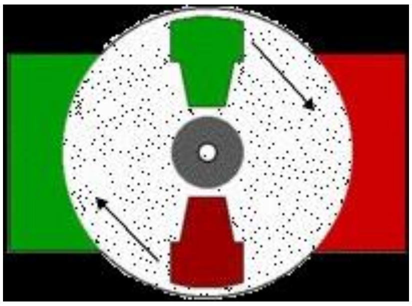


Figure 5.13 DC Motor Rotation

Construction and Working

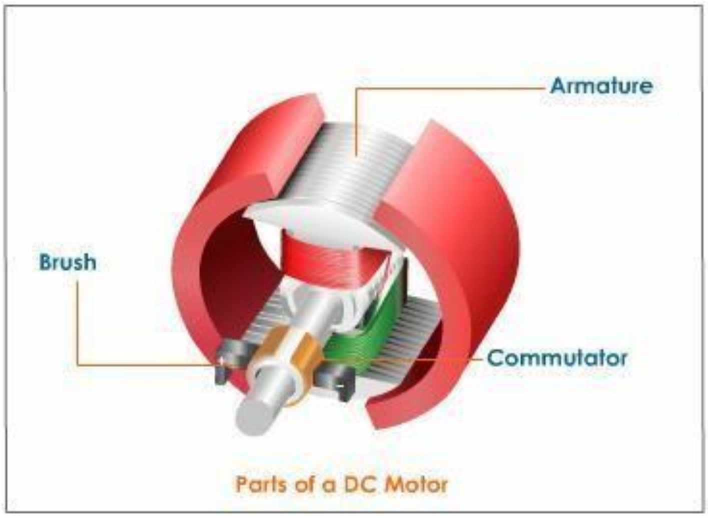


Figure 5.14 Working and construction

Parts of a DC Motor

Armature

A D.C. motor consists of a rectangular coil made of insulated copper wire wound on a soft iron core. This coil wound on the soft iron core forms the armature. The coil is mounted on an axle and is placed between the cylindrical concave poles of a magnet.

Commutator

A commutator is used to reverse the direction of flow of current. Commutator is a copper ring split into two parts C_1 and C_2. The split rings are insulated form each other and mounted on the axle of the motor. The two ends of the coil are soldered to these rings. They rotate along with the coil. Commutator rings are connected to a battery. The wires from the battery are not connected to the rings but to the brushes which are in contact with the rings.

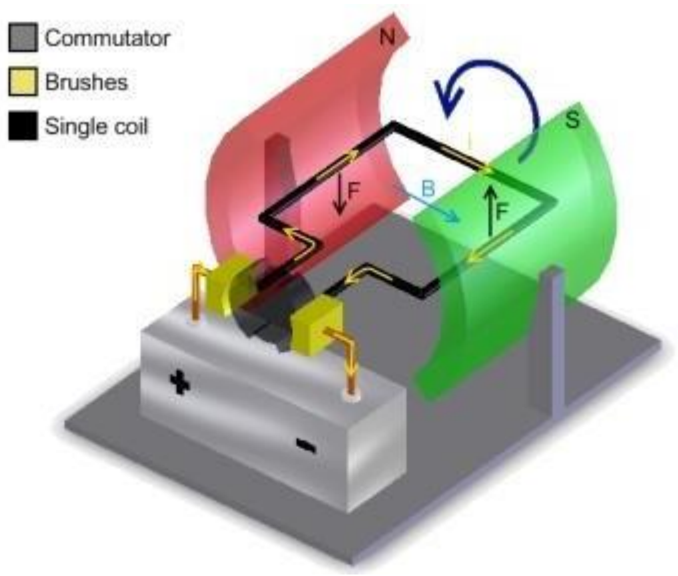


Figure 5.15 DC Motor parts

Brushes

Two small strips of carbon, known as brushes press slightly against the two split rings, and the split rings rotate between the brushes.

The carbon brushes are connected to a D.C. source.

Working of a DC Motor

When the coil is powered, a magnetic field is generated around the armature. The left side of the armature is pushed away from the left magnet and drawn towards the right, causing rotation.

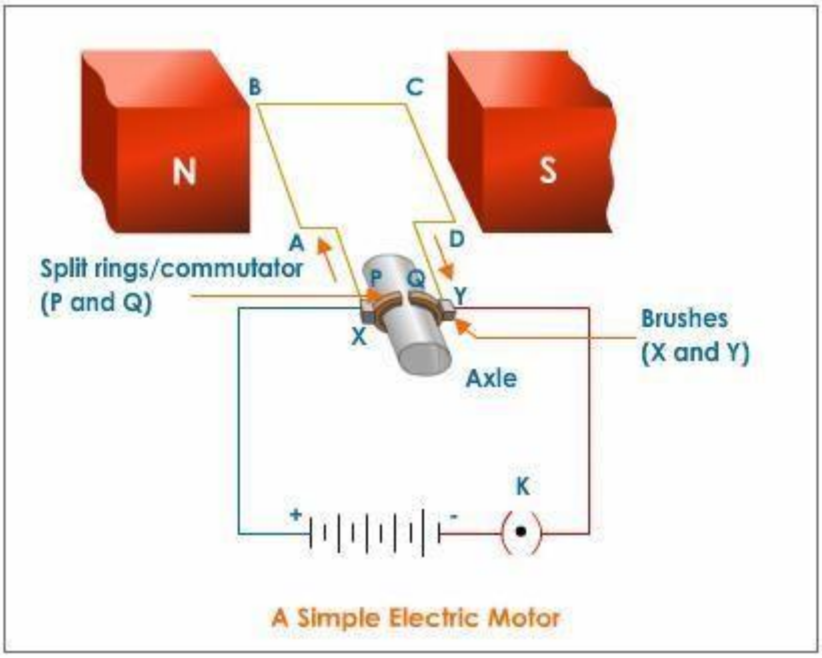


Figure 5.16 Working of a DC Motor

When the coil turns through 90^0, the brushes lose contact with the commutator and the current stops flowing through the coil.

Now when the coil turns through 180^0, the sides get interchanged. As a result the commutator ring C_1 is now in contact with brush B_2 and commutator ring C_2 is in contact with brush B_1. Therefore, the current continues to flow in the same direction.

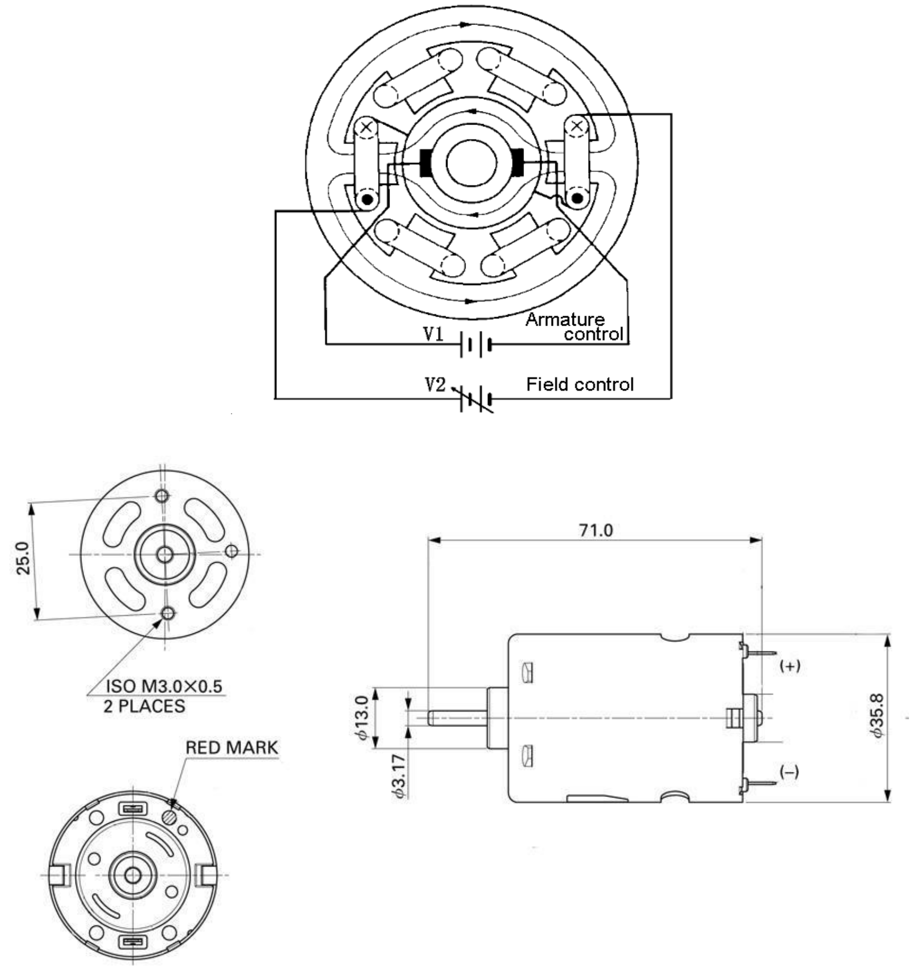


Figure 5.17 Rotation of Motor

PARAMETRS OF THE DC MOTRS

1. Direction of rotation
2. Motor Speed
3. Motor Torque
4. Motor Start and Stop

Direction of Rotation

A DC Motor has two wires. We can call them the positive terminal and the negative terminal, although these are pretty much arbitrary names (unlike a battery where these polarities are vital and not to be mixed!). On a motor, we say that when the + wire is connected to + terminal on a power source, and the - wire is connected to the - terminal source on the same power source, the motor rotates clockwise (if you are looking towards the motor shaft). If you reverse the wire polarities so that each wire is connected to the opposing power supply terminal, then the motor rotates counter clockwise. Notice this is just an arbitrary selection and that some motor manufacturers could easily choose the opposing convention. As long as you know what rotation you get with one polarity, you can always connect in such a fashion that you get the direction that you want on a per polarity basis.

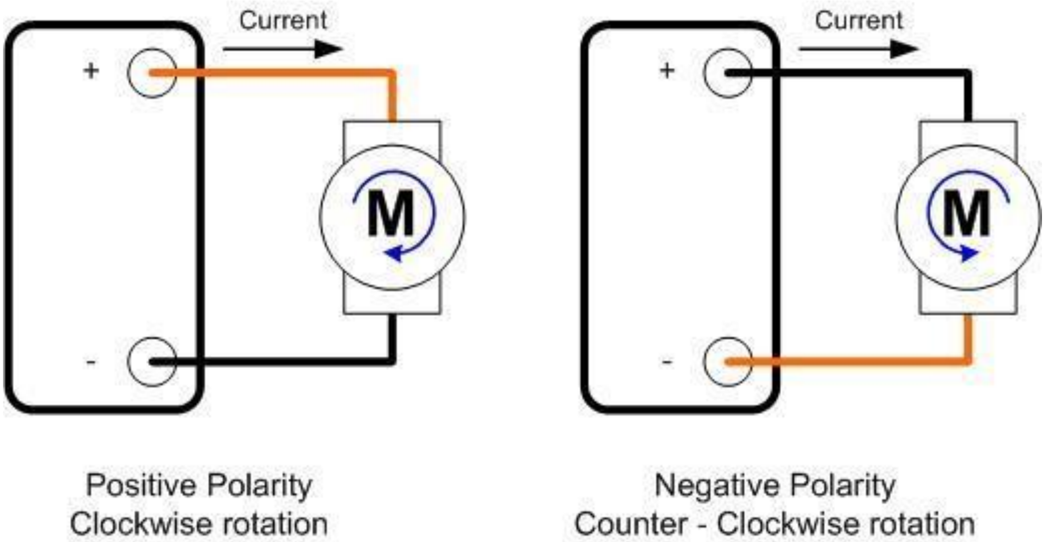


Figure 5.18 Polarity

DC Motor Rotation vs Polarity

Facts:

- DC Motor rotation has nothing to do with the voltage magnitude or the current magnitude flowing through the motor.
- DC Motor rotation does have to do with the voltage polarity and the direction of the current flow.

DC Motor Speed

Whereas the voltage polarity controls DC motor rotation, voltage magnitude controls motor speed. Think of the voltage applied as a facilitator for the strengthening of the magnetic field. In other words, the higher the voltage, the quicker will the magnetic field become strong. Remember that a DC motor has an electromagnet and a series of permanent magnets. The applied voltage generates a magnetic field on the electromagnet portion. This electromagnet field is made to oppose the permanent magnet field. If the electromagnet field is very strong, then both magnetic entities will try to repel each other from one side, as well as atract each other from the other side. The stronger the induced magnetic field, the quicker will this separation/attaction will try to take place. As a result, motor speed is directly proportional to applied voltage.

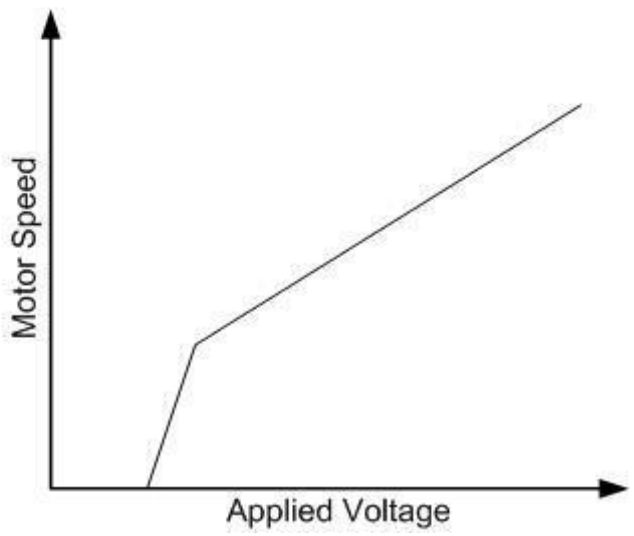


Figure 5.19 Motor speed

Motor Speed Curve

One aspect to have in mind is that the motor speed is not entirely lineal. Each motor will have their own voltage/speed curve. One thing I can guarantee from each motor is that at very low voltages, the motor will simply not move. This is because the magnetic field strength is not enough to overcome friction. Once friction is overcome, motor speed will start to increase as voltage increase.The following video shows the concept of speed control and offers some ideas on how this can be achieved.

Motor Torque

In the previous segment I kind of described speed as having to do with the strength of the magnetic field, but this is in reality misleading. Speed has to do with how fast the magnetic field is built and the attraction/repel forces are installed into the two magnetic structures. Motor strength, on the other hand, has to do with magnetic field strength. The stronger the electromagnet attracts the permanent magnet, the more force is exerted on the motor load.Per example, imagine a motor trying to lift 10 pounds of weight. This is a force that when multiplied by a distance (how much from the ground we are lifting the load) results in WORK. This WORK when exerted through a predetermined amount of time (for how long we are lifting the weight) gives us power. But whatever power came in, must come out as energy can not be created or destroyed. So that you know, the power that we are supplying to the motor is computed by

$$P = IV$$

Where P is power, I is motor current and V is motor voltage

Hence, if the voltage (motor speed) is maintained constant, how much load we are moving must come from the current. As you increase load (or torque requirements) current must also increase.

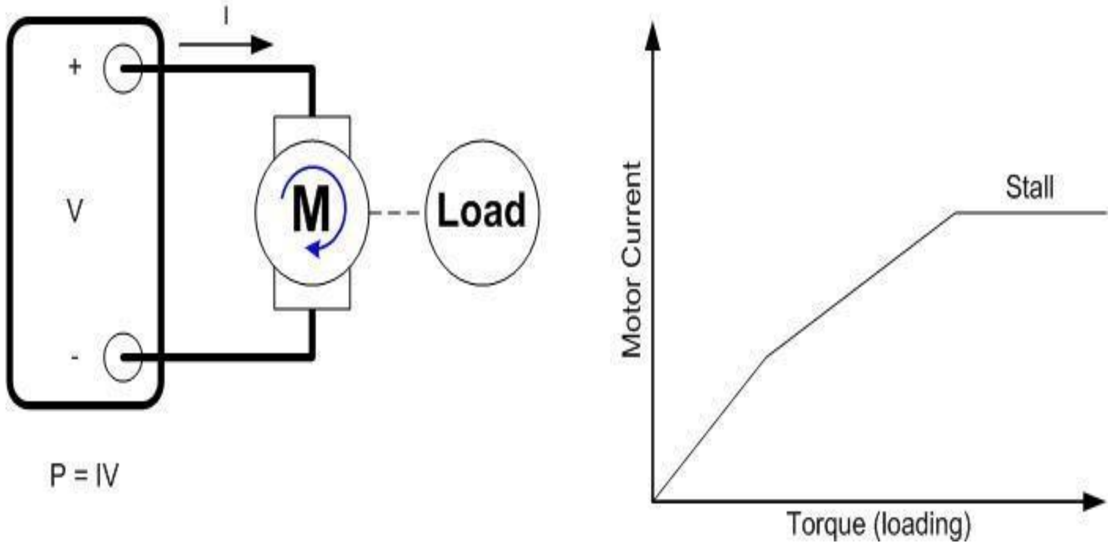


Figure 5.20 Motor curve

Motor Loading

One aspect about DC motors which we must not forget is that loading or increase of torque can not be infinite as there is a point in which the motor simply can not move. When this happens, we call this loading "Stalling Torque". At the same time this is the maximum amount of current the motor will see, and it is refer to Stalling Current. Stalling deserves a full chapter as this is a very important scenario that will define a great deal of the controller to be used. I promise I will later write a post on stalling and its intricacies.

Motor Start and Stop

You are already well versed on how to control the motor speed, the motor torque and the motor direction of rotation. But this is all fine and dandy as long as the motor is actually moving. How about starting it and stopping it? Are these trivial matters? Can we just ignore them or should we be careful about these aspects as well? You bet we should!

Starting a motor is a very hazardous moment for the system. Since you have an inductance whose energy storage capacity is basically empty, the motor will first act as an inductor. In a sense, it should not worry us too much because current can not change abruptly in an inductor, but the truth of the matter is that this is one of the instances in which you will see the highest currents flowing into the motor. The start is not necessarily bad for the motor itself as in fact the motor can easily take this Inrush Current. The power stage, on the other hand and if not properly designed for, may take a beating.Once the motor has started, the motor current will go down from inrush levels to whatever load the motor is at. Per example, if the motor is moving a few gears, current will be proportional to that load and according to torque/current curves.Stopping the motor is not as harsh as starting. In fact, stopping is pretty much a breeze. What we do need to concern ourselves is with how we want the motor to stop. Do we want it to coast down as energy is spent in the loop, or do we want the rotor to stop as fast as possible? If the later is the option, then we need braking. Braking is easily accomplished by shorting the motor outputs. The reason why the motor stops so fast is because as a short is applied to the motor terminals, the Back EMF is shorted. Because Back As already said, H-bridge can be made with the help of transistors as well as MOSFETs, the only thing is the power handling capacity of the circuit. If motors are needed to run with high current then lot of dissipation is there. So, head sinks are needed to cool the circuit. Now you might be thinking in why i did not discuss the cases like High side left on and Low side left on or high side right on and low side right on. Clearly seen in the diagram, you don't want to burn your power supply by shorting them. So that is why those combinations are not discussed in the truth table.

CHAPTER 6

RESULTS

6.1 Results

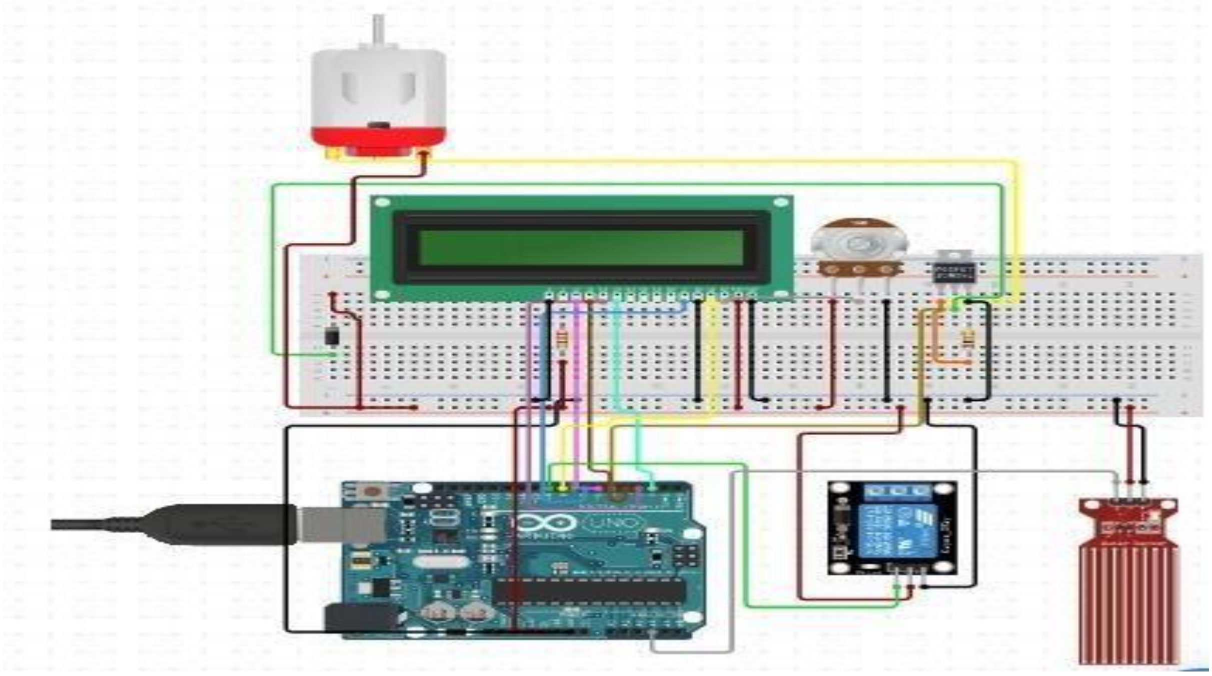
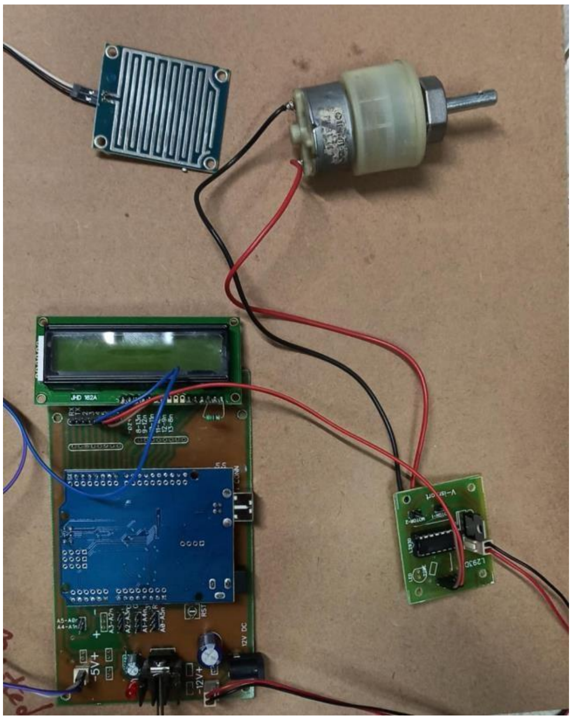


Figure 6.1 schematic diagram



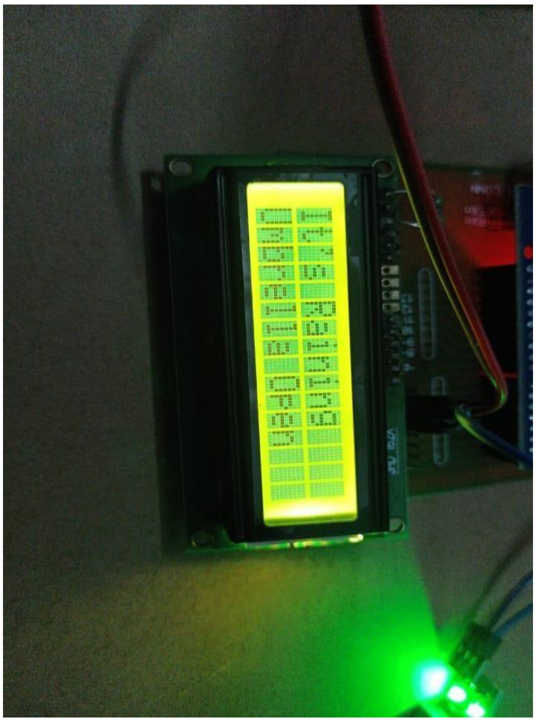


Figure 6.1 (a) setup components

Figure 6.1(b) result on LCD Display

The smart system consisting of a rack is fixed to umbrella such that when a sensor senses the exceeding value then the it will give some information to the arduino so that the arduino will take the information and makes some processing then send to the LCD display the LCD display will show the result of the sensor so that the paddle which is connected is start working by use of DC motor the DC motor will start rotating so the paddle will start opening so the rack which is fixed to the umbrella is start functioning. so the umbrella is opened

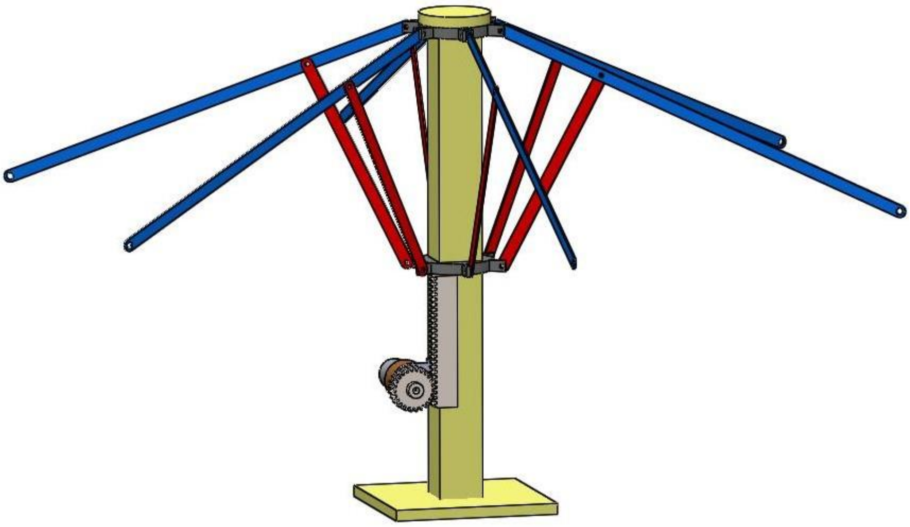


Figure 6.3 setup of automatic rainwater sensing umbrella

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion:

After completing the present work with all the process involved in designing the automated umbrella which can be of reasonable cost analysis and effective way to providing the shelter and safety to the object and goods with high intensity of power and this type of umbrella should be used in the various place in village to protect crops and the vehicles, etc. It can be applying at large scale area for reduced human work. It included the rack and pinion gear which changes the motion rotation into reciprocating. It also used in future after some of modification algorithm can be developed to coordinate Working system of rain to identify weather condition based on set value. The designed system prototype can be used along with the renewable source of energy. The designed model is not only smart but also intelligent as it will take decision about folding and unfolding of umbrella. The system can be control by Arduino and dc motor used for folding the roof with two or more switch fix along with the knowledge of next future direction.

7.2 Future Scope

The automatic rain water sensing umbrella can be implemented for the following

- Sheltering the houses
- Covering the farm lands in heavy rains

References

[1] Alkali, A.H., Dada, E.G., Kida, A.M. and Ali, A.A.,. "Sunlight and Rainfall Activated Retractable Roof", International Journal of Computer Engineering and Applications, 12(11), pp.1-12, 2018

[2] Maryam OmarOmar Bin SaminOmar Bin SaminImran and AhmedImran Ahmed, "Smartshed: An Automatic Shed System Based On Rain, Temperature And Light Intensity",Confernce: International Conference on Science, Innovation and Management (ICSIM 2019)At: Bangkok, Thailand April 2019.

[3] Abhijit G Kalbande, "Mart Automation System Using Arduino and Rain Drop Sensor", International Journal Of Current Engineering And Scientific Research (Ijcesr), Volume-4, Issue-6, 2017.

[4] https://patents.google.com/patent/US7909048B2/en

[5] Reddy, P.A., Prudhvi, G.S., Reddy, P.S.S. and Ramesh, S.S., Automatic rain sensing car wiper 2018. [6] Balathandapani, R., Boopathi, D., Jotheeshwaran, S., Arundeva, G. and Saranya, C. "Automatic rain water and crop saving system using embedded technology". International Journal of Science, Engineering and Technology Research (IJSETR), 4(3), 2015.

[7] Ajay, A., Shivashankar, P.S. and Sunil, D.M.,"Agriculture Crop Protection With Rain Water Harvesting And Power Generation.

[8] Mazidi, M.A., Mazidi, J.G. and McKinlay, R.D., The 8051 microcontroller and embedded systems: using Assembly and C (Vol. 626). Pearson/Prentice Hall. 2006.

[9] Alazzawi, L. and Chakravarty, A., "Design and implementation of a reconfigurable automatic rain sensitive windshield wiper". International Journal of Advances in Engineering & Technology, 8(2), p.73, 2015.

A

PROJECTREPORT

On

WIRELESS BLACKBOX FOR CARS USING SENSORS AND GPS MODULE

Submitted by

1) Mr.P.Phaneendra (17K81A0440)

2) Mr.Ch.Kali (17K81A0409)

3) Mr.Md.Saif (17K81A0432)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Mrs.P.Kiranmayee

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST.MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad – 500100

JUNE 2021

ST. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA & NAAC A+ Accredited

2020-2021

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the project entitled "WIRELESS BLACKBOX FOR CARS USING SENSORS AND GPS MODULE", is being submitted by **Mr.P.Phaneendra(17K81A0440), Mr.Ch.Kali(17K81A0432), Mr.Md.Saif(17K81A0432)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by her. The result embodied in this report have been verified and found satisfactory.

PROJECT GUIDE

Mrs.P.Kiranmayee

Associate Professor

Department of ECE

HEAD OF THE DEPARTMENT

Dr.B.Harikrishna

Professor

Department of ECE

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **CHANDUPATLA KALI** WITH ROLL NO.**17K81A0409, MOHAMMED SAIF** WITH ROLL NO.**17K81A0432**, **PULLURI PHANEENDRA** WITH ROLL NO.**17K81A0440**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**WIRELESS BLACK BOX FOR CARS USING SENSORS AND GPS MODULE**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT

DIRECTOR

TRAININGS & PLACEMENTS

LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014

Email : contact@lasyainfotech.com, ov@lasyainfotech.com

Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of '**Bachelor of Technology in Department of Electronics and Communication Engineering**', session: 2017 – 2021, **St.Martin's Engineering College**, here by declare that the project work entitled "**Wireless Blackbox for cars using sensors and gps module"** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. The result embodied in this project report has not been submitted in any university for award of any degree.

P. Phaneendra	**Ch. Kali**	**Md. Saif**
17K81A0440	**17K81A0409**	**17K81A0432**

// ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would b incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna**, **Head & Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our department project coordinator Dr.A.Anand kumar** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide, **Mrs.P.Kiranmayee, Assistant Professor**, Department of ECE and our class project coordinator **Mrs.A.Sowmya, Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

1. P. Phaneendra
2. Ch. Kali
3. Md. Saif

ABSTRACT

The main purpose of this wireless black box project is to develop a vehicle black box system that can be installed into any vehicle all over the world. This paradigm is often designed with minimum range of circuits. Wireless black box is basically a device that will indicate all the parameters of a vehicle crash and will also store and display its parameters such as temperature, location, vibration, alcohol limit etc. At the time of accident, the message will be sent from the system built inside the car to the registered mobile numbers such as emergency numbers of police stations, hospitals, family members, owner etc. We have used various types of sensors like temperature sensor, which is used to measure temperature. Vibration sensor measures vibrations felt by the car during accident. Alcohol sensor is located on the steering wheel which will indicate whether the driver is drunk. Gyroscope sensor is used to indicate tilt during the accident. GSM module, GPS module are some of the devices used in this project which helps in accomplishing the output.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

GPS - Global Positioning System

GSM - Global System for Mobile communication

IDE - Integrated Development Environment

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The main purpose of this wireless black box project is to develop a vehicle black box system that can be installed into any vehicle all over the world. This paradigm is often designed with minimum range of circuits. Wireless black box is basically a device that will indicate all the parameters of a vehicle crash and will also store and display its parameters such as temperature, location, vibration, alcohol limit etc. At the time of accident, the message will be sent from the system built inside the car to the registered mobile numbers such as emergency numbers of police stations, hospitals, family members, owner etc. We have used various types of sensors like temperature sensor, which is used to measure temperature. Vibration sensor measures vibrations felt by the car during accident. Alcohol sensor is located on the steering wheel which will indicate whether the driver is drunk. Gyroscope sensor is used to indicate tilt during the accident. GSM module, GPS module are some of the devices used in this project which helps in accomplishing the output. If any accident occurs, this wireless device will send a message from mobile phone and indicating the position of vehicle by tracking the location through GPS system to family member, emergency service so that they can provide ambulance and necessary treatment.

1.2 OBJECTIVES OF THE STUDY

The primary objective for doing our project was to introduce a blackbox system to alert the registered mobile numbers when an accident has occurred in an efficient and effective manner. To achieve this, the following objectives have to be completed.

- Studying the problem of providing help and medical aid to the victims when an accident has occurred at any location.
- Studying how fast the medical assistance is provided to the victims.
- Develop a solution to that problem.
- Studying the technologies that can solve the problem.
- Designing and developing a system for solving the problem.
- Testing and maintaining the implemented system

1.3 SCOPE OF THE STUDY

Road accidents are a human tragedy. Deaths are due to non timely availability of helps/medical aid after accidents.

If the person who has suffered the accident receives medical help in time he can survive the accident and many important lives can be saved. The system is easy to build and compact in size so that it can be easily installed in any vehicle.

Our project plays crucial role at the time of accident. The range of this system is extended to motorcycles and motorvehicles.

1.4 MATERIAL REQUIREMENT

1.4.1 HARDWARE REQUIREMENT

- Arduino Uno
- GPS Receiver
- GSM
- Temperature sensor LM35
- Accelerometer ADXL345
- Gas sensor MQ3
- Vibration sensor
- Motor driver L293D
- Motor

1.4.2 SOFTWARE REQUIREMENT

- Arduino IDE
- Circuito.io

1.5 PROCUMENT OF EQUIPMENT

We brought all the required hardware components in an electronic manufacturing shop and before assembling and interfacing hardware components to Arduino UNO micro-controller, we connected individual components to Arduino in circuito.io tool and understood working of every individual component.

Then, we assembled all components and interfaced to Arduino UNO micro-controller using connecting wires through bread board and we wrote program to work all the hardware components in Arduino IDE we understood the working of all components in a practical way.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH AREA

The Black Box concept is derived from the aviation industry, a flight recorder, colloquially known as a black box; although it is now orange-colored for easy search, is an electronic recording device placed in an aircraft for the purpose of facilitating the investigation of aviation accidents and incidents.

Figure 2.1 Flight Recorder

A flight recorder is an electronic recording device placed in an aircraft for the purpose of facilitating the investigation of aviation accidents and incidents. Flight recorders are also known by the misnomer black box—they are, in fact, painted bright orange in color to aid in their recovery after accidents.

An FDR has historically been one of two types of "flight recorder" carried on aircraft, the other being a cockpit voice recorder. Where both types of recorder are fitted, they are now sometimes combined into a single unit (ICAO Definition: *Combination recorders*). Combination recorders need to meet the flight recorder equipage requirements as specifically detailed in ICAO Annex 6 - Operation of Aircraft.

In existing system most of people associate black box with airplanes but they are no longer just key tool in investigation airplane accident. Presently tracking system introduced in vehicle to avoid accident and save the peoples life. But this system still installed in some of high-end vehicles only because this system is too expensive for most of the vehicle user.

Presently tracking system introduced in vehicle to avoid accident and save the peoples life. But this system still installed in some of high end vehicles only because this system are too expensive for most of the vehicle user.

2.2 REVIEW ON RELATED LITERATURE

Existing system is a device installed to record vehicle crashes or accidents. Electronically sensed problems in the engine or a sudden change in wheel speed trigger's this device

It do not track the vehicle movement or the driver's action and do record several types of important data only few seconds before the collision, due to the existing system is designed to monitor activation of airbags

2.3 CONCLUSIONS ON REVIEWS

- Post-Accident Detection Systems: All the existing systems are in such a way that the family members of the accident victim will come to know about the incident after a long duration which leads to death of the victim.
- Lack of Intelligence in the detection systems: Due to the lack of intelligence the person life may get into risk

In our project we are introducing accident detection and alarm system which is expected to save peoples life by detecting the accidents occurred and provides help by tracing the location of the motorcycle riders with the help of GPS technology.

The purpose of the project is to find the accident location using GPS module and to send this location by means of sending a message using GSM module to the pre-coded number.

CHAPTER 3

PROJECT DESIGN

3.1 OVERVIEW OF THE DESIGN

Four sensors that are temperature sensor LM35, vibration sensor, gas sensor MQ3, accelerometer ADXL345 are the primary inputs. LCD display, GPS6MV2, GSM SIM800L are the primary outputs. To indicate motion of car we are using a motor.

Initially the threshold detection level of sensors value is predefined. When the device is switched on, it displays “Wireless Blackbox”. Now the user is prompted to give an input such that it exceeds the threshold of the sensors. Once this happens the SMS is sent to registered mobile number with the location of the accident place.

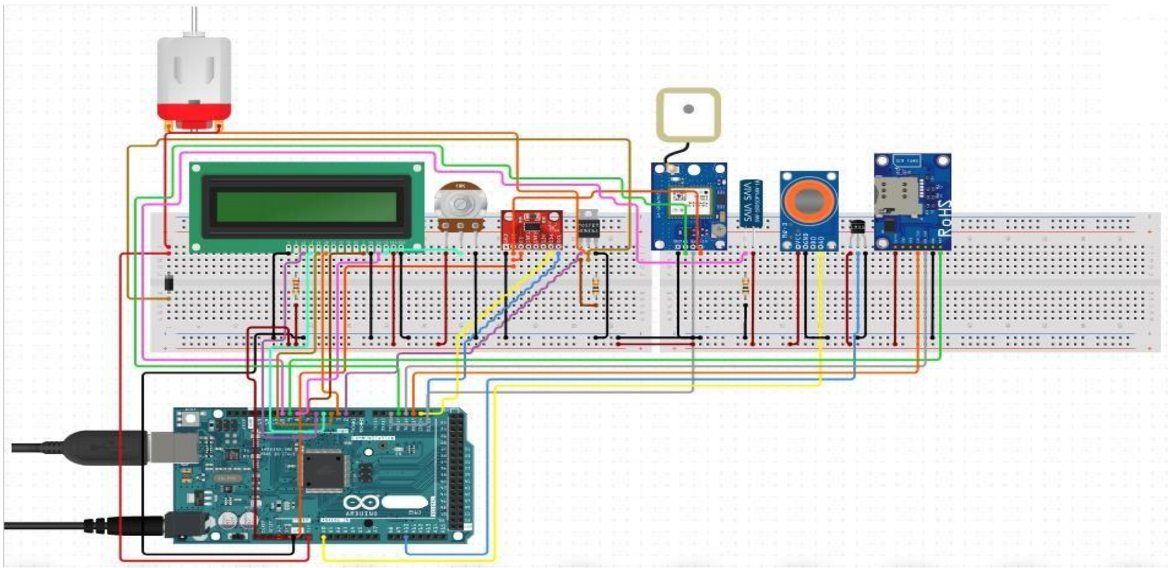


Figure 3.1: Overview of the design

3.2 EQUIPMENT ANALYSIS

3.2.1 Arduino Uno:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

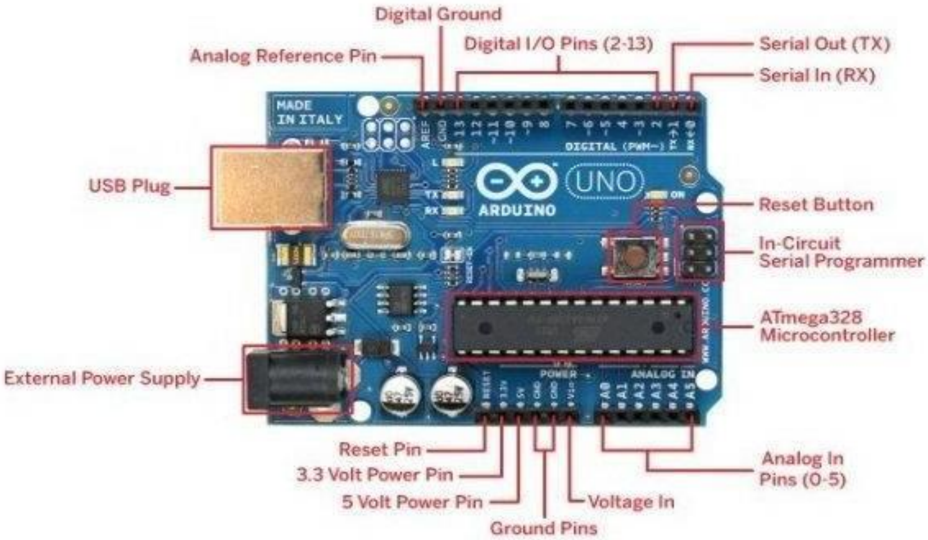


Figure 3.2: Arduino Board Description

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

Board Types :

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header

Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.1: Arduino boards based on ATMEGA328 microcontroller

Board Name	**Operating Volt**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad	3.3V	8MHz	14	6	6	1	Native USB

Arduino USB							

Table 3.2: Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 3.3: Arduino boards based on ATMEGA2560 microcontroller

Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)**

	Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the

	brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.
16	**AREF**

	AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

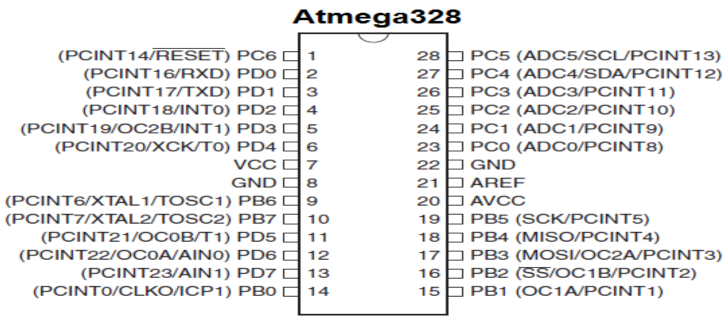


Figure 3.3: Pin description of Atmega 328

3.2.2 GPS Receiver:

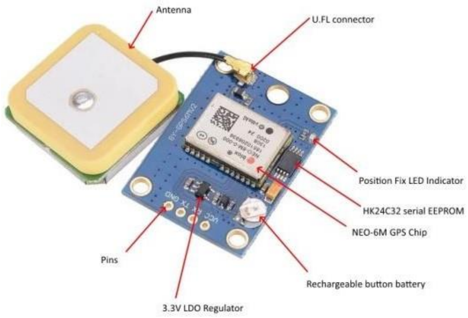


Figure 3.4: GPS6MV2 Receiver

The **NEO-6MV2** is a **GPS** (Global Positioning System) module and is used for navigation. The module simply checks its location on earth and provides output data which is longitude and latitude of its position.It is from a family of stand-alone GPS receivers featuring the high performance u-blox 6 positioning engine. These flexible and cost effective receivers offer numerous connectivity options in a miniature (16 x 12.2 x 2.4 mm) package. The compact architecture, power and memory options make **NEO-6 modules** ideal for **battery operated mobile devices** with very strict cost and space constraints. Its Innovative design gives **NEO-6MV2** excellent navigation performance even in the most challenging environments.

The module has four output pins and we will describe the function each pin of them below. The powering of module and communication interface is done through these four pins.

Pin Name	Description
VCC	Positive power pin
RX	UART receive pin
TX	UART transmit pin
GND	Ground

Features and Electrical Characteristics:

- Standalone GPS receiver
- Anti-jamming technology
- UART Interface at the output pins (Can use SPI ,I2C and USB)
- Under 1 second time-to-first-fix for hot and aided starts
- Receiver type: 50 Channels - GPS L1 frequency - SBAS (WAAS, EGNOS, MSAS)
- Time-To-First-fix: For Cold Start 32s, For Warm Start 23s, For Hot Start <1s
- Maximum navigation update rate: 5Hz
- Default baud rate: 9600bps
- EEPROM with battery backup
- Sensitivity: -160dBm
- Supply voltage: 3.6V
- Maximum DC current at any output: 10mA
- Operation limits: Gravity-4g, Altitude-50000m, Velocity-500m/s
- Operating temperature range: -40ºC TO 85°C

This module is one of popular GPS modules on the market and is also cheap to buy. The location data provided by it is accurate enough to satisfy most applications. And for it to be included in smart phones and tablets design points out its efficiency. This module is famous among hobbyist and engineers altogether who want to work on applications involving navigation.

3.2.3 GSM:

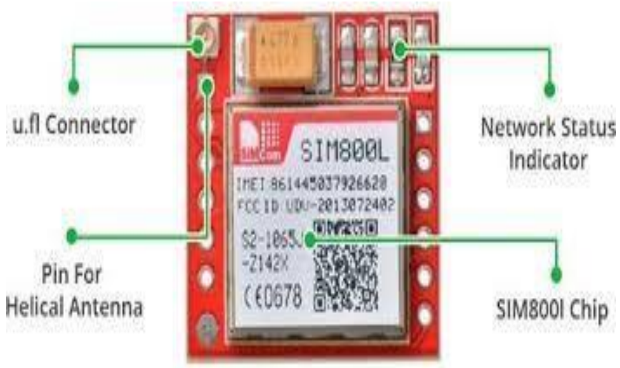


Figure 3.5: GSM SIM800L

At the heart of the module is a SIM800L GSM cellular chip from SimCom. The operating voltage of the chip is from 3.4V to 4.4V, which makes it an ideal candidate for direct LiPo battery supply. This makes it a good choice for embedding into projects without a lot of space.

All the necessary data pins of SIM800L GSM chip are broken out to a 0.1″ pitch headers. This includes pins required for communication with a microcontroller over UART. The module supports baud rate from 1200bps to 115200bps with Auto-Baud detection.

The module needs an external antenna to connect to a network. The module usually comes with a Helical Antenna and solders directly to NET pin on PCB. The board also has a U.FL connector facility in case you want to keep the antenna away from the board.

There's a SIM socket on the back! Any activated, 2G micro SIM card would work perfectly. Correct direction for inserting SIM card is normally engraved on the surface of the SIM socket.

This module measures only 1 inch² but packs a surprising amount of features into its little frame. Some of them are listed below:

- Supports Quad-band: GSM850, EGSM900, DCS1800 and PCS1900
- Connect onto any global GSM network with any 2G SIM
- Make and receive voice calls using an external 8Ω speaker & electret microphone
- Send and receive SMS messages

- Send and receive GPRS data (TCP/IP, HTTP, etc.)
- Scan and receive FM radio broadcasts
- Transmit Power:
    - Class 4 (2W) for GSM850
    - Class 1 (1W) for DCS1800
- Serial-based AT Command Set
- FL connectors for cell antennae
- Accepts Micro SIM Card

We cannot directly connect Rx pin on module to Arduino's digital pin as Arduino Uno uses 5V GPIO whereas the SIM800L module uses 3.3V level logic and is NOT 5V tolerant. This means the Tx signal coming from the Arduino Uno must be stepped down to 3.3V so as not to damage the SIM800L module. There are several ways to do this but the easiest way is to use a simple resistor divider.

3.2.4 Temperature sensor LM35:

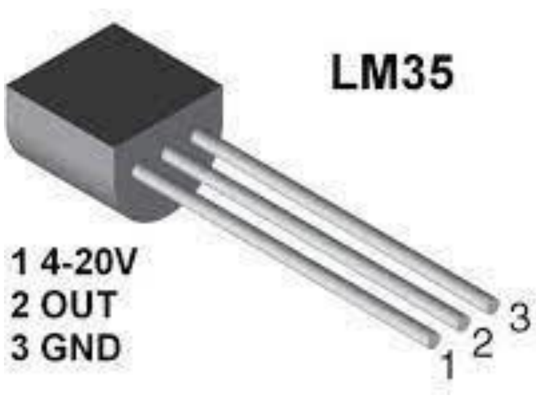


Figure 3.6: Temperature sensor LM35

Pin Configuration:

Pin Number	Pin Name	Description
1	Vcc	Input voltage is +5V for typical applications
2	Analog Out	There will be increase in 10mV for raise of every 1°C. Can range from -1V(-55°C) to 6V(150°C)
3	Ground	Connected to ground of circuit

LM35 Regulator Features:

- Minimum and Maximum Input Voltage is 35V and -2V respectively. Typically 5V.
- Can measure temperature ranging from -55°C to 150°C
- Output voltage is directly proportional (Linear) to temperature (i.e.)
- There will be a rise of 10mV (0.01V) for every 1°C rise in temperature.
- ±0.5°C Accuracy
- Drain current is less than 60uA
- Low cost temperature sensor
- Small and hence suitable for remote applications
- Available in TO-92, TO-220, TO-CAN and SOIC package

LM35 is a precession Integrated circuit Temperature sensor, whose output voltage varies, based on the temperature around it. It is a small and cheap IC which can be used to measure temperature anywhere between -55°C to 150°C. It can easily be interfaced with any Microcontroller that has ADC function or any development platform like Arduino.

The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly proportional to the Centigrade temperature. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55°C to 150°C temperature range.

3.2.5 Alcohol sensor MQ-3 :

Figure 3.7: Alcohol sensor MQ-3

Pin Name	Description
VCC	This pin powers the module, typically the operating voltage is +5V
GND	Used to connect the module to system ground
Digital Out (DO)	You can also use this sensor to get digital output from this pin, by setting a threshold value using the potentiometer
Analog Out (AO)	This pin outputs 0-5V analog voltage based on the intensity of the gas

Features of MQ-3 Alcohol Sensor:

- Sensor Type - Semiconductor
- Easy SIP header interface
- Compatible with most of the microcontrollers
- Low-power standby mode
- Requires heater voltage
- Good sensitivity to alcohol gas
- Fast response and High sensitivity
- Long life and low cost
- Requires simple Drive circuit

Specifications of MQ-3 Gas Sensor:

- Power requirements: 5 VDC @ ~165 mA (heater on) / ~60 mA (heater off)
- Current Consumption: 150mA
- DO output: TTL digital 0 and 1 (0.1 and 5V)

- AO output: 0.1- 0.3 V (relative to pollution)
- The maximum concentration of a voltage of about 4V
- Detecting Concentration: 0.05-10mg/L Alcohol
- Interface: 1 TTL compatible input (HSW), 1 TTL compatible output (ALR)
- Heater consumption: less than 750mW
- Operating temperature: 14 to 122 °F (-10 to 50°C)
- Load resistance: 200kΩ
- Sensitivity S: Rs(in air)/Rs(0.4mg/L Alcohol)≥5
- Sensing Resistance Rs: 2KΩ-20KΩ(in 0.4mg/l alcohol)
- Dimensions: 32 x 22 x 16 mm
- MQ-3 module is suitable for detecting Alcohol, Benzine, CH_4, Hexane, LPG, CO. Sensitive material of MQ-3 gas sensor is SnO_2, which with lower conductivity in clean air. When the target alcohol gas exist, the sensor's conductivity is more higher along with the gas concentration rising. MQ-3 gas sensor has high sensitity to Alcohol, and has good resistance to disturb of gasoline, smoke and vapor.
- This sensor provides an analog resistive output based on alcohol concentration. When the alcohol gas exist, the sensor's conductivity gets higher along with the gas concentration rising.

3.2.6 Accelerometer ADXL345 :

Figure 3.8: Accelerometer ADXL345

The **ADXL345** is a small, low power, complete 3-axis MEMS accelerometer modules with both I2C and SPI interfaces. The ADXL345 board feature on-board 3.3V voltage regulator and level shifter, which makes it simple to interface with 5V microcontrollers such as the Arduino.

ADXL345 Module Features & Specifications:

- 3V-6V DC Supply Voltage
- On-board LDO Voltage regulator
- Built-in Voltage level convertor (MOSFET based)
- It can be interface with 3.3V or 5V Microcontroller.
- Ultra-Low Power: 40uA in measurement mode, 0.1uA in standby@ 2.5V
- Tap/Double Tap Detection
- Free-Fall Detection
- SPI and I2C interfaces
- Measuring Range: ±16g
- Measuring Values (-16g to +16g):
- X: -235 to +270
- Y: -240 to +260
- Z: -240 to +270

This **ADXL345 Accelerometer module** consists of an ADXL345 Accelerometer IC, Voltage Regulator IC, Level Shifter IC, resistors, and capacitors in an integrated circuit. Different manufacturers use a different voltage regulator IC.

ADXL345 IC from Analog Devices is the brain of this module. The ADXL345 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of ±16g.

ADXL345 Accelerometer module consists of 8 pins. Using the ADXL345 module with a microcontroller is very easy. Connect VCC and GND pins to 5V and GND pins of Microcontroller. Also, connect SCL and SDA pins to the SCL and SDA pins of Arduino.

The basic structure of the accelerometer consists of fixed plates and moving plates. When the acceleration is applied on an axis capacitance between fixed plates and moving plates is changed. This results in a sensor output voltage amplitude, which is proportional to the acceleration.

3.2.7 Vibration sensor :

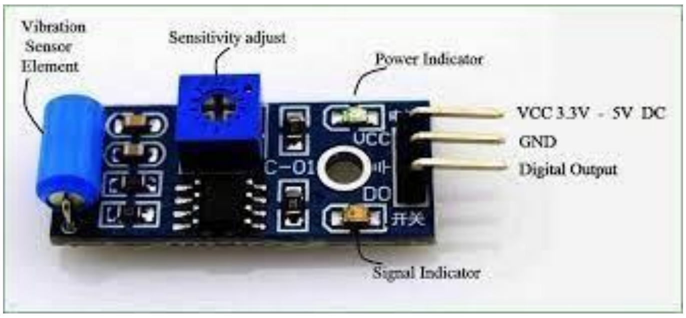


Figure 3.9: Vibration sensor

A vibration sensor is a device that measures the amount and frequency of vibration in a given system, machine, or piece of equipment. Those measurements can be used to detect imbalances or other issues in the asset and predict future breakdowns.

3.2.8 LCD 16X2 Display:

Figure 3.10: LCD display

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.

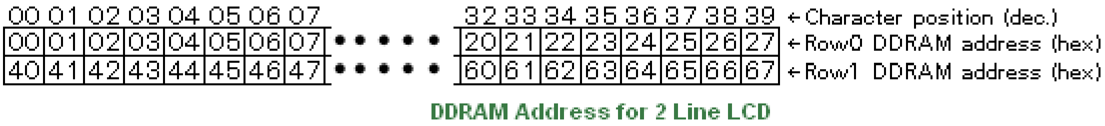


Figure 3.11: DDRAM addresses of 2 Line LCD

16 X 2 ALPHANUMERIC LCD MODULE FEATURES:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

3.2.9 Motor driver L293D :

Figure 3.12: Motor driver L293D

L293D is basically a high current dual motor driver/controller Integrated Circuit (IC). It is able to drive load having current up to 1A at the voltage ranging from 4.5V to 36V. Motor driver usually act as current amplifier because they receive a low current signal as an input and provides high current signal at the output.

L293D is basically a motor driver or controller. It has two built-in H-bridge circuits which are able to control two DC motors simultaneously in both clockwise and counter clockwise direction. It acts as an current high amplifier because it take low current signal at its input and provides higher current signal at the output in order to drives different load e.g. stepper motor & DC motors. Its features include large input voltage supply range, large output current high noise immunity input signals etc. Its common real-life applications include stepping motor drivers, relay drivers, DC motor drivers etc. L-293D motor driver/controller is shown in the figure given below.

- L-293D has sixteen (16) pins, having different individual functions.
- All of these sixteen pins along with their serial no, are given in the table shown below.
- L 293D each pin has different tasks to perform when they are in working condition.
- The tasks assigned to each pin are given in the table shown below.

L293D Pins	
Pin. No	Pin Name
1	Enable 1,2
2	Input 1
3	Output 1
4	GND
5	GND
6	Output 2
7	Input 2
8	V_{CC2}
9	Enable 3,4
10	Input 3
11	Output 3
12	GND
13	GND
14	Output put 4
15	Input 4
16	V_{CC1}

Table 3.4: L293D Pin Functions

3.2.10 Motor:

Figure 3.13: Motor

Motor’s usually operates on this higher current. L-293D has to built-in H-Bridge driver circuits and is able to control two DC motors at a time in both clockwise and counter clockwise direction. It has two enable pins and they should be kept high in order to control the motor. By changing the polarity of applied signal motor can be rotated in either clockwise or counter clockwise direction.

3.2.11 Arduino IDE:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments.

All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows. Wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers.

- The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money
- Getting Started with Arduino and Genuino products: -

Install the Arduino Software (IDE) on Windows PCs-
This document explains how to install the Arduino Software (IDE) on Windows machines.

✓ Download the Arduino Software (IDE)

✓ Proceed with board specific instructions.

HOW TO DOWNLOAD THE ARDUINO SOFTWARE (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

INSTALLATION:

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Figure 3.14: Standard USB Cable (A plug to B plug)

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Figure 3.15: Standard USB Cable (an A to Mini-B plug)

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

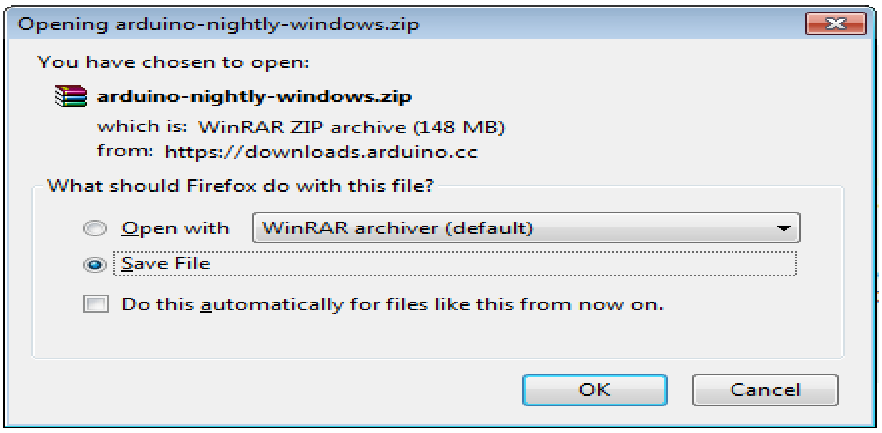


Figure 3.16: Download Arduino IDE Software

Step 3 – Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 – Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

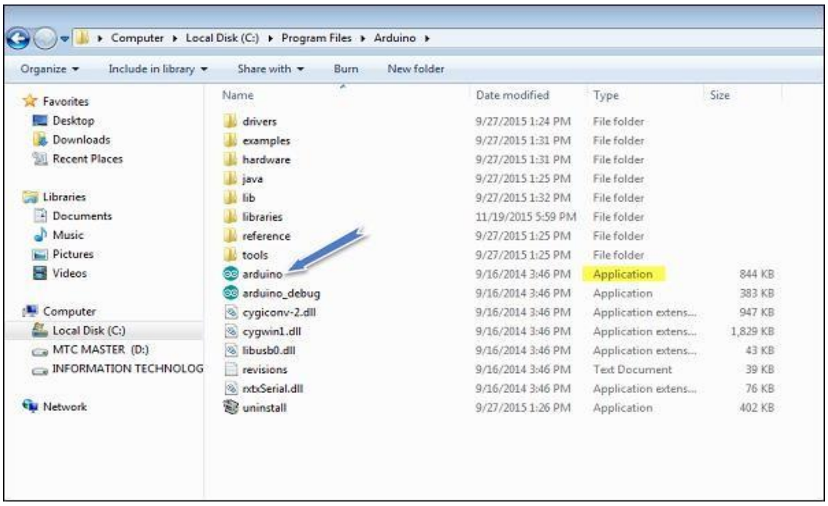


Figure 3.17: Launch Arduino IDE

Step 5 – Open your first project.

Once the software starts, you have two options –

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**

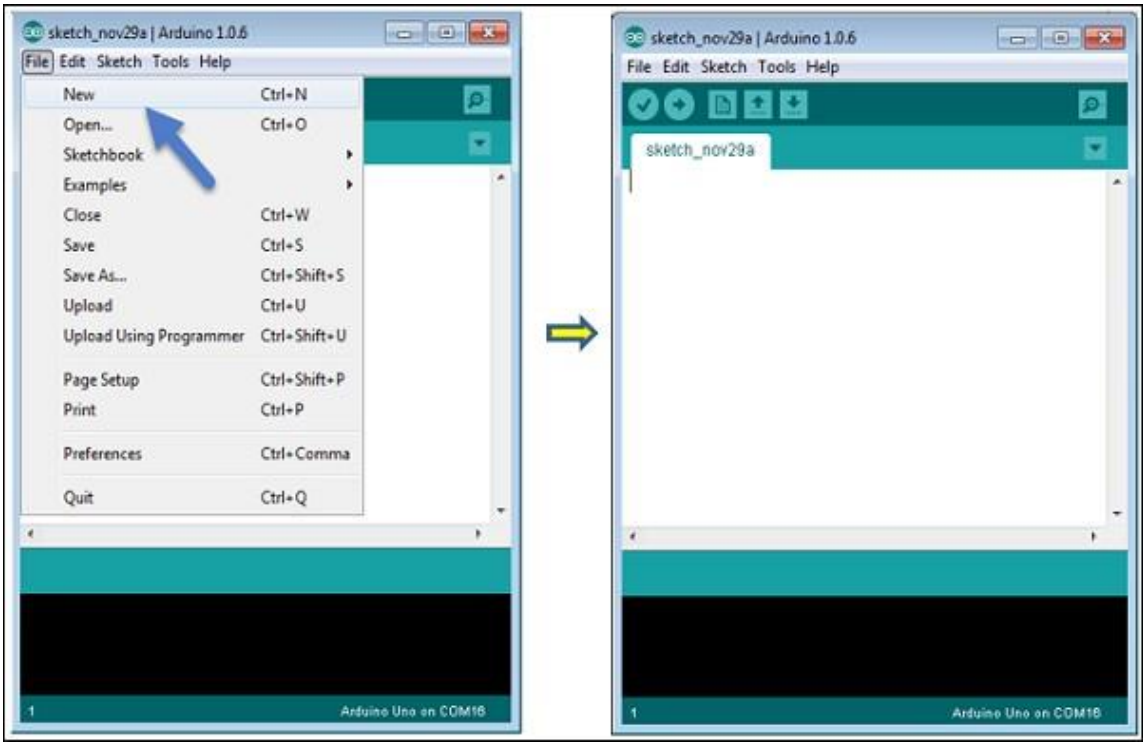


Figure 3.18: Open your first project.

To open an existing project example, select File → Example → Basics → Blink.

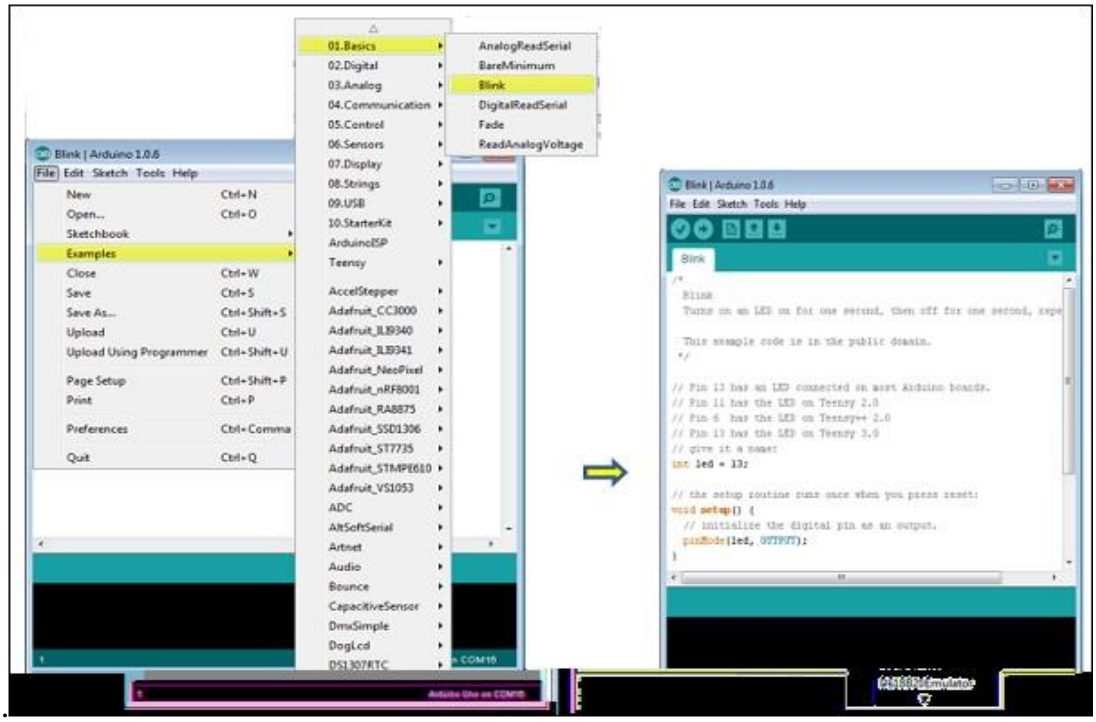


Figure 3.19: Open your first program.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

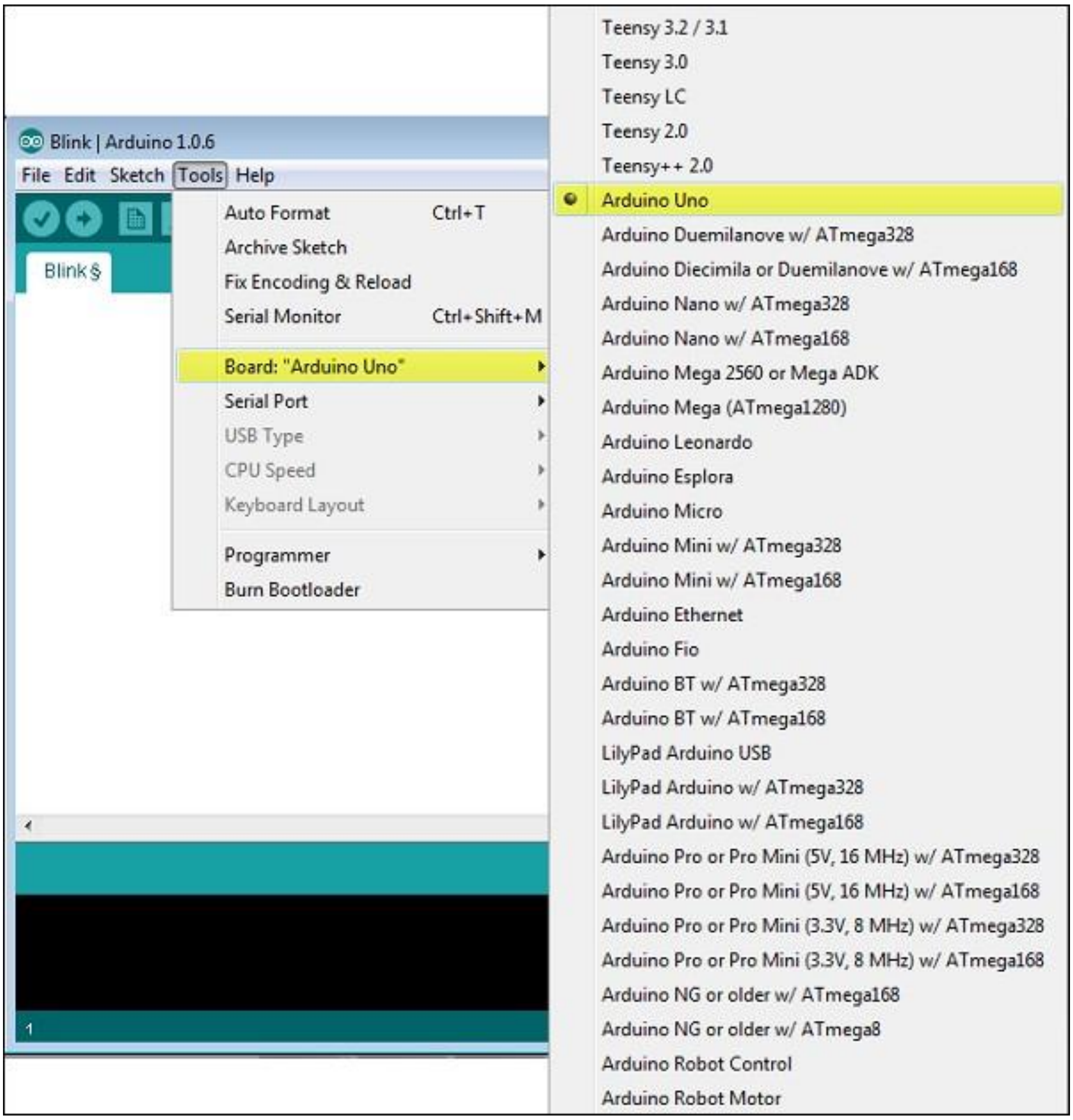


Figure 3.20: Select your Arduino board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

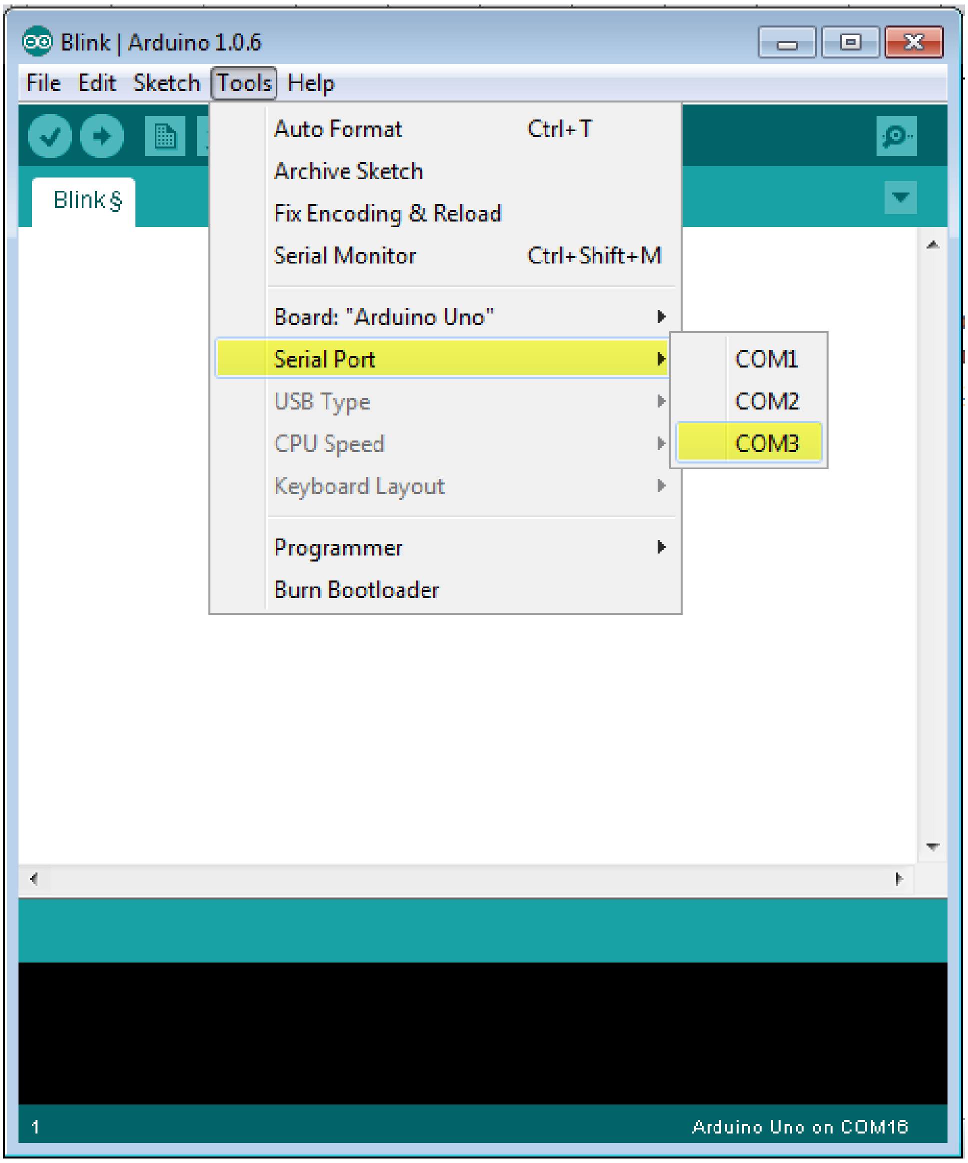


Figure 3.21: Select your serial port

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

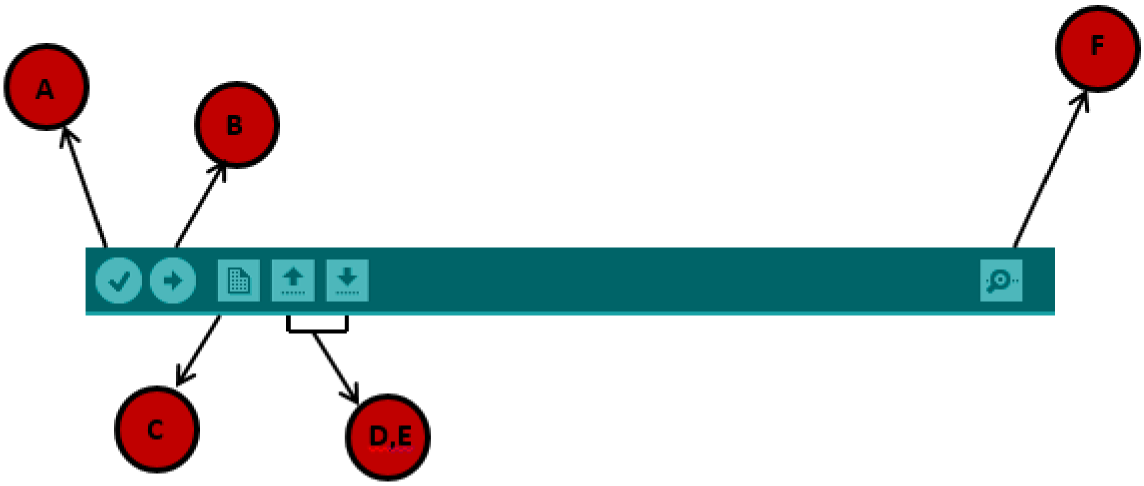


Figure 3.22: Upload the program to your board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from board & send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

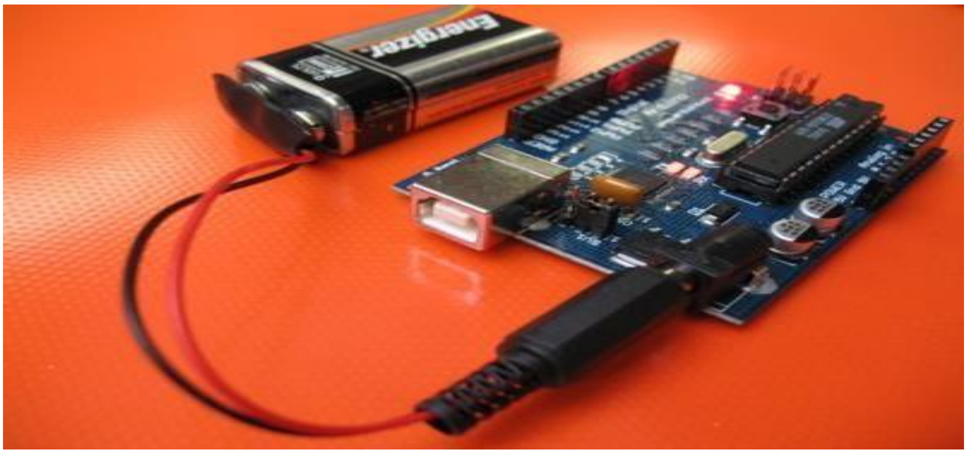


Figure 3.23: Connecting a Battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called "sketches", but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

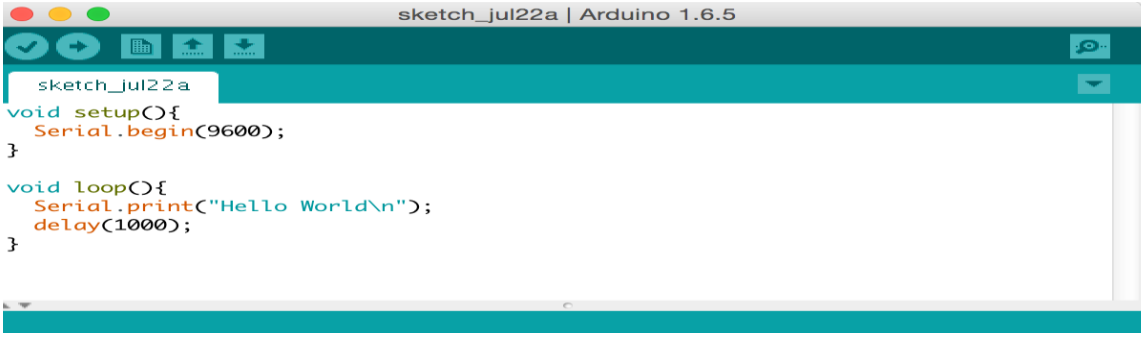


Figure 3.24: Example Program

3.2.12 Circuito.io:

Figure 3.25: circuito tool to design schematic diagram

Circuito.io is an online tool for designing complete electronic circuits. The Circuito app generates instant and accurate schematics and code for your electronic circuit. You select the major building blocks, and it computes all the electrical requirements for your selection.

It has a fantastic interface that allows you to drag and drop different parts together. It also has three different sections that one needs to work on before testing, and the first is the Bill of Materials (BoM) called DESIGN.

Here, you check out all the materials available and you then select your preferred options. You choose the components you want, or you think you need and move on to the next section. The next section being the wiring tool which will process, add all necessary additional items required and in return give a well-labeled wiring diagram.

Another exciting aspect about circuito.io is that it has an interface that allows you to step through each building component, guiding you through the creation of the circuit; This simply means that you are not working with a static diagram rather one that can move in different directions.

It does not end there though, and it only gets better with circuito.io. After building with the aid of the diagram, one can move on to the CODE module; this is a part of the platform that gives examples on how to program every piece used in the previous steps. The platform will solve the problem of hundreds of Arduino users, and it will be a lot easier to program your circuit after going through sample sketches for various pieces. It even acts as a tutorial method for beginners. So if you are stuck on what Arduino project to work on, circuito.io might just be the app for you.

3.3 DEFINE THE MODULES

3.3.1 GSM Module

Figure 3.26: SIM800L PINOUT

- NET is a pin where you can solder Helical Antenna provided along with the module.
- VCC supplies power for the module. This can be anywhere from 3.4V to 4.4 volts. Remember connecting it to 5V pin will likely destroy your module! It doesn't even run on 3.3 V! An external power source like Li-Po battery or DC-DC buck converters rated 3.7V 2A would work.
- RST (Reset) is a hard reset pin. If you absolutely got the module in a bad space, pull this pin low for 100ms to perform a hard reset.
- RxD (Receiver) pin is used for serial communication.
- TxD (Transmitter) pin is used for serial communication.
- GND is the Ground Pin and needs to be connected to GND pin on the Arduino.
- RING pin acts as a Ring Indicator. It is basically the 'interrupt' out pin from the module. It is by default high and will pulse low for 120ms when a call is received. It can also be configured to pulse when an SMS is received.
- DTR pin activates/deactivates sleep mode. Pulling it HIGH will put module in sleep mode, disabling serial communication. Pulling it LOW will wake the module up.
- MIC± is a differential microphone input. The two microphone pins can be connected directly to these pins.
- SPK± is a differential speaker interface. The two pins of a speaker can be tied directly to these two pins.

3.3.2 GPS Module

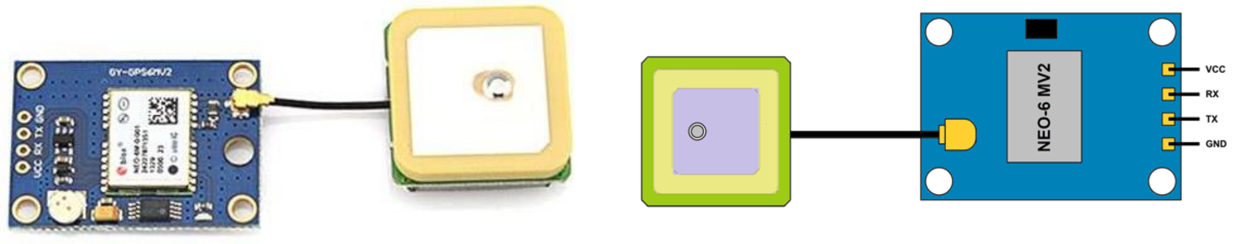


Figure 3.27: GPS6MV2 Receiver

The module has four output pins and we will describe the function each pin of them below. The powering of module and communication interface is done through these four pins.

Pin Name	**Description**
VCC	Positive power pin
RX	UART receive pin
TX	UART transmit pin
GND	Ground

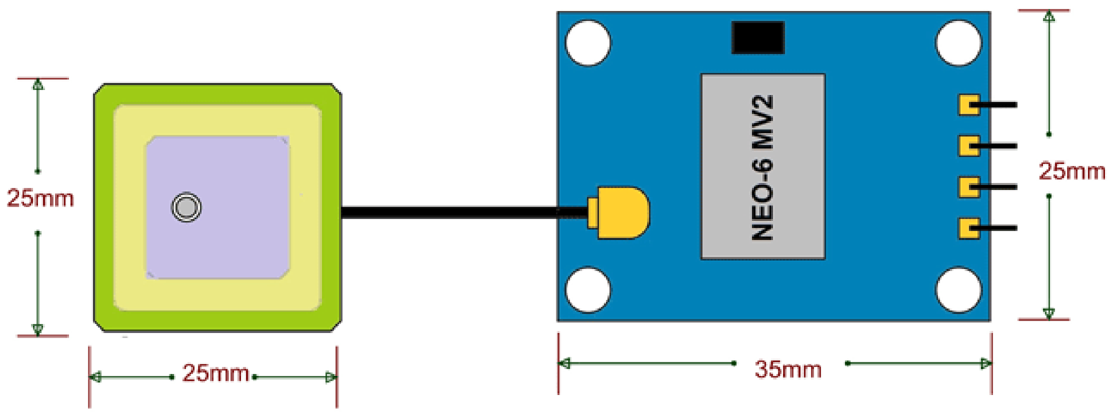


Figure 3.27: GPS6MV2 2-D model

3.4 MODULE FUNCTIONALITIES

3.4.1 GSM Functioning:

GSM is an open, digital cellular technology used for transmitting mobile voice and data services. GSM is a digital mobile telephone system that is widely used in Europe and other parts of the world. GSM uses a variation of Time Division Multiple Access (TDMA) and is the most widely used of the three digital wireless telephone technologies (TDMA, GSM and CDMA).

Computers use AT commands to control modems. Both GSM modems and dial-up modems support a common set of standard AT commands. GSM modem can be used just like a dialup modem. In addition to the standard AT commands, GSM modems support an extended set of AT commands. These extended AT commands are defined in the GSM standards. With the extended AT commands, various things can be done such as - Sending SMS messages, Monitoring the signal strength etc.

For sending AT commands and communicating with the SIM800L module, we will use the serial monitor. The sketch below will enable the Arduino to communicate with the SIM800L module on serial monitor. Before we proceed with detailed breakdown of code, connect your Arduino to PC, compile below code and upload it to the Arduino.

The sketch starts by including a SoftwareSerial.h library and initializing it with the Arduino pins to which Tx and Rx of SIM800L module is connected.

In setup function: we initialize a serial communication link between Arduino, Arduino IDE and SIM800L module at a baud rate of 9600.

Now that we have established a basic connection, we will try to communicate with the SIM800L module by sending AT commands.

- AT – It is the most basic AT command. It also initializes Auto-baud'er. If it works you should see the AT characters echo and then OK, telling you it's OK and it's understanding you correctly! You can then send some commands to query the module and get information about it such as

- AT+CSQ – Check the 'signal strength' – the first # is dB strength, it should be higher than around 5. Higher is better. Of course it depends on your antenna and location!

- AT+CCID – get the SIM card number – this tests that the SIM card is found OK and you can verify the number is written on the card.

- AT+CREG? Check that you're registered on the network. The second # should be 1 or 5. 1 indicates you are registered to home network and 5 indicates roaming network. Other than these two numbers indicate you are not registered to any network.

- ATI – Get the module name and revision

- AT+COPS? – Check that you're connected to the network, in this case BSNL

- AT+COPS=? – Return the list of operators present in the network.

- AT+CBC – will return the lipo battery state. The second number is the % full (in this case its 93%) and the third number is the actual voltage in mV (in this case, 3.877 V)

- AT+CMGF=1 – Selects SMS message format as text. Default format is protocol data unit (PDU).

- AT+CMGS=+ZZxxxxxxxxxx – Sends SMS to the phone number specified. The text message entered followed by a 'Ctrl+z' character is treated as SMS. 'Ctrl+z' is actually a 26^{th} non-printing character described as 'substitute' in ASCII table. So, we need to send 26_{DEC} ($1A_{HEX}$) once we send a message.

- AT+CMGF=1 – Selects SMS message format as text. Default format is Protocol Data Unit (PDU)

- AT+CNMI=1,2,0,0,0 – specifies how newly arrived SMS messages should be handled. This way you can tell the SIM800L module either to forward newly arrived SMS messages directly to the PC, or to save them in message storage and then notify the PC about their locations in message storage.

3.4.2 GPS Functioning:

The Global Positioning System satellites transmit signals to equipment on the ground. GPS receivers passively receive satellite signals; they do not transmit. GPS receivers require an unobstructed view of the sky, so they are used only outdoors and they often do not perform well within forested areas or near tall buildings. GPS operations depend on a very accurate time reference, which is provided by atomic clocks on board. Each GPS satellite transmits data that indicates its location and the current time.

All GPS satellites synchronize operations so that these repeating signals are transmitted at the same instant. The signals, moving at the speed of light, arrive at a GPS receiver at slightly different times because some satellites are further away than others. The distance to the GPS satellites can be determined by estimating the amount of time it takes for their signals to reach the receiver. When the receiver estimates the distance to at least four GPS satellites, it can calculate its position in three dimensions.

There are at least 24 operational GPS satellites at all times plus a number of spares. The satellites, operated by the U.S. Department of Defence, orbit with a period of 12 hours (two orbits per day) at a height of about 11,500 miles travelling at near 2,000mph. Ground stations are used to precisely track each satellite's orbit.

Start by connecting the patch antenna to the U.FL connector. Remember to thread the U.FL cable through one of the mounting holes for robust connection. The module usually comes with header pins unsoldered. So, you'll need to solder them.

Now, connect Tx and Rx pin on module to digital pin#2 and #3 respectively on Arduino; as we'll be using software serial to talk to the module. Next, connect VCC pin to the 5V pin on the arduino and GND to ground.

$	Every NMEA sentence starts with $ character.
GPRMC	Global Positioning Recommended Minimum Coordinates
123519	Current time in UTC – 12:35:19
A	Status A=active or V=Void.
4807.038,N	Latitude 48 deg 07.038′ N
01131.000,E	Longitude 11 deg 31.000′ E
022.4	Speed over the ground in knots
084.4	Track angle in degrees True
003.1,W	Magnetic Variation
*6A	The checksum data, always begins with *

- gps.speed.value() function returns current ground speed in 100ths of a knot.
- gps.course.value() function returns current ground course in 100ths of a degree.
- gps.satellites.value() function returns the number of visible, participating satellites.

- gps.hdop.value() function returns horizontal diminution of precision.

3.4.3 Project Code:

```
#include <Wire.h>

#include <Adafruit_Sensor.h>

#include <Adafruit_ADXL345_U.h>

/* Assign a unique ID to this sensor at the same time */

Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);

#define TempPin A3

#include <LiquidCrystal.h>

#include<SoftwareSerial.h>

SoftwareSerial mySerial(2,3);

LiquidCrystal lcd(13,12,11,10,9,8);//RS,EN,D4,D5,D6,D7

int TempValue;

int m1=4,m2=5;

int vib=6,alco=7;

void sendmsg(const char* num,String msg,char* extr)

{

  mySerial.print("AT+CMGS=\"");

  mySerial.print(num);

  mySerial.println("\"");delay(800);

  mySerial.println(msg);Serial.println(extr);delay(800);

  mySerial.write(0x1a);delay(2000);

}

void serialFlush()

{
```

```
  while(Serial.available() > 0)
  {
    char tt = Serial.read();
  }
}
const char* number = "8801359523\0";  /*Give the required mobile num to send the msg*/
//const char* number1 = "9603986537\0";
char gps_location[40];
void gps()
{
  lcd.clear();
  lcd.setCursor(0, 1);lcd.print("READING GPS");delay(50);
   char findValue[]="$GPRMC\0";
   while(1)
   {
    if(Serial.find(findValue,sizeof(findValue)))
    {
      for(int i=0;i<14;i++)
      {
        while(!Serial.available());
        char ch = Serial.read();
      }
      for(int i=0;i<24;i++)
      {
        while(!Serial.available());
        gps_location[i] = Serial.read();
```

```
      }
       break;
    }
   }
   lcd.setCursor(0, 1);lcd.print("READING GPS DONE");delay(2000);
   delay(1000);
   serialFlush();
}
char t;
void setup()
{
  pinMode(m1,OUTPUT);
 pinMode(m2,OUTPUT);
 digitalWrite(m1,LOW);
 digitalWrite(m2,LOW);
 pinMode(vib,INPUT);
 pinMode(alco,INPUT);
 lcd.begin(16,2);
 lcd.setCursor(0,0);lcd.print("Wireless Black Box");
 lcd.setCursor(0,1);lcd.print("For Car");
 delay(2000);
 Serial.begin(9600);
  mySerial.begin(9600);
  mySerial.println("AT");delay(1000);
  mySerial.println("AT+CMGF=1");delay(1000);
  mySerial.println("AT+CNMI=1,2,0,0");delay(1000);
```

```
 mySerial.println("AT+CSMP=17,167,0,16");delay(1000);

 delay(1000);

/* Initialise the sensor */

if(!accel.begin())

{

/* There was a problem detecting the ADXL345 ... check your connections */

lcd.clear();lcd.setCursor(0,0);

lcd.print("No Acclerometer");

while(1);

}

/* Set the range to whatever is appropriate for your project */

accel.setRange(ADXL345_RANGE_16_G);

}

void loop(void)

{

/* Get a new sensor event */

sensors_event_t event;

accel.getEvent(&event);

TempValue = analogRead(TempPin); // Getting LM35 value and saving it in variable

 float TempCel = ( TempValue/1024.0)*500; // Getting the celsius value from 10 bit analog
value

 float TempFarh = (TempCel*9)/5 + 32; // Converting Celsius into Fahrenhiet

 lcd.clear();lcd.setCursor(0,0);lcd.print("Temp: ");lcd.print(TempCel);

 lcd.setCursor(0,1);lcd.print("X: ");lcd.print(event.acceleration.x);lcd.print(" Y:
");lcd.print(event.acceleration.z);

 delay(100);

 digitalWrite(m1,LOW);
```

```
 digitalWrite(m2,HIGH);
delay(1000);
if(digitalRead(vib)==HIGH)
{
 String msg = "Accident Occured at Location:";
 lcd.clear();lcd.setCursor(0,0);lcd.print("Accident Occured");
 delay(100);
 gps();
 delay(500);
 sendmsg(number,msg,gps_location);
 delay(500);
 digitalWrite(m1,LOW);
 digitalWrite(m2,LOW);
 delay(3000);
}
if(digitalRead(alco)==HIGH)
{
 String msg = "Alcohol Consumed";
 char location[] = ".\0";
 lcd.clear();lcd.setCursor(0,0);lcd.print("Alcohol Dete");
 delay(100);
 sendmsg(number,msg,location);
 delay(500);
 digitalWrite(m1,LOW);
 digitalWrite(m2,LOW);
 delay(3000);
```

```
 }
 if(TempCel>40)
 {
  lcd.clear();lcd.setCursor(0,0);lcd.print("High Temperature");
  delay(1000);
 }
}
```

CHAPTER 4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM

The Black Box concept is derived from the aviation industry, a flight recorder, colloquially known as a black box. With the advancement in technology and cost coming down, in our project we attempt to build similar device for our car which can send the location to pre-coded number when an accident occurs

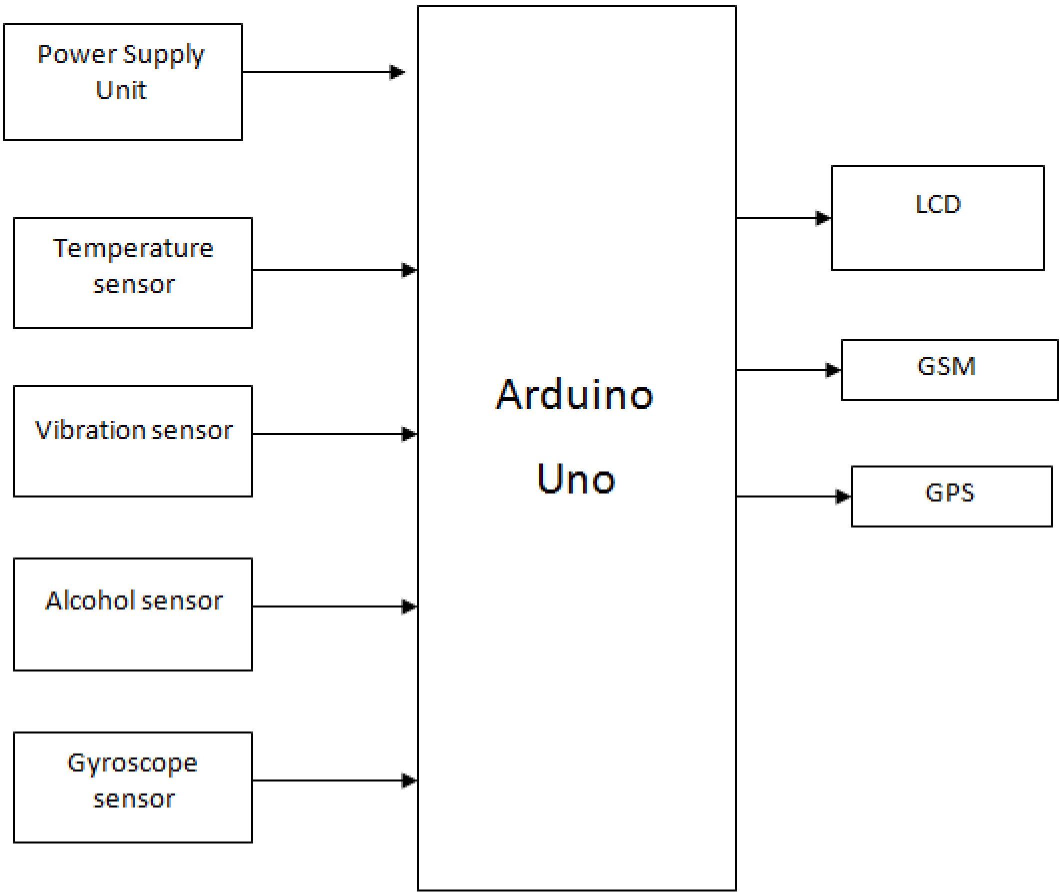


Figure 4.1: Block diagram of the blackbox

Four sensors that are temperature sensor LM35, vibration sensor, gas sensor MQ3, accelerometer ADXL345 are the primary inputs. LCD display, GPS6MV2, GSM SIM800L are the primary outputs. To indicate motion of car we are using a motor.

4.2 FLOW CHART

.

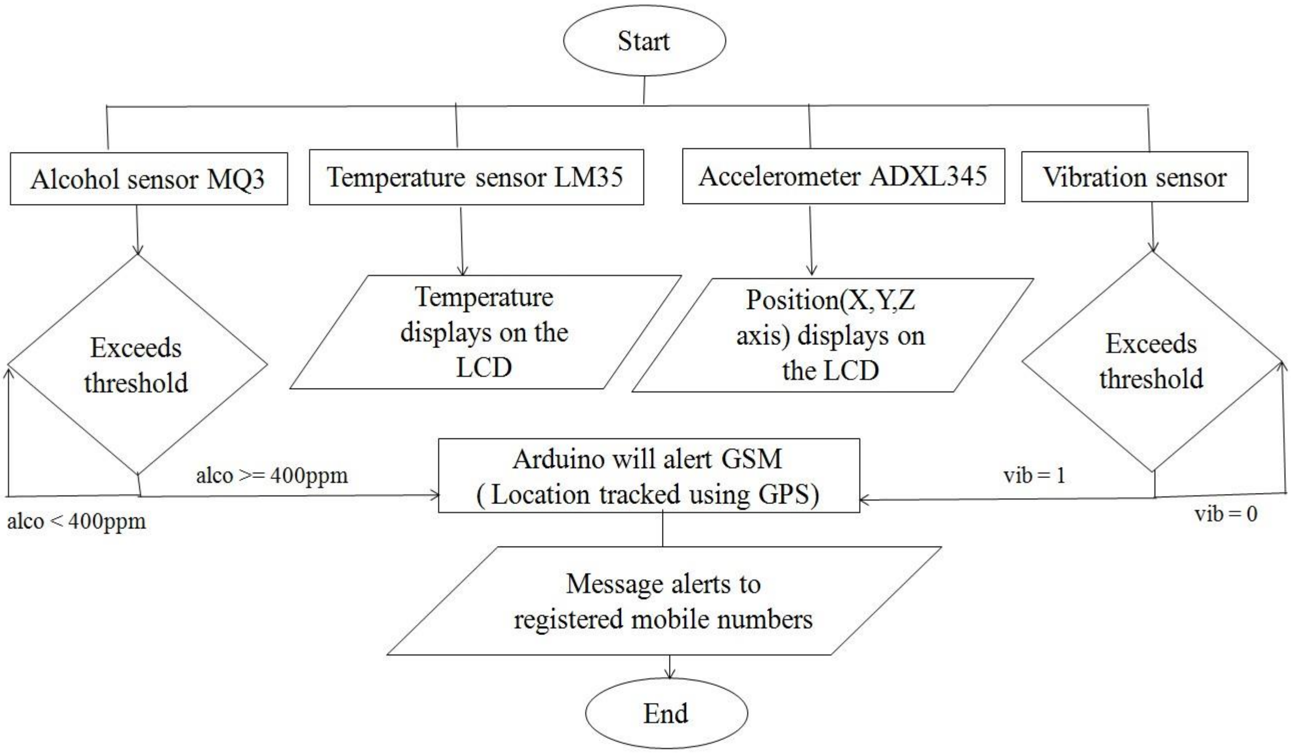


Figure 4.2: Flow chart of the system

This flow chart gives a brief idea as to how the project “Wireless Blackbox for cars using sensors and gps module” works.

Initially the threshold detection level of sensors value is predefined. When the device is switched on, it displays “Wireless Blackbox”. Now the user is prompted to give an input such that it exceeds the threshold of the sensors. Once this happens the SMS is sent to registered mobile number with the location of the accident place.

The GPS receiver helps to collect the location of the place. The GSM module sends the SMS.

4.3 HARDWARE SETUP

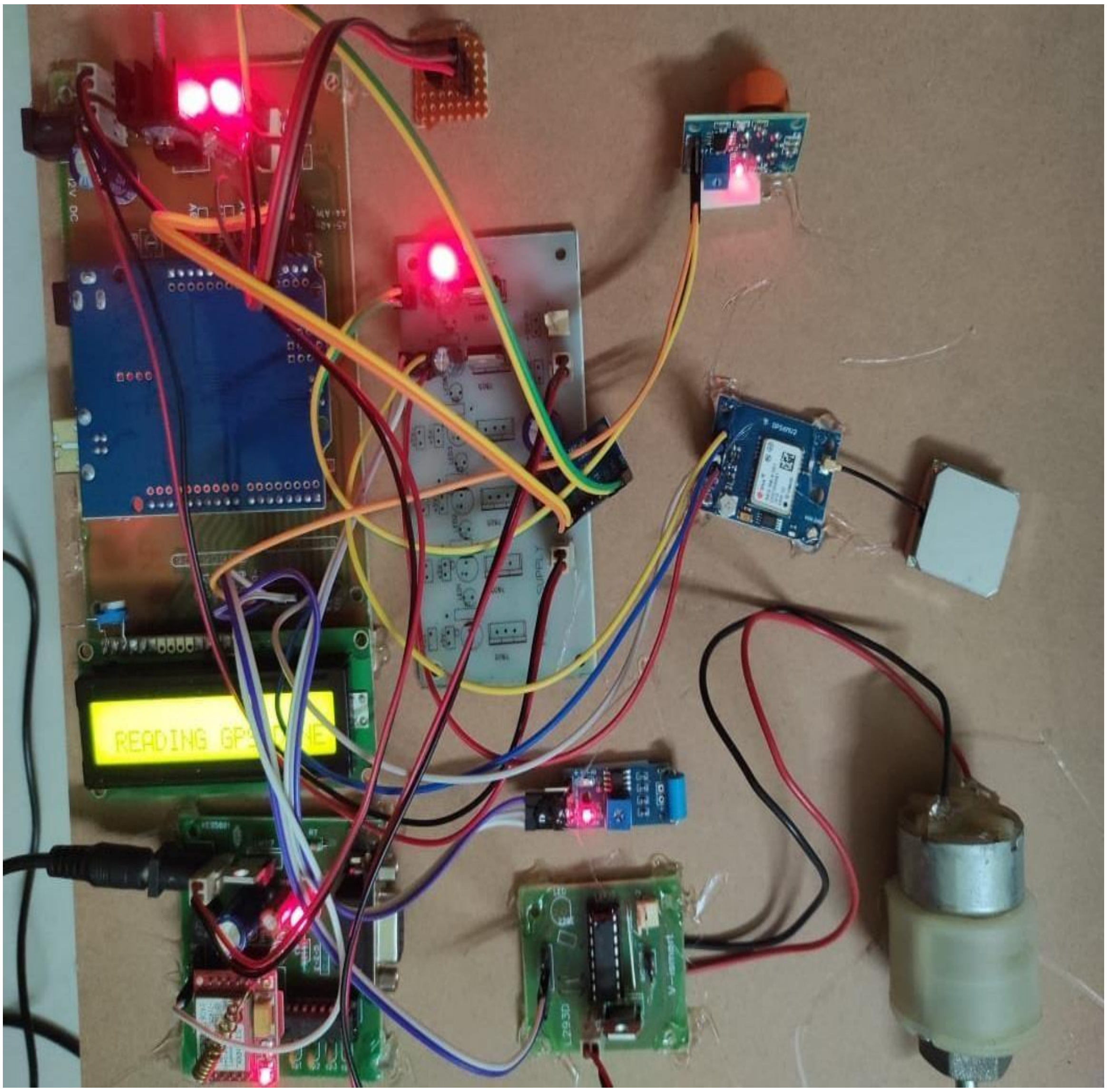


Figure 4.3: Hardware Setup

This hardware setup gives a brief idea as to how the components of the project “Wireless Blackbox for cars using sensors and gps module” were connected.

4.4 RESULT

After successfully developing and implementing the project, whenever the threshold limit of the alcohol sensor MQ3 exceeded or vibration is detected by the vibration sensor the following results (alert messages) were obtained.

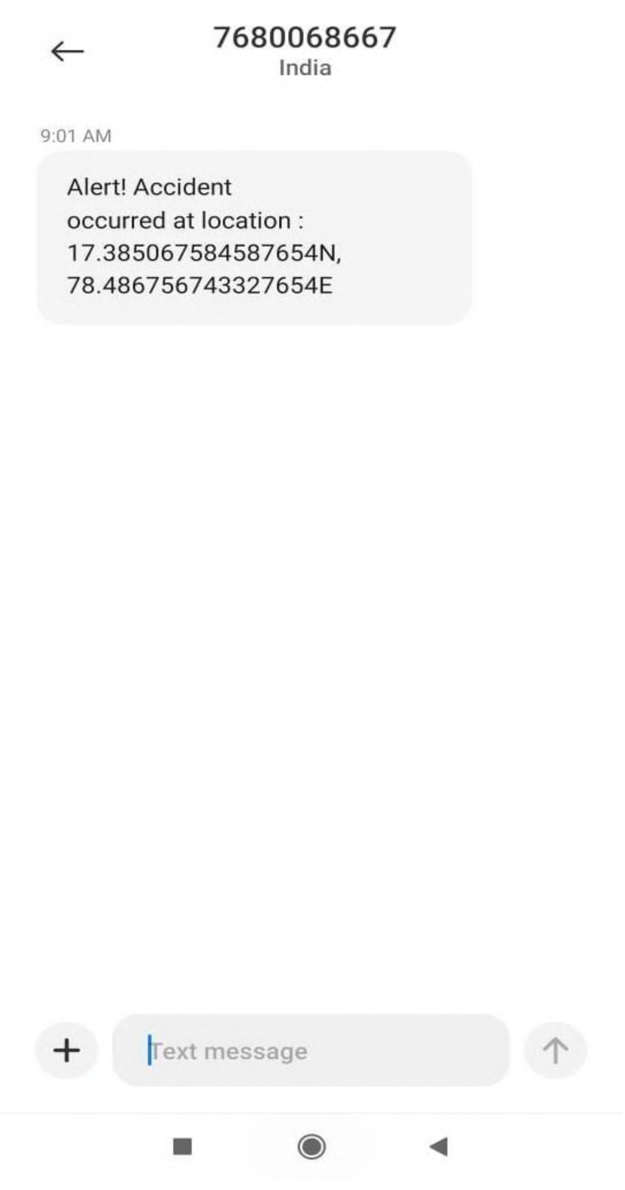


Figure 4.4: Alert message received when alcohol limit exceeded

Figure 4.5 Alert message received when accident occurred

CHAPTER 5

PROJECT TESTING

Input	Output
LM35 senses temperature	Temperature displayed on LCD
MQ3 detects alcohol & alcohol concentration exceeds 400pm	Sends alert message
Vibration sensor detects vibrations	Sends alert message
ADXL345 detects tilt	X,Y,Z co-ordinates(position) are displayed on the LCD

Table 5.0: Analyzing experimental data

- Initially the threshold detection level of sensors value is predefined.
- When the device is switched on, it displays “Wireless Blackbox”.
- Now the user is prompted to give an input such that it exceeds the threshold of the sensors.
- Once this happens the SMS is sent to registered mobile number with the location of the accident place.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION:

The sole purpose of this project was to develop a blackbox system for car accidental monitoring and alerting. The system has successfully overcome the drawbacks of the existing system by introducing alert messages.

Finally, We conclude that system wireless black box using sensors, GSM and GPS tracking has been developed for car accidental monitoring and alerting which gives an intelligent solution to the problem.

6.2 FUTURE ENHANCEMENT:

In future, by applying ultrasonic sensors features we can detect the distance of a vehicles moving nearby our vehicles. This system can be interfaced with vehicle airbag system that prevents vehicle occupants from striking interior objects such as steering wheel or window.

This can also be developed by interconnecting a camera to the controller module that takes the photograph of the accident spot that makes tracking easier.

PUBLICATION

We have published a paper on our project titled “Wireless Blackbox for cars using sensors and gps module” at Online Mega International Conference on “Smart Modernistic in Electronics and Communication” **(ICSMEC-21)** organized by Department of Electronics & Communication Engineering, St. Martin’s Engineering College **(ID :** ICSMEC21-0070)

REFERENCES

[1] Mr.Dinesh Kumar HSDK, Shreya Gupta, Sumeet Kumar, Sonali Srivastava, “ Accident Detection and Reporting System Using GPS and GSM Module”, Journal of Emerging Technologies and Innovative Research (JETIR), Volume 2, Issue 5, May 2015

[2] Shailesh Bhavthankar and Prof. H. G. Sayyed, “Wireless System for Vehicle Accident Detection and Reporting using Accelerometer and GPS”, International Journal of Scientific& Engineering Research (IJSER), Volume 6, Issue 8, August 2015

[3] P. Kaliuga Lakshmi and C. Thangamani , “An Efficient Vehicle Accident Detection Using Sensor Technology”, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET),Volume 5, Issue 3, March 2016

[4]. M. Lu, W. chen, X. Shen, H.C. Lam and J. Liu, "Positioning and tracking construction vehicle in highly dense urban areas and building Construction sites," Automation in Construction, vol. 16, issue 5.pp.647-656, August 2007

[5] https://nevonprojects.com/wireless-black-box-for-cars-using-sensors-gps-module/

[6]https://www.researchgate.net/publication/314836750_Installation_Of_Black_Box_In_Automobile#:~:text=5.2Disadvantages%201.As%20the,Initial%20cost%20is%20high

[7] https://www.howstuffworks.com/gadgets/other-gadgets/black-box.html

[8] https://spectrum.ieee.org/aerospace/aviation/beyond-the-black-box/0

BIBILOGRAPHY

[1] Abdallah Kassem, Rabih Jabr, Ghady Salamouni and Ziad Khairallah Maalouf, "Vehicle Black Box System", IEEE International Systems Conference, April 2008

[2] P. Ajay Kumar Reddy, P.Dileep Kumar, K. Bhaskarreddy, E. Venkataramana and M.Chandrasekhar Reddy, "Black Box For Vehicles", International Journal of Engineering Inventions (IJEI), Volume 1, Issue 7, October 2012

[3] Sri Krishna Chaitanya Varma, Poornesh, Tarun Varma and Harsha , "Automatic Vehicle Accident Detection And Messaging System Using GPS and GSM Modems", International Journal of Scientific & Engineering Research (IJSER), Volume 4, Issue 8, August 2013

[4] Shaik Khadar Basha and P Sireesh Babu , "Wireless Black Box Report for Tracking of Accidental Monitoring In Vehicles", International Journal Of Professional Engineering Studies (IJPRES),Volume 1, Issue 2, Dec 2013

[5] Rajashri R. Lokhande and Sachin P. Gawate, "Design & Implementation of Vehicle Black Box For Driver Assistance And Alert", IOSR Journal of Computer Science (IOSR-JCE), 2014

[6] Prof. Ashish B. Dudhale, Steve Felix S, Harsha Phatak and Sayali Jathar, "Car Black Box System for Accident Prediction and Crash Recovery", International Journal of Engineering Science and Computing (IJESC), May 2014

[7] Ramchandra Patil and Shivaraj Hublikar, "Design and Implementation of Car Black Box with Collision Avoidance System using ARM" ,International Journal of Innovative Technology and Exploring Engineering (IJITEE), Volume4, Issue3, August 2014

[8] Vikram Singh Kushwaha , DeepaYadav , Abusayeed Topinkatti and Amrita Kumari, "Car Accident Detection System Using GPS and GSM", International Journal of Emerging Trend in Engineering and Basic Sciences (IJEEBS),Volume 2 , Issue 1, Jan- Feb 2015

[9] https://news.discovery.com/tech/beyond-the-black-box.html

[10] https://article.timesofindia.indiatimes.com/2012-07-14/india

APPENDIX

Figure 5.0: Ideal View of Blackbox System

A

PROJECT REPORT

On

ULTRASONIC NAVIGATION STICK FOR THE BLIND USING ARDUINO

Submitted by

Ms B.Pratyusha (17K81A0405)

Ms R.Sai Spurthy (17K81A0441)

Ms S.Manvitha (17K81A0446)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Mrs.A.Soumya, M.Tech

(Assistant Professor)

DEPARTMENT OF ELCTRONICS AND

COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad - 500100

NBA & NAAC A+ ACCREDITED

BONAFIDE CERTIFICATE

THIS IS TO CERTIFY THAT THE MAJOR-PROJECT WORK ENTITLED "**ULTRASONIC NAVIGATION STICK FOR THE BLIND USING AUDINO**" IS A BONAFIDE WORK CARRIED OUT BY **B.PRATYUSHA**, **R.SAI SPURTHY, S.MANVITHA**, BEARING ROLL NO: 17K81A0405, 17K81A0441, 17K81A0446 IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF TECHNOLOGY IN ELECTRONICS & COMMUNICATION ENGINEERING BY THE JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD DURING THE ACADEMIC YEAR 2020-2021.

THE RESULTS EMBODIED IN THIS REPORT HAVE NOT BEEN SUBMITTED TO ANY OTHER UNIVERSITY OR INSTITUTION FOR THE AWARD OF ANY DEGREE.

Internal guide
Mrs A.Soumya
Assistant Professor

Head of the department
Dr.B.Hari krishna
Professor

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **B. PRATYUSHA** WITH ROLL NO.**17K81A0405, MANVITHA SEDHARI** WITH ROLL NO.**17K81A0446**, **R.SAI SPURTHY** WITH ROLL NO.**17K81A0441**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**ULTRASONIC NAVIGATION STICK FOR THE BLIND USING ARDUINO**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of **Bachelor of Technology** in Department of **'Electronics and Communication Engineering**', session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled '**Ultrasonic stick for the blind using Arduino'** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

B.Pratyusha	**R.Sai Spurthy**	**S.Manvitha**
(17K81A0405)	**(17K81A0441)**	**(17K81A0446)**

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr. B. Hari Krishna**, **Head &Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our **Department Project Coordinator Dr. A. Anand Kumar** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our class project coordinator and our project internal guide **Mrs.A.Soumya,** Assistant Professor, Department of ECE for guidance and encouragement in carrying out this project work.

1. B.Pratyusha (17K81A0405)
2. R.Sai Spurthy (17K81A0441)
3. S.Manvitha (17K81A0446)

ABSTRACT

Blind stick is an innovative stick designed for visually disabled people for improved navigation. We here propose an advanced blind stick that allows visually challenged people to navigate with ease using advanced technology. The blind stick is integrated with ultrasonic sensor along with light and water sensing. Our proposed project first uses ultrasonic sensors to detect obstacles ahead using ultrasonic waves. On sensing obstacles the sensor passes this data to the microcontroller. The microcontroller then processes this data and calculates if the obstacle is close enough. If the obstacle is not that close the circuit does nothing. If the obstacle is close the microcontroller sends a signal to sound a buzzer. It also detects and sounds a different buzzer if it where detects water and alerts the blind. One more feature is that it allows the blind to detect if there is light or darkness in the room. The system has one more advanced feature integrated to help the blind find their stick if they forget they kept it. A wireless RF based remote is used for this purpose. Pressing the remote button sounds a buzzer on the stick which helps the blind person to find their stick. Thus this system allows for obstacle detection as well as finding stick if misplaced by visually disabled people.

TABLE OF CONTENTS

.

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.0 Overview of the project:

Blindness is a condition of lacking the visual perception due to physiological or neurological factors partially or fully. The main concept of the paper is to provide an electronic aid as guidance to overcome the lacking of their visualization power by proposing a simple, efficient, configurable electronic guidance system for blind and visually impaired pedestrians. Ultrasonic Sensor is the proposed electronic aid which senses the obstacles in its path by continuously transmitting the ultrasonic waves.

When an obstacle appears in its vicinity then the ultrasonic waves get reflected to the system immediately. And then ultrasonic receiver senses these ultrasonic waves. This method supports the microcontroller to obtain the information from ultrasonic waves and then it alerts the blind pedestrians through voice message.

The advantage of our proposed system is its voice-based announcement for easy navigation which can assist a blind pedestrian to pass through a busy road. Moreover, this system is an auditory guidance system for the visually impaired pedestrians using ultrasonic-to-audio signal transformation

In this project, the system is designed in such a way that it gathers data about the environment via ultrasonic sensors and extracts the visual information from that data. This visual information is then transformed into an audio signal immediately and the blind pedestrian can recognize the environmental information through binaural sound generated by the system.

The whole operation of the project can be described by using block diagram which is a graphical method for explaining the concept of the system without the presence of the individual components within the project. Figure 1 represents the basic block diagram of the whole system of project.

1.1 Objectives of the Study:

The main objective is to help visually challenged people to navigate with ease using advance technology. In this technology controlled world, where people strive to live independently, this project proposes an ultrasonic stick for blind people to help them gain personal independence. Since this is economical and not bulky, one can make use of it easily.

1.2 Scope of the Study:

The system can be supplemented with ultrasonic sensor which can be used in detecting obstacles and we can also provide a buzzer for alerting a person.We can also add speakers to store voice notes and play accordingly.

1.3 Material Requirement:

Component	Specification	Quantity
Ultrasonic Sensor	HC -SR04	1
Arduino	UNO	1
Buzzer	5-Volt	1
LCD	16 x 2	1
Battery	9 Volt	1

Table 1 : Materials Required for the project

1.4 Procument of Equipment:

This project is developed in such a way that it is efficient and affordable in a simple way. So,all the components required to implement the system is available in the market in low prize. Firstly,we made a list of the components and looked for the places and prioritized the places with good quality and of low cost.Next we ordered all the required things . After receiving all the equipments, we started the implementation of the project.

1.5 Organization of Chapters:

1.5.0 Introduction:

Visually impaired persons have difficulty to interact and feel their environment. They have little contact with surroundings.

Physical movement is a challenge for visually impaired persons, because it can become tricky to distinguish obstacles appearing in front of them, and they are not able to move from one place to another. They depend on their families for mobility and financial support. Their mobility opposes them from interacting with people and social activities.

In the past, different systems are designed with limitations without a solid understanding of the

nonvisual perception. Researchers have spent the decades to develop an intelligent and smart stick to assist and alert visually impaired persons from obstacles .Over the last decades, research has been conducted for new devices to design a good and reliable system for visually impaired persons to detect obstacles and warn them at danger places.

Smart walking stick is specially designed to detect obstacles which may help the blind to navigate care-free. The audio messages will keep the user alert and considerably reduce accidents. A voice enabled automatic switching is also incorporated to help them in private space as well. This system presents a concept to provide a smart electronic aid for blind people, both in public and private space The proposed system detects the obstacle images which are present in outdoor and indoor with the help of a sensor. The Stick measures the distance between the objects and smart walking stick by using an ultrasonic sensor. When any objects or obstacles come in range of an ultrasonic sensor then the buzzer beeps. The smart walking stick is a simple and purely mechanical device to detect the obstacles on the ground. This device is light in weight and portable. But its range is limited due to its own size. It provides the best travel aid for the person. The blind person can move from one place to another independently without the others help. The main aim of the system is to provide a efficient navigation aid for the blind persons which gives a sense of vision by providing the information about their surroundings and objects around them.

1.5.1 Literature Survey:

Nowadays, the wearable health monitoring system is the main application of Internet of things Likewise lots of wearable devices are designed for visually impaired people. Few systems are discussed here. In sensor assisted stick for the blind people describes about a wearable equipment which consists of a light weight blind stick.

the obstacle detection circuit based on a sensor. It is mainly developed to help the blind person to move alone safely from one place to another and to avoid any obstacles that may be encountered. The device detects the fixed as well as moving objects and thus it may help to avoid accidents. The main component for the working of this system is the infrared sensor which is used to scan a predetermined area around the blind person by emitting-reflecting waves. The reflected signals are received from the objects are used as inputs to the microcontroller and then used for determining the direction and distance of the objects around the blind person.

The main objective of this is to provide an application for blind people to detect the obstacles in various directions, detecting pits and manholes on the ground to make free to walk.

An innovative stick is designed for the visually disabled people for their easy navigation. The blind stick is able to detect the water by integrating with ultrasonic sensor. In this system, the ultrasonic sensors are used to detect obstacles by using ultrasonic waves. By sensing the obstacles the sensor passes the received data to the microcontroller. The microcontroller processes the data and calculates if the obstacle is close enough to the person. If the obstacle is not close to the microcontroller, the circuit does not do anything. If the obstacle is close enough to the microcontroller, it sends a signal to buzzer. The system also detects water and provides different sounds and alerts the blind person.

Multitasking stick is designed to indicate safe path to visually disable people. The micro-controller based automated hardware allows a blind person to detect obstacles in front of them. The hardware part consists of a micro-controller which was incorporated with an ultrasonic sensor, voice play back module and additional equipment. The ultrasonic waves are used to detect the obstacles. The temperature sensors are provided to detect the fire or high temperature area. The presence of water is detected using the current sensing principle. The acknowledgement from the sensing obstacle is received through the voice play back module. The system is provided with RF module to find the misplaced stick. These features allow the blind people to move from one place to another independently and easily.

1.5.2 Project Design:

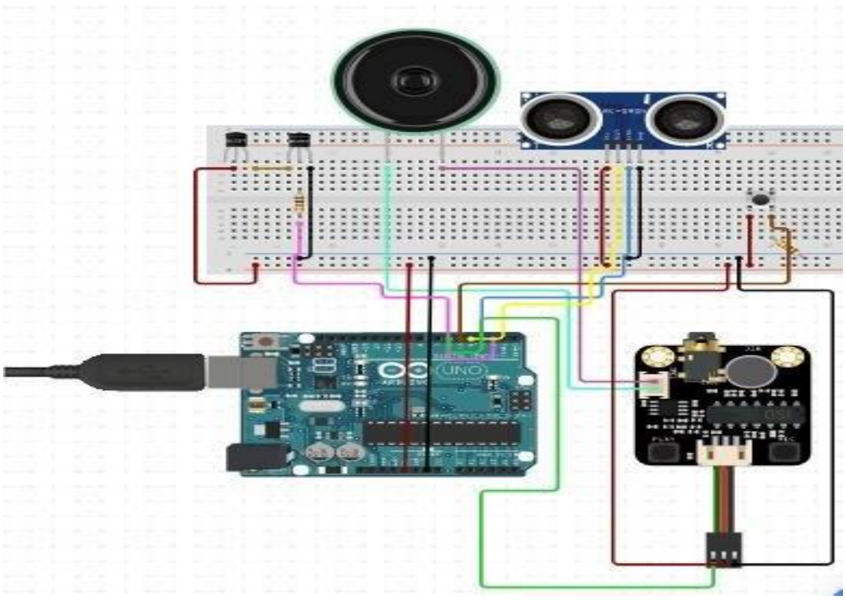


Figure 1.0 : Schematic design of the Project

1.5.3 Project Implementation:

The proposed system provides improvements to the existing system design. It tries to make the existing system more efficient, convenient and user- friendly. The implementation of the proposed design of the stick requires following hardware components:

1. Ultrasonic sensor
2. Arduino
3. Buzzer
4. APR9600

1.5.4 Project Testing:

The purpose of the testing phase is to evaluate and test declared requirements, features, and expectations regarding the project prior to its delivery in order to ensure the project matches initial requirements stated in specification documents.

1.5.5 Conclusion and Future Enhancement:

The smart walking stick, constructed with at most accuracy, will help the blind people to move from one place to another without others help. This could also be considered a crude way of giving the blind a sense of vision. This stick reduces the dependency of visually impaired people on other family members, friends and guide dogs while walking around. The smart stick detects objects or obstacles in front of users and feeds warning back, in the form of beep. The advantage of the system lies in the fact that it can prove to be a low cost solution to millions of blind person worldwide.Hence, we can conclude that there are a number of techniques for making a ultrasonic blind walking stick for blind people. The advantage of the system lies in the fact that it can prove to be a very low cost solution to millions of blind person worldwide. Our prototype can detect the obstacles of 3 metres range.The system can be supplemented with actual GPS module used in cars and we can provide a vibrator for the partially deaf person. This makes the system further more compact. A wall following function can also be added so that the user can walk straight along a corridor in an indoor environment.

CHAPTER II

CHAPTER II
LITERATURE SURVEY

2.0 Literature Review on Research Area:

The authors Ashraf Anwar and Sultan Aljahdali in their paper used ultrasonic sensor for detecting the obstacle coming in the path of the blind person, also they have used heat and moisture sensor for giving more details to the person with the stick about the road or path. For alarming the blind person they are using different buzzers and sensors that will tell the person that something is there in front of him/her. The only disadvantage of this system is that GPS is not included in this system.

In the paper named 'Implementation and Design of Smart Blind Stick for Obstacle Detection and Navigation System' whose authors are K.S.Manikanta, T. S. S. Phani , A. Pravin used Buzzers and speakers in their model which helped the visually impaired person to stay alert on the roads without the help of others. As moisture sensor is not included in the model so it cannot detect water present in the path of the person using that stick.

2.1 Review on Related Literature:

This section describes appropriate related works on the development of smart canes intended for visually-impaired people. According to [1], technology can help in reducing manybarriers that people with disabilities face. These kinds of technologies are referred to as assistive technology (AT). There are many types of disabilities, including physical disabilities, hearing-impaired, and visually-impaired. A T has been utilized in assisting them [4, 5, 6, 9]. However,developing an AT is expensive [3], making their selling price high.

The GuideCane [9] is designed to help the visually-impaired users navigate safely and quickly. and other hazards. GuideCane is used like the widely used white cane, where the user holds the GuideCane in front of the user while walking. The GuideCane is considerably heavier than the white cane, because it uses a s ervo motor. The wheels are equipped with encoders to determine the relative motion. The servo motor, controlled by the built-in computer, can steer the wheels left and right relative to the cane. To detect obstacles, the GuideCane is equipped with ten ultrasonic mini joystick located at the handle allows the user to specify a desired direction of motion. GuideCane is far heavier than the ordinary white cane and also it is hard to keep because it cannot be folded.

2.2 Conclusions on Reviews:

So,There by we conclude that after going through all the previously published papers ,we decided to implement an efficient and affordable smart stick in a simplest way.

For this,we'll add a sensor to detect the obstacle and also a buzzer so that it can alert the person using the kit and we can also supplement it with APR9600,so that he can add some voice notes.

CHAPTER III

CHAPTER III
PROJECT DESIGN

3.0 Overview of the Design:

The aim of this project is to investigate the development of a navigation aid for blind and visually impaired People. It is based on a microcontroller with synthetic speech output. This aid is portable and gives information to the user about urban walking routes to point out what decisions to make. On the other hand, and in order to reduce navigation difficulties of the blind, an obstacle detection system using ultrasounds added to this device. Blind stick is an innovative stick designed for visually disabled people for improved navigation. We here propose an advanced blind stick that allows visually challenged people to navigate with ease using advanced technology. The blind stick is integrated with ultrasonic sensor and a buzzing system. The brain of the circuit is Arduino Uno MCU board (Board1). Ultrasonic sensor "HC-SR04" is used for obstacle detection using ultrasonic waves. This sensor require a power supply of 3.3V each to operate up to a distance of 3 m and can detect obstacles within an average angle of 25 degrees in the sphere.

POWER SUPPLY

3.1 Block Diagram

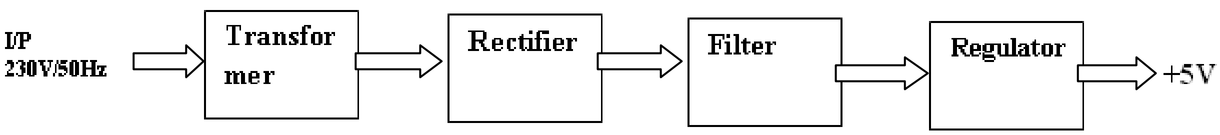


Figure 3.1 Power Supply

Circuit Diagram

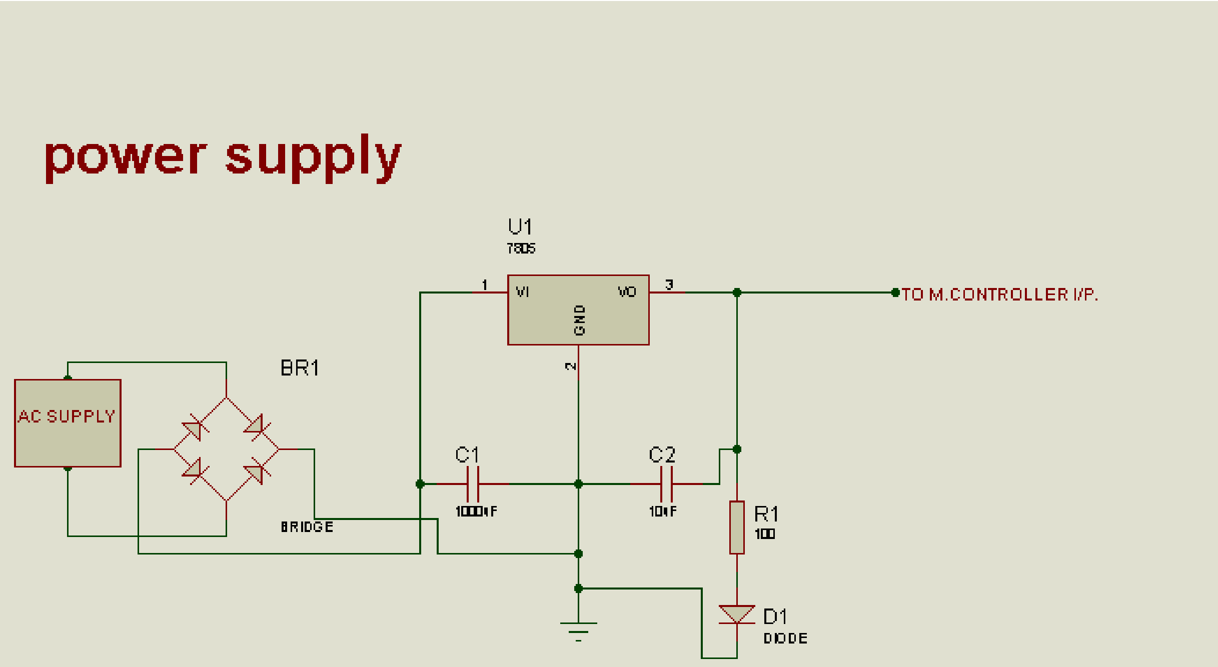


Figure 3.2 Circuit Diagram

Description

3.2.1 Transformer

A **transformer** is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

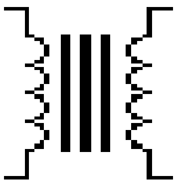


Figure: Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

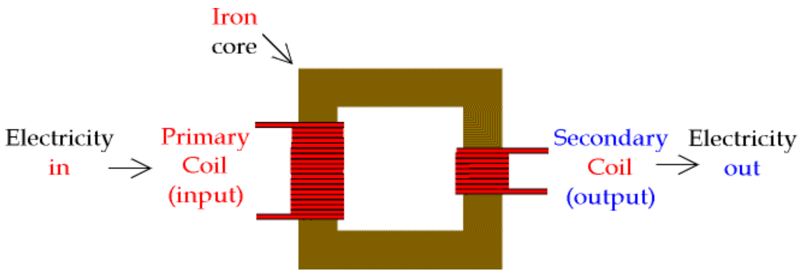


Figure : Transformer

Basic Principle:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

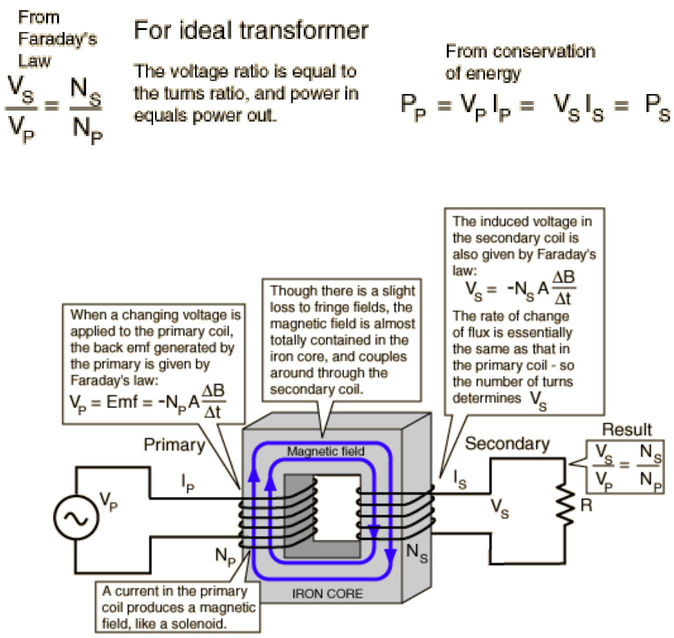


Figure: Basic Principle

Transformer Working

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

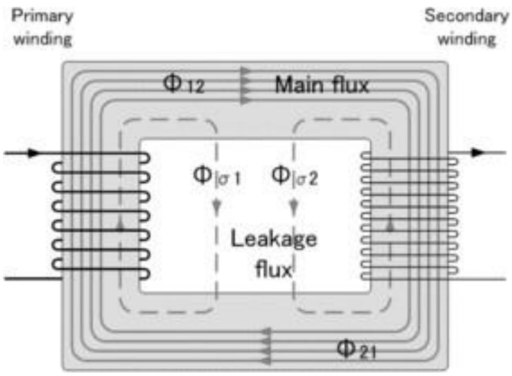


Figure: Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f.

The iron core is laminated to prevent 'eddy currents' flowing in the core. These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

Classification of Transformer:

- Step-Up Transformer
- Step-Down Transformer

Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn's ratio of the two sets of windings determines the amount of voltage transformation.

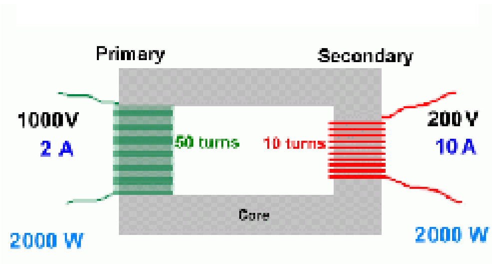


Figure: Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant.

Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single-phase step-down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase step up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

Step-Up Transformer:

A step-up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply.

A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

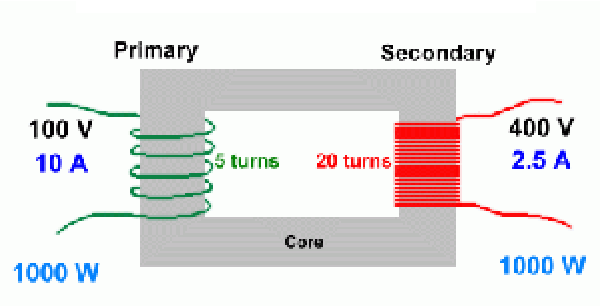


Figure: Step-Up Transformer

Applications

Generally, these **Step-Up Transformers** are used in industries applications only.

Turns Ratio and Voltage

The ratio of the number of turns on the primary and secondary coils determines the ratio of the voltages...

$$\frac{V_s}{V_p} = \frac{N_s}{N_p}$$

...where V_p is the primary (input) voltage, V_s is the secondary (output) voltage, N_p is the number of turns on the primary coil, and N_s is the number of turns on the secondary coil.

3.2.2 Diodes

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

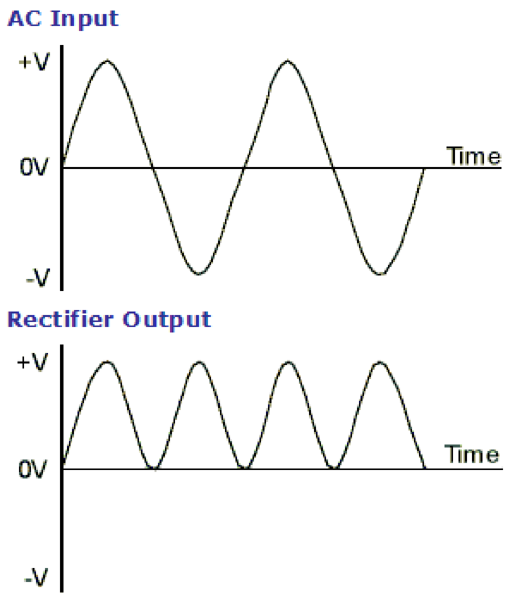


Figure Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased.

While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

Capacitor Filter

The **capacitor-input filter**, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

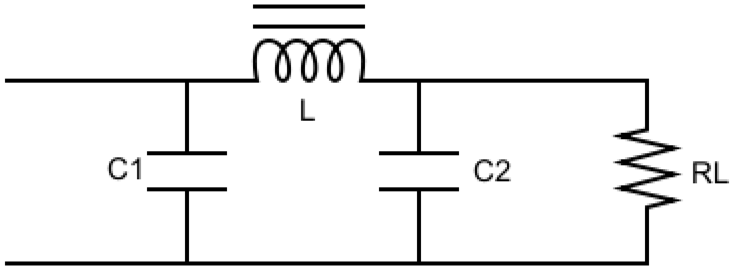


Figure: Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result the DC component flows through the inductor while the AC component is blocked.

1. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result only the DC component appears across the load RL.

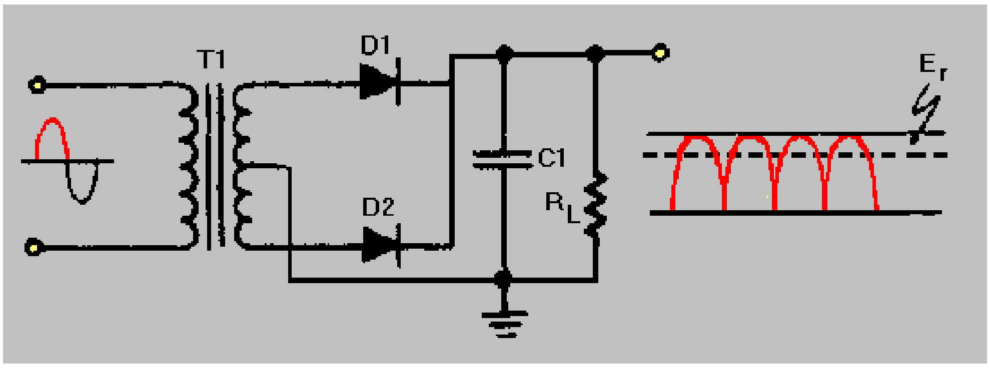


Figure: Centered Tapped Full-Wave Rectifier with a Capacitor Filter

Voltage Regulator:

A **voltage regulator** is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components.

Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the p ositive series and 'xx'indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx'indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consists the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

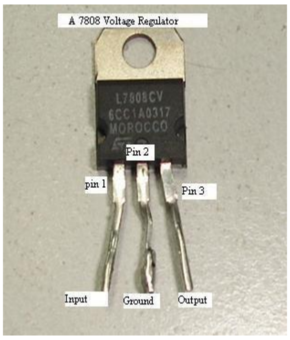


Figure : Regulator

3.1 Equipment Analysis:

Component	Specification	Quantity
Ultrasonic Sensor	HC -SR04	1
APR9600	------	1
Arduino	UNO	1
Buzzer	5-Volt	1
LCD	16 x 2	1
Battery	9 Volt	1

Table 2 : Equipment Analysis

3.2 Define the Modules:

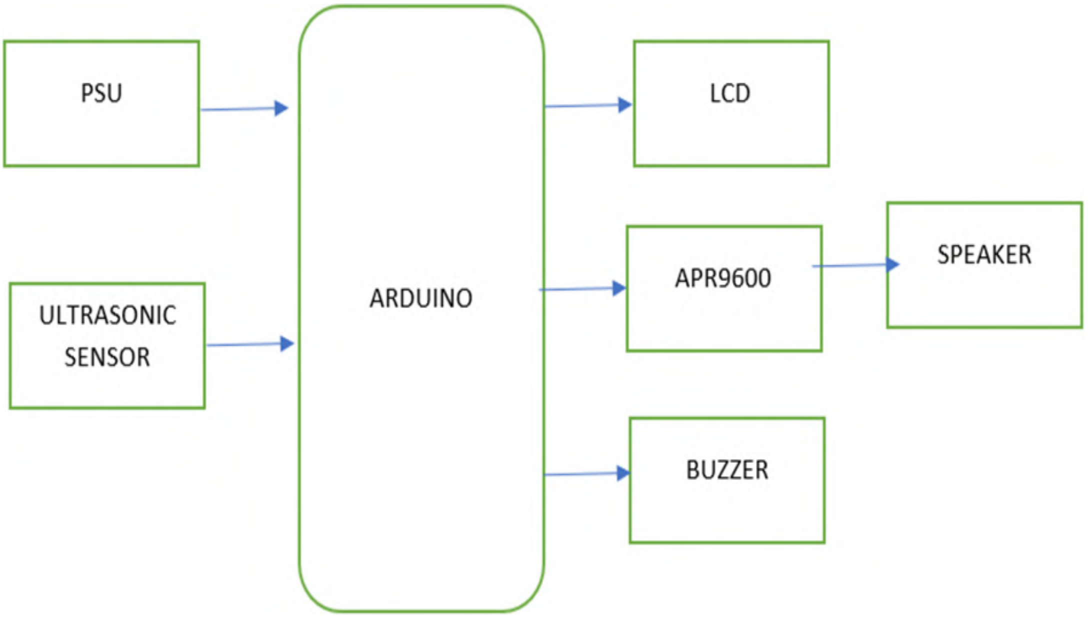


Figure 3.2.0: Block Diagram

Arduino Uno:

Arduino/Genuino **Uno** is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.

Table 3.2.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.2.2 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

3.2.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

3.2.4 Arduino boards based on AT91SAM3X8E microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

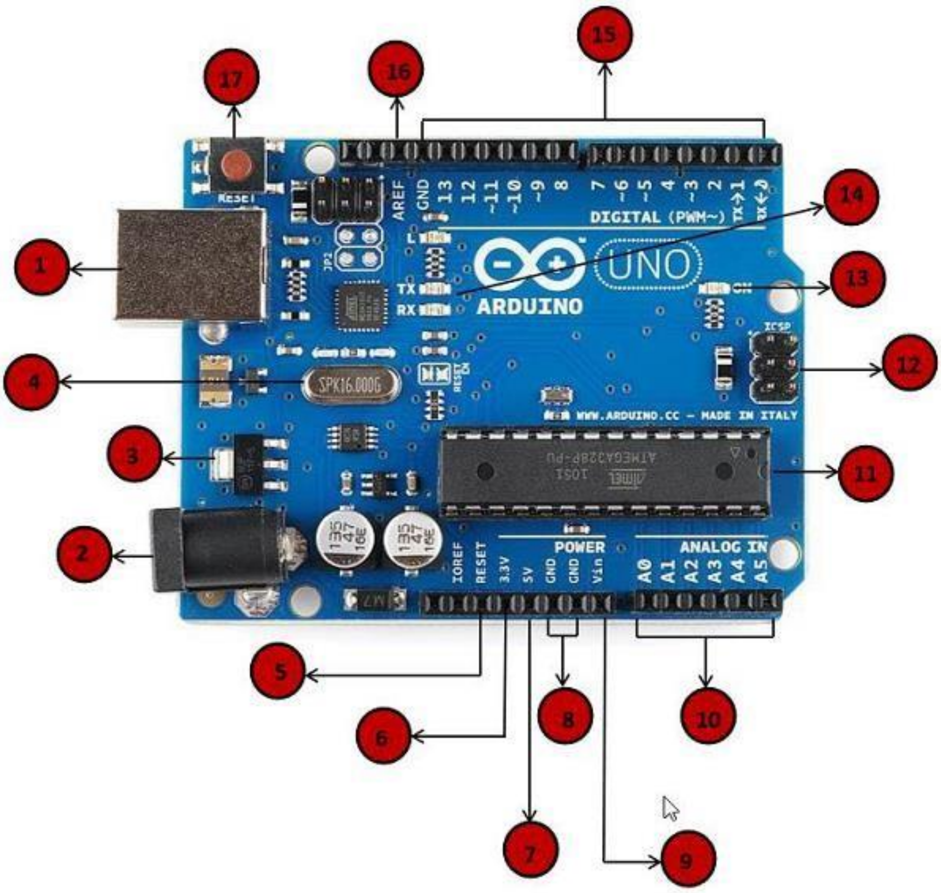


Figure : Arduino

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an

	"expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Ultrasonic Sensor:

An **ultrasonic sensor** is an electronic device that measures the distance of a target object by emitting **ultrasonic** sound waves, and converts the reflected sound into an electrical signal. **Ultrasonic** waves travel faster than the speed of audible sound (i.e. the sound that humans can hear).

Buzzer:

A buzzer or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices,timers,and confirmation of user input such as a mouse click or keystroke.

APR9600:

APR9600 is a low-cost high performance sound record/replay IC incorporating flash analogue storage technique. .... Total sound recording time can be varied from 32 seconds to 60 seconds by changing the value of a single resistor.

3.3 Module Functionalities:

Arduino Uno:

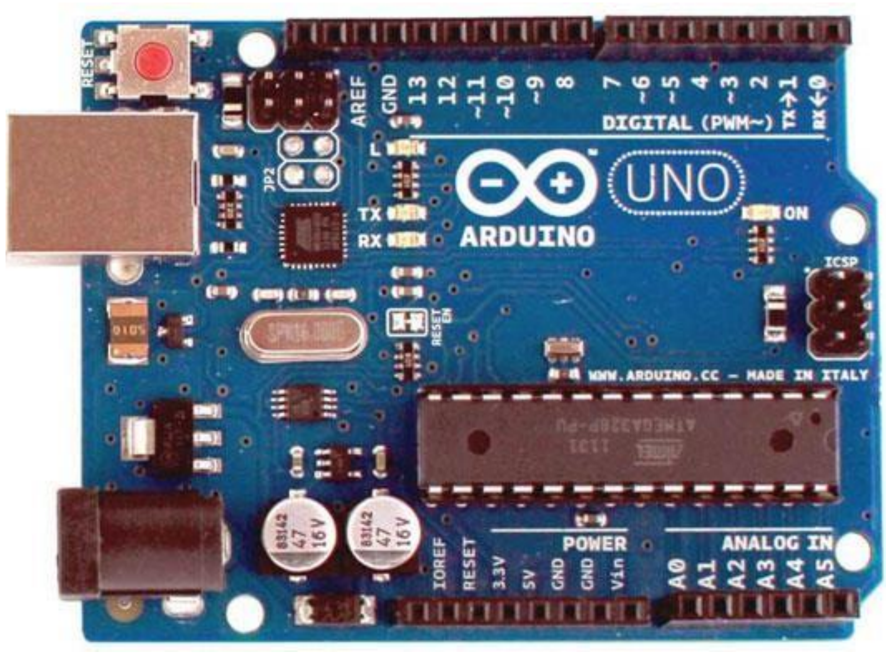


FIGURE 3.3 : Arduino

The Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.

It contains everything needed to support the microcontroller. Page to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

The Arduino Uno is programmed using the Arduino Software(IDE), our Integrated Development Environment common to all our boards and running both online and offline.

Ultrasonic Sensor:

Figure : 3.4

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

Ultrasonic Sensor Pin Configuration

Pin Number	**Pin Name**	**Description**
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz

Equivalent distance measuring Sensors

US transmitter Receiver pair, IR sensor module, IR sensor pair, IR Analog distance sensor,

HC-SR04 Ultrasonic Sensor - Working

As shown above the **HC-SR04 Ultrasonic (US) sensor** is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below

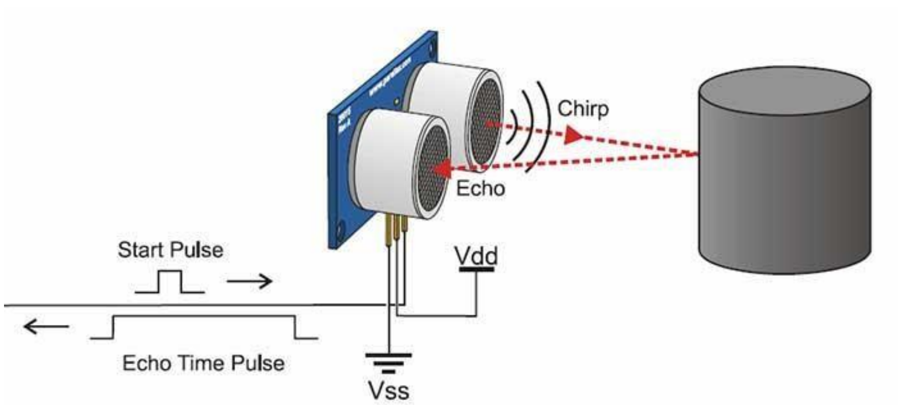


Figure : 3.5

Ultrasonic sensors work by sending out a sound wave at a frequency above the range of human hearing. The transducer of the sensor acts as a microphone to receive and send the ultrasonic sound. Our ultrasonic sensors, like many others, use a single transducer to send a pulse and to receive the echo. The sensor determines the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse.

The working principle of this module is simple. It sends an ultrasonic pulse out at 40kHz which travels through the air and if there is an obstacle or object, it will bounce back to the sensor. By calculating the travel time and the speed of sound, the distance can be calculated.

Ultrasonic sensors are a great solution for the detection of clear objects. For liquid level measurement, applications that use infrared sensors, for instance, struggle with this particular use case because of target translucence.

For presence detection, ultrasonic sensors detect objects regardless of the color, surface, or material (unless the material is very soft like wool, as it would absorb sound.)

To detect transparent and other items where optical technologies may fail, ultrasonic sensors are a reliable choice.

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

How to use the HC-SR04 Ultrasonic Sensor

HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used.

Power the Sensor using a regulated +5V through the Vcc ad Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off

This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor.

The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information the distance is measured as explained in the above heading.

Applications

- Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
- Used to measure the distance within a wide range of 2cm to 400cm
- Can be used to map the objects surrounding the sensor by rotating it
- Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water

Buzzer

1. Magnetic Transducer

Magnetic transducers contain a magnetic circuit consisting of a iron core with a wound coil and a yoke plate, a permanent magnet and a vibrating diaphragm with a movable iron piece. The diaphragm is slightly pulled towards the top of the core by the magnet's magnetic field. When a positive AC signal is applied, the current flowing through the excitation coil produces a fluctuating magnetic field, which causes the diaphragm to vibrate up and down, thus vibrating air. Resonance amplifies vibration through resonator consisting of sound hole(s) and cavity and produces a loud sound.

2. Magnetic Buzzer (Sounder)

Buzzers like the TMB-series are magnetic audible signal devices with built-in oscillating circuits. The construction combines an oscillation circuit unit with a detection coil, a drive coil and a magnetic transducer. Transistors, resistors, diodes and other small devices act as circuit devices for driving sound generators. With the application of voltage, current flows to the drive coil on primary side and to the detection coil on the secondary side.

The amplification circuit, including the transistor and the feedback circuit, causes vibration. The oscillation current excites the coil and the unit generates an AC magnetic field corresponding to an oscillation frequency.

This AC magnetic field magnetizes the yoke comprising the magnetic circuit. The oscillation from the intermittent magnetization prompts the vibration diaphragm to vibrate up and down, generating buzzer sounds through the resonator.

Specifications:

Rated Voltage: A magnetic buzzer is driven by 1/2 square waves (V o-p).
Operating Voltage: For normal operating. But it is not guaranteed to make the minimum Sound Pressure Level (SPL) under the rated voltage.
Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.
Direct Current Resistance: The direct current resistance is measured by ammeter directly.
Sound Output: The sound output is measured by decibel meter. Applying rated voltage and 1/2 square waves, and the distance of 10 cm.
Rated Frequency: A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.
Operating Temp: Keep working well between -30℃ and +70℃.
Driving methods: AX series with built drive circuit will be the best choice when we cannot provide frequency signal to a buzzer, it only needs direct current.
Dimension: Dimension affects frequency, small size result in high frequency.
Voltage: Depend on V o-p (1/2 square waves)
Fixed methods: From the highest cost to the lowest- DIP, wires/ connector, SMD.
Soldering methods: AS series is soldered by hand, the frequency is lower because of the holes on the bottom. On the other hand, we suggest AC series for the reflow soldering, the reliability is better.

How to choose a buzzer

There are many different kinds of buzzer to choose, first we need to know a few parameters, such as voltage, current, drive method, dimension, mounting type, and the most important thing is how much SPL and frequency we want.

Operating voltage: Normally, the operating voltage for a magnetic buzzer is from 1.5V to 24V, for a piezo buzzer is from 3V to 220V. However, in order to get enough SPL, we suggest giving at least 9V to drive a piezo buzzer.

Consumption current: According to the different voltage, the consumption current of a magnetic buzzer is from dozens to hundreds of mill amperes; oppositely, the piezo type saves much more electricity, only needs a few mill amperes, and consumes three times current when the buzzer start to work.

Driving method: Both magnetic and piezo buzzer have self drive type to choose. Because of the internal set drive circuit, the self drive buzzer can emit sound as long as connecting with the direct current. Due to the different work principle, the magnetic buzzer need to be driven by 1/2 square waves, and the piezo buzzer need square waves to get better sound output.

Dimension: The dimension of the buzzer affects its SPL and the frequency, the dimension of the magnetic buzzer is from 7 mm to 25 mm; the piezo buzzer is from 12 mm to 50 mm, or even bigger.

Connecting way: Dip type, Wire type, SMD type, and screwed type for big piezo buzzer are usually seen.

Sound Pressure Level (SPL): Buzzer is usually tested the SPL at the distance of 10 cm, if distance double, the SPL will decay about 6 dB; oppositely, the SPL will increase 6 dB when the distance is shortened by one time. The SPL of the magnetic buzzer can reach to around 85 dB/ 10 cm; the piezo buzzer can be designed to emit very loud sound, for example, the common siren, are mostly made of piezo buzzer.

Figure 3.6 Buzzer

APR9600 voice features :

- ✓ Single-chip, high quality voice recording & playback solution
- o No external Ics required
- o Minimum external component

- o Random access of multiple fixed duration messages
- o Sequential access of multiple variable duration messages
- ✓ User friendly, easy to use operation
- o Programming & development systems not required
- o Level activated recording & edge activated playback switches
- ✓ Low power consumption
- o Operating current: 25mA typically
- o Standby current: 1uA typically
- o Automatic power down
- ✓ Chip enable pin for simple message expansion

General Description:

The APR9600 device offers true single chip voice recording, non volatile storage, and playback capability for 40 to 60 seconds.

The device supports both random and sequential access of multiple messages. Sample rates are user selectable, allowing designers to customize their design for unqiue quality and storage time needs. Integrated output amplifier, microphone amplifier, and AGC circuits greatly simplify system design. The device is ideal for use in portable voicce recorders, toys, and many other consumer and industrial applications.

APLUS integrated achieves these high levels of storage capability by using its proprietary analog/multilevel storage technology enables the APR9600 device to reproduce voice signals in their natural form. It eliminates the need for encoding and compression, which often introduce distortion.

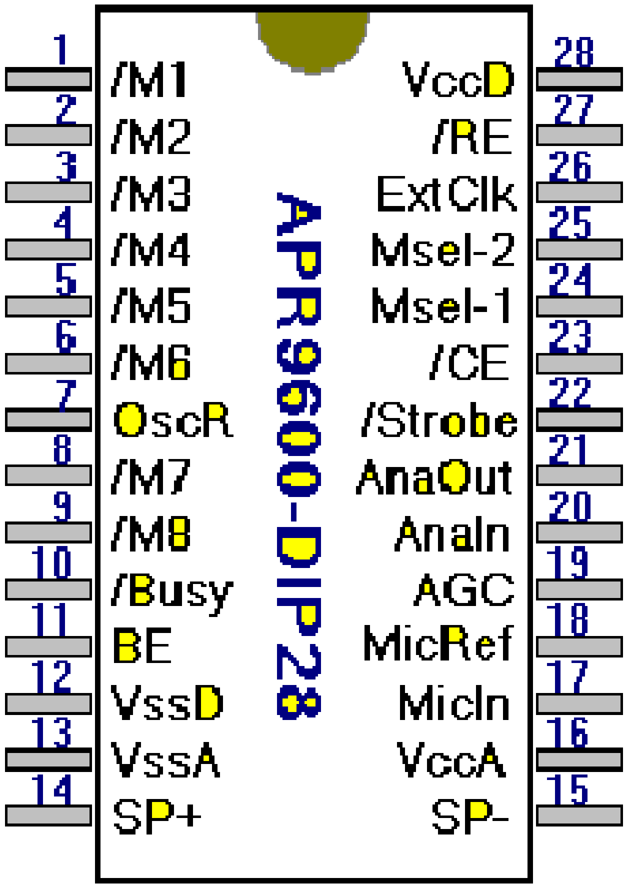


Figure : 3.7 The APR9600 DIP

Functional Description:

APR9600 block diagram is included in order to describe the device's internal architure. At the left hand side of the diagram are the analog inputs. A differntial microphone amplifier, including integrated AGC, is included on-chip for applications requiring use. The amplified microphone signals fed into the device by connecting the ANA_OUT pin to the ANA_IN pin through an external DC blocking capacitor. Recording can be fed directly into the ANA_IN pin through a DC blocking capacitor, however, the connection between ANA_IN and ANA_OUT is still required for playback. The next block encountered by the input signal is the internal anti alising filter.

The filter automatically adjust its response according to the sampling frequency selected so Shannon;s Sampling Theorem is satisfied. After anti aliasing filtering is accomplished the signal is ready to be clocked into the memory arry. This storage is accomplished through a combination of the sample and hold circuit and the analog write/read circuit. These circuits are clocked by either the internal oscillator or an external clock source. When playback is desired the previously stored recording is retrived from memory, low pass filtered, and amplified as shown on the right hand side of the diagram. The signal can be heard by connecting a speaker to the SP+ and SP- pins.

Chip wide management is accomplished through the device control block shown in the upper right hand corner. Message management is provided through the message control block represented in the lower center of the block diagram.

APR9600 Block Diagram:

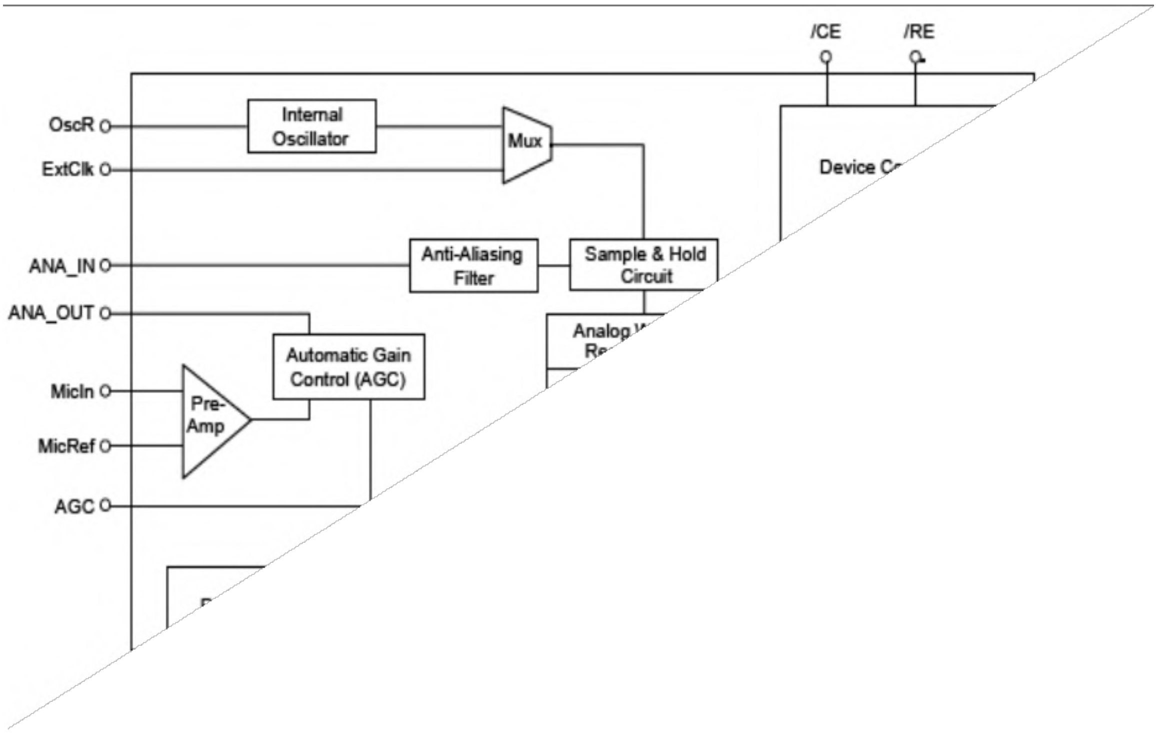


Figure :3.8

Message Management:

Message Management General Description:

Playback and record operations are managed by on-chip circuitry. These are several avaliable messaging modes depending upon desired operation. These message modes determine message management style, message length, and external parts count. Therefore, the designer must select the appropriate operating mode before behining the design.

Operating modes do not affect voice quality for information on factors affecting quality refer to the sampling rate & voice quality section. The device supports five message management modes (defined by the MSEL1, MSEL2 and /M8_OPTION pins).

Random access mode with 2, 4, or 8 fixed duration messages tape mode, with multiple variable duration messages, provides two options:

- Auto rewind
- Normal

Table 3.3 Modes

Mode	MSEL1	MSEL2	/M8_OPTION
Random access 2 fixed duration messages	0	1	Pull this pin to Vcc through 100k resistor
Random access 4 fixed duration messages	1	0	Pull this pin to Vcc through 100k resistor
Random access 8 fixed duration messages	1	1	The /M8 message trigger becomes input pin
Tape mode, Auto rewind operation	0	0	0
Tape mode, Normal operation	0	0	1

Modes cannot be mixed. Switching of modes after the device has recorded an intial message is not recommended. If modes are switched after an intial recording has been made some unpredictable message fragments from the previous mode may remain present, and be audible on playback, in the new mode. These fragments will disappear after a record operation in the newly selected mode. Table1 defines the recording necessary to choose the desired mode.

An important feature of the APR9600 message management capabilities is the ability prompt the user to change in the device's status through the use of "beeps" superimposed on the device's output. This feature is enabled by asserting a logic high level on the BE pin.

Random Access Mode:

Random access mode supports 2, 4, or 8 message segments of fixed duration. As suggested recording or playback can be made randomly in any of the selected messages. The length of each message segment is the total recording length available (as defined by the selected sampling rate) divided by the total number of segments enabled (as decoded in table1). Random access mode provides easy indexing to message segments.

Functional Description of Recording in Random Access Mode:

On power up, the deviceis ready to record or playback in amy of the enabled message segments. To record, /CE must be set low to enble the device and /RE must be set low to enable recording. You intiate recording by applying a low level on the message trigger pin that represents the message segment you intend to use. The message trigger pins are labeled /M1_MESSAGE - /M8_OPTION on pins 1-9 (excluding pin 7) for message segments 1-8 respectively.

Note: Message trigger pins of M1_MESSAGE, /M2_NEXT, /M7_END, and /M8_OPTION, have expanded names to represent the different functionality that these pins assume in the other pins with the same functinality as /M3, /M4, /M5, and /M6. For a more through explanationtable that appears. When actual recording begins the device responds with a single beep (if the BE pin is high to enable the beep tone) at the speaker outputs to indicate that it has started recording. Recording continues as long as the message pin stays low. The rising edge of the same message trigger pin during record stops the recording operation (indicated with a single beep). If the message trigger pin is held low beyond the end of the maximum allocated duration, recording stops automatically (indicated with two beeps), regardless of the state of the message trigger pin. The chip then enters low power mode until the message trigger pin returns high. After the message trigger pin returns to high, the chip enters standby mode. Any subsequent high to low transition on the same message trigger pin will intiate recording from the beginning of the same message segment. The entire previous message is then overwritten by the new message, regardless of the duration of the new message. Transition on any other message trigger pin orthe /RE pin during the record operation are ignored until after the device enters standby mode.

Functional Description of Playback Random Access Mode:

On power up, the device is ready to record or playback, in any of the enabled message segments. To playback, /CE must be set low to enable thye device and /RE must be set high to disable recording & enable playback. You intiate playback by applying a high to low edge on the message trigger pin that represents the message segment you intended to playback. Playback will continue until the end of the message is reached. If a high to low edge occurs on the same message trigger pin during playback, playback of the current message stops immediately. If a different message trigger pin pulses during playbck, playback of the current message stops immediately (indicated by one beep) and playback of the new message segment begins.

A delay equal to 8,400 cycles of the sample clock will be encountered before the device staqrts playing the new meesage. If a message trigger pin is held low, the selected message is playback repeatedly as long as the trigger pin stays low. A period of silence, of a duration equal to 8,400 cycles of the sampling clock, will be inserted during looping as an indicator to the user of the transition between the end and the beginning of the message.

Tape Mode:

Tape mode manages messages sequentially much like traditional cassette tape recorders. Within tape mode two options exist, auto rewind and normal. Auto rewind mode configures the device to automatically rewind or playback of the message. In tape mode, using either option, messages must be recorded or playback sequentially, much like a traditional cassette tape recorder.

Function Description of Recording in Tape Mode using the Auto Rewind Option:

On power up, the device is ready to record or playback, starting at the first address in the memory arry. To record, /CE must be set low to enable the device and /RE must be set low to enable recording. A falling edge of the /M1_MESSAGE pin initiates voice recording (indicated by one beep).A subsequent rising edge of the /M1_MESSAGE pin during recording stops the recording (also indicated by one beep). If the M1_MESSAGE pin is held low beyond the end of the available memory, recording will stop automatically (indicated by two beeps). The device will then assert a logic low on the /M7_END pin until the /M1 Message pin is released.

The device returns to standby mode when the /M1_MESSAGE pin goes high again.After recording is finished the device will automatically rewind to the beginning of the most recently recorded message and wait for the next user input. The auto rewind function is convenient because it allows the user to immediately playback and review the message without the need to rewind. However, caution must be practiced because a subsequent record operation will overwrite the last recorded message unless the user remembers to pulse the /M2_Next pin in order to increment the device past the current message. A subsequent falling edge on the /M1_Message pin starts a new record operation, overwriting the previously existing message. You can preserve the previously recorded message by using the /M2_Next input to advance to the next available message segment. To perform this function, the /M2_NEXT pin must be pulled low for at least 400 cycles of the sample clock.The auto rewind mode allows the user to record over the just recorded message simply by initiating a record sequence without first toggling the /M2_NEXT pin.

To record over any other message however requires a different sequence. You must pulse the /CE pin low once to rewind the device to the beginning of the voice memory.

The /M2_NEXT pin must then be pulsed low for the specified number of times to move to the start of the message you wish to overwrite. Upon arriving at the desired message a record sequence can be initiated to overwrite the previously recorded material.After you overwrite the message it becomes the last available message and all previously recorded messages following this message become inaccessible.

If during a record operation all of the available memory is used, the device will stop recording automatically,(double beep) and set the /M7_END pin low for a duration equal to 1600 cycles of the sample clock.Playback can be initiated on this last message, but pulsing the /M2_Next pin will put the device into an "overflow state".Once the device enters an overflow state any subsequent pulsing of /M1_MESSAGE or /M2_NEXT will only result in a double beep and setting of the /M7_END pin low for a duration equal to 400 cycles of the sample clock. To proceed from this state the user must rewind the device to the beginning of the memory array. This can be accomplished by toggling the /CE pin low or cycling power. All inputs, except the /CE pin,are ignored during recording.

Function Description of playback in Tape Mode using Auto Rewind Option:

On power-up, the device is ready to record or playback,starting at the first address in the memory array.Before you can begin playback, the /CE input must be set to low to enable the device and /RE must be set to high to disable recording and enable playback. The first high to low going pulse of the /M1_MESSAGE pin initiates playback from the beginning of the current message; on power up the first message is the current message. When the /M1_MESSAGE pin pulses low the second time, playback of the current Message stops immediately. When the /M1_MESSAGE pin pulses low a third time, playback of the current message starts again from its beginning. If you hold the /M1_MESSAGE pin low continuously the same message will play continuously in a looping fashion. A 1,540ms period of silence is inserted during looping as an indicator to the user of the transition between the beginning and end of the message.Note that in auto rewind mode the device always rewinds to the beginning of the current message. To listen to a subsequent message the device must be fast forwarded past the current message to the next message. This function is accomplished by toggling the /M2_NEXT pin from high to low. The pulse must be low for least 400 cycles of the sampling clock.

After the device is incremented to the desired message the user can initiate playback of the message with the playback sequence described above.

A special case exists when the /M2_NEXT pin goes low during playback. Playback of the current message will stop, the device will beep,advance to the next message and initiate playback of the next message. (Note that if /M2 Next goes low when not in playback mode, the device will prepare to play the next message, but will not actually initiate playback).

If the /CE pin goes high during playback, playback of the current message will stop, the device will beep,reset to the beginning of the first message, and wait for a subsequent playback command.When you reach the end of the memory array, any subsequent pulsing of /M1_MESSAGE or /M2_NEXT will only result in a double beep. To proceed from this state the user must rewind the device to the beginning of the memory array. This can be accomplished by toggling the /CE pin low or cycling power.

Functional Description of Recording in Tape Mode using the Normal Option:

On power-up, the device is ready to record or playback,starting at the first address in the memory array.Before you can begin recording, the /CE input must be set to low to enable the device and /RE must be set to low to enable recording. On a falling edge of the /M1_MESSAGE pin the device will beep once and initiate recording. A subsequent rising edge on the /M1 Message pin will stop recording and insert a single beep. If the M1_MESSAGE pin is held low beyond the end of the available memory, recording Stops automatically, and two beeps are inserted; regardless of the state of the /M1_MESSAGE pin. The device returns to the standby mode when the /M1_MESSAGE pin is returned high.A subsequent falling edge on the /M1_MESSAGE pin starts a new record operation in the memory array immediately fol- lowing the last recorded message, thus preserving the last recorded message.To record over all previous messages you must pulse the /CE pin low once to reset the device to the beginning of the first message. You
can then initiate a record sequence, as described above, to record a new message. The most recently recorded message will become the last recorded message and all previously recorded messages following this message will become inaccessible.

If you wish to preserve any current messages it is recommend that the Auto Rewind option be used instead of the Normal option. If the Normal option is necessary the following sequence can be used. To preserve current messages you must fast forward past the messages you want to keep before you can record a new message.

To fast forward when using the Normal option you must switch to play mode and listen to messages sequentially until you arrive at the beginning of the
message you wish to overwrite.

At this stage you should switch back to record mode and overwrite the desired message.The most recently recorded message will become the last recorded message and all previously recorded messages following this message will become inaccessible. All inputs, except /CE, are ignored during recording.

Functional Description of Playback in Tape Mode using the Normal Option:

On power-up, or after a low to high transition on /RE the device is ready to record or playback starting at the first address in the memory array. Before you can begin playback ofmessages, the /CE input must be set to low to enable the device and /RE must be set to high to enable playback.

The first high to low going pulse of the /M1_MESSAGE pin initiates playback from the beginning of the current message. When the /M1_MESSAGE pin pulses from high to low a second time, playback of the current message stops immediately. When the /M1_MESSAGE pin pulses from high to low a third time, playback of the next message starts again from the beginning. If you hold the /M1_MESSAGE pin low continuously, the current message and subsequent messages play until the one of the following conditions is met: the end of the memory array is reached, the last message is reached,the /M1_message pin is released. If the last recorded message has already played, any further transitions on the /M1_MESSAGE pin will initiate a double beep for warning and the /M7_END pin will go low. To exit this state you must pulse the /CE pin high and then low once during standby to reset the pointer to the beginning of the first message.

3.4 Related Designs/Graphs:

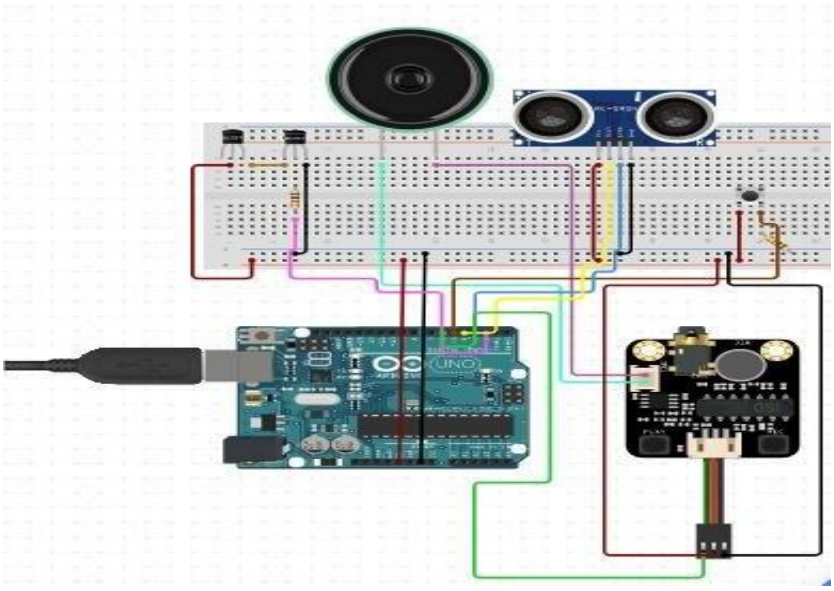


Figure :3.4.1 Schematic Diagram

CHAPTER IV

CHAPTER IV

PROJECT IMPLEMENTATION

To complete the project successfully, the process has to done in a disciplined way step by step. Different steps involved in the implementing the project are

- Project Initiation.
- Project Planning.
- Project Execution.
- Project Monitoring and Controlling.
- Project Closing.

Stages in Detail

- **Project Initiation** : This step involves initializing the project i.e **Project initiation** is the first phase of the project management life cycle and in this stage, companies decide if the project is needed and how beneficial it will be for them. The two metrics that are used to judge a proposed project and determine the expectations from it are the business case and feasibility study.

- **Project Planning:** Project planning is a discipline for stating how to complete a project within a certain timeframe, usually with defined stages, and with designated resources. One view of project

- Planning divides the activity into: Setting objectives (these should be measurable) **Planning** the **schedule**. Making supporting plans.

- **Project Execution: Project execution** is the stage of the **project** where everything your team has planned is put into action.

- **Project Monitoring and Controlling:**The **Monitoring and Controlling** process oversees all the tasks and metrics necessary to ensure that the approved and authorized **project** is within scope, on time, and on budget so that the **project** proceeds with minimal risk. **Monitoring and Controlling** process is continuously performed throughout the life of the **project.**

- **Project Closing**: This is the last step in implementing stages. In this phase results are obtained if all above stages are implemented perfectly.

Results

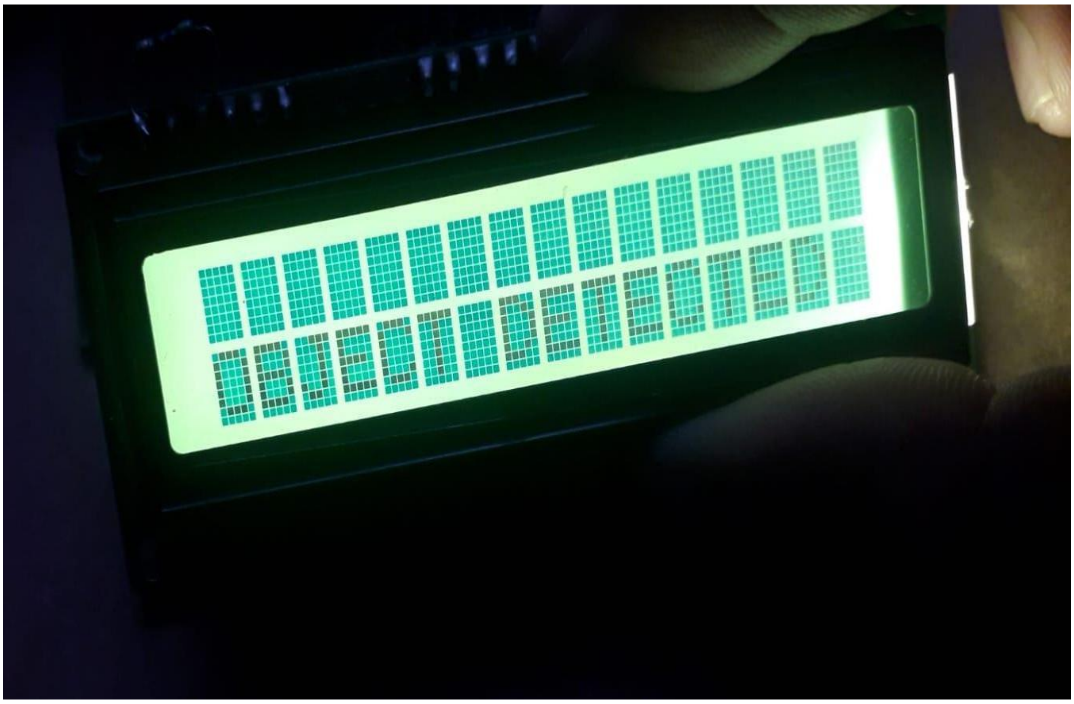


Figure 4.2.1 : When obstacle is detected

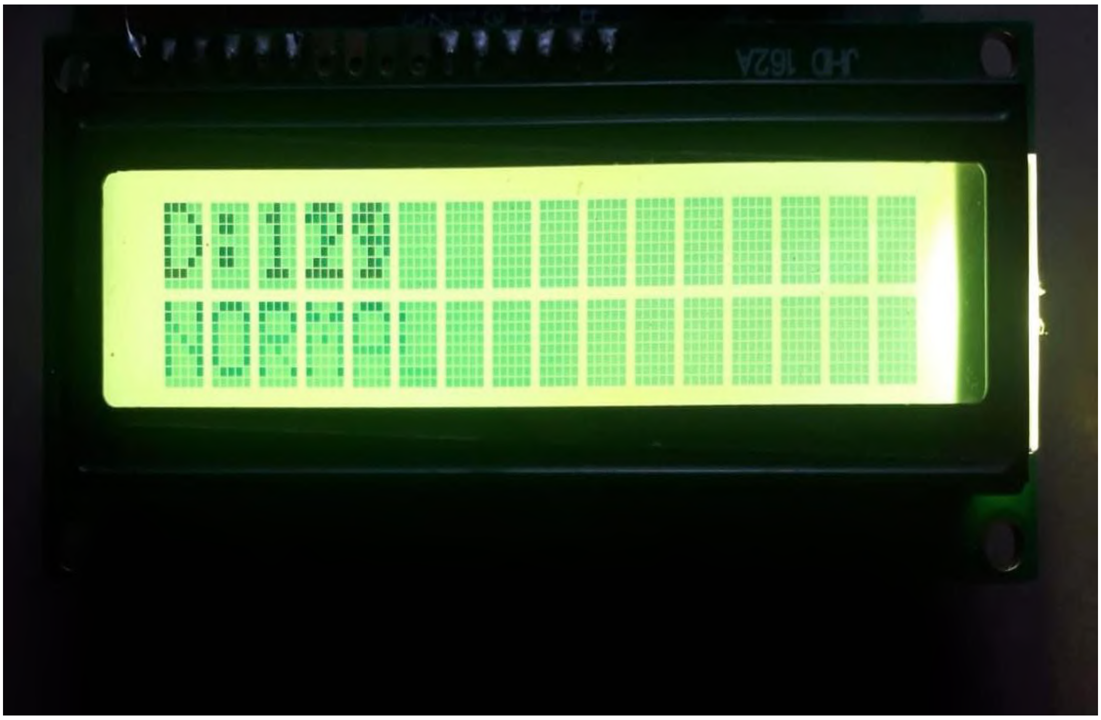


Figure 4.2.2 : When obstacle is not detected

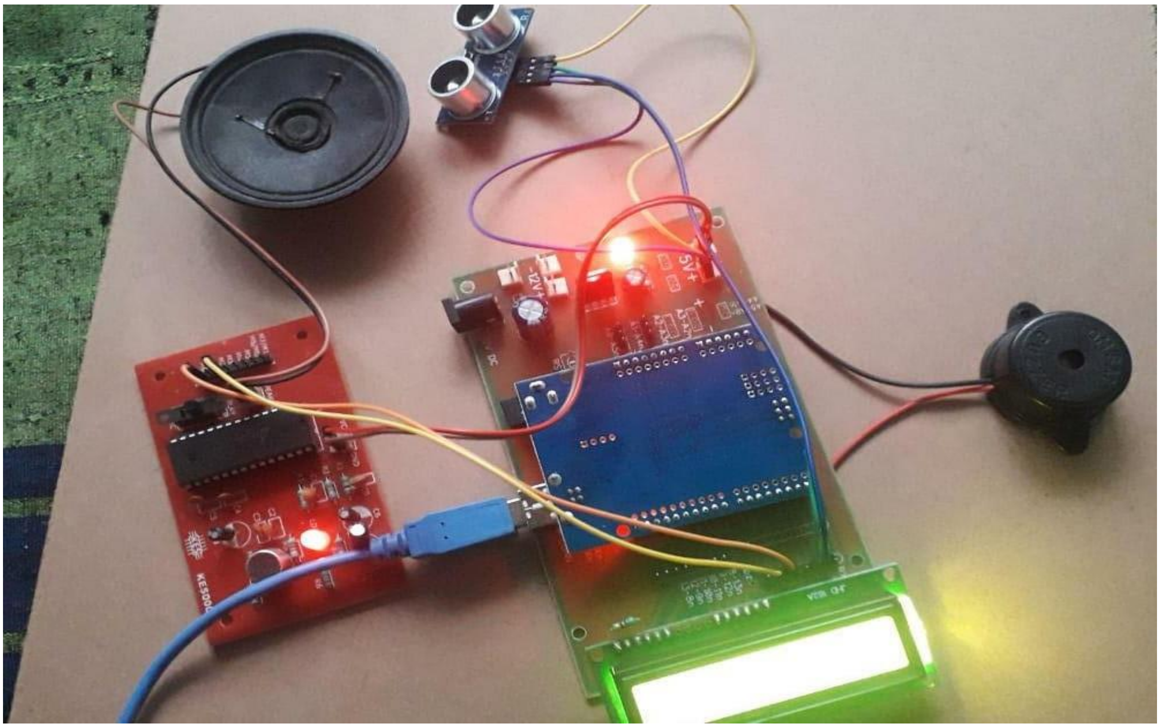


Figure 4.2.3 : Overall Output of the Project

CHAPTER V

CHAPTER V
PROJECT TESTING

5.0 Overview of Testing Methods

- It is observed that the performance difference between the estimated and the actual performance is very minute.
- It worked better for all the inputs provided.
- Estimated result is obtained.
- No Errors Occurred during the functioning of the kit.
- The person using the kit was able to know the presence of obstacles infront of him by using the stick.
- Not only the blind person but also person who is both blind and deaf can use this kit as we are having buzzer which vibrats when the obstacle is found.

CHAPTER VI

CHAPTER VI

CONCLUSION AND FUTURE ENHANCEMENT

PUBLICATION:

- Online Mega International Conference on “Smart Modernistic in Electronics and Communication” (ICSMEC-21) organized by Department of Electronics & Communication Engineering, St. Martin’s Engineering College
- ID : ICSMEC21-0073

REFERENCES :

[1] Shinohara, K. ―Designing assistive technology for blind users‖ In Proceedings of the 8th International ACM SIGACCESS conference on Computers and accessibility, 2006.

[2] Benjamin J. M., Ali N. A., Schepis A. F., ―A Laser Cane for the Blind‖ Proceedings of the San Diego Biomedical Symposium, Vol. 12, 53-57.

[3] Johann B., Iwan U., ―The Guide Cane — A Computerized Travel Aid for the Active Guidance of Blind Pedestrians‖ Proceedings of the IEEE International Conference on Roboticsand Automation, Albuquerque, NM, 1283-1288, 1997.

[4] Madad A. Shah, Sayed H. Abbas, Shahzad A. Malik , ―Blind Navigation via a DGPSbasedHand-held Unit‖ Australian Journal of Basic and Applied Sciences, 4(6): 1449-1458, 2010.

[5] Kulyukin V., Gharpure C., and Nicholson J., ―Robo-Cart: Toward Robot-Assisted Navigationof Grocery Stores by the Visually Impaired,‖ IEEE/RSJ International Conference.

[6] R. Dhanuja, F. Farhana and G. Savitha, “Smart Blind Sitck Using Arduino”, in International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 03 March 2018.

[7] D. E. Gbenga, A. I. Shani and A. L. Adekunle, “Smart Walking Stick for Visually Impaired People Using Ultrasonic Sensors and Arduino”, in International Journal of Engineering and Technology (IJET), Vol 9 No 5 Oct-Nov 2017.

[8] V. Jeevana, R. K. Sundar, K. Pravin S. Preethi and R. Karthik, “Design of Intelligent Stick - Guide for the Blind”, in IJSRD - International Journal for Scientific Research & Development,Vol. 6, Issue 01, 2018.une. 2017).

BIBILOGRAPHY

Author	**Title of Publication**	**Date of Publication**
Simon Monk	Programming Arduino	29 June 2016
Nithish Sukhija	Smart stick for blind Man	14 April 2017

Table 6.1

APPENDICES

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);//rs//en//d7//d6//d5//d4

const int trigPin = 2;
const int echoPin = 3;
int buz = 7,v1=5,v2=6;
long duration;
int distance;


void setup() {


  pinMode(v1, OUTPUT);
  pinMode(v2, OUTPUT);
  //pinMode(r1, INPUT);
  pinMode(buz, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  digitalWrite(buz,HIGH);
  digitalWrite(v1,HIGH);
  digitalWrite(v2,HIGH);



  Serial.begin(9600);

  lcd.begin(16,2);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("ULTRA SONIC NAVIGATION FOR BLIND");

  delay(500);


}


void loop() {


   lcd.clear();lcd.setCursor(0, 0);lcd.print("D:");lcd.print(distance);lcd.print("       ");
   digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)
  // Displays the distance on the Serial Monitor
  Serial.print("Distance: ");
```

A

MAJOR PROJECT REPORT

On

QUADRIPLEGICS WHEELCHAIR CONTROL BY HEAD MOTION USING ACCELOROMETER

Submitted by

Mr. S. Shivakumar Reddy(17K81A0449)
Mr. S. Satish Kumar Reddy(17K81A0450)
Ms. S. Rajitha(17K81A0451)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Esteemed Guidance of

Mr. Joel Josephson

Associate Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE
An Autonomous Institute

Dhulapally, Secunderabad – 500 100

JUNE 2021

BONAFIDE CERTIFICATE

This is to certify that the project entitled "QUADRIPLEGICS WHEELCHAIR CONTROL BY HEAD MOTION USING ACCELOROMETER", is being submitted by **Mr. S. Shivakumar Reddy(17K81A0449),Mr. S. Satish Kumar Reddy(17K81A0450), Ms. S. Rajitha (17K81A0451)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bona fide work carried out by them. The result embodied in this report have been verified and found satisfactory.

PROJECT GUIDE	**HEAD OF THE DEPARTMENT**
Mr. JOEL JOSEPHSON	**Dr. B. HARI KRISHNA**
Associate Professor	**Professor**
Department of ECE	**Department of ECE**

Internal Examiner External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT S. SHIVAKUMAR REDDY WITH ROLL NO. 17K81A0449, S. SATISH KUMAR REDDY WITH ROLL NO 17K81A0450, S. RAJITHA WITH ROLL NO 17K81A0451 OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST.MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, WE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "QUADRIPLEGICS WHEELCHAIR CONTROL BY HEAD MOTION USING ACCELOROMETER" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH HIS SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com Website : www.lasyainfotech.com
contact:7330666881/82/83/84/86

DECLARATION

We, the student of **Bachelor of Technology** in Department of ELECTRONICS AND COMMUNICATION ENGINEERING, session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled "QUADRIPLEGICS WHEELCHAIR CONTROL BY HEAD MOTION USING ACCELOROMETER" is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

Mr. S. Shivakumar Reddy(17K81A0449)

Mr. S. Satish Kumar Reddy(17K810450)

Ms. S. Rajitha(17K81A0451)

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to our **College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr. P. Santosh Kumar Patra**, **our beloved Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr. B. HARI KRISHNA**, **Head &Professor**, Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We proud to record that had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in our memory for years to come.

We would like to express our gratitude to our dept. project coordinator **Dr. A. ANAND, Professor** for his involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time to come.

Finally, we highly grateful and obliged to our project guide **Mr. Joel Josephson, Associate Professor,** for his advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

S. SHIVAKUMAR REDDY (17K81A0449)
S. SATISH KUMAR REDDY (17K81A0450)
S. RAJITHA (17K81A0451)

ABSTRACT

The problems faced by paralyzed and handicapped people are many among which the move any of their body parts except head. This robotic wheelchair project aims at designing a wheelchair that will work crucial one is independent mobility. They need external help from other people to execute and perform their daily activities. Electric and advanced wheelchairs are manufactured and designed with an aim to aid paraplegics. But such electric wheelchairs cannot by patients having high range of impairment like persons who cannot on the basis of head movement of the patient. For normal movement of wheelchair we use hand force.

Traditional Wheelchairs though have certain limitations with the flexibility, heavy weight of the chair and limited functions. Tremendous developments have been made in the field of wheelchair technology. Be that as it may, even these noteworthy advancements couldn't help the quadriplegics to explore wheelchair freely. Medical gadgets intended to support the Paraplegic and Quadriplegic patients are exceptionally muddled, once in a while accessible and costly. We go for planning a straightforward financially effective programmed wheelchair utilizing MEMS technology for quadriplegics with head and neck versatility. The control system interprets the situation of the user's head into speed and directional control of the wheelchair. The system is divided into two main units: MEMS Sensor and programmed Arduino Micro Controller. The MEMS sensor senses the change in direction of head and likewise the signal is given to microcontroller. Depending on the direction of the Acceleration, microcontroller controls the wheel chair directions like LEFT, RIGHT, FRONT, and BACK with the aid of DC motors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT:

Introduction:

Now a days, many elderly and physically disabled people need wheelchairs to make their personal mobility easy so for them wheelchair is a suitable device. By using wheelchair they can move from one place to another effortlessly. By using standard wheelchairs, it requires another person's help or by self-propelling. These problems can be overcome by using electric wheelchairs or joystick controlled wheelchair. Generally, Electric Wheelchairs cost ranges from 50000 and more [1]. This Head-motion controlled Wheelchair is way cheaper than Electric or Joystick controlled wheelchair available in market. The Head-motion Controlled Wheelchair is far better than conventional Wheelchairs. The conventional joystick wheelchair requires hand motion for controlling which are not suitable for physically handicapped peoples. But Head motion controlled wheelchair is suitable for those whose hands or fingers are paralyzed as it requires head motion. Rather than these, the head-motion controlled wheelchair is far cheaper than joystick wheelchairs. This paper investigates the research work of Head-motion controlled wheelchair for the user interaction. Gesture type, interface, technology used, user types, issues, problems, advantages and final result have been listed and described to give background of Wheel Chair based technology development.

Quadriplegics are people who can't utilize any of the furthest points. The purposes behind such diminished movement prospects can be unique: stroke, joint pain, hypertension, degenerative sicknesses of bones and joints and instances of loss of motion and birth abandons. Likewise, quadriplegia shows up as a result of mishaps or age. The patients with such serious incapacities can't play out their ordinary activities, for example, taking care of, and development through space. Contingent upon the seriousness of the in capacity, a patient can hold opportunity of development to a specific level by utilizing diverse clinical gadgets. Versatility has gotten significant for a decent personal satisfaction.

Arranging a system with free versatility for such debilitated people is our point in this endeavour. We target planning a straightforward savvy wheelchair constrained by head level-opponent which additionally incorporates patient monitoring framework and obstacle detection framework. This wheelchair incorporates ADXL335 sensor, battery, dc motor, and Arduino uno microcontroller. ADXL335 which is used for perceiving the tilt. This sensor finds the tilt and alters the Course of the wheel chair relying upon tilt.

1.2 OBJECTIVES OF THE STUDY:

The main objective of the project is to design a wheelchair tilt communicator system that could operate the wheelchair of the handicapped person with the help of tilt of head movements. This system could be used by physically disabled persons who cannot move their hands or legs but make head and eye motions. This wheelchair could be operated in any direction using head tilt movements by the handicapped person. Design and development of Head motion controlled wheelchair has been achieved using tilt sensors and wireless modules. The system is implemented practically and works well with a person sitting on it. This wheelchair is aimed to be designed at a lower cost as compared to the other versions available in the market. The head motion controlled wheelchair designed using tilt communicator system turns out to be a great use for quadriplegic patients and disabled people having more than 45% or more disability as this could be operated easily through head gestures.

1.3 SCOPE OF THE STUDY:

- In the race of man versus machine head motion controlled system comes as an example companionship of man and machine.

- To avoid physical hardship an accelerometer is used due to which the slight movement of head turns the wheelchair into the desired direction. It is designed to be characterized by low price and higher reliability.

- Quadriplegics are persons who are not able to use any of the extremities. The reasons for such decreased motion possibilities can be different: stroke, arthritis, high blood pressure, degenerative diseases of bones and joints and cases of paralysis and birth defects. Also, quadriplegia appears as a consequence of accidents or age. The patients with such severe disabilities are not able to perform their everyday actions, such as: feeding, toilette usage and movement through space.

- Depending on the severity of the disability, a patient can retain freedom of movement to a certain level by using different medical devices.

1.4 MATERIAL REQUIREMENT:

1.4.1 HARDWARE REQUIREMENT:

- ARDUINO UNO
- ACCELOROMETER (ADXL335)
- DC MOTORS
- MOTOR DRIVER(L293D)
- LCD (16*2)
- BATTERY

1.4.2 SOFTWARE REQUIREMENT:

- Proteus software
- Arduino software

1.5 PROCUMENT OF EQUIPMENT:

We brought all the required hardware components in an electronic manufacturing shop and before assembling and interfacing hardware components to Arduino UNO micro-controller, we connected individual components to Arduino in Proteus Software and understood working of every individual component in Proteus Software.

Now, we assembled all components and interfaced to Arduino UNO micro-controller using connecting wires through bread board and we wrote program to work all the hardware components in Arduino IDE we understood the working of all components in a practical way.

We were able to monitor the toll booth system with better efficiency, with less time delay and the results were also satisfactory and we successfully got the output in mobile telnet app and LCD screen. We would like to improve the project in future for further developments.

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH AREA:

When an unfortunate event affects the motor capacity of a person, it is necessary to use devices like wheelchairs that offer a means of displacement for patients with motors problems of the lower limbs. Tremendous leaps have been made in the field of wheelchair technology However, even these significant advances haven't been able to help quadriplegics field of wheelchair technology. However, even these significant advances haven't been able to help quadriplegics navigate wheelchair unassisted. Some patients that cannot manipulate the wheelchair with their arms due to a lack of force

2.2 REVIEW ON RELATED LITERATURE:

According to research paper, "HAND GESTURE RECOGNITION: A LITERATURE REVIEW"[2], it focuses on human computer interaction. It is a survey of recent hand gesture recognition systems. Key issues of hand gesture recognition system are presented with challenges of gesture system. Review methods of recent postures and gestures recognition system presented as well. Orientation histogram method applied in this have some problems which are; similar gestures might have different orientation histograms and different gestures could have similar orientation histograms, besides that, the proposed method achieved well for any objects that dominate the image even if it is not the hand gesture.

- According to research paper, titled "IR Sensor-Based Gesture Control Wheelchair" [3] published in IEEE, it works on the principle of gesture recognition by using Infrared sensors. In this method, IR sensors are used for identifying the simple gestures to control the powered wheelchair to move in any direction. In the proposed prototype system, a gesture pad that includes IR sensors, MCU and power management circuit is designed for gesture recognition and identification and a controller for driving motors is implemented. The main problem that comes with IR is during daylight, its sensitivity is reduced, and hence causing problem for processing the further programs. And it is difficult to recognize exact gestures using IR sensors
- In research paper published, entitled, "Electric Powered Wheelchairs",[4] its aim was to review the concepts and previous work on velocity control, traction control, suspension control, stability control, stair-climbing wheelchairs, and wheelchair navigation. The information gathered in this study is

intended to promote awareness of the status of contemporary powered wheelchair control technology and increase the functional mobility of people who use EPWs. But, it has a major disadvantage of cost efficiency. It's quite expensive as compared to normal wheelchair.

• In an article [5], from Int. Journal of Engineering Research and Applications, aims at gesture control wheelchair using hand movements. According to this article, with hand movements, wheelchair's direction can be changed. But it has disadvantage over paralyzed patients, who cannot move their hands.

• We are using accelerometer, mounted to the earpiece, so that with movement in head, or say, by moving

2.3 CONCLUSION ON REVIEWS:

The existing systems include hand gesture based, accelerometer and voice controlled systems etc. The hand gesture based system used the transfer of hand gesture information commands to move the wheel chair. The accelerometer and voice controlled system used voice recognition kit and MEMS motion sensor to drive the wheel chair.

The issues with the existing system are:

1) Unable to adapt to external conditions.

2) Less identification accuracy.

3) Classification techniques employed are complex.

4) Time consuming and costly.

CHAPTER-3

PROJECT DESIGN

3.1 OVERVIEW OF THE DESIGN:

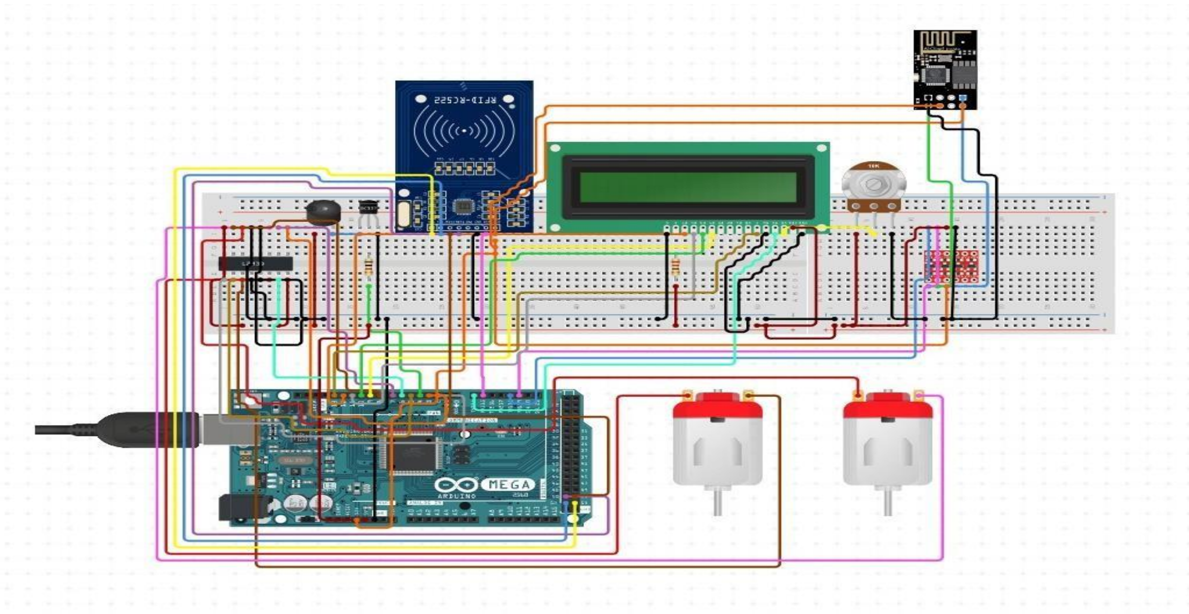


Fig.3.1 Overview of the design

3.2 EQUIPMENT ANALYSIS:

3.2.1 ARDUINO:

3.2.1.1 INTRODUCTION:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

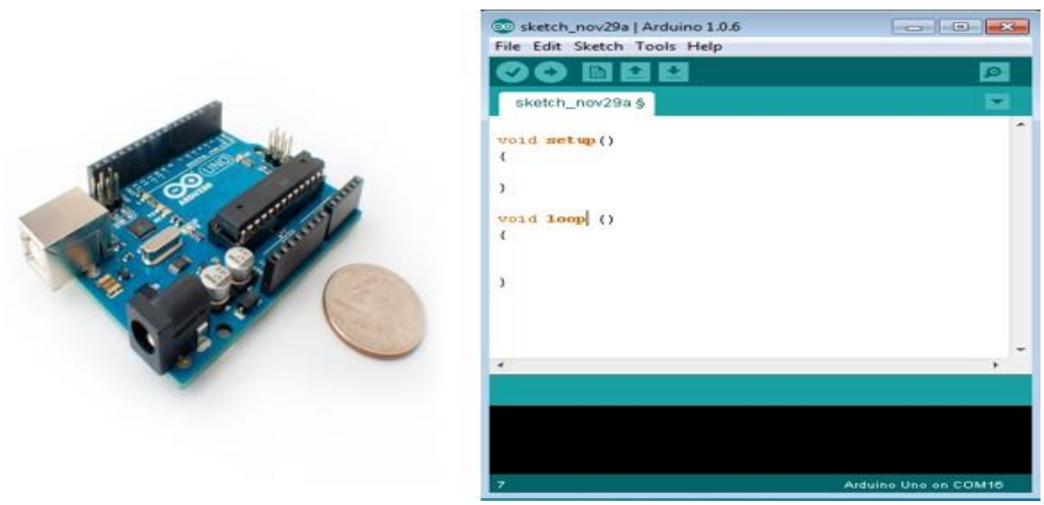


Fig 3.2.1.1 Arduino Uno

3.2.1.2 BOARD TYPES

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header

Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.2.1.2.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	**Operating Volt**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB

Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 3.2.1.2.2 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 3.2.1.2.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table 3.2.1.2.4 Arduino boards based on AT91SAM3X8E microcontroller

3.2.1.3 BOARD DESCRIPTION:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

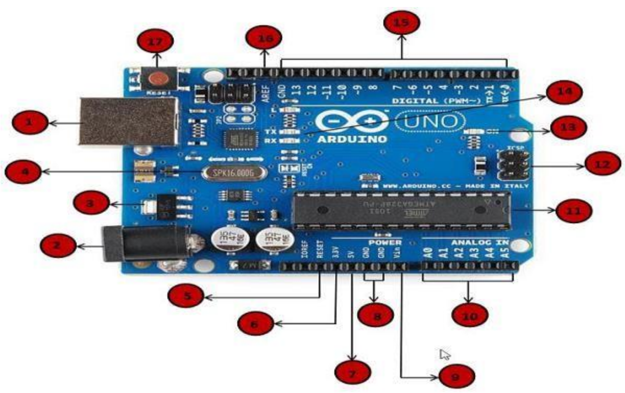


Fig.3.2.1.3 Board Description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).

3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC

	your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

3.2.1.4 ARDUINO FAMILY

Arduino makes several different boards, each with different capabilities. In addition, part of being open-source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

Fig.3.2.1.4: Arduino Family

3.2.1.5 SHIELDS:

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

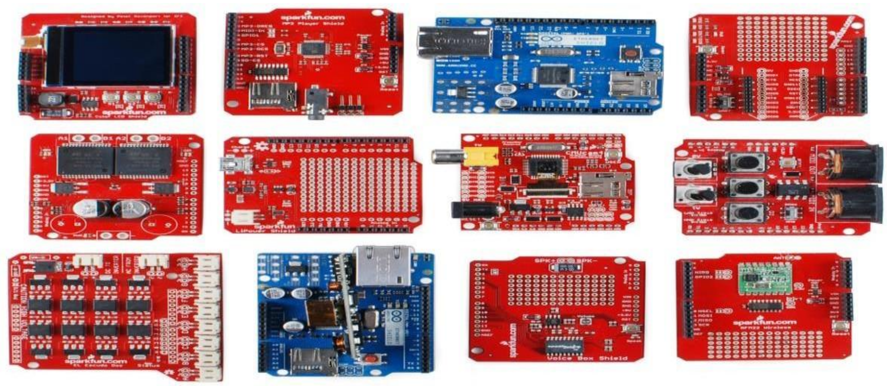


Fig.3.2.1.5: Arduino Shields

3.2.1.6 PIN DESCRIPTION OF ATMEGA328:

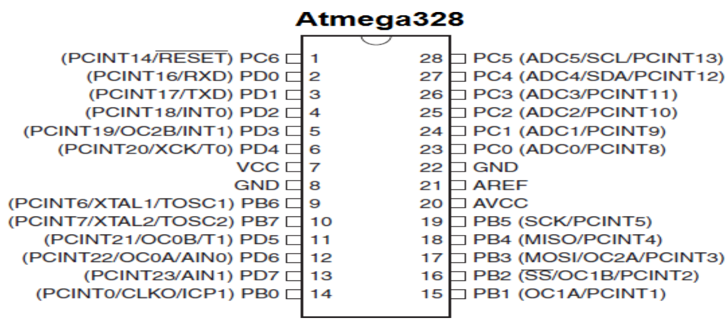


Fig.3.2.1.6 Pin description of ATMEGA328

3.2.1.7 APPLICATIONS:

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

3.2.2 16*2 LCD SCREEN DISPLAY

3.2.2.1 INTRODUCTION

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

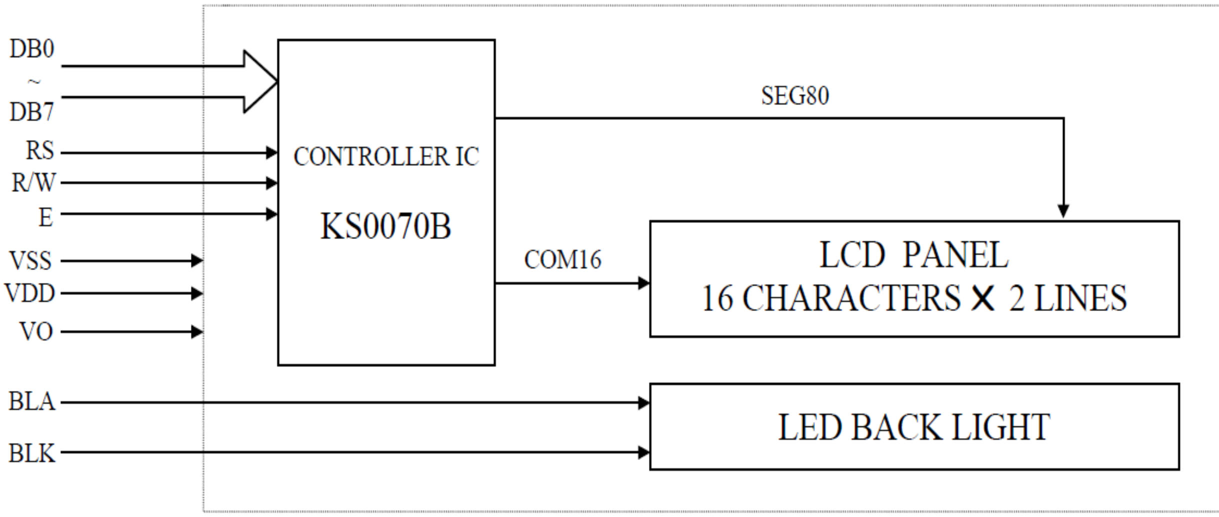


fig.3.2.2.1 Lcd block diagram

3.2.2.2 DISPLAY DATA RAM:

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.

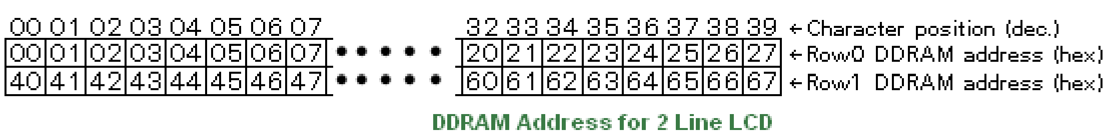


Fig.3.2.2.2 DDRAM addresses of 2 Line LCD

3.2.2.3 CHARACTER GENERATOR ROM:

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM.

3.2.2.4 BUS FLAG:

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e. BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

3.2.2.5 INSTRUCTION REGISTER (IR) AND DATA REGISTER (DR):

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g. LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD.
Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

3.2.2.6 16 X 2 ALPHANUMERIC LCD MODULE FEATURES:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible

- Connector for standard 0.1-pitch pin headers

3.2.2.7 CIRCUIT DESCRIPTION:

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly. The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

3.2.3 L293D

3.2.3.1 INTRODUCTION

L293D is basically a high current dual motor driver/controller Integrated Circuit (IC). It is able to drive load having current up to 1A at the voltage ranging from 4.5V to 36V. Motor driver usually act as current amplifier because they receive a low current signal as an input and provides high current signal at the output

L293D is basically a motor driver or controller. It has two built-in H-bridge circuits which are able to control two DC motors simultaneously in both clockwise and counter clockwise direction. It acts as an current high amplifier because it take low current signal at its input and provides higher current signal at the output in order to drives different load e.g. stepper motor & DC motors. Its features include large input voltage supply range, large output current high noise immunity input signals etc. Its common real-life applications include stepping motor drivers, relay drivers, DC motor drivers etc.

L-293D motor driver/controller is shown in the figure given below.

Fig. 3.2.3.1 L293D Motor Driver

3.2.3.2 L293D PINS:

- L-293D has sixteen (16) pins, having different individual functions.
- All of these sixteen pins along with their serial no, are given in the table shown below.

3.2.3.3 L293D PIN FUNCTIONS:

- L 293D each pin has different tasks to perform when they are in working condition.
- The tasks assigned to each pin are given in the table shown below.

L293D Pins	
Pin. No	Pin Name
1	Enable 1,2
2	Input 1
3	Output 1
4	GND
5	GND
6	Output 2
7	Input 2
8	Vcc2
9	Enable 3,4
10	Input 3
11	Output 3
12	GND
13	GND
14	Output put 4
15	Input 4
16	Vcc1

Table 3.2.3.3 L293D Pin Functions

3.2.3.4 L293D PINOUT:

- Pinout diagram of any device presents the pins configuration through a completely labelled diagram.

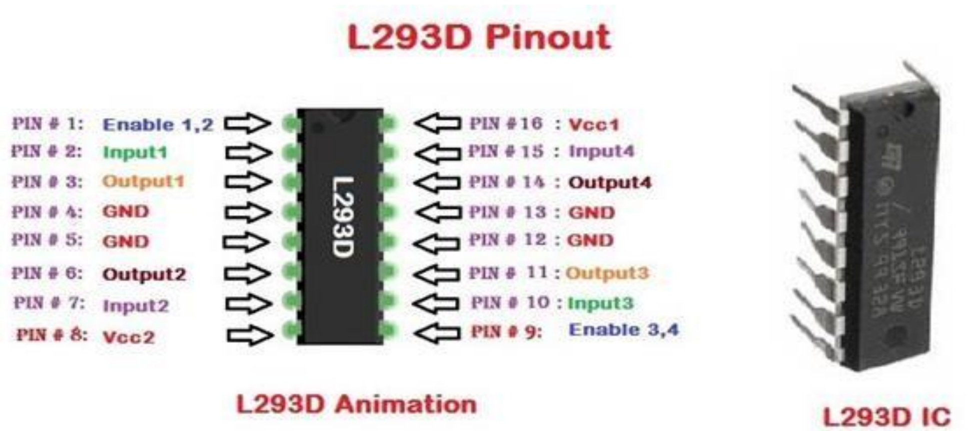


Fig. 3.2.3.4 L293D Pinout

3.2.4 MOTOR

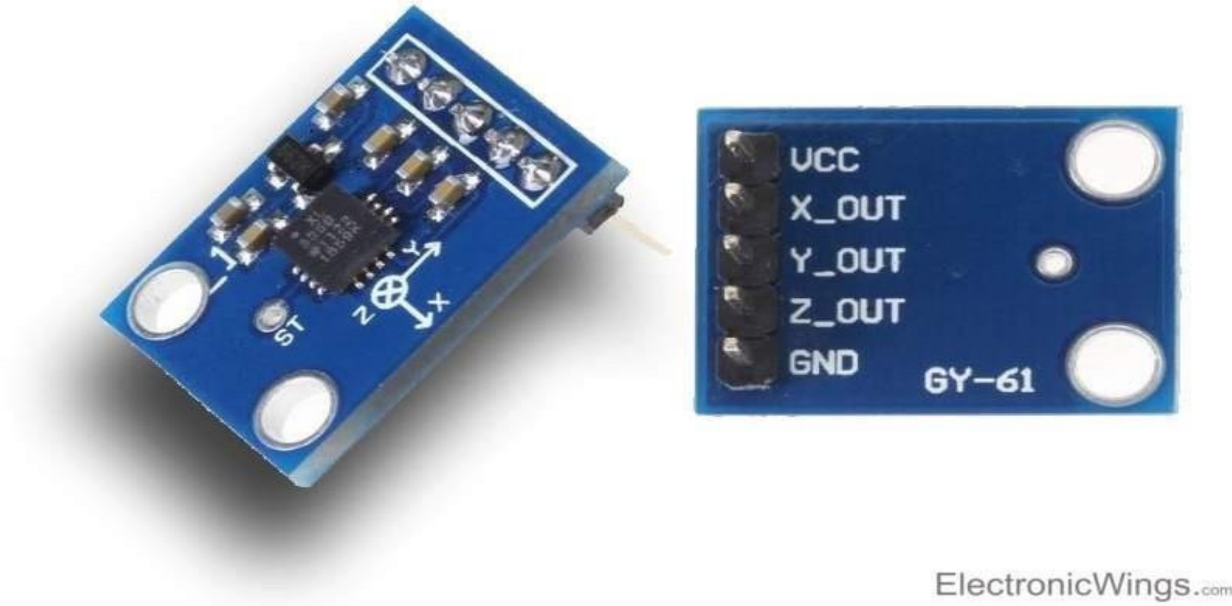
Motors usually operates on this higher current. L-293D has to built-in H-Bridge driver circuits and is able to control two DC motors at a time in both clockwise and counter clockwise direction. It has two enable pins and they should be kept high in order to control the motor. By changing the polarity of applied signal motor can be rotated in either clockwise or counter clockwise direction. If L 293D enable pin is high, its corresponding driver will provide the desired out. If the enable pin is low, there will be no output. L-293D has different features including internal ESD protection, large voltage supply range, large output current per channel, high noise immunity input etc. L 293D plays a vital role in electronics era and has several different applications e.g. relay drivers, DC motor drivers, stepping motor drivers etc. The further detail about L 293D motor driver/controller will be given later in this tutorial.

3.2.5 ACCELOROMETER:

3.2.5.1 INTRODUCTION:

An accelerometer is a device that measures the vibration, or acceleration of motion of a structure. The force caused by vibration or a change in motion (acceleration) causes the mass to "squeeze" the piezoelectric material which produces an electrical charge that is proportional to the force exerted upon it. Since the charge is proportional to the force, and the mass is a constant, then the charge is also proportional to the acceleration. These sensors are used in a variety of ways from space stations to handheld devices, and there's a good chance you already own a device with an accelerometer in it. For example, almost all smartphones today house an accelerometer. They help the phone know whether it undergoes acceleration in any direction, and it’s the reason why your phone’s display switches on when you flip it. In an industry setting, accelerometers help engineers understand a machine's stability and enable them to monitor for any unwanted forces/vibrations
An accelerometer is a **transducer** that is used to measure the physical or measurable acceleration that is made by an object. Co-ordinate acceleration cannot be measured using this device.

An accelerometer is an electro-mechanical device that is used to measure the specific force of an object, a force obtained due to the phenomenon of weight exerted by an object that is kept in the frame of reference of the accelerometer. In the case of static acceleration, the device is mainly used to find the degrees at which an object is tilted with respect to the ground. In dynamic acceleration, the movement of the object can be foreseen.



.32

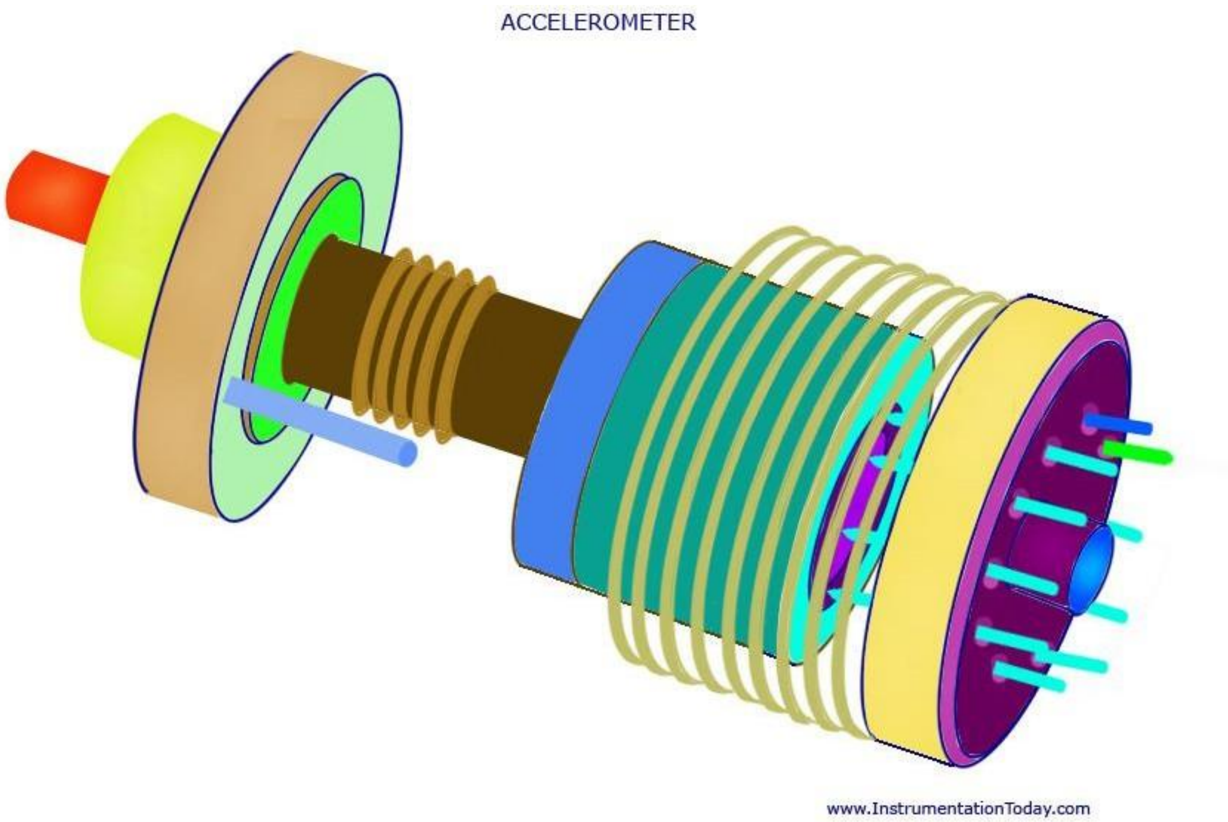


Fig.3.2.5.1: Accelorometer

3.2.5.2. Working

The most commonly used device is the piezoelectric accelerometer. As the name suggests, it uses the principle of piezoelectric effect. The device consists of a piezoelectric quartz crystal on which an accelerative force, whose value is to be measured, is applied.

Due to the special self-generating property, the crystal produces a voltage that is proportional to the accelerative force. The working and the basic arrangement is shown in the figure below.

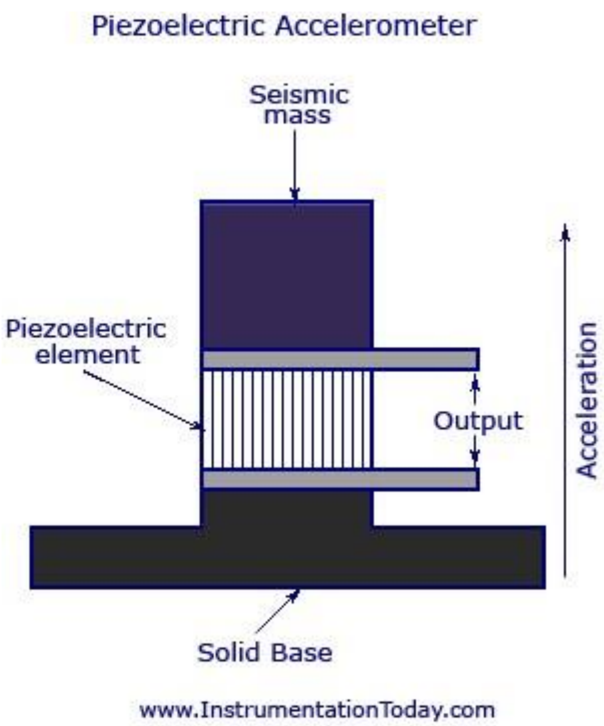


Fig 3.2.5.2. Piezoelectric Accelerometer

As the device finds its application as a highly accurate vibration measuring device, it is also called a vibrating sensor. Vibration sensors are used for the measurement of vibration in bearings of heavy equipment and pressure lines. The piezoelectric accelerometer can be classified into two. They are high impedance output accelerometer and low impedance accelerometer.

In the case of high impedance device, the output voltage generated in proportion to the acceleration is given as the input to other measuring instruments. The instrumentation process and signal conditioning of the output is considered high and thus a low impedance device cannot be of any use for this application. This device is used at temperatures greater than 120 degree Celsius.

The low impedance accelerometer produces a current due to the output voltage generated and this charge is given to a FET transistor so as to convert the charge into a low impedance output. This type of accelerometers is most commonly used in industrial applications.

3.2.5.3. Types

In industrial applications, the most commonly used components to convert the mechanical action into its corresponding electrical output signal are piezoelectric, piezoresistive and capacitive in nature. Piezoelectric devices are more preferred in cases where it is to be used in very high temperatures, easy mounting and also high frequency rang e up to 100 kilohertz. Piezoresistive devices are used in sudden and extreme vibrating applications. Capacitive accelerometers are preferred in applications such as a silicon-micro machined sensor material and can operate in frequencies up to 1 kilohertz. All these devices are known to have very high stability and linearity. Nowadays, a new type of accelerometer called the **Micro Electro-Mechanical System (MEMS) Accelerometer** is being used as it is simple, reliable and highly cost effective. It consists of a cantilever beam along with a seismic mass which deflects due to an applied acceleration. This deflection is measured using analog or digital techniques and will be a measure of the acceleration applied.

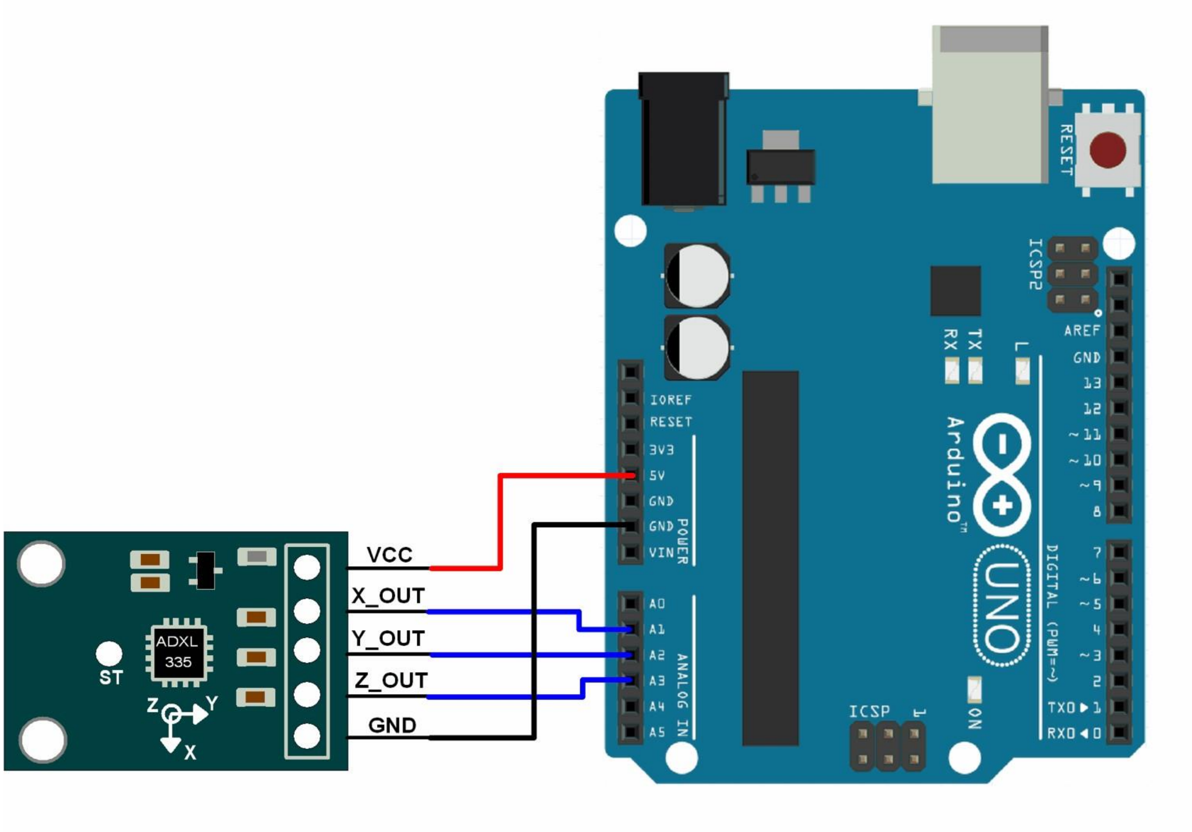


Fig 3.2.5.3. Interfacing accelerometer with Arduino uno

3.2.5.4. Accelerometer Specifications

Frequency Response – This parameter can be found out by analyzing the properties of the quartz crystal used and also the resonance frequency of the device.

1. Accelerometer Grounding – Grounding can be in two modes. One is called the Case Grounded Accelerometer which has the low side of the signal connected to their core. This device is susceptible to ground noise. Ground Isolation Accelerometer refers to the electrical device kept away from the case. Such a device is prone to ground produced noise.

2. Resonant Frequency – It should be noted that the resonant frequency should be always higher the the frequency response.

3. Temperature of Operation – An accelerometer has a temperature range between -50 degree Celsius to 120 degree Celsius. This range can be obtained only by accurate installment of the device.

4. Sensitivity – The device must be designed in such a way that it has higher sensitivity. That is, even for a small accelerative force, the electrical output signal should be very high. Thus a high signal can be measured easily and is sure to be accurate.
5. Axis – Most of the industrial applications requires only a 2-axis accelerometer. But if you want to go for 3D positioning, a 3-axis accelerometer will be needed.
6. Analog/Digital Output – You must take special care in choosing the type of output for the device. Analog output will be in the form of small changing voltages and digital output will be in PWM mode.

3.2.5.5. Device Selection

Selection of the device depends on the following factors.

1. Selection depends on the range of frequency you need.
2. Depends on the size and shape of the object whose acceleration is to be measured.
3. Whether the measurement environment is dirty or clean.
4. Depends on the range of vibration that is to be measured.
5. Depends on the range of temperature in which the device will have to work.
6. Depends on whether the device is to be case grounded or grounded isolated.

3.2.5.6. Applications

1. Machine monitoring.
2. Used to measure earthquake activity and aftershocks.
3. Used in measuring the depth of CPR chest compression.
4. Used in Internal Navigation System (INS). That is, measuring the position, orientation, and velocity of an object in motion without the use of any external reference.
5. Used in airbag shooting in cars and vehicle stability control

6. Used in video games like PlayStation 3, so as to make the steering more controlled, natural and real.

7. Used in camcorder to make images stable.

3.2.6 SOFTWARE EXPLANATION:

- Arduino software
- Proteus simulation

3.2.6.1 ARDUINO SOFTWARE:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are

completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

3.2.6.2 WHY ARDUINO?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handy board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and

wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other micro controller plat forms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows. Wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++

libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

- Open source and extensible hardware - The plans of the Arduino boards are published under

a Creative Commons license, so experienced circuit designers c an make their own version
Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money

- Getting Started with Arduino and Genuino products: -

Install the Arduino Software (IDE) on Windows PCs-
This document explains how to install the Arduino Software (IDE) on Windows machines.

✓ Download the Arduino Software (IDE)

✓ Proceed with board specific instructions.

3.2.6.3 HOW TO DOWNLOAD THE ARDUINO SOFTWARE (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.
When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

3.2.6.4 INSTALLATION:

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.3.2.6.4.1 Standard USB Cable (A plug to B plug)

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig. 3.2.6.4.2 Standard USB Cable (an A to Mini-B plug)

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

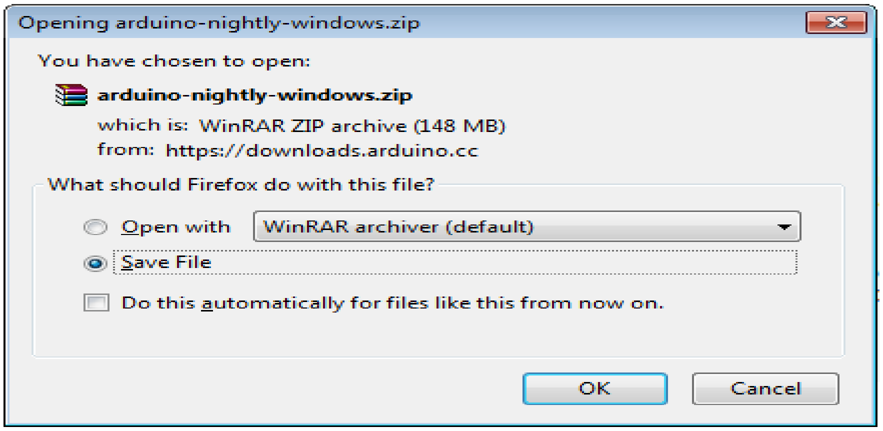


Fig. 3.2.6.4.3 Download Arduino IDE Software

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

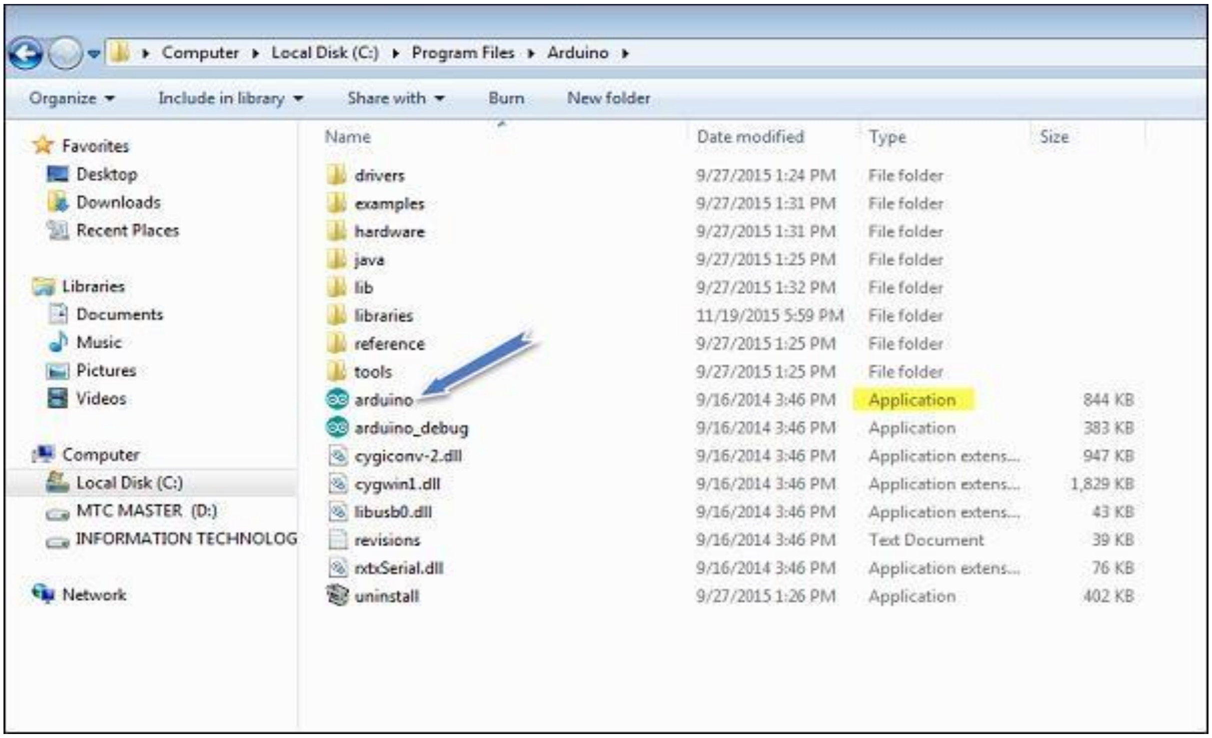


Fig. 3.2.6.4.4 Launch Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**

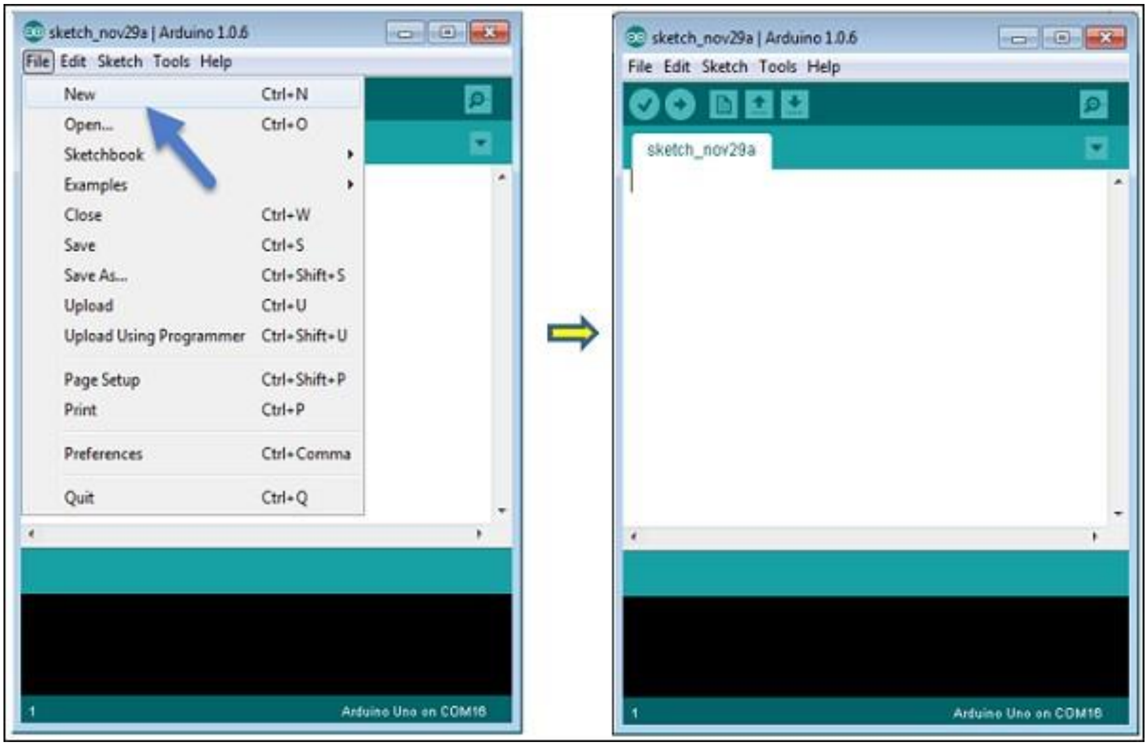
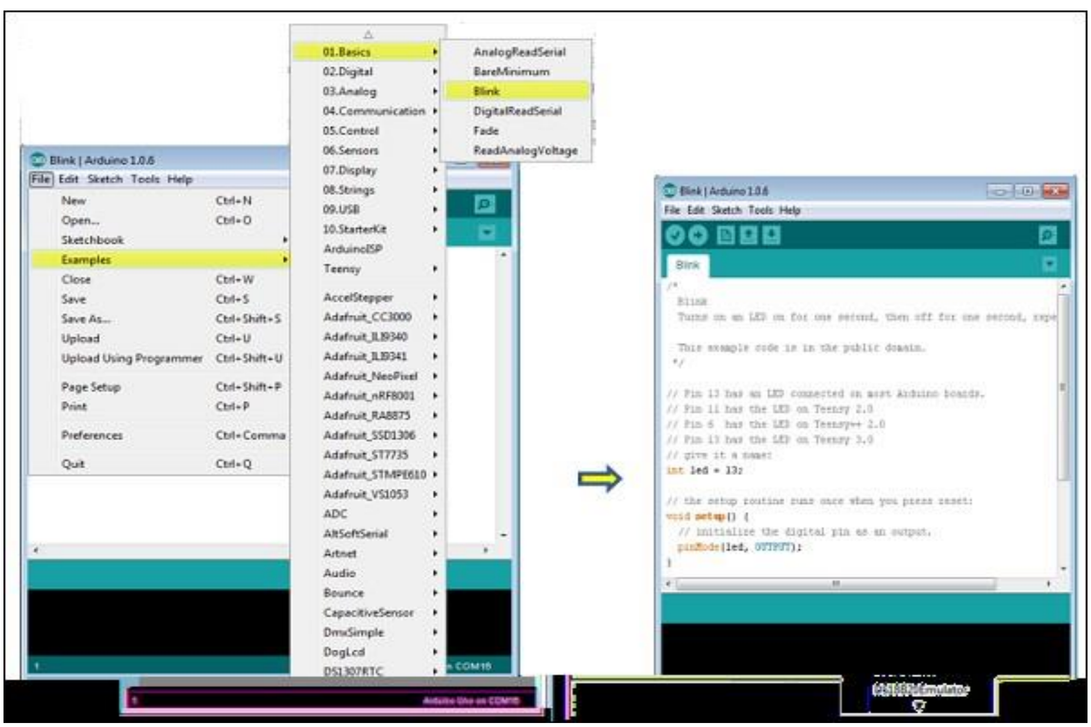


Fig. 3.2.6.4.5 Open your first project.

To open an existing project example, select File → Example → Basics → Blink.



.

Fig. 3.2.6.4.6 Open your first program.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

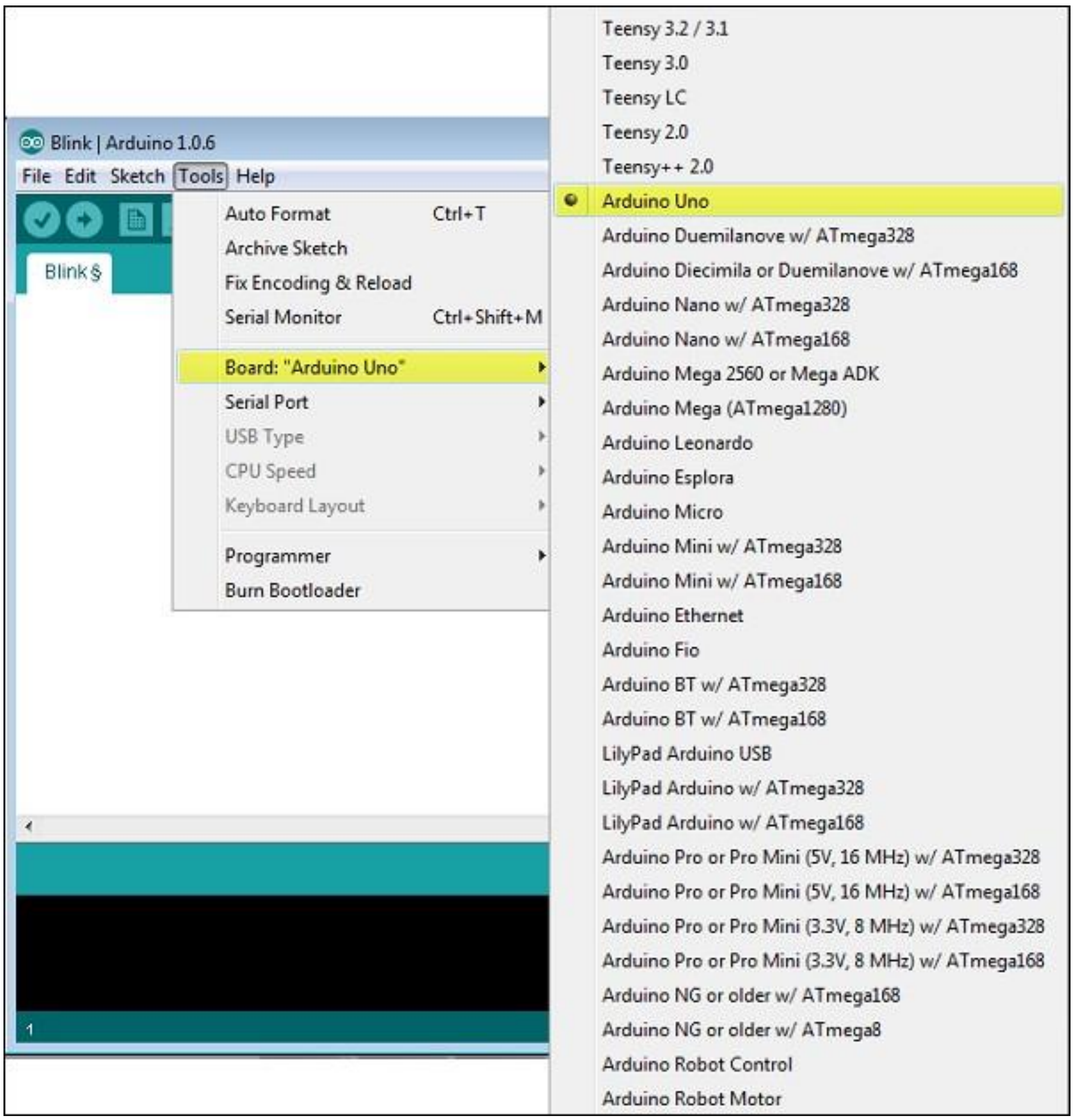


Fig. 3.2.6.4.7 Select your Arduino board.

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

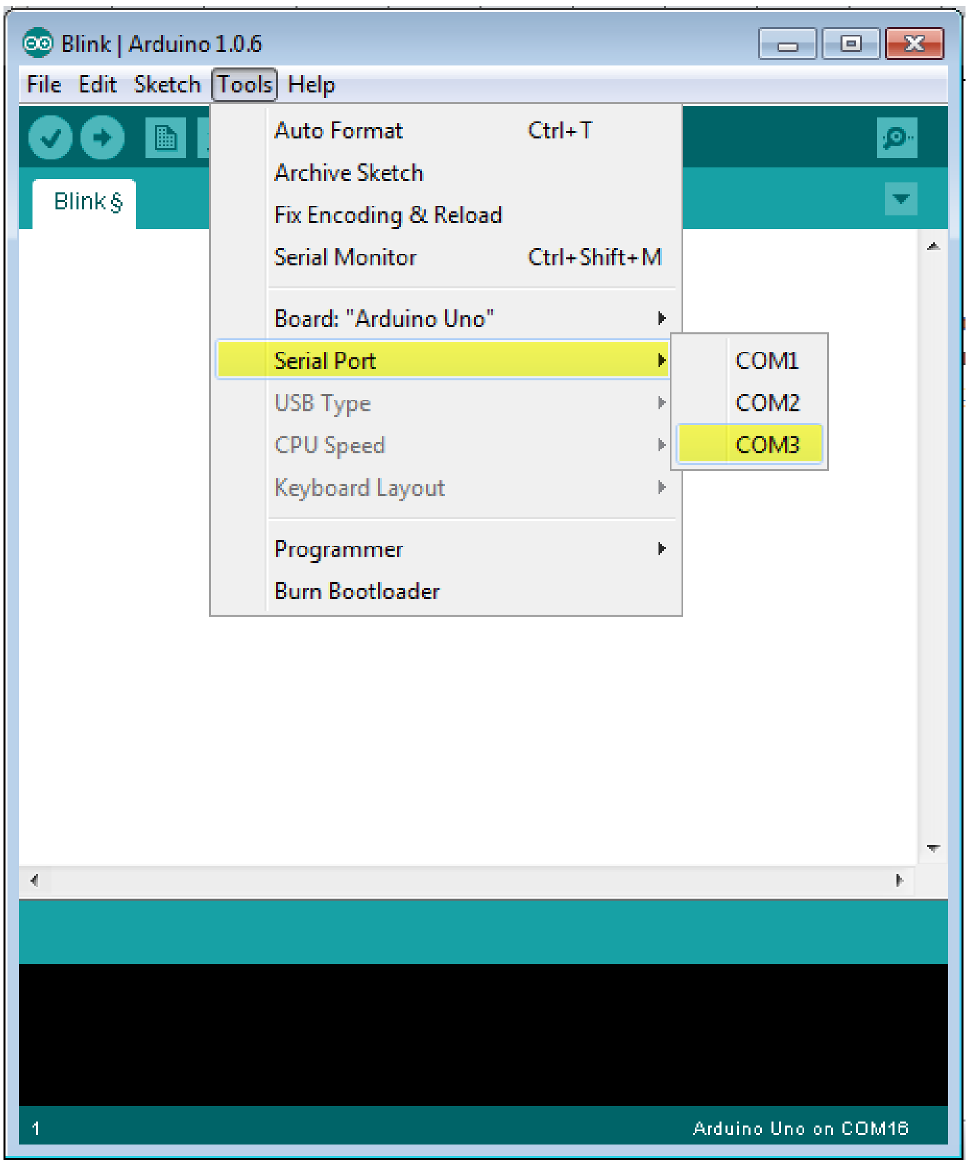


Fig. 3.2.6.4.8 Select your serial port.

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

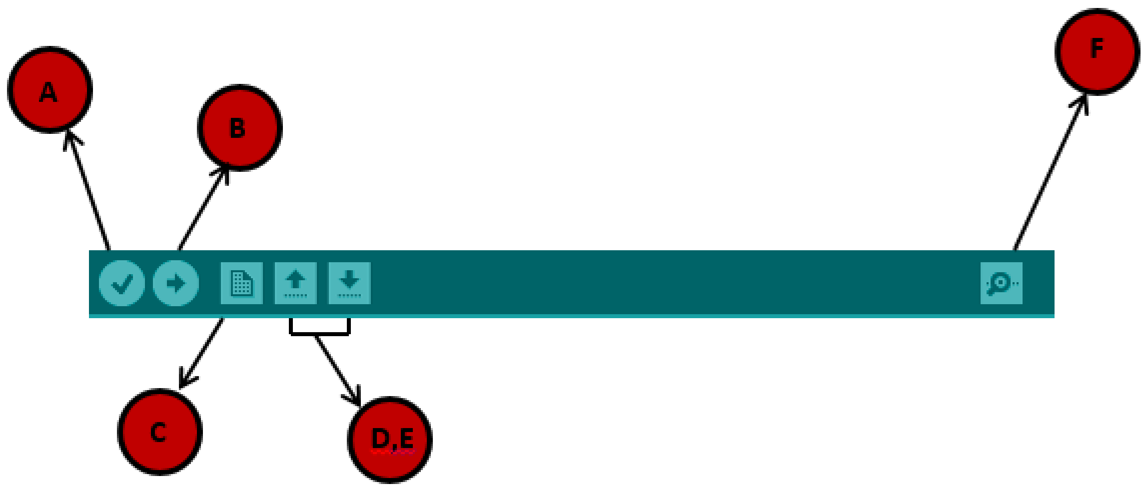


Fig. 3.2.6.4.9 Upload the program to your board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from board & send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

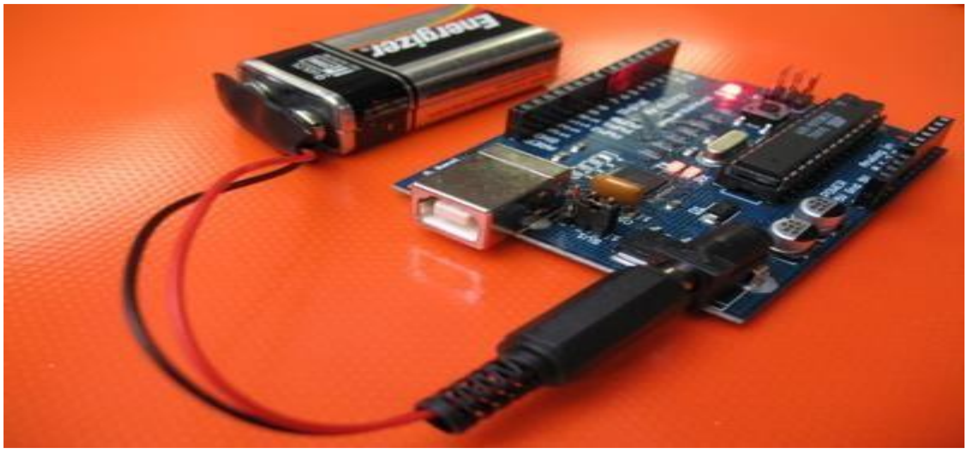


Fig. 3.2.6.4.10 Connecting a Battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called “sketches”, but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

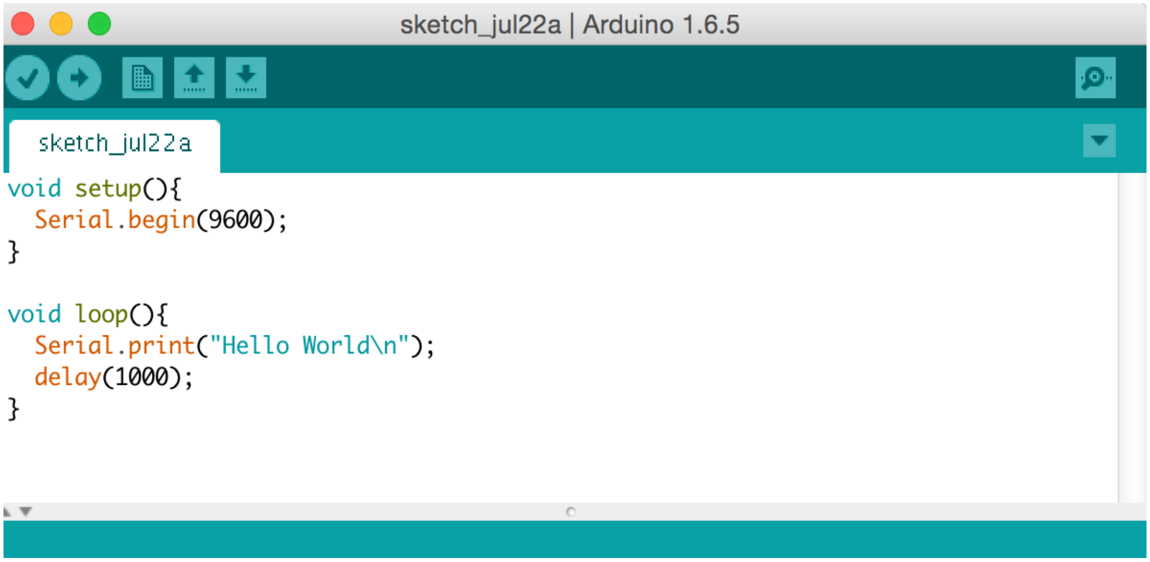


Fig. 3.2.6.4.11 Example Program

3.2.7 PROTEUS:

PROTEUS:

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline "From concept to completion".

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages.

It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design:

Step 1: Open ISIS software and select New design in File menu

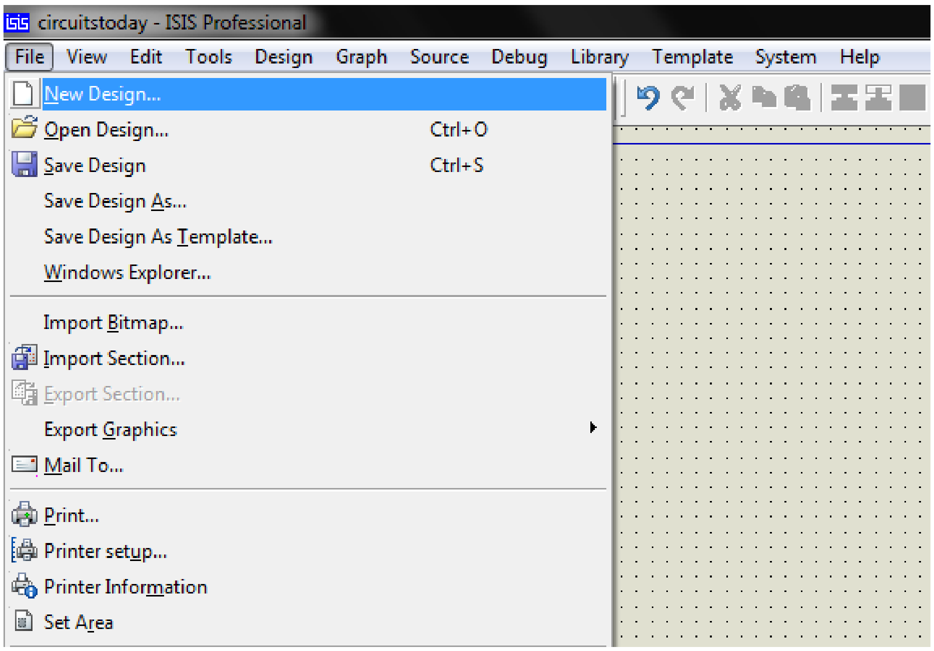


Fig. 3.2.7.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

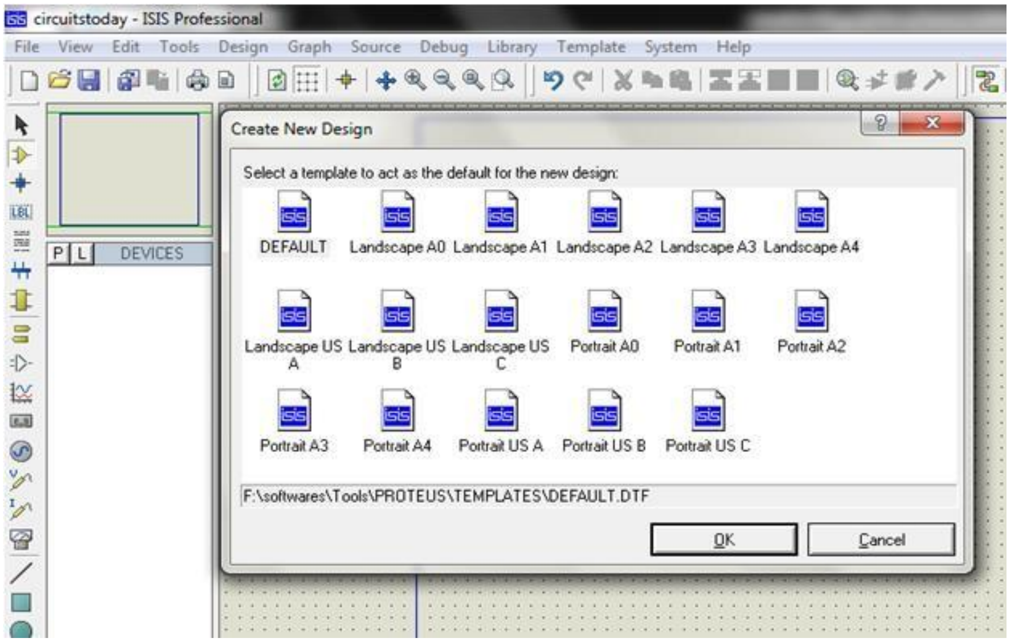


Fig. 3.2.7.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

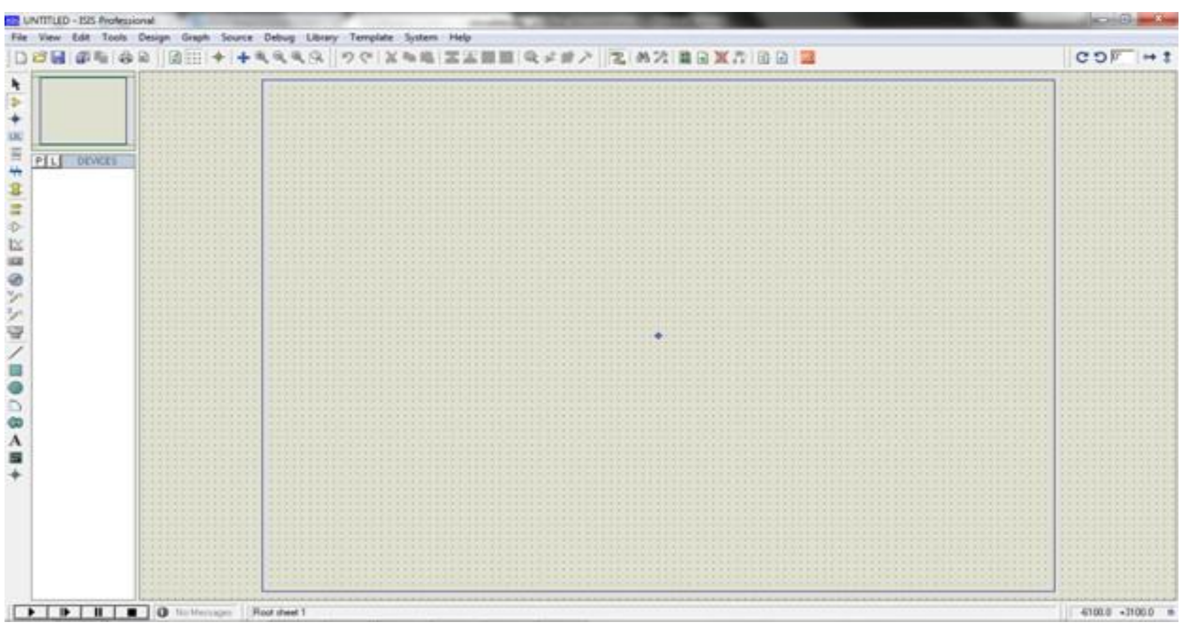


Fig. 3.2.7.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

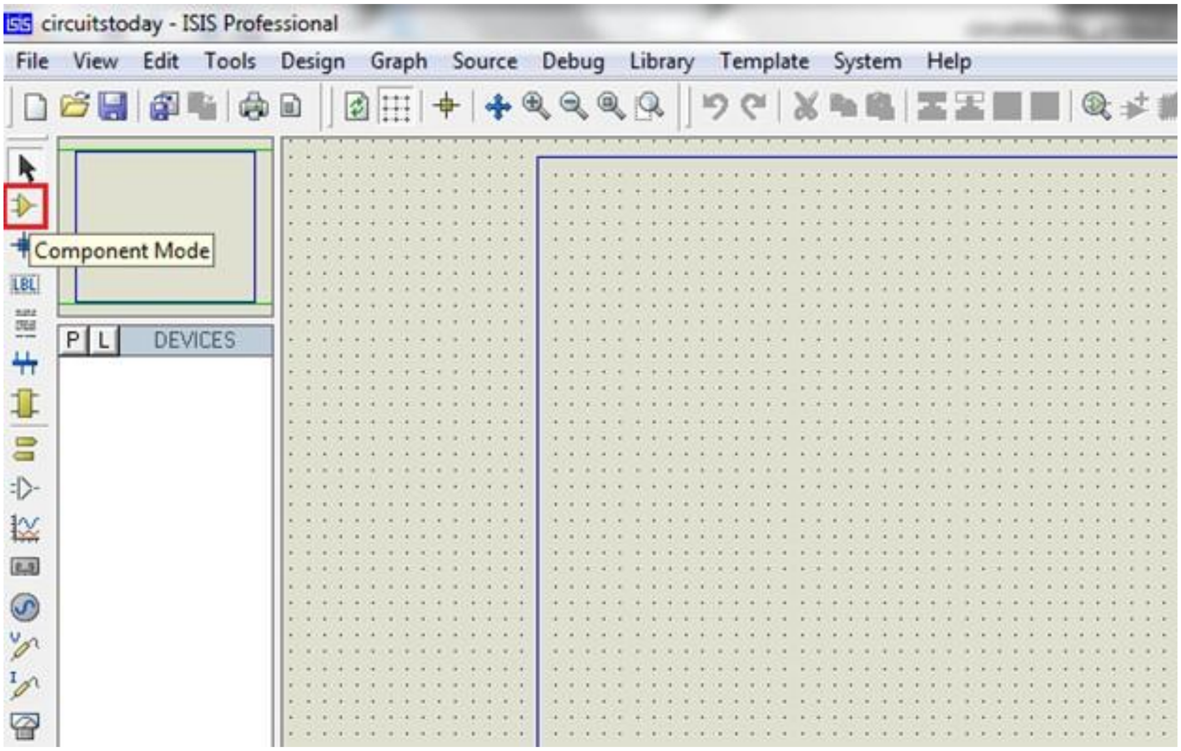


Fig. 3.2.7.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

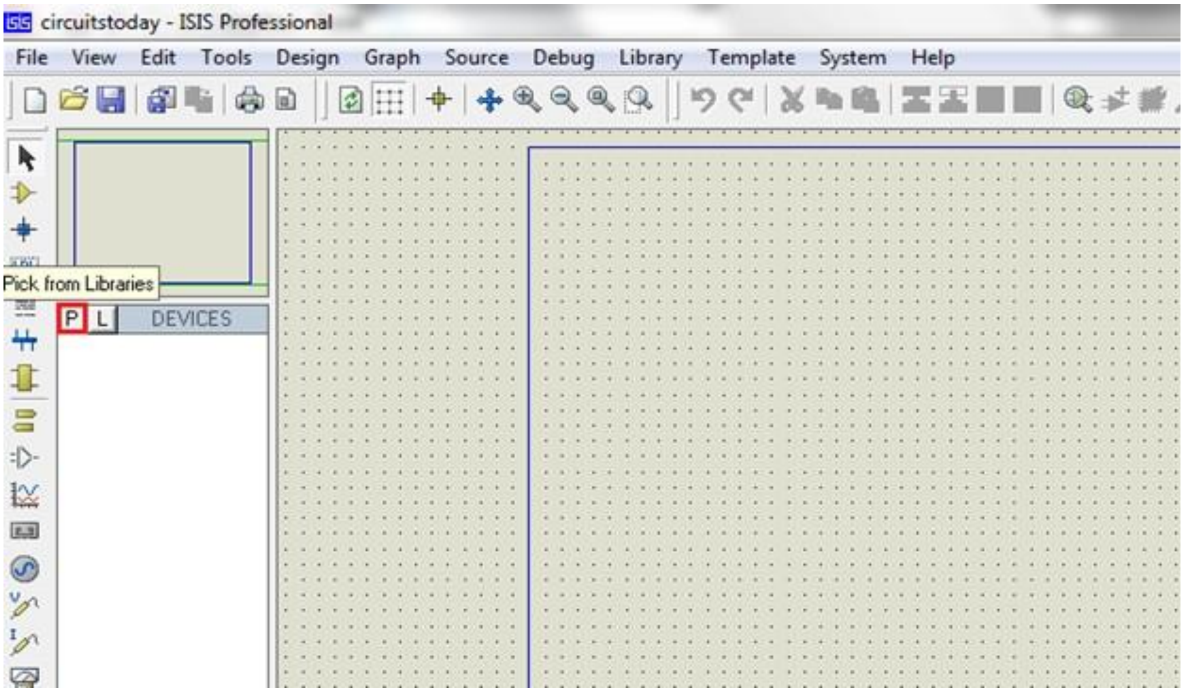


Fig. 3.2.7.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

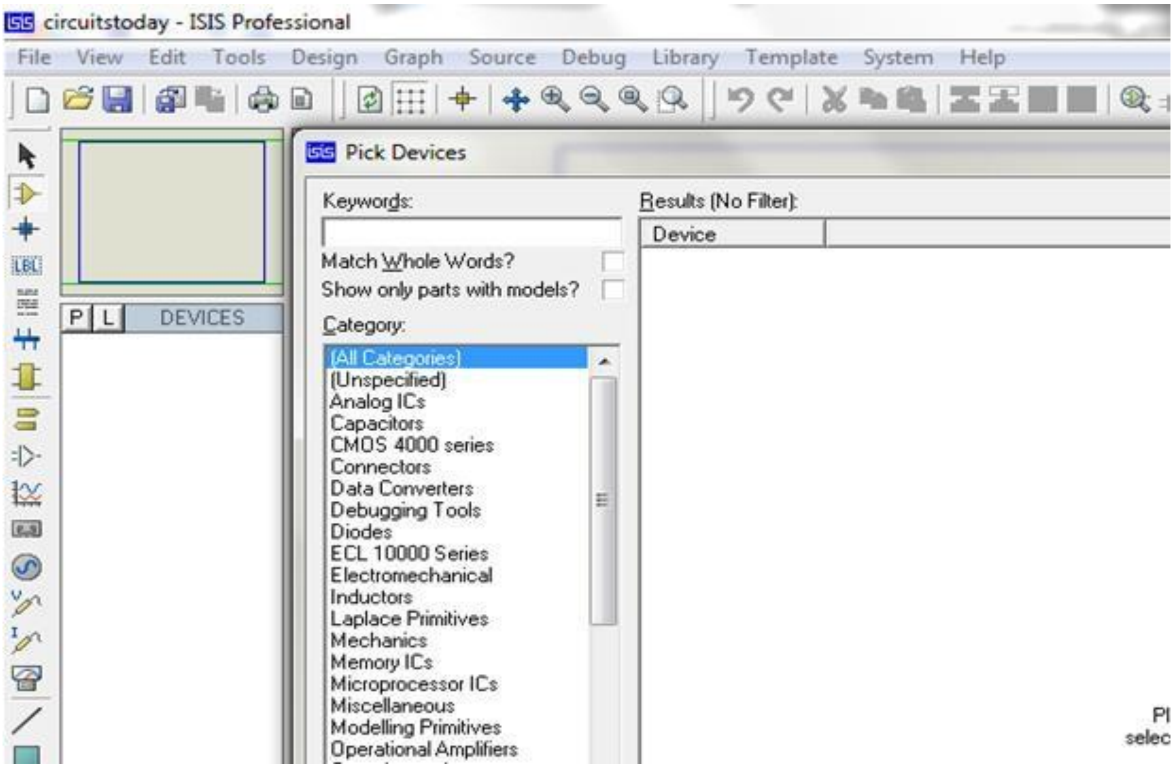


Fig. 3.2.7.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

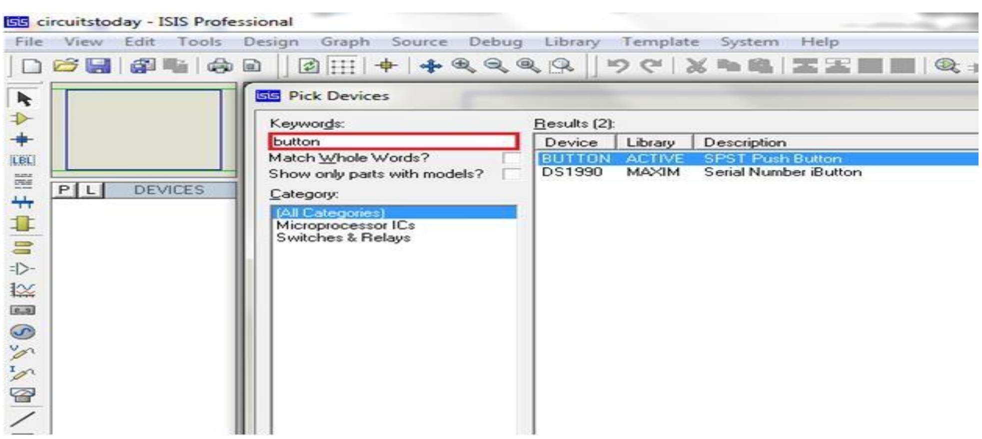


Fig. 3.2.7.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

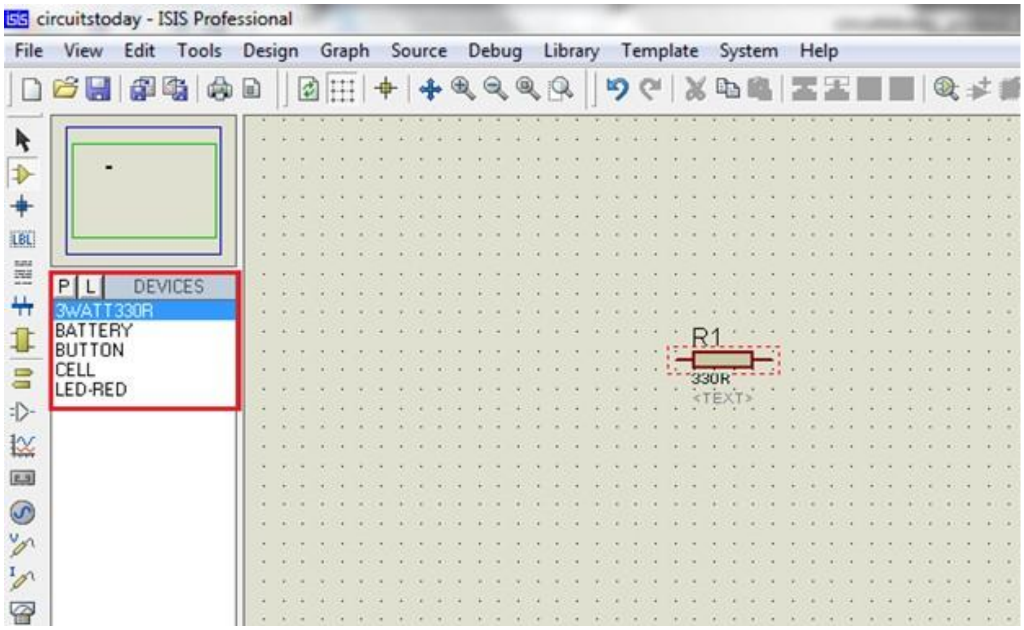


Fig. 3.2.7.8 Component Selection

Place all the required components and route the wires i.e., make connections. Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

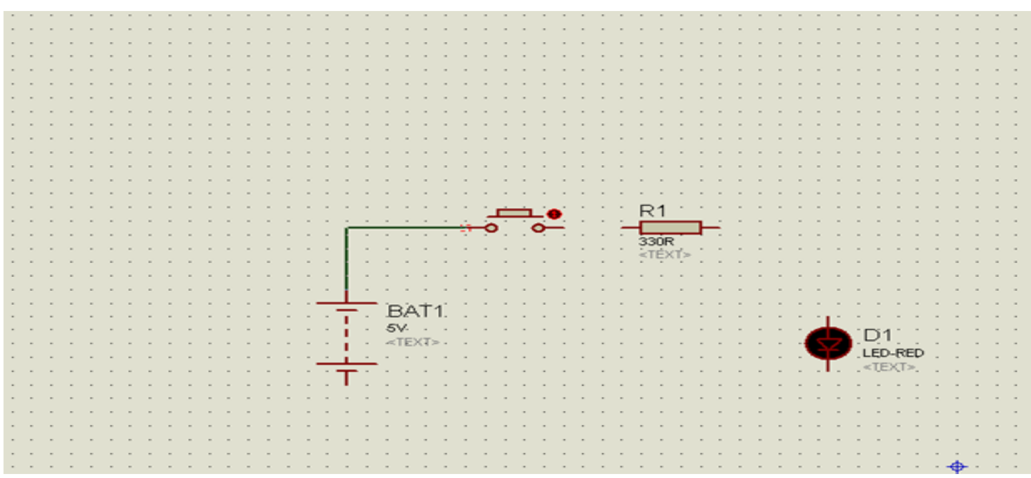


Fig. 3.2.7.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

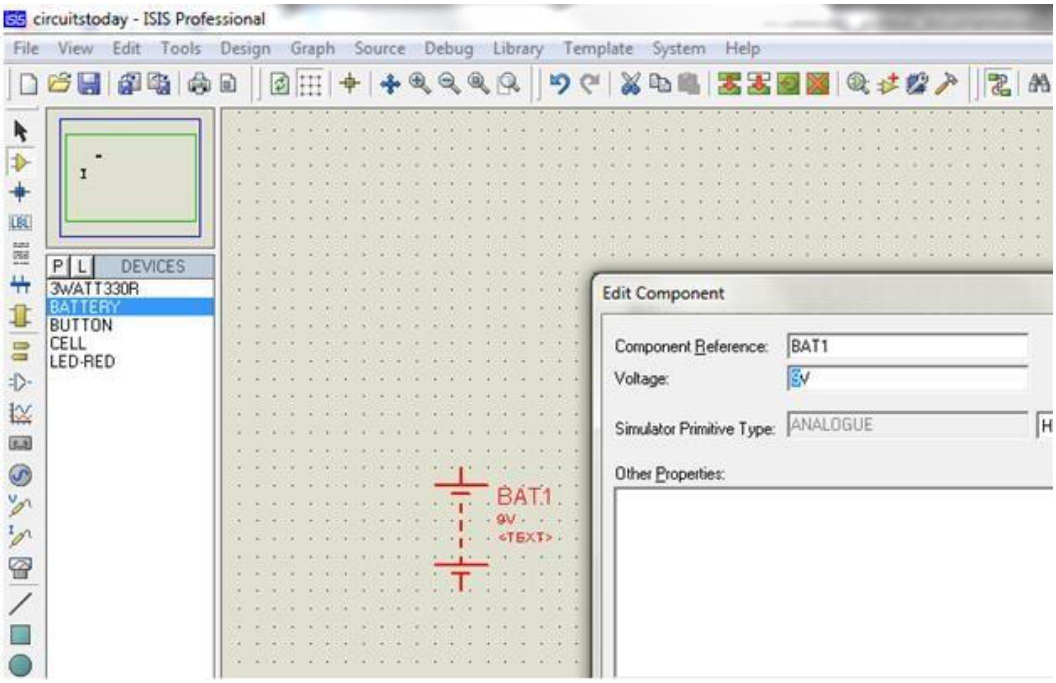


Fig. 3.2.7.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

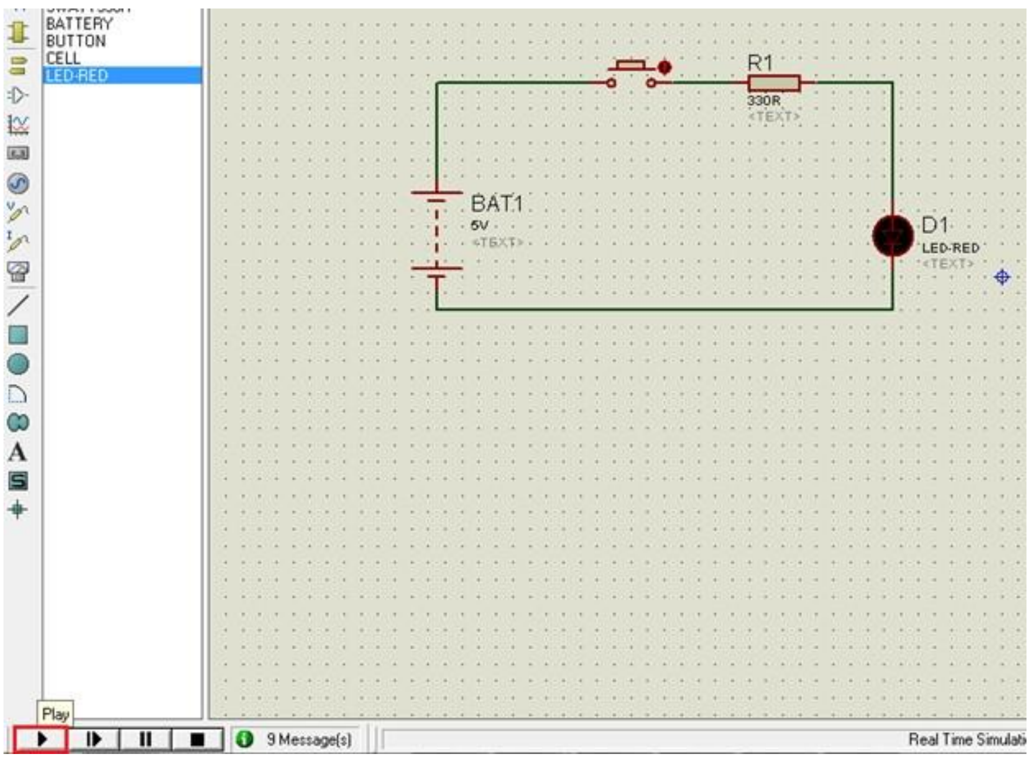


Fig. 3.2.7.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

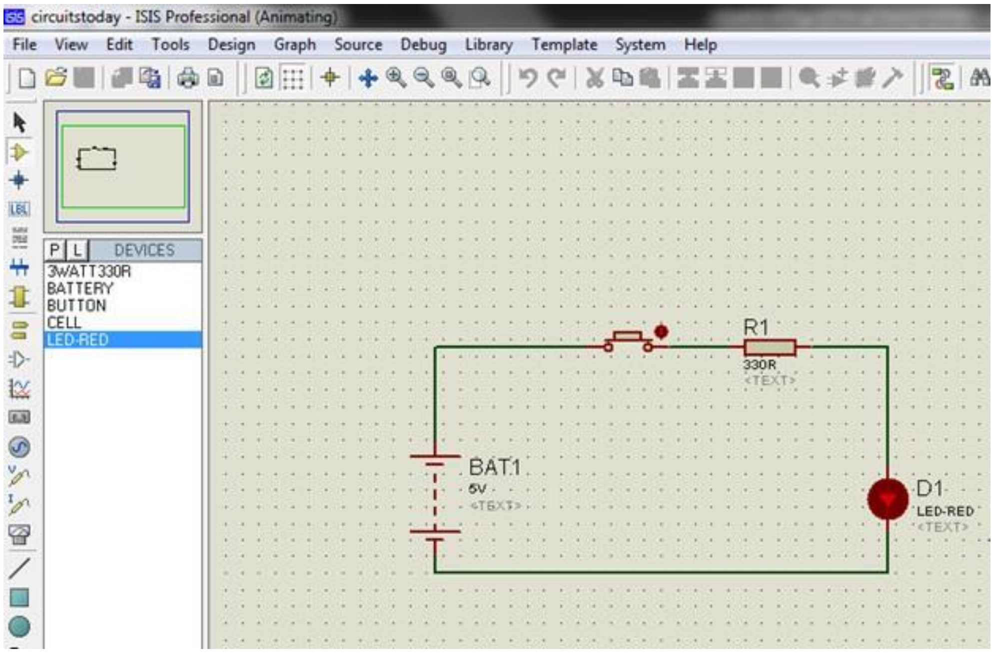


Fig . 3.2.7.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

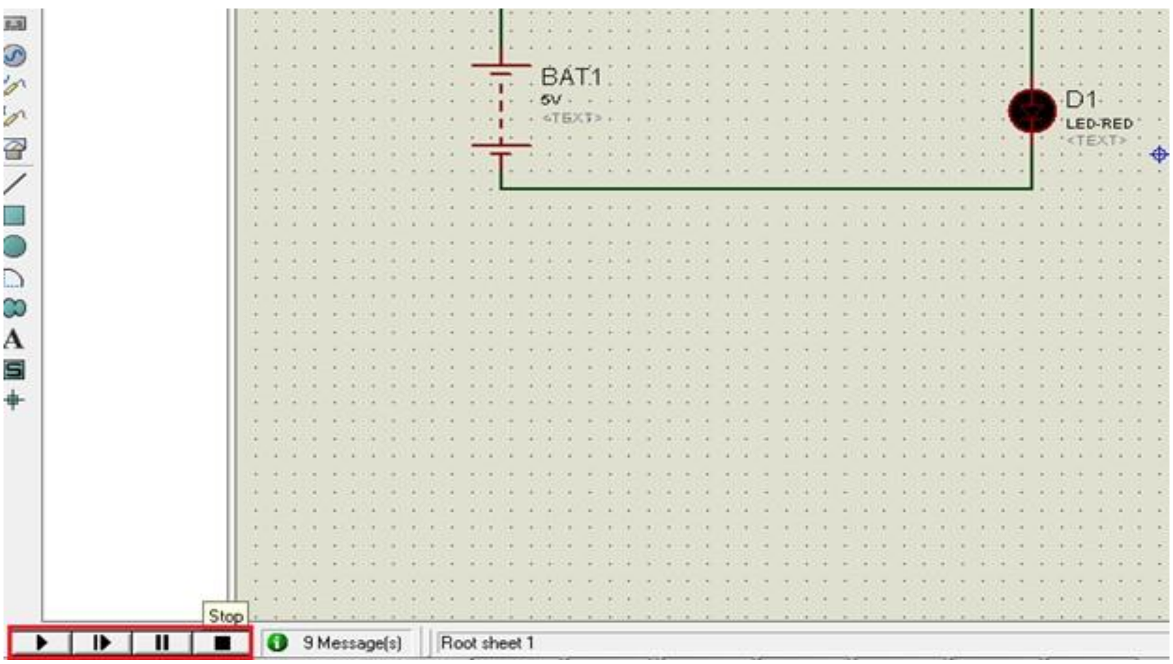


Fig. 3.2.7.13 Simulation Step-Pause-Stop Buttons

According to its developers, Things Speak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. Things Speak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates.

Things Speak was originally launched by iot Bridge in 2010 as a service in support of IoT applications.

Things Speak has integrated support from the numerical computing software MATLAB from MathWorks, allowing Things Speak users to analyse and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license from Math works.[citation needed].

Things Speak has a close relationship with Math works, Inc. In fact, all of the Things Speak documentation is incorporated into the Math works' Matlab documentation site and even enabling registered Math works user accounts as valid login credentials on the Things Speak website. The terms of service and privacy policy of ThingSpeak.com are between the agreeing user and Math works, Inc.

Setup Things Speak

Create a free MathWorks account or sign into Things Speak using an existing account.

Select the Things Speak channel you want your data to stream into. See Collect Data in a New Channel for help creating a new channel.

Record the following for the selected channel:

Channel ID, which is listed at the top of the channel view.

Write API key, which can be found on the API Keys tab of your channel view.

CHAPTER-4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM OF PROJECT:

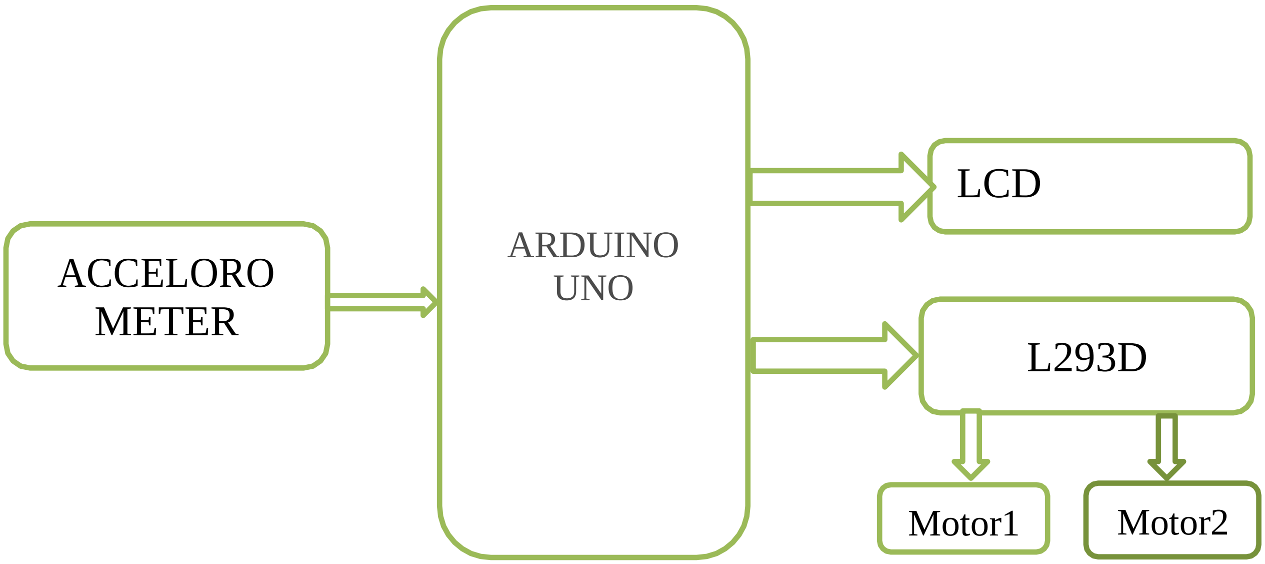


Fig.4.1: Block diagram of the project

4.2 FLOW CHART

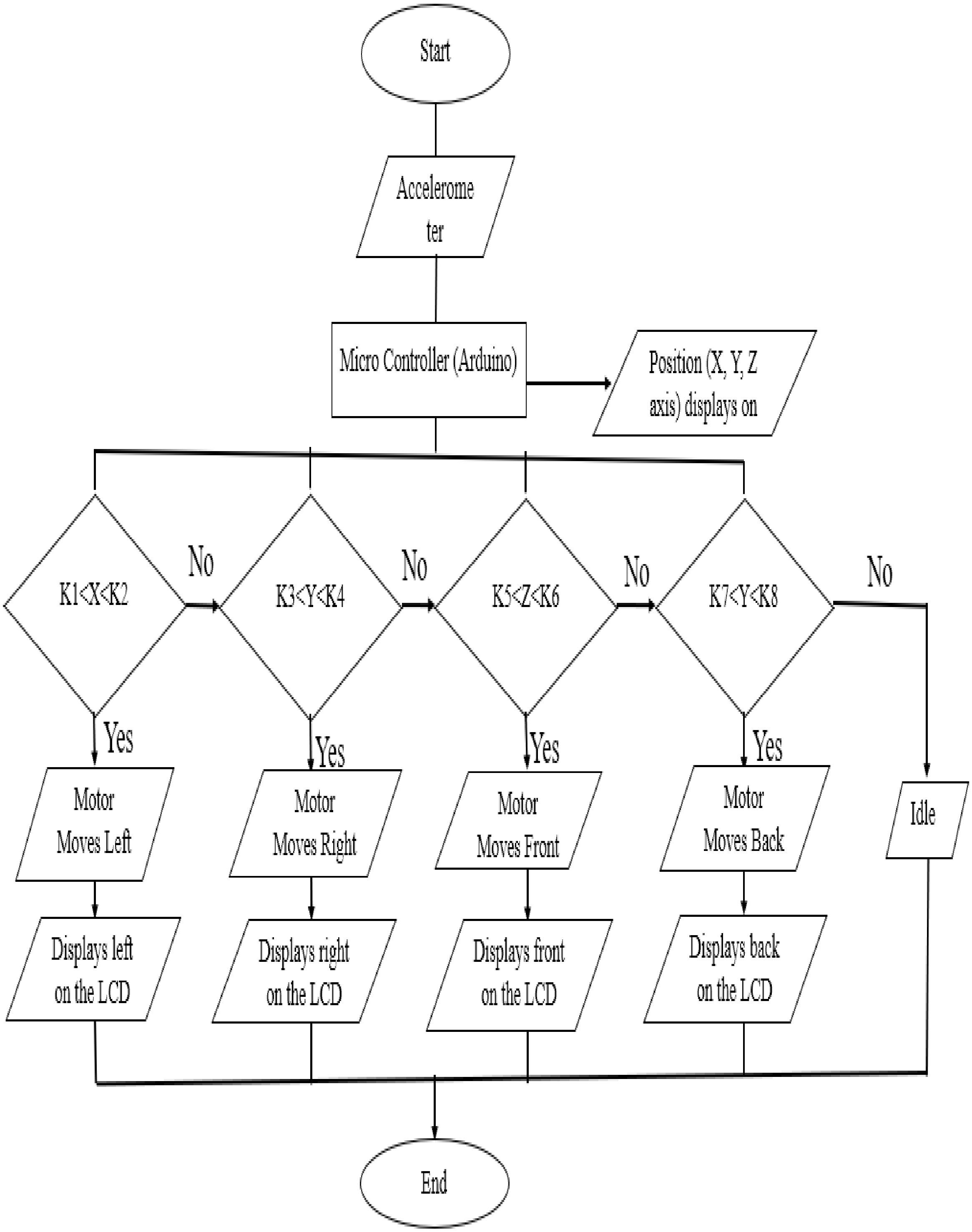


Fig.4.2 Flow Chart

4.3 IMPLEMENTATION STAGES

Stepwise implementation of Quadriplegics wheelchair Control by Head Motion using Accelorometer is given as follows

Step1: Switch on power supply

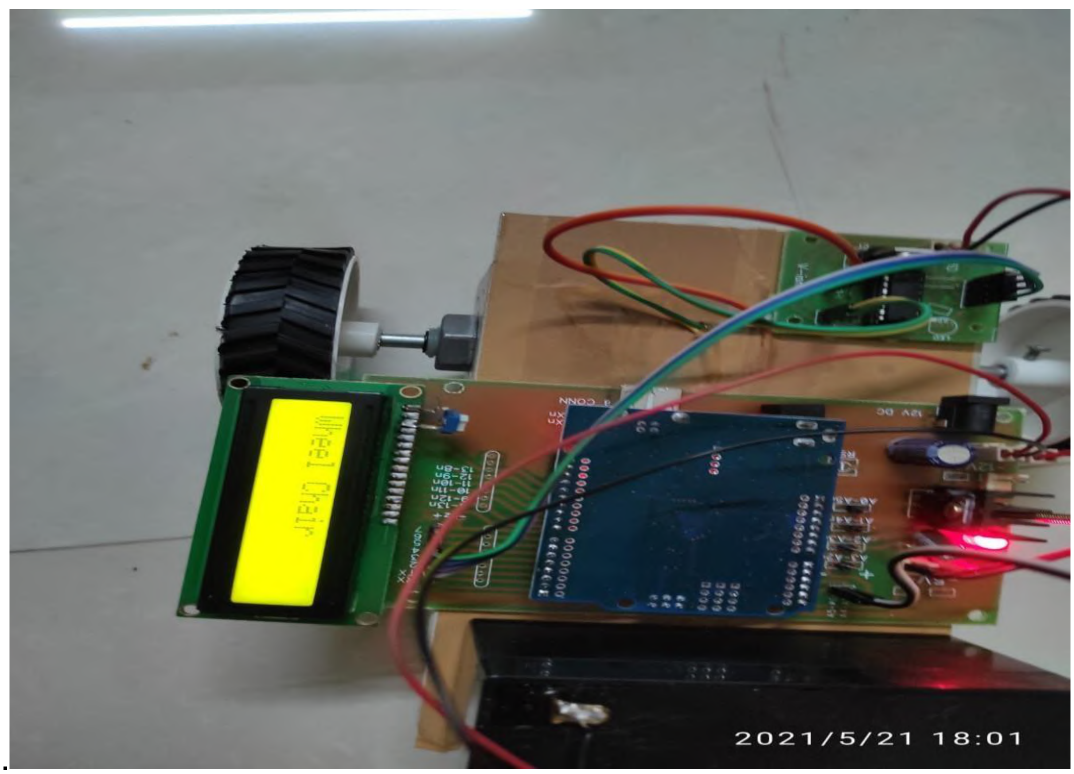
Step2: Place Accelorometer on head of the disabled person and if any tilt occurs then accelerometer will takes corresponding x, y, z values and sends to Arduino uno.

Step3: Micro controller will compares the x, y, z values with original values and then sends a message to LCD that contains x, y, z values**.**

Step4: At the same time the micro controller will sends the commands to dc motors ,then dc motors will acts correspondingly.

Step5: Then with the help of dc motors the chair will move left, right, front or back

4.4 RESULTS



.

Fig.4.4.1. wheelchair moving towards left

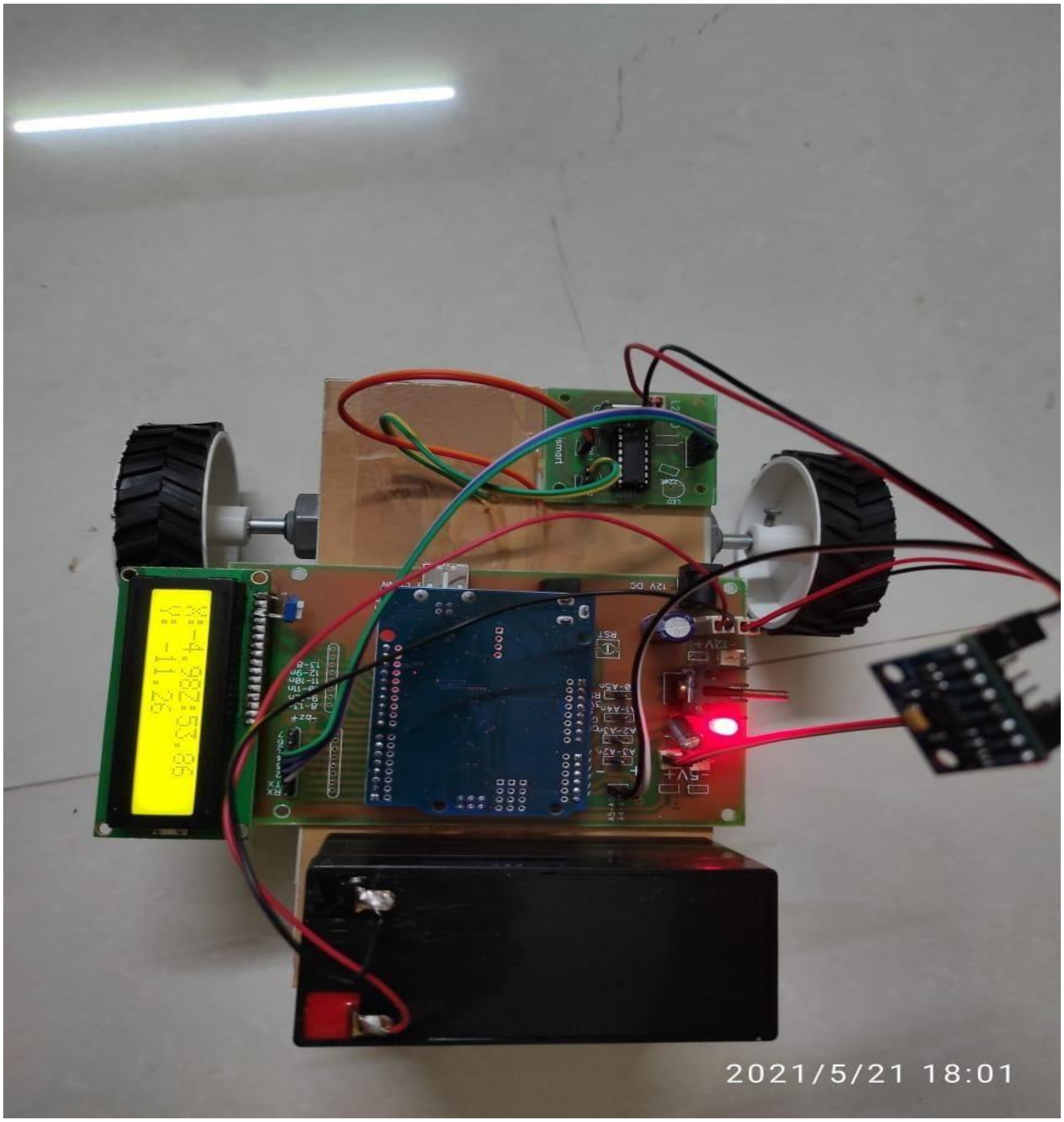


Fig 4.4.2: Wheelchair moving towards right

CHAPTER-5

PROJECT TESTING

5.1 OVERVIEW OF TESTING METHODS

5.1 SOFTWARE TESTING

Software Testing is a process of executing the application with intent to find any software bugs. It is used to check whether the application met its expectations and all the functionalities of the application is working.

The final goal of testing is to check whether the application is behaving in the way it is supposed to under specified conditions. All aspects of the code are examined to check the quality of application. The primary purpose of testing is to detect software failures so that defects may be uncovered and corrected. The test cases are designed in such way that scope of finding the bugs is maximum.

5.2 TESTING LEVELS:

There are various testing levels based on the specificity of test.

• **Unit testing:** Unit testing refers to tests conducted on a section of code in order to verify the functionality of that piece of code. This is done at the function level.

• **Integration Testing:** Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Its primary purpose is to expose the defects associated with the interfacing of modules.

• **System Testing**: System testing tests a completely integrated system to verify that the system meets its requirements.

• **Acceptance testing**: Acceptance testing tests the readiness of application, satisfying all requirements.

• **Performance testing:** Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or device such as response time or millions of instructions per second etc. 62

5.3 SYSTEM TEST CASES

A test case is a set of test data, preconditions, expected results and post conditions, developed for a test scenario to verify compliance against a specific requirement. We have designed and executed a few test cases to check if the project meets the functional requirements.

5.4 CODE IMPLEMENTATION

```
#include<SoftwareSerial.h>

#include<LiquidCrystal.h>

Liquid Crystal lcd(13,12,11,10,9,8);

#include <Wire. h>

#include <Adafruit_Sensor. h>

#include <Adafruit_ADXL345_U.h>

/* Assign a unique ID to this sensor at the same time */

Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);


int m1_1=2,m1_2=3,m2_1=4,m2_2=5;

void setup()

{

  pinMode(m1_1,OUTPUT);

  pinMode(m1_2,OUTPUT);

  pinMode(m2_1,OUTPUT);

  pinMode(m2_2,OUTPUT);

  digitalWrite(m1_1,LOW);

  digitalWrite(m1_2,LOW);

  digitalWrite(m2_1,LOW);

  digitalWrite(m2_2,LOW);

  lcd.begin(16,2);

  lcd.clear();lcd.setCursor(0,0);lcd.print("Wheel Chair");
```

```
  delay(2000);

  Serial.begin(9600);

  /* Initialise the sensor */

if(!accel.begin())

{

/* There was a problem detecting the ADXL345 ... check your connections */

lcd.print("NO Acclerometer");

while(1);

}

/* Set the range to whatever is appropriate for your project */

accel.setRange(ADXL345_RANGE_16_G);

}

void loop(void)

{

/* Get a new sensor event */

sensors_event_t event;

accel.getEvent(&event);

/* Display the results (acceleration is measured in m/s^2) */

lcd.clear();lcd.setCursor(0,0);

lcd.print("X:"); lcd.print(event.acceleration.x);lcd.print("Z:"); lcd.print(event.acceleration.z);

lcd.setCursor(0,1);

lcd.print("Y: "); lcd.print(event.acceleration.y);

delay(100);
```

```
digitalWrite(m1_1,LOW);
digitalWrite(m1_2,LOW);
digitalWrite(m2_1,LOW);
digitalWrite(m2_2,LOW);
delay(1000);
if(event.acceleration.z >=49 && event.acceleration.z <50 )
{
  lcd.clear();lcd.setCursor(0,0);
  lcd.print("Forward");
  delay(100);
  digitalWrite(m1_1,LOW);
  digitalWrite(m1_2,HIGH);
  digitalWrite(m2_1,LOW);
  digital Write(m2_2,HIGH);
  delay(2000);
}
if(event. acceleration. y >=9 && event. acceleration. y <10 )
{
  lcd. clear();lcd.setCursor(0,0);
  lcd.print("Backward");
  delay(100);
  digitalWrite(m1_1,HIGH);
  digitalWrite(m1_2,LOW);
```

```
  digitalWrite(m2_1,HIGH);
  digitalWrite(m2_2,LOW);
  delay(2000);
}
if(event.acceleration.x >=5 && event.acceleration.x <6 )
{
  lcd.clear();lcd.setCursor(0,0);
  lcd.print("Left");
  delay(100);
  digitalWrite(m1_1,LOW);
  digitalWrite(m1_2,LOW);
  digitalWrite(m2_1,LOW);
  digitalWrite(m2_2,HIGH);
  delay(2000);
}
if(event.acceleration.y >=1 && event.acceleration.y <2)
{
  lcd.clear();lcd.setCursor(0,0);
  lcd.print("Right");
  delay(100);
  digitalWrite(m1_1,LOW);
  digitalWrite(m1_2,HIGH);
  digitalWrite(m2_1,LOW);
```

```
  digitalWrite(m2_2,LOW);
delay(2000);
}
}
```

CHAPTER-6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The proposed framework is expected to make a financially savvy wheelchair to assist quadriplegic with peopling who think that it's hard to move freely. The structure uses head advancement to control the wheelchair. The tilt focuses made are distinguished and voltages are created by accelerometer. These voltages are taken by microcontroller which in this way controls the course of wheelchair.

This venture work was carried on to satisfy the prerequisite of older and impair individuals, giving the autonomous route utilizing head movement-controlled wheelchair. It causes them to move effectively as like ordinary people do.+

The circuit works appropriately to the order given by the user and the wheelchair is moved in understanding to the head signal given by the individual. Head movement-controlled wheelchair incorporates obstacle detection, which assists with distinguish in g the hindrance and stops the wheelchair. Monitoring of the patient's wellbeing condition is possible with the assistance of patient monitoring framework.

6.2 FUTURE ENHANCEMENT

- We can make a wheelchair which can be worked by a remote. Output of device are often applied to wireless transmitter circuit and might received at chair circuit by receiver electronic equipment. So wireless operation can reduce wiring arrangements
- Instead of using MEMS we can use eye retina using optical sensor to move wheelchair in different direction. Using tissue layer movement we'd be ready to drive a chair.
- Researchers are going on improvement of handicap wheelchair using nervous system of human..
- The structured wheelchair can be broadened utilizing sun powered boards which is increasingly productive. Sun powered board itself stimulates the wheelchair. But solely the downside is setup of panel and conjointly weight of chair will increase.

- Voice monitoring helps the disabled person to determine the obstacle by acknowledging with alarm signals. That can act as motor kill switch.

PUBLICATION

Submitted Paper in the Conference ICISECT - 21 with Paper ID ICSMEC21- 0065and got Acceptance for the Paper

REFERENCES

1. Satish Kumar, Dheeraj, Neeraj, Sandeep Kumar. Design and Development Of Head Motion Controlled Wheelchair, IJAET,2015.
2. S. Shaheen, A. Umakeswari., Intelligent Wheelchair for People with Disabilities, International Journal of Engineering and Technology (IJET), Vol.5, No.1, pp.391-397, Feb-Mar2013.
3. Aleksandar, P. Branko, D.: Wheelchair control by head motion. Serbian Journal of Electrical Engineering, 10(1), 2013, p. 135151.
4. Pushpendra Jhaand PreetiKhurana, Hand Gesture Controlled Wheelchair, IJCTA, 9(41), 2016, pp.243-249.
5. Prof. Vishal V. Pande, Nikita S.Ubale, Darshana, P. Masurkar, Nikita, R. Ingole, Pragati, P. Mane, Hand Gesture Based Wheelchair Movement Control for Disabled Person Using MEMS., Int. Journal of Engineering Research and Applications, www.ijera.com, ISSN : 2248-9622, Vol. 4, Issue 4(Version 4), April 2014,pp.152-158.
6. Vijendra.P. Meshram and pooja. A. Rajkumar, International Journal of Advanced Research in Computer Science and Software Engineering, vol.5, issue.1, January 2015, pp.641-646.
7. Narendar Kumar and Vidhi, Two Dimension Head Movements Based Smart Wheelchair Using Accelerometer, International Journal of Scientific Engineering and Research, vol. 2, issue. 7, July 2014, pp. 9-11.

BIBILOGRAPHY

[1] E.J Rechy Ramirez, “Head Movements Based Control of an Intelligent Wheel Chair in an Indoor Environment”, IEEE International Conference on Robotics and Biometrics, pp.1464-1469, Dec.2012, doi:10.1109/ROBIO.2012. 6491175.

[2] Narendar Kumar and Vidhi, “Two Dimension Head Movements Based Smart Wheelchair Using Accelerometer”, International Journal of Scientific Engineering and Research, vol. 2, issue.7, July 2014, pp.9- 11.

[3] V. Kumar, Vignesh S.N and Barathi Kannan K, “Head Motion Controlled Robotic Wheelchair”, International Journal of Emerging Technology and innovative Engineering, vol.1, issue.3, March 2015, pp.176-179.

[4] Vijendra. P. Meshram and pooja. A. Rajkumar, “International Journal of Advanced Research in Computer Science and Software Engineering”, vol.5,issue.1,January 2015,pp.641-646.

APPENDICES

This survey is the accomplishment of the task where gesture controlled user interface for elderly and disable people has been. From this survey it has been identified that elderly and disable needs more technology support using their nature behaviour, considering their limitations. We can use affordable technology for daily activities. The wheelchair is fully capable of carrying the load up to 110Kg, and moving in accordance to the head gesture given by the person who is using the wheel chair. Certain improvisation and improvement can be done to make the wheelchair more reachable to those whose whole body is paralyzed. Certain eyes gesture or brain signals reader can be imparted on the wheelchair system so as to make it better. For now, it works for all kind of disabled or elderly person, and even for those patients whose whole body is paralyzed but still head movement is possible.

A

PROJECT REPORT

On

UNPAIRED IMAGE DENOISING

Submitted by

1) Mr.B.ShivaPrasad Reddy (17K81A0406)

2)Ms.G.Grishma(17K81A0415)

3) Ms.T.Pooja Singh(17K81A0452)

in partial fulfillment for the award

of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Ms.T.Sowmya.

Assistant Professor.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

NBA &NAAC A+ Accredited

Dhulapally, Secunderabad – 500 100

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

NBA &NAAC A+ Accredited

Dhulapally, Secunderabad-500 100

2020-2021.

Department of Electronics & Communication Engineering.

BONAFIDE CERTIFICATE

This is to certify that the project entitled **UNPAIRED IMAGE DENOISING** is being submitted by **1. B.ShivaPrasad Reddy (17K81A0406), 2. G.Grishma (17K81A0415), 3. T.Pooja Singh (17K81A0452) in** partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in electronics and communication engineering** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

PROJECT GUIDE	HEAD OF THE DEPARTMENT
Ms. T. SOWMYA Assistant Professor Department of ECE	Dr. B. HARI KRISHNA Professor Department of ECE

EXTERNAL EXAMINER

Place:

Date:

DECLARATION

We, the student of **Bachelor of Technology** in Department of 'ELECTRONICS AND COMMUNICATION ENGINEERING', session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled Unpaired Image Denoising is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university foraward of any degree.

B.ShivaPrasad Reddy	**(17K81A0406)**
G.Grishma	**(17K81A0415)**
T.Pooja Singh	**(17K81A0452)**

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT BANALA SHIVAPRASAD REDDY WITH ROLL NO.17K81A0406, GRISHMA GOLLA WITH ROLL NO.17K81A0415, T.POOJA SINGH WITH ROLL NO.17K81A0452, OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "UNPAIRED IMAGE DENOISING" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna**, **Head & Professor**, the Dept. of electronics and communication engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept. Project Coordinator Dr.A.Anand** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide, **T.Sowmya, Assistant Professor**, department of ECE and our class project coordinator **A. Soumya, Assistant professor**, department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. B.ShivaPrasad Reddy (17K81A0406)
2. G.Grishma (17K81A0415)
3. T.Pooja Singh (17K81A0452)

ABSTRACT

Deep learning approaches in image processing predominantly resort to supervised learning. A majority of methods for image denoising are no exception to this rule and hence demand pairs of noisy and corresponding clean images. Only recently has there been the emergence of methods such as Noise2Void, where a deep neural network learns to denoise solely from noisy images. However, when clean images that do not directly correspond to any of the noisy images are actually available, there is room for improvement as these clean images contain useful information that fully unsupervised methods do not exploit. In this paper, we propose a method for image denoising in this setting. First, we use a flow-based generative model to learn a prior from clean images. We then use it to train a denoising network without the need for any clean targets. We demonstrate the efficacy of our method through extensive experiments and comparison.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF OUTPUT SCREENS

CHAPTER 1

INTRODUCTION

1.0 OVERVIEW OF THE PROJECT

First, we use a flow-based generative model to learn a prior from clean images. We then use it to train a denoising network without the need for any clean targets. We demonstrate the efficacy of our method through extensive experiments and comparisons.

1.1 OBJECTIVES OF THE STUDY

Each method has its own advantages and disadvantages. In this project, we summarize some important objective in the field of image denoising. First, we give the formulation of the image denoising problem, and then we present several image denoising techniques. In addition, we discuss the characteristics of these techniques. Finally, we provide several promising directions for future research.

1.2 SCOPE OF THE STUDY

With the explosion in the number of digital images taken every day, the demand for more accurate and visually pleasing images is increasing. However, the images captured by modern cameras are inevitably degraded by noise, which leads to deteriorated visual image quality. Therefore, work is required to reduce noise without losing image features (edges, corners, and other sharp structures).

Image de noising is a classical yet still active topic in low level vision since it is an indispensable step in many practical applications. The goal of image denoising is to recover a clean image x from a noisy observation y which follows an image degradation model y = x + v. One common assumption is that v is additive white Gaussian noise (AWGN) with standard deviation σ. From a Bayesian viewpoint, when the likelihood is known, the image prior modeling will play a central role in image denoising. Over the past few decades, various models have been exploited for modeling image priors, including nonlocal self-similarity (NSS) models sparse models, gradient models and Markov Noise corrupts virtually any image captured through a camera. The degradation due to noise is typically captured in the equation: Y = X + N where X is a clean image, N is noise and Y is the corresponding noisy version of X. Image denoising methods attempt to recover the clean image from its noisy version. Traditional methods such as BM3D, NSCR, WNNM rely on the self-similarity of image patches to denoise solely from noisy images. Methods such as that use deep learning have been proposed for image denoising. Although they achieve state-of-the-art performance along with excellent test times, they are all discriminative models. As a result, they require pairs of noisy images and their corresponding clean images. Recently, deep learning methods like Noise2Noise and Noise2Void have been proposed that use statistical properties of noisy image patches to eliminate noise. While these methods do not need any clean images, in situations where they are available, they cannot utilize the valuable information available in the clean images.

Despite their high denoising quality, most of the image prior-based methods typically suffer from two major drawbacks. First, those methods generally involve a complex optimization problem in the testing stage, making the de noising process time – consuming. Thus, most of the prior based methods can hardly achieve high performance without sacrificing computational efficiency. Second, the models in general are non-convex and involve several manually chosen parameters, providing some leeway to boost denoising performance.

To overcome the limitations of prior-based approaches, several discriminative learning methods have been recently developed to learn image prior models in the context of truncated inference procedure. The resulting models are able to get rid of the iterative optimization procedure in the test phase. Schmidt and Roth proposed a cascade of shrinkage fields (CSF) method that unifies the random field-based model and the unrolled half-quadratic optimization algorithm into a single learning framework. Chen et al. proposed a trainable nonlinear reaction diffusion (TNRD) model which learns a modified fields of experts [12] image prior by unfolding a fixed number of gradient descent inference steps. Some of the other related work can be found .Although CSF and TNRD have shown promising results toward bridging the gap between computational efficiency and denoising quality, their performance are inherently restricted to the specified forms of prior. To be specific, the priors adopted in CSF and TNRD are based on the analysis model, which is limited in capturing the full characteristics of image structures. In addition, the parameters are learned by stage-wise greedy training plus joint fine-tuning among all stages, and many handcrafted parameters are involved. Another no negligible drawback is that they train a specific model for a certain noise level, and are limited in blind image de noising.

Deep learning methods have been successfully applied in various computer vision tasks, including image classification and object detection and have dramatically improved the performance of these systems, setting the new state-of-the-art. Recently, very promising results have also been reported for image processing applications such as image restoration super-resolution and optical flow. The significant boost in performance achieved by deep networks can be mainly attributed to their advanced modeling capabilities, thanks to their deep structure and the presence of non-linearities that are combined with discriminative learning on large training datasets. However, most of the current deep learning methods developed for image restoration tasks are based on general network architectures that do not fully exploit problem-specific knowledge. It is thus reasonable to expect that incorporating such information could lead to further improvements in performance. Only very recently, Schmidt and Roth [34] and Chen and Pock [6] introduced deep networks whose architecture is specifically tailored to certain image restoration problems. However, even in these cases, the resulting models are local ones and do not take into account the inherent non-local self-similarity property of natural images. On the other hand, conventional methods that have exploited this property have been shown to gain significant improvements compared to standard local approaches. A notable example is the Block Matching and 3D Collaborative Filtering (BM3D) method which is a very efficient and highly engineered approach that held the state-of-the-art record in image denoising for almost a decade.

CHAPTER 2

LITERATURE REVIEW

Recently, deep learning methods like Noise 2 Noise and Noise 2 Void have been proposed that use statistical properties of noisy image patches to eliminate noise. While these methods do not need any clean images, in situations where they are available, they cannot utilize the valuable information available in the clean images. Corresponding author, email id: priyathamkat@smail.iitm.ac.in Ground truth Noisy input Our output Fig. 1: Sample result from our method. Observe that the fine details in the tree are restored without any noticeable blur even when the noise level in the input is high ($\sigma = 35$). Image taken from BSD68. Another important class of methods are prior based. Priors are crucial for obtaining a reasonable answer out of all the possible solutions for an ill-posed problem such as image denoising. With handcrafted priors, these methods can be used when clean images are not available. However, these priors have been criticized as they are often chosen for their computational or analytical convenience rather than accuracy. Deep learning has allowed for constructing more accurate priors. Deep image prior claims that the architecture of a convolutional neural network alone can act as a prior for natural images. Though the results are good, it is surprising as there is no mathematical justification for why this prior works. Going further, Chen et al have used a GAN to explicitly construct a prior for realistic noise which they use for denoising. In this paper, we propose an approach for denoising using another class of generative models, called flow-based generative models .These models learn an invertible transformation from a complex distribution like images to a simple one like the Gaussian distribution. They have been successfully used to generate realistic images. An example of this is the work by King ma et al., where in they use flow based models to generate photorealistic face images using the Celeb A HQ dataset.

CHAPTER 3

PROJECT DESIGN

3.1 OVERVIEW OF THE DESIGN

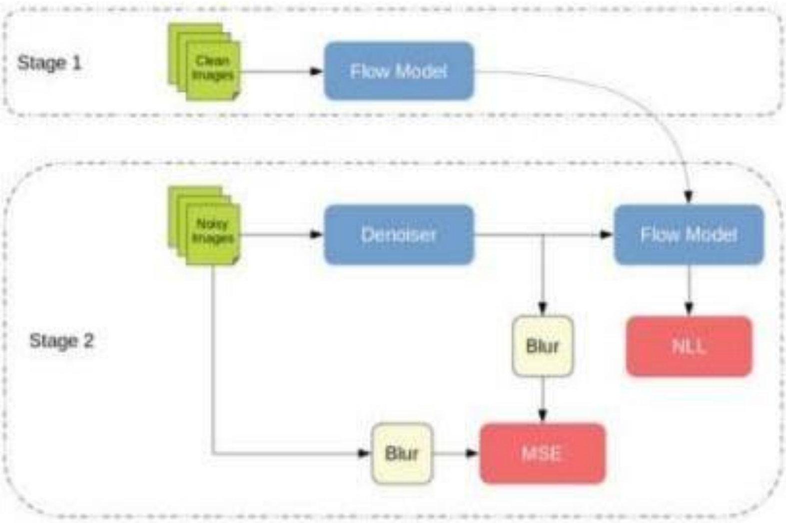


Fig 3.1 Proposed method.

Above figure. An illustration of our method. In the first stage, we train a flow-based model to learn a prior distribution on clean images. Next, we use this prior along with weak supervision train a denoising network. In this section we describe our two-stage approach to using the log-likelihood as a prior for image denoising.

Stage 1: Training the Flow model First, we train a flow-based model based on clean images to learn a transformation from clean images to the standard multivariate Gaussian random variable. Due to structure of the flow-based model as described in (2) and the tractable probability density of a Gaussian random variable, we can evaluate (6) for any given image and obtain the likelihood that the image is clean. Concretely, we train a flow-based model h to minimize the following objective:

$$-\log P(\boldsymbol{X}) = -\log P(\boldsymbol{Z}) + \sum_{i=1}^{n} \log \left| \frac{\mathrm{d}\boldsymbol{Z_i}}{\mathrm{d}\boldsymbol{Z_{i-1}}} \right| \quad (10)$$

$$= \frac{1}{2}\|\boldsymbol{Z}\|_2^2 + \sum_{i=1}^{n} \log \left| \frac{\mathrm{d}\boldsymbol{Z_i}}{\mathrm{d}\boldsymbol{Z_{i-1}}} \right| + C \quad (11)$$

where C is a constant that normalizes the Gaussian distribution. It has no bearing on the training and hence can be eliminated. Note that once the training in Stage 1 is complete, h is fixed during Stage 2.

Stage 2: Training the Denoiser Given a noisy image Y , the posterior distribution for the corresponding clean image X is

$$P(\boldsymbol{X} \mid \boldsymbol{Y}) = \frac{P(\boldsymbol{Y} \mid \boldsymbol{X})P(\boldsymbol{X})}{P(\boldsymbol{Y})} \qquad (12)$$

To obtain the maximum a posteriori (MAP) estimate of the clean image, the denominator can be ignored and the numerator or equivalently its log value is maximized.

$$\arg\max_{\boldsymbol{X}} \log P(\boldsymbol{X} \mid \boldsymbol{Y}) = \arg\max_{\boldsymbol{X}} \log P(\boldsymbol{Y} \mid \boldsymbol{X}) + \log P(\boldsymbol{X}) \qquad (13)$$

Assuming additive white Gaussian noise, log P(Y |X) is simply the negative of the squared error between Y and X. Using the flow model h trained in Stage 1, we can also compute the prior log-likelihood of X. Based on (13), we can formulate a loss function (note the change of signs as by convention, we want to minimize this loss) for the denoiser d as follows:

$$(Y - X)\ 2 - \lambda \log P(X) \qquad (14)$$

where λ is a hyper parameter that controls the relative importance of the conditional and the prior probability distributions. To be mathematically precise, λ depends on the noise level in the image. From our experiments, we also find that the performance of the denoiser is very sensitive to the choice of λ. This poses a challenge as we want to train a single denoiser for a range of noise levels. To reduce the dependency of λ on the noise level, we modify the first term in (14) to instead measure the squared error between blurred versions of X and Y. Intuitively speaking, we are training d to copy only the low frequency information from the input Y while adding details that make the output X to look cleaner. The flow model h dictates what details are added to Y. The final form of the loss function we use for the denoiser d is (B(Y) − B(X))2 − λ log P(X) (15) Here B is a local mean filter, the size of which is chosen to be 3 × 3, as that gave the best performance on the validation set.

3.2 BLOCK DIAGRAM

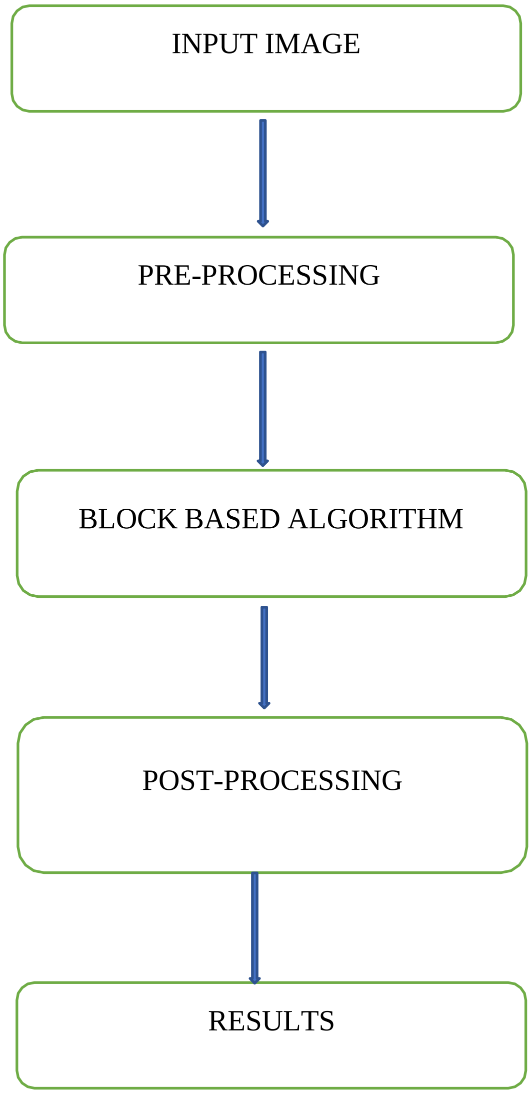


Fig 3.2 Block diagram

3.3 FLOW CHART

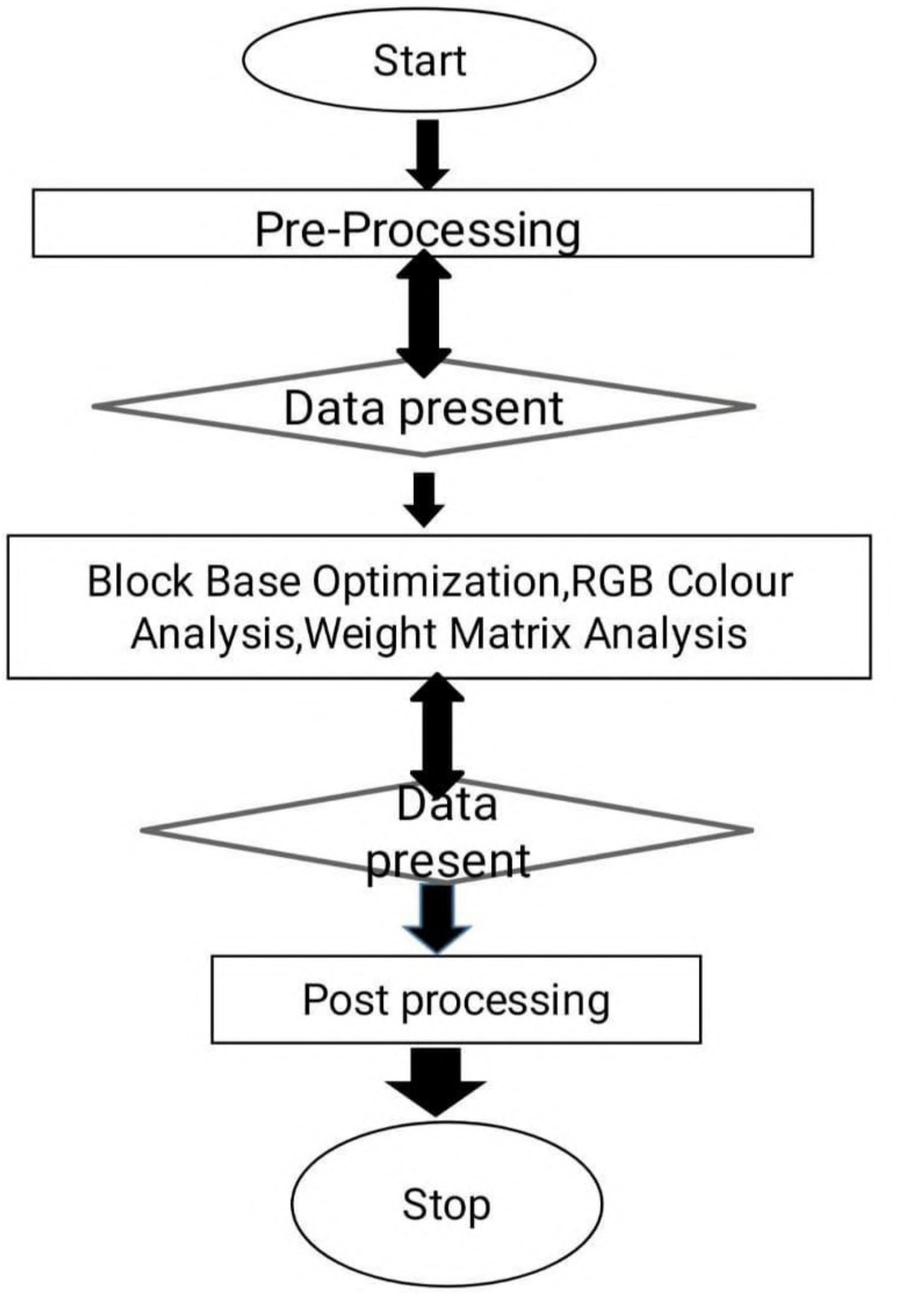


Fig 3.3 Flow chart

3.4 FLOW-BASED GENERATIVE MODELS:

Flow-based generative models [13, 14] learn the bijective transformation from a high-dimensional, complicated random variable X to a latent random variable Z. Typically, X represents images in a dataset while Z is assumed to be a standard normal random vector

$$\boldsymbol{Z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \qquad (1)$$

$$\boldsymbol{X} = h(\boldsymbol{Z}) \qquad (2)$$

To learn the transformation h, the following unbiased estimate of the negative log-likelihood of X is minimized:

$$\frac{1}{N}\sum_{i=1}^{N} - \log P(\boldsymbol{x_i}) \qquad (3)$$

Here, xi are samples from the dataset. Using the standard rules of random variable transformation, log P(X) can be written as

$$\log P(\boldsymbol{X}) = \log P(\boldsymbol{Z}) - \log \left| \frac{\mathrm{d}h}{\mathrm{d}\boldsymbol{x}} \right| \qquad (4)$$

where dh dx is the determinant of the Jacobian of h. This term can be further decomposed when h is a composition of several other functions as is typical in a deep neural network.

$$\boldsymbol{X} = \boldsymbol{Z_0} \xrightarrow{h_1} \boldsymbol{Z_1} \xrightarrow{h_2} \boldsymbol{Z_2} \dots \xrightarrow{h_n} \boldsymbol{Z_n} = \boldsymbol{Z} \qquad (5)$$

$$\log P(\boldsymbol{X}) = \log P(\boldsymbol{Z}) - \sum_{i=1}^{n} \log \left| \frac{\mathrm{d}\boldsymbol{Z_i}}{\mathrm{d}\boldsymbol{Z_{i-1}}} \right| \qquad (6)$$

To make the computation of the right hand side of (6) tractable, flow-based models restrict the class of transformations to those for which the Jacobian is a triangular (or even a diagonal) matrix. A simple example is the following additive coupling layer

$$\boldsymbol{y_{p_1}} = \boldsymbol{x_{p_1}} \qquad (7)$$

$$\boldsymbol{y_{p_2}} = \boldsymbol{x_{p_2}} + m(\boldsymbol{x_{p_1}}) \qquad (8)$$

where x, y are the inputs and outputs of the layer respectively; p1, p2 is a partition of the features along the channel dimension and m is an arbitrary transformation. For this layer, it is easy to see that the Jacobian is

$$\begin{bmatrix} \boldsymbol{I_{p_1}} & 0 \\ \frac{\mathrm{d}m(\boldsymbol{x_{p_1}})}{\mathrm{d}\boldsymbol{x_{p_1}}} & \boldsymbol{I_{p_2}} \end{bmatrix} \tag{9}$$

where Ip1 , Ip2 are identity matrices that are of the same size as the partitions p1, p2. Conveniently, the determinant of the matrix in (9) is simply 1 and hence it is ideal for use in a flow-based model. Unlike in we do not require invertible transformations as there is no need for sampling when we are only learning a prior. Nevertheless, in our work we use the layers and formulation of flow-based models proposed .

CHAPTER 4

INTRODUCTION TO IMAGE PROCESSING

4.1 IMAGE:

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Fig 4.1 General image.

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

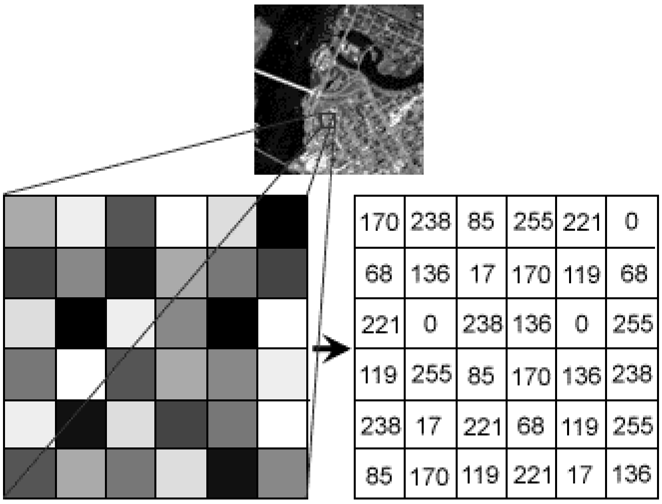


Fig 4.2 Image pixel

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

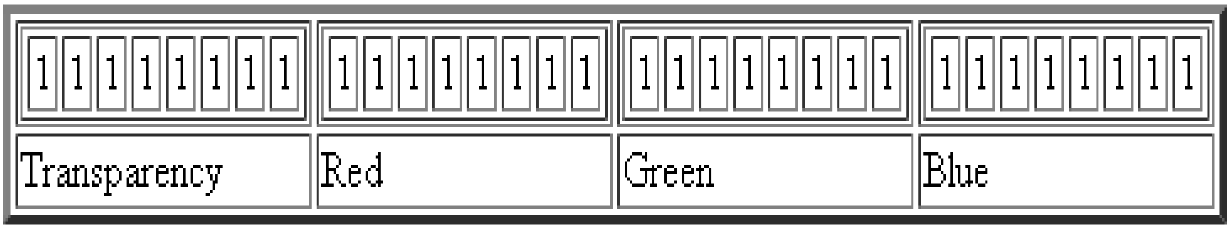


Fig 4.3. Transparency image

4.2 IMAGE FILE SIZES:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

4.3 IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

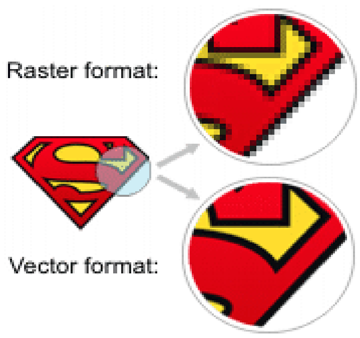


Fig4.4 Resolution image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

4.3.1 RASTER FORMATS:

These formats store images as bitmaps (also known as pix maps).

- **JPEG/JFIF:**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

- **EXIF:**

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image meta data between digital cameras and editing and viewing software. The meta data are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

- **TIFF:**

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

- **PNG:**

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

- **GIF:**

GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

- **BMP:**

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

4.3.2 VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM:**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

- **SVG:**

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

4.4 IMAGE PROCESSING:

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

4.5 FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:

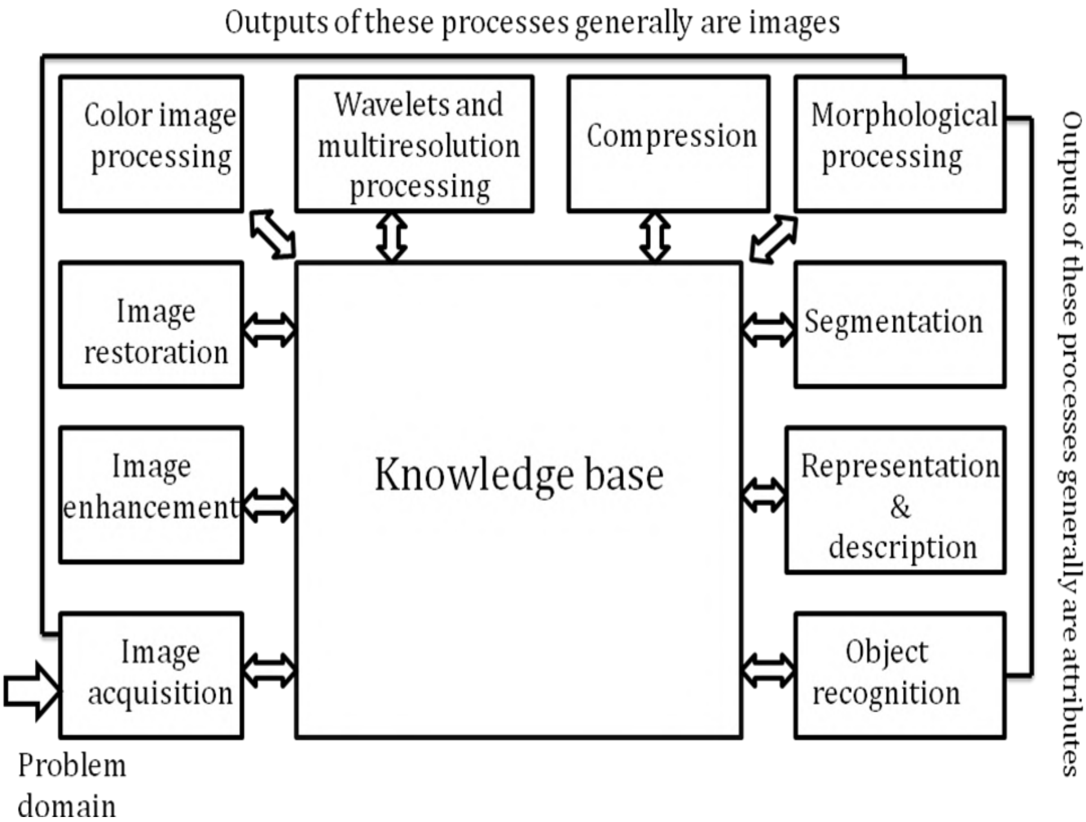


Fig 4.5 Image fundamental

4.5.1 IMAGE ACQUISITION:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Fig 4.5.1 Digital camera image

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Fig 4.5.2 digital camera cell

4.5.2 IMAGE ENHANCEMENT:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because "it looks better." It is important to keep in mind that enhancement is a very subjective area of image processing.

Fig 4.5.3 Image enhancement

4.5.3 IMAGE RESTORATION:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Fig 4.5.4 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, where removal of image blur by applying a deblurring function is considered a restoration technique.

4.5.4 COLOR IMAGE PROCESSING:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Fig 4.5.5 Color & Gray scale image

4.5.5 WAVELETS AND MULTIRESTORATION PROCESSING:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform based image processing since the late1950's, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

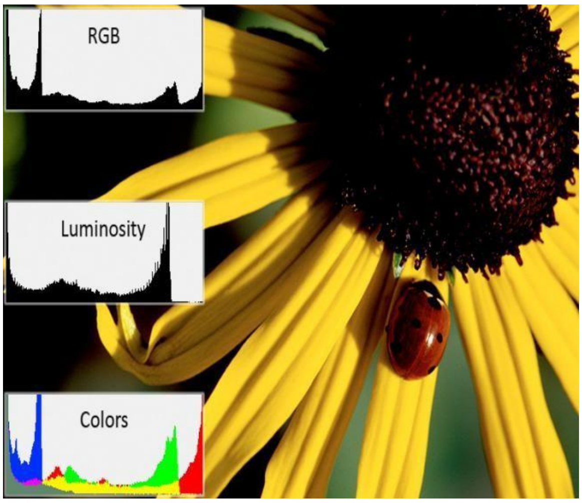


Fig 4.5.6 rgb histogram image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called **Multiresolution** theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

4.5.6 COMPRESSION:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

4.5.7 MORPHOLOGICAL PROCESSING:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Fig 4.5.7 blur to de blur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x, y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

4.5.8 SEGMENTATION:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

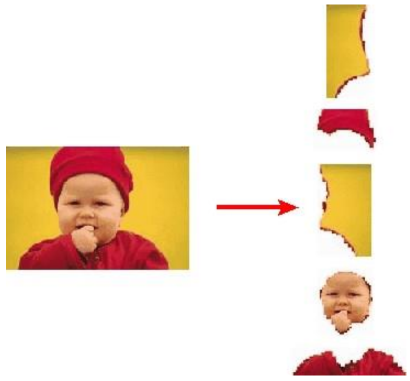


Fig 4.5.8 Image segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

4.5.9 REPRESENTATION AND DESCRIPTION:

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

4.5.10 OBJECT RECOGNITION:

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

4.5.11 KNOWLEDGEBASE:

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an inter related to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

4.6 COMPONENTS OF AN IMAGE PROCESSING SYSTEM:

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

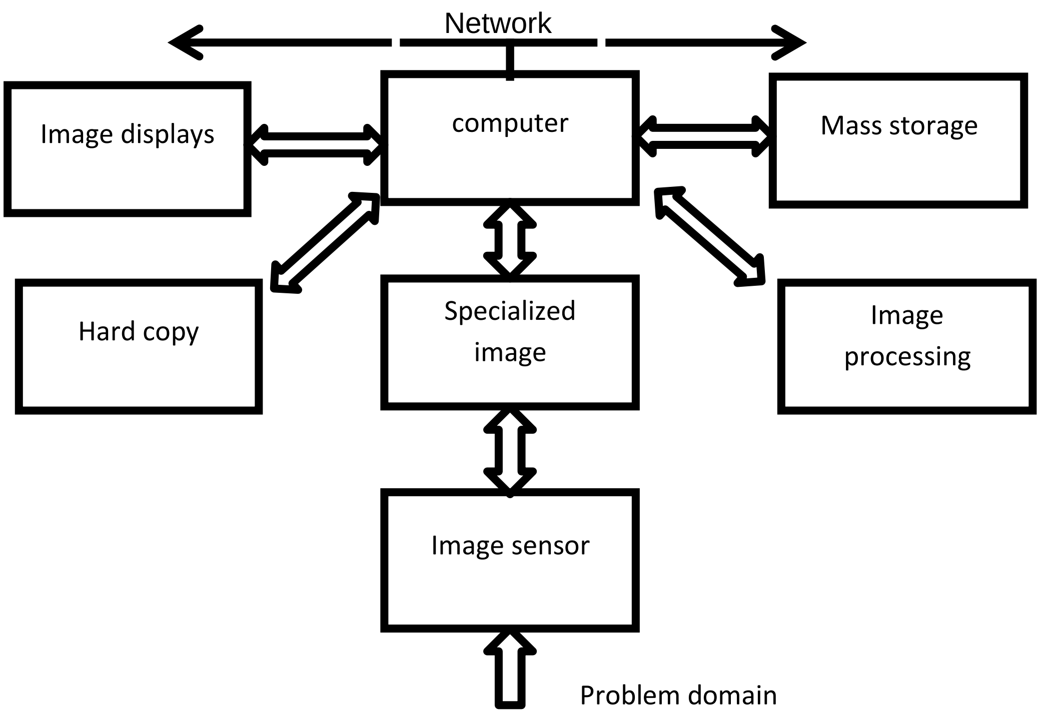


Fig 4.6 Component of image processing

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 1.24 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

- **Image Sensors:**

With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

- **Specialized Image Processing Hardware:**

Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit

(ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

- **Computer:**

  The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

- **Image Processing Software:**

  Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

- **Mass storage:**

  Mass storage capability is a must in image processing applications. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage for image processing applications fall into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), G bytes (meaning giga, or one billion, bytes), and T bytes (meaning tera, or one trillion, bytes).

  One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig. 1.24. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in “jukeboxes” are the usual media for archival applications.

- **Image Displays:**

Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are there requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

- **Hardcopy:**

Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

- **Network:**

Networking is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of optical fiber and other broadband technologies.

Color and texture are two low-level features widely used for image classification, indexing and retrieval. Color is usually represented as a histogram, which is a first order statistical measure that captures global distribution of color in an image One of the main drawbacks of the histogram- based approaches is that the spatial distribution and local variations in color are ignored. Local spatial variation of pixel intensity is commonly used to capture texture information in an image. Grayscale Co-occurrence Matrix (GCM) is a well-known method for texture extraction in the spatial domain. A GCM stores the number of pixel neighborhoods in an image that have a particular grayscale combination. Let I be an image and let p and Np respectively denote any arbitrary pixel and its neighbor in a given direction. If GL denotes the total number of quantized gray levels and gl denotes the individual gray levels, where, gl ∈{0, . . .,GL _ 1}, then each component of GCM can be written as follows:

$$\mathrm{gcm}(i, j) = \mathrm{Pr}((\mathrm{gl}\ \ \mathrm{gl}$$

GCM(i, j) is the number of times the gray level of a pixel p denoted by glp equals i, and the gray level of its neighbor Np denoted by glNp equals j, as a fraction of the total number of pixels in the image. Thus, it estimates the probability that the gray level of an arbitrary pixel in an image is i, and that of its neighbor is j. One GCM matrix is generated for each possible neighborhood direction, namely, 0, 45, 90 and 135.Average and range of 14 features like Angular Second Moment, Contrast, Correlation, etc., are generated by combining all the four matrices to get a total of 28 features. In the GCM approach for texture extraction, color information is completely lost since only pixel gray levels are considered.

To incorporate spatial information along with the color of image pixels, a feature called color correlogram has recently been proposed. It is a three dimensional matrix that represents the probability of finding pixels of any two given colors at a distance ‘d’ apart Auto correlogram is a variation of correlogram, which represents the probability of finding two pixels with the same color at a distance ‘d’ apart. This approach can effectively represent color distribution in an image. However, correlogram features do not capture intensity variation. Many image databases often contain both color as well as gray scale images. The color correlogram method does not constitute a good descriptor in such databases.

Another method called Color Co-occurrence Matrix (CCM) has been proposed to capture color variation in an image. CCM is represented as a three-dimensional matrix, where color pair of the pixels p and Np are captured in the first two dimensions of the matrix and the spatial distance ‘d’ between these two pixels is captured in the third dimension. This approach is a generalization of the color correlogram and reduces to the pure color correlogram for d = 1. CCM is generated using only the Hue plane of the HSV (Hue, Saturation and Intensity Value) color space. The Hue axis is quantized into HL number of levels. If individual hue values are denoted by hl, where $\mathrm{hl} \in \{0, \ldots, \mathrm{HL}-1\}$, then each component of CCM can be written as follows:

$$\mathrm{ccm}(i, j) = \Pr((\mathrm{hl}_{\mathrm{p}}, \mathrm{hl}_{N_{\mathrm{p}}}) = (i, j))$$

Four matrices representing neighbors at angles 0, 90, 180 and 270 are considered. This approach was further extended by separating the diagonal and the non-diagonal components of CCM to generate a Modified Color Co-occurrence Matrix (MCCM). MCCM, thus, may be written as follows: MCCM = (CCMD; CCMND)

Here, CCMD and CCMND correspond to the diagonal and off-diagonal components of CCM. The main drawback of this approach is that, like correlogram, it also captures only color information and intensity information is completely ignored.

An alternative approach is to capture intensity variation as a texture feature from an image and combine it with color features like histograms using suitable weights . One of the challenges of this approach is to determine suitable weights since these are highly application-dependent. In certain applications like Content-based Image Retrieval (CBIR), weights are often estimated from relevance feedback given by users.

While relevance feedback is sometimes effective, it makes the process of image retrieval user-dependent and iterative. There is also no guarantee on the convergence of the weight-learning algorithms. In order to overcome these problems, researchers have tried to combine color and texture features together during extraction.

Proposed two approaches for capturing color and intensity variations from an image using the LUV color space. In the Single-channel Co-occurrence Matrix (SCM), variations for each color channel, namely, L, U and V are considered independently. In the Multi- channel Co-occurrence Matrix (MCM), variations are captured taking two channels at a time – UV, LU and LV. Since the LUV color space separates out chrominance (L and U) from luminance (V), SCM in effect, generates one GCM and two CCMs from each image independently. As a result, correlation between the color channels is lost

However, in MCM, the count of pair wise occurrences of the values of different channels of the color space is captured. Thus, each component of MCM can be written as follows:

mcmUV(i; j) = Pr((up; vNp) = (i; j))

mcmLU(i; j) = Pr((lp; uNp) = (i; j))

mcmLV(i; j) = Pr((lp; vNp) = (i; j))

Here, mcm UV(i, j) is the number of times the U chromaticity value of a pixel p denoted by up equals i, and the V chromaticity value of its neighbor Np denoted by vNp equals j, as a fraction of the total number of pixels in the image. Similarly, mcmLU(i, j) and mcmLV(i, j) are defined. One MCM matrix is generated for each of the four neighborhood directions, namely, 0, 45, 90 and 135.

Deng and Manjunath (2001) proposed a two-stage method called JSEG, which combines color and texture after image segmentation. In the first stage, colors are quantized to the required levels for differentiating between various regions of an image. Pixel values of the regions are then replaced by their quantized color levels to form a color map. Spatial variation of color levels between different regions in the map is viewed as a type of texture composition of the image.

Yu et al. (2002) suggested the use of color texture moments to represent both color and texture of an image. This approach is based on the calculation of Local Fourier Transformation (LFT) coefficients. Eight templates equivalent to LFT are operated over an image to generate a characteristic map of the image. Each template is a 3 · 3 filter that considers eight neighbors of the current pixel for LFT calculation. First and second order moments of the characteristic map are then used to generate a set of features.

In this paper, we propose an integrated approach for capturing spatial variation of both color and intensity levels in the neighborhood of each pixel using the HSV color space. In contrast to the other methods, for each pixel and its neighbor, the amount of color and intensity variation between them is estimated using a weight function. Suitable constraints are satisfied while choosing the weight function for effectively relating visual perception of color and the HSV color space properties. The color and intensity variations are represented in a single composite feature known as Integrated Color and Intensity Co-occurrence Matrix (ICICM). While the existing schemes generally treat color and intensity separately, the proposed method provides a composite view to both color and intensity variations in the same feature. The main advantage of using ICICM is that it avoids the use of weights to combine individual color and texture features. We use ICICM feature in an image retrieval application from large image databases.

Early result on this work was reported in (Vadivel et al., 2004a). In the next section, we describe the proposed feature extraction technique after introducing some of the properties of the HSV color space. Choice of quantization levels for color and intensity axes, selection of parameter values and a brief overview of the image retrieval application.

Integrated color and intensity co-occurrence matrix:

We propose to capture color and intensity variation around each pixel in a two-dimensional matrix called Integrated Color and Intensity Co-occurrence Matrix (ICICM). This is a generalization of the Grayscale Co-occurrence Matrix and the Color Co-occurrence Matrix techniques. For each pair of neighboring pixels, we consider their contribution to both color perception as well as gray level perception to the human eye. Some of the useful properties of the HSV color space and their relationship to human color perception are utilized for extracting this feature. In the next sub-section, we briefly explain relevant properties of the HSV color space. In the subsequent subsection, we describe how the properties can be effectively used for generating ICICM.

HSV color space:

HSV Color space: Basically there are three properties or three dimensions of color that being hue, saturation and value HSV means Hue, Saturation and Value. It is important to look at because it describes the color based on three properties. It can create the full spectrum of colors by editing the HSV values. The first dimension is the Hue. Hue is the other name for the color or the complicated variation in the color. The quality of color as determined by its dominant wavelength. This Hue is broadly classified into three categories. They are primary Hue, Secondary Hue and Ternary Hue. The first and the foremost is the primary Hue it consists of three colors they are red, yellow and blue. The secondary Hue is formed by the combination of the equal amount of colors of the primary Hue and the colors of the secondary Hue which was formed by the primary Hue are Orange, Green and violet. The remaining one is the ternary Hue is formed by the combination of the primary Hue and the secondary Hue. The limitless number of colors are produced by mixing the colors of the primary Hue in different amounts. Saturation is the degree or the purity of color. Then the second dimension is the saturation. Saturation just gives the intensity to the colors. The saturation and intensity drops just by mixing the colors or by adding black to the color. By adding the white to the color in spite of more intense the color becomes lighter. Then finally the third dimension is the Value. The value is the brightness of the color. When the value is zero the color space is totally black with the increase in the color there is also increase in the brightness and shows the various colors. The value describes the contrast of the color. That means it describes the lightness and darkness of the color. As similar to the saturation this value consists of the tints and shades. Tints are the colors with the added white and shades are the colors with the added black.

Properties of the HSV color space:

Sensing of light from an image in the layers of human retina is a complex process with rod cells contributing to scotopic or dim-light vision and cone cells to photopic or bright-light vision (Gonzalez and Woods, 2002). At low levels of illumination, only the rod cells are excited so that only gray shades are perceived. As the illumination level increases, more and more cone cells are excited, resulting in increased color perception. Various color spaces have been introduced to represent and specify colors in a way suitable for storage, processing or transmission of color information in images. Out of these, HSV is one of the models that separate out the luminance component (Intensity) of a pixel color from its chrominance components (Hue and Saturation). Hue represents pure color, which is perceived when incident light is of sufficient illumination and contains a single wavelength. Saturation gives a measure of the degree by which a pure color is diluted by white light. For light with low illumination, corresponding intensity value in the HSV color space is also low.

The HSV color space can be represented as a Hexa cone, with the central vertical axis denoting the luminance component, I (often denoted by V for Intensity Value). Hue, is a chrominance component defined as an angle in the range [0,2p] relative to the red axis with red at angle 0, green at 2p/3, blue at 4p/3 and red again at 2p. Saturation, S, is the other chrominance component, measured as a radial distance from the central axis of the hexacone with value between 0 at the center to 1 at the outer surface. For zero saturation, as the intensity is increased, we move from black to white through various shades of gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the color represented by its hue. When saturation is near 0, all the pixels in an image look alike even though their hue values are different.

As we increase saturation towards 1, the colors get separated out and are visually perceived as the true colors represented by their hues. Low saturation implies presence of a large number of spectral components in the incident light, causing loss of color information even though the illumination level is sufficiently high. Thus, for low values of saturation or intensity, we can approximate a pixel color by a gray level while for higher saturation and intensity, the pixel color can be approximated by its hue. For low intensities, even for a high saturation, a pixel color is close to its gray value. Similarly, for low saturation even for a high value of intensity, a pixel is perceived as gray. We use these properties to estimate the degree by which a pixel contributes to color perception and gray level perception.

One possible way of capturing color perception of a pixel is to choose suitable thresholds on the intensity and saturation. If the saturation and the intensity are above their respective thresholds, we may consider the pixel to have color dominance; else, it has gray level dominance. However, such a hard thresholding does not properly capture color perception near the threshold values. This is due to the fact that there is no fixed level of illumination above which the cone cells get excited. Instead, there is a gradual transition from scotopic to photopic vision. Similarly, there is no fixed threshold for the saturation of cone cells that leads to loss of chromatic information at higher levels of illumination caused by color dilution. We, therefore, use suitable weights that vary smoothly with saturation and intensity to represent both color and gray scale perception for each pixel.

NON INTERVAL QUANTIZATION:

Due to the large range for each component by directly calculating the characteristics for the retrieval then the computation will be very difficult to ensure rapid retrieval. It is essential to quantify HSV space component to reduce computation and improve efficiency. At the same time, because the human eye to distinguish colors is limited, do not need to calculate all segments. Unequal interval quantization according the human color perception has been applied on H , S ,V components.

Based on the color model of substantial analysis, we divide color into eight parts. Saturation and intensity is divided into three parts separately in accordance with the human eyes to distinguish. In accordance with the different colors and subjective color perception quantification, quantified hue (H), saturation (S) and value (V).

In accordance with the quantization level above, the H, S, V three-dimensional feature vector for different values of with different weights to form one dimensional feature vector and is given by the following equation:

$$G = Qs*Qv*H+Qv*s+V$$

Where Qs is the quantized series of S and Qv is the quantized series of V. And now by setting Qs = Qv = 3, Then G = 9H+3S+V

$$H = \begin{cases} 0 & \text{if } h \in [316,20] \\ 1 & \text{if } h \in [21,40] \\ 2 & \text{if } h \in [41,75] \\ 3 & \text{if } h \in [76,155] \\ 4 & \text{if } h \in [156,190] \\ 5 & \text{if } h \in [191,270] \\ 6 & \text{if } h \in [271,295] \\ 7 & \text{if } h \in [296,315] \end{cases} \qquad S = \begin{cases} 0 & \text{if } s \in [0,0.2) \\ 1 & \text{if } s \in [0.2,0.7) \\ 2 & \text{if } s \in [0.7,1) \end{cases} \qquad V = \begin{cases} 0 & \text{if } v \in [0,0.2) \\ 1 & \text{if } v \in [0.2,0.7) \\ 2 & \text{if } v \in [0.7,1) \end{cases}$$

In this way three component vector of the HSV from one dimensional vector, Which quantize the whole color space for the 72 kinds of the main colors. So we can handle 72 bins of one dimensional histogram. This qualification is effective in reducing the images by the effect of the light intensity, but also reducing the computational time and complexity.

IMAGE RETRIEVAL:

Image retrieval is nothing but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords or the descriptions to the images so that the retrieval can be performed. Manual image annotation is time consuming, expensive and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc..,

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others. one can today generate volumes of images with content as diverse as family get-togethers and national park visits. Low-cost storage and easy Web hosting has fueled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchable image data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real-world image search system designers.

As far as technological advances are concerned, growth in Content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real world implications, applications, and constraints of the technology. Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future.

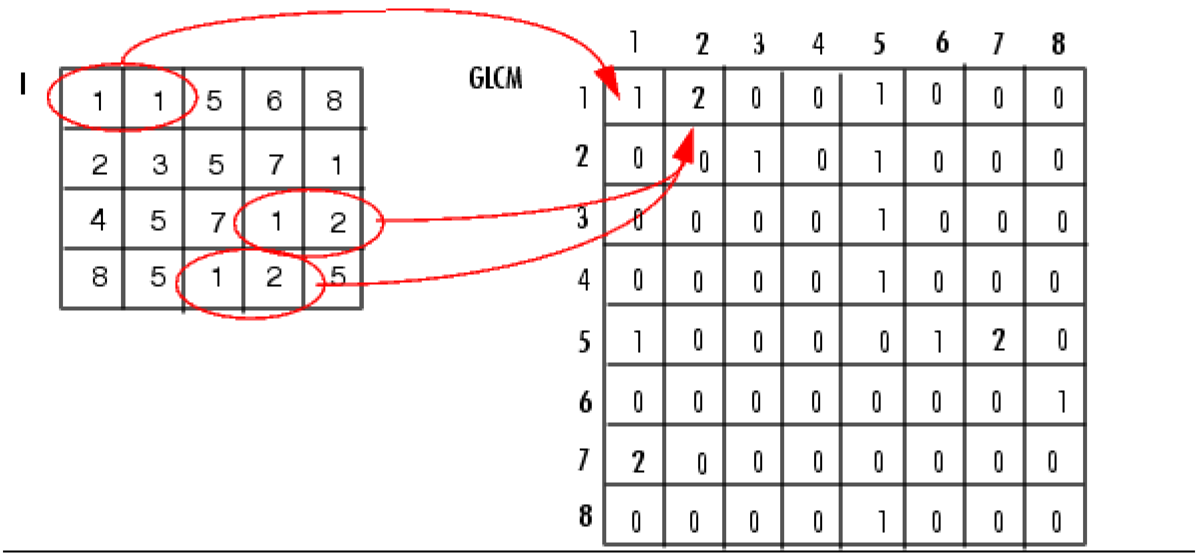
An image retrieval system designed to serve a personal collection should focus on features such as personalization, flexibility of browsing, and display methodology. For example, Google's Picasa system [Picasa 2004] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods. The notable few applications include an FPGA implementation of a color histogram-based image retrieval system [Kotoulas and Andreadis 2003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 2003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 2002].

Discussion. Regardless of the nature of the collection, as the expected user-base grows, factors such as concurrent query support, efficient caching, and parallel and distributed processing of requests become critical. For future real-world image retrieval systems, both software and hardware approaches to address these issues are essential. More realistically, dedicated specialized servers, optimized memory and storage support, and highly parallelizable image search algorithms to exploit cluster computing powers are where the future of large-scale image search hardware support lies.

OVERVIEW OF TEXTURE:

We all know about the term Texture but for defining it is a hard time. One can differentiate the two different Textures by recognizing the similarities and differences. Commonly there are three ways for the usage of the Textures.

Based on the Textures the images can be segmented. To differentiate between already segmented regions or to classify them. We can reproduce Textures by producing the descriptions. The texture can be analyzed in three different ways. They are Spectral, Structural and Statistical:



4.7 Overview of Texture

CHAPTER-5

DIGITAL IMAGE PROCESSING

Digital image processing

Background:

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An important characteristic underlying the design of image processing systems is the significant level of testing & experimentation that normally is required before arriving at an acceptable solution. This characteristic implies that the ability to formulate approaches &quickly prototype candidate solutions generally plays a major role in reducing the cost & time required to arrive at a viable system implementation.

What is DIP

An image may be defined as a two-dimensional function f(x, y), where x & y are spatial coordinates, & the amplitude off at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital image by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensor, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis& computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting & somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low-level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low- level process is characterized by the fact that both its inputs & outputs are images.

Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images. Finally higher- level processing involves "Making sense" of an ensemble of recognized objects, as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

What is an image?

An image is represented as a two dimensional function f(x, y) where x and y are spatial co-ordinates and the amplitude of 'f' at any pair of coordinates (x, y) is called the intensity of the image at that point.

Gray scale image:

A grayscale image is a function I (xylem) of the two spatial coordinates of the image plane I(x, y) is the intensity of the image at the point (x, y) on the image plane. I (xylem) takes non-negative values assume the image is bounded by a rectangle [0, a] ×[0, b]I: [0, a] × [0, b] → [0, info)

Color image:

It can be represented by three functions, R (xylem) for red, G (xylem) for green *and* B (xylem) for blue.

An image may be continuous with respect to the x and y coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate’s values is called sampling. Digitizing the amplitude values is called quantization.

Coordinate convention:

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image f(x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size M X N. The values of the coordinates (xylem) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates. In many image processing books, the image origin is defined to be at (xylem)=(0,0).The next coordinate values along the first row of the image are (xylem)=(0,1).It is important to keep in mind that the notation (0,1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Following figure shows the coordinate convention. Note that x ranges from 0 to M-1 and y from 0 to N-1 in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (xylem) the toolbox uses the notation (race) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate topples, (alb), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentation refers to the coordinates. Less frequently the toolbox also employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refers to rows. This is the opposite of our use of variables x and y.

Image as Matrices:

The preceding discussion leads to the following representation for a digitized image function:

f (0,0) f(0,1)… ....f(0,N-1)

f(1,0) f(1,1) ....f(1,N-1)

f(xylem) = f (M-1,0) f(M-1,1)… .... f(M-1,N-1)

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements.A digital image can be represented naturally as a MATLAB matrix:

f(1,1) f(1,2) ……. f(1,N)

f(2,1) f(2,2) ....f(2,N)

. . .

f = f (M,1) f(M,2).......... f(M,N)

Where f (1,1) = f(0,0) (note the use of a monoscope font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation f(p ,q) denotes the element located in row p and the column q. For example f(6,2) is the element in the sixth row and second column of the matrix f. Typically we use the letters M and N respectively to denote the number of rows and columns in a matrix. A 1xN matrix is called a row vector whereas an Mx1 matrix is called a column vector. A 1x1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using mono-scope characters. We use conventional Roman, italic notation such as f(x ,y), for mathematical expressions.

Reading Images:

Images are read into the MATLAB environment using function imread whose syntax is

Imread (‘filename’)

Table 5.1 Format Names

Format Name	**Description**	**Recognized Extension**
TIFF	Tagged Image File Format	.tif, .tiff

JPEG	Joint Photograph Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	.xwd

Here filename is a spring containing the complete of the image file(including any applicable extension).For example the command line >> f = I m read ('8. jpg');

Reads the JPEG (above table) image chest Xray into image array f. Note the use of single quotes (') to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output If a semicolon is not included. MATLAB displays the results of the operation(s) specified in that line. The prompt symbol (>>) designates the beginning of a command line, as it appears in the MATLAB command window.

Data Classes:

Although we work with integers coordinates the values of pixels themselves are not restricted to be integers in MATLAB. Table above list various data classes supported by MATLAB and IPT are representing pixels values. The first eight entries in the table are refers to as numeric data classes. The ninth entry is the char class and, as shown, the last entry is referred to as logical data class.

All numeric computations in MATLAB are done in double quantities, so this is also a frequent data class encounter in image processing applications. Class unit 8 also is encountered frequently, especially when reading data from storages devices, as 8 bit images are most common representations found in practice. These two data classes, classes logical, and, to a lesser degree, class unit 16 constitute the primary data classes on which we focus. Many IPT functions however support all the data classes listed in table. Data class double requires 8 bytes to represent a number uint8 and int 8 require one byte each, uint16 and int16 requires 2bytes and unit 32.

Table 5.2 Data Classes

Name	Description
Double	double precision, floating-point numbers the Approximate
Unit 8	unsigned 8_bit integers in the range[0,255] (1 byte per element)
Unit 16	unsigned 16_bit integers in the range[0,65535] (2 bytes per element)
Unit 32	unsigned 32_bit integers in the range[0,4294967295] (4 byte per element)
Int 8	signed 8_bit integers in the range[-128,127] (1 byte per element)
Int 16	unsigned 16_bit integers in the range[32768,32767] (2 byte per element)
Int 32	unsigned 32_bit integers in the range[-2147483648,21474833647] (one byte per element)
Single	single precision floating-point numbers with values in the approximate range (4 byte per element)
Char	Character(2 bytes per elements)
Logical	Values are 0 to 1(1 bytes per element)

Int 32 and single required 4 bytes each. The char data class holds characters in Unicode representation. A character string is merely a 1*n array of characters logical array contains only the values 0 to 1,with each element being stored in memory using function logical or by using relational operators.

Image Types:

The toolbox supports four types of images:

1 .Intensity images

2. Binary images

3. Indexed images

4. R G B images.

Most monochrome image processing operations are carried out using binary or intensity images, so our initial focus is on these two image types. Indexed and RGB colour images.

Intensity Images:

An intensity image is a data matrix whose values have been scaled to represent intentions. When the elements of an intensity image are of class unit8, or class unit 16, they have integer values in the range [0,255] and [0, 65535], respectively. If the image is of class double, the values are floating point numbers. Values of scaled, double intensity images are in the range [0, 1] by convention.

Binary Images:

Binary images have a very specific meaning in MATLAB.A binary image is a logical array 0s and1s.Thus, an array of 0s and 1s whose values are of data class, say unit8, is not considered as a binary image in MATLAB .A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, we create an array B using the statement.

B=logical (A)

If A contains elements other than 0s and 1s.Use of the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.Using relational and logical operators also creates logical arrays. To test if an array is logical we use the I logical function is logical(c). If c is a logical array, this function returns a 1.Otherwise returns a 0. Logical array can be converted to numeric arrays using the data class conversion functions.

Indexed Images:

An indexed image has two components:

A data matrix integer, x

A color map matrix, map

Matrix map is an m*3 arrays of class double containing floating point values in the range [0, 1].The length m of the map are equal to the number of colors it defines. Each row of map specifies the red, green and blue components of a single color. An indexed images uses "direct mapping" of pixel intensity values color map values. The color of each pixel is determined by using the corresponding value the integer matrix x as a pointer in to map. If x is of class double ,then all of its components with values less than or equal to 1 point to the first row in map, all components with value 2 point to the second row and so on. If x is of class units or unit 16, then all components value 0 point to the first row in map, all components with value 1 point to the second and so on.

RGB Image:

An RGB color image is an M*N*3 array of color pixels where each color pixel is triplet corresponding to the red, green and blue components of an RGB image, at a specific spatial location. An RGB image may be viewed as “stack” of three gray scale images that when fed in to the red, green and blue inputs of a color monitor

Produce a color image on the screen. Convention the three images forming an RGB color image are referred to as the red, green and blue components images. The data class of the components images determines their range of values. If an RGB image is of class double the range of values is [0, 1].

Similarly the range of values is [0,255] or [0, 65535].For RGB images of class units or unit 16 respectively. The number of bits use to represents the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8bit image, the corresponding RGB image is said to be 24 bits deep.

Generally, the number of bits in all component images is the same. In this case the number of possible color in an RGB image is (2^b) ^3, where b is a number of bits in each component image. For the 8bit case the number is 16,777,216 colors**.**

CHAPTER 6

SOFTWARE INTRODUCTION

6.1. INTRODUCTION TO MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M

– files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

6.2 The MATLAB system:

The MATLAB system consists of five main parts

- **Development Environment**:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library**:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language**:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

- **Graphics:**

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files. Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

6.3 GRAPHICAL USER INTERFACE (GUI):

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files. A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed. To launch GUIDE from the MATLAB command window, type guide filename**.** Where filename is the name of an existing FIG-file on the current path. If filename is omitted,

GUIDE opens a new (i.e., blank) window.

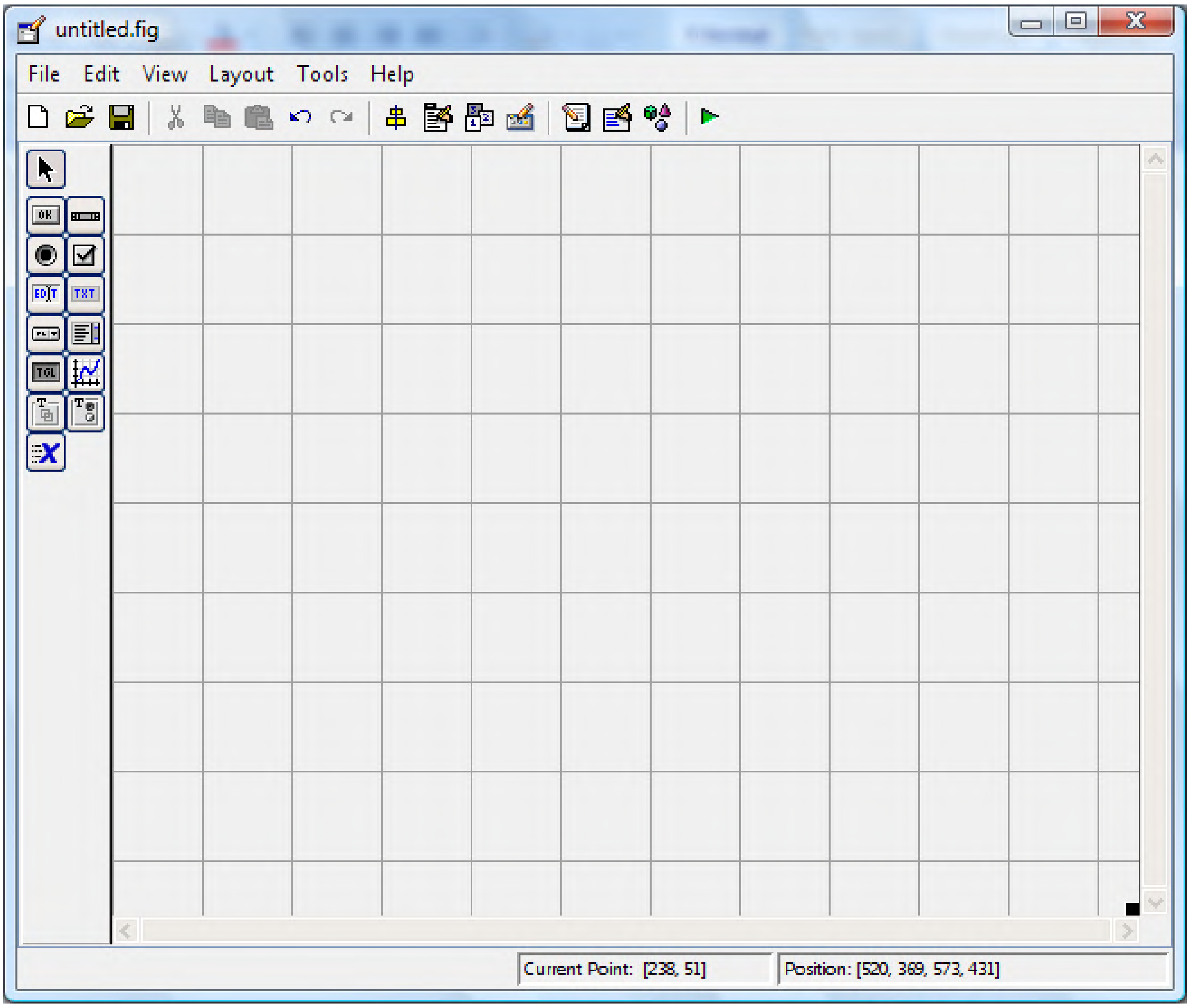


Figure 6.3 GUI

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

6.4 Getting Started:

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator,

and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos. At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

6.4.1 Introduction - describes the components of the MATLAB system.

6.4.2 Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

6.4.3 Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

6.4.4 Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

5.4.5 Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

6.5 DEVELOPMENT ENVIRONMENT

6.5.1 INTRODUCTION

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

6.5.2 Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop. On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop. You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

6.5.3 QUITTING MATLAB:

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

6.5.4 MATLAB DESKTOP:

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries. You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts. You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

6.5.5 DESKTOP TOOLS:

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser
-

Command Window:

Use the Command Window to enter variables and run functions and M-files.

Command History:

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs:

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad:

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents. To open the Help browser, click the help button in the toolbar, or type help browser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator:

Use to Help Navigator to find information. It includes:

Product filter -Set the filter to show documentation only for the products you specify.

Contents tab -View the titles and tables of contents of documentation for your products.

Index tab -Find specific index entries (selected keywords) in the Math Works documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

Display Pane:

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser:

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path:

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser:

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces. To view the workspace and information about each variable, use the Workspace browser, or use the functions who and who's. To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function. The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor:

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger:

Use the Editor/Debugger to create and debug M-files, which are programs you write to run MATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint. If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

6.6 MANIPULATING MATRICES

6.6.1 ENTERING MATRICES

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example. You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

  Start by entering Durer's matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon; to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

  To enter Durer's matrix, simply type in the Command Window

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

MATLAB displays the matrix you just entered.

```
A =

   16    3    2   13

    5   10   11    8

    9    6    7   12

    4   15   14    1
```

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

5.6.2 EXPRESSIONS

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

Variables:

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example, num_students = 25 Creates a 1-by-1 matrix named num students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers:

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

6.6.3 OPERATORS

Expressions use familiar arithmetic operators and precedence rules.

Table 6.6.3 Operators

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

6.6.4 FUNCTIONS

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun,   For a list of more advanced mathematical and matrix functions, type help specfun help elmat. Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Table 6.6.4 Functions

Pi	3.14159265...
I	Imaginary unit, √-1
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

6.7 GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

6.7.1 CREATING GUIs WITH GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

6.7.2 GUI DEVELOPMENT ENVIRONMENT

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

The Implementation of a GUI:

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file -contains a complete description of the GUI figure and all of its children (UI controls and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the call backs, which are defined as sub functions. This M-file is referred to as the application M-file in this documentation. Note that the application M-file does not contain the code that lays out the UI controls; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

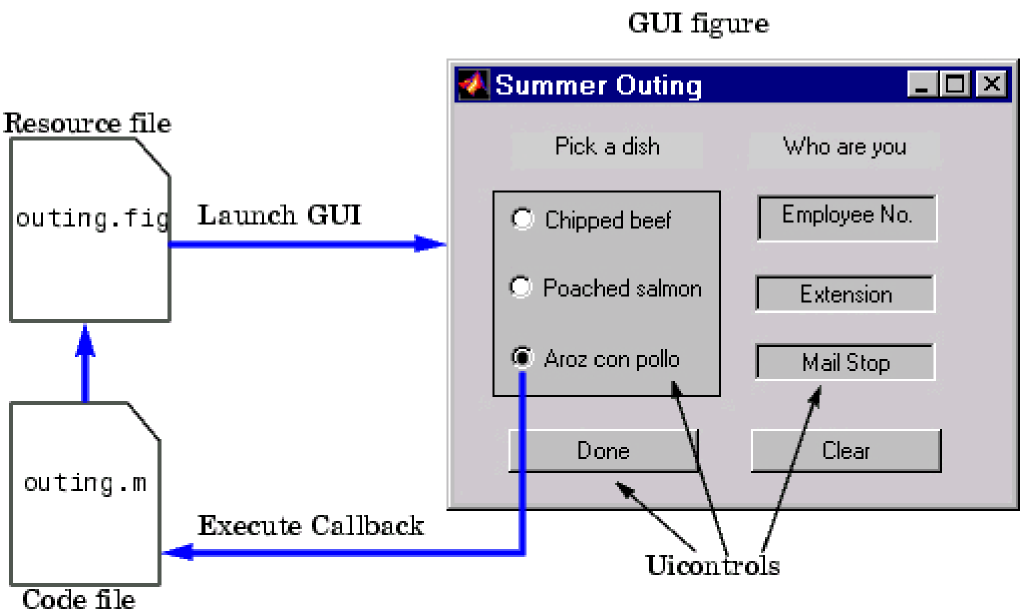


Figure 6.7.2 graphical user blocks

6.7.3 FEATURES OF THE GUIDE-GENERATED APPLICATION M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information). The automatically inserted sub function prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no control creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

6.7.4 BEGINNING THE IMPLEMENTATION PROCESS

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques

used in the application M-file.

Application Examples - a collection of examples that illustrate techniques

which are useful for implementing GUIs.

Command-Line Accessibility:

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as find object, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

6.7.5 USER INTERFACE CONTROL

The Layout Editor Component palette contains the user interface controls that you can use in your GUI. These components are MATLAB UI control objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons

- List boxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons:

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set:

String - set this property to the character string you want displayed on the push button.

Tag-GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback:

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons:

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback:

The call back routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File:

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)

button_state = get(h,'Value');

if button_state == get(h,'Max')

    % toggle button is pressed

elseif button_state == get(h,'Min')

    % toggle button is not pressed

end
```

Adding an Image to a Push Button or Toggle Button:

Assign the C Data property an m-by-n-by-3 array of RGB values that define a true color image. For example, the array defines 16-by-128 true color image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);

a(:,:,2) = rand(16,128);

a(:,:,3) = rand(16,128);

set(h,'C Data',a)
```

Radio Buttons:

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive with in a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior:

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following sub function, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)

set(off,'Value',0)
```

Obtaining the Radio Button Handles:

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)

off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];

mutual_exclude(off)

% Continue with callback.
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes:

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes). The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)
if (get(h,'Value') == get(h,'Max'))
   % then checkbox is checked-take approriate action
else
   % checkbox is not checked-take approriate action
end
```

Edit Text:

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user. To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)
user_string = get(h,'string');
% proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component:

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (is nan) and displays an error dialog (error dlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)

user_entry = str2double(get(h,'string'));

if isnan(user_entry)

    errordlg('You must enter a numeric value','Bad Input','modal')

end

% proceed with callback...
```

Triggering Callback Execution:

On UNIX systems, clicking on the menu bar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menu bar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text:

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it.

Frames:

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only UI controls can appear within frames (axes cannot).

Placing Components on Top of Frames:

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes:

List boxes display a list of items and enable users to select one or more items. The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1. The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty. The List box Top property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. List box Top is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Non integer values are fixed to the next lowest integer

Single or Multiple Selection:

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type:

List boxes differentiate between single and double clicks on an item and set the figure Selection Type property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution:

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a done button (push button) and program its callback routine to query the list box Value property (and possibly the figure Selection Type property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback sub function from the application M-file. Leave the callback sub function stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus:

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu:

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item. This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

switch val

case 1

% The user selected the first item

case 2

% The user selected the second item

% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

string_list = get(h,'String');

selected_string = string_list{val}; % convert from cell array to string

% etc.
```

Enabling or Disabling Controls:

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

On-The control is operational

Off-The control is disabled and its label (set by the string property) is grayed out. Inactive-The control is disabled, but its label is not grayed out. When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure Window Button Down FCN callback executes. Then the control's Button Down FCN callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes:

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks:

Axes are not UI control objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes Button Down FCN property to define the callback.

6.7.6 PLOTTING TO AXES IN GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes:

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example, axes (handles.axes1) makes the axes whose Tag property axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axes. See GUI with Multiple Axes for example that uses two axes.

Figure:

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control.

CHAPTER 7

RESULTS

Figure 7.1 Input Color Image

We evaluate our method on the BSD68 dataset for different noise levels and compare it with BM3D, Noise2Void, deep image prior.

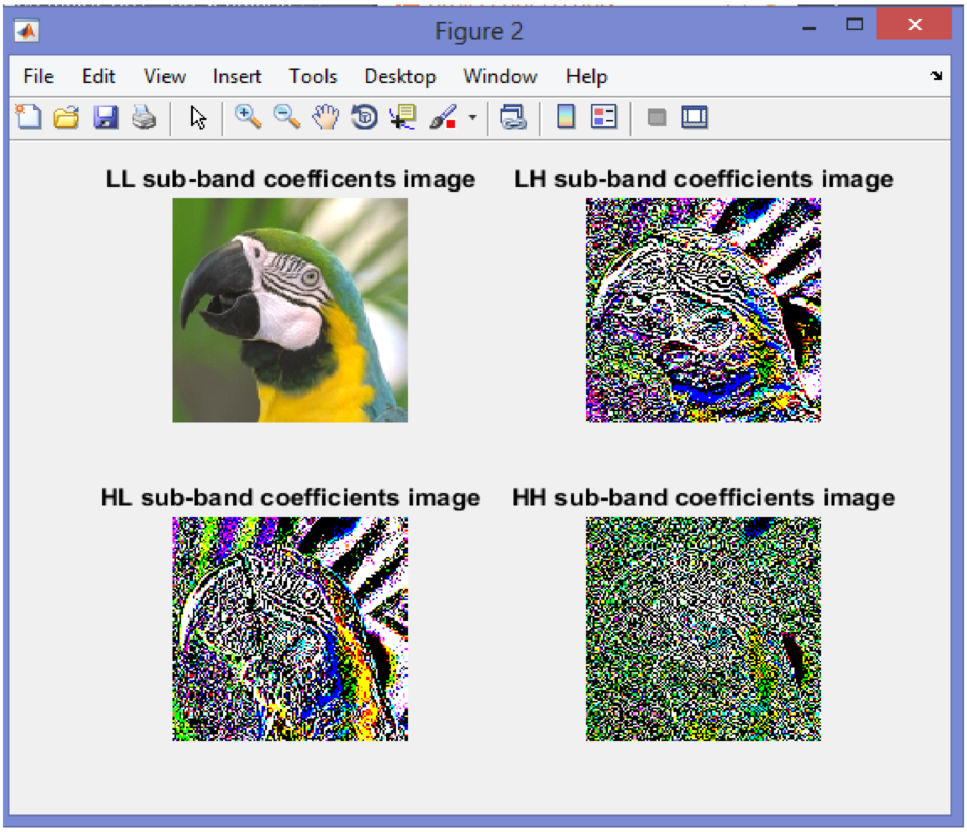


Figure 7.2 Image Coefficients

All comparisons are made using either results reported in the respective papers or those obtained from running the code that the authors have generously shared. average PSNR values of different methods for images from BSD68.

Figure 7.3 Reconstructed Image

Although PSNR is not an accurate metric for perceptual quality, our method performs competitively with Noise2Void and is better than Deep image prior.

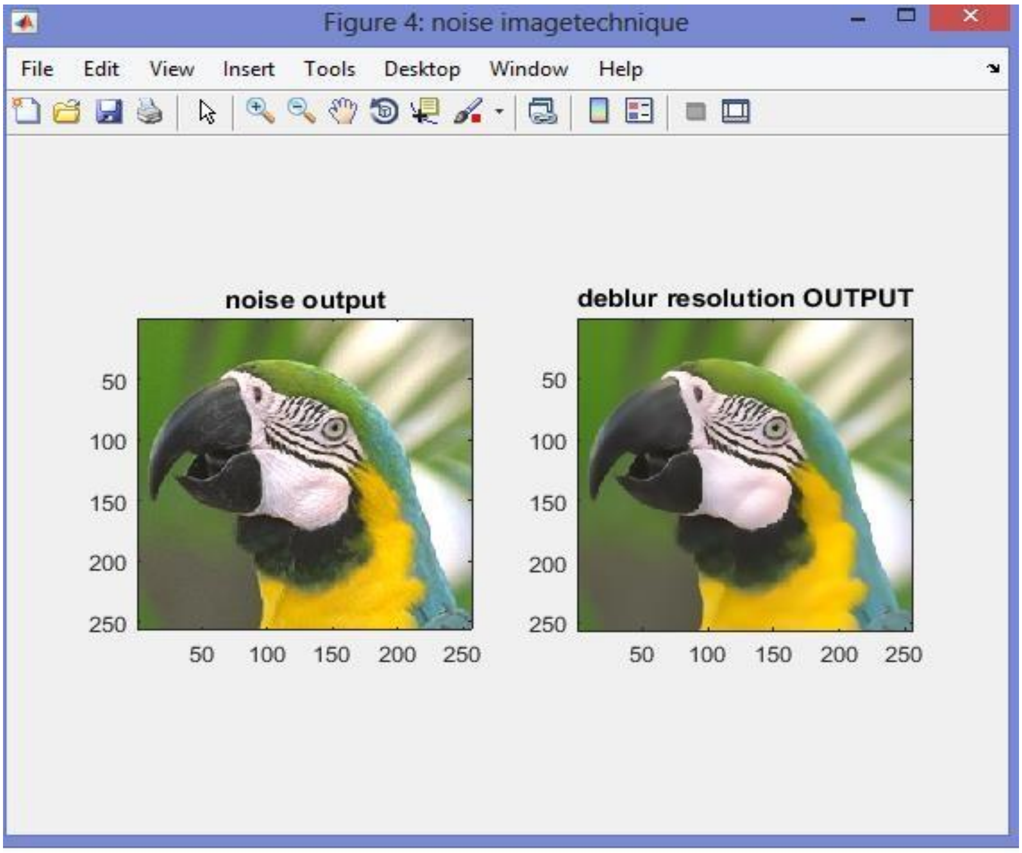


Figure 7.4 Noise output and DeBlur Resolution Output Image

Our method is able to remove noise effectively without blurring any textures, details or sharp edges (this is obvious in the sky in the first set of images). Deep image prior produces outputs that still have visible noise.

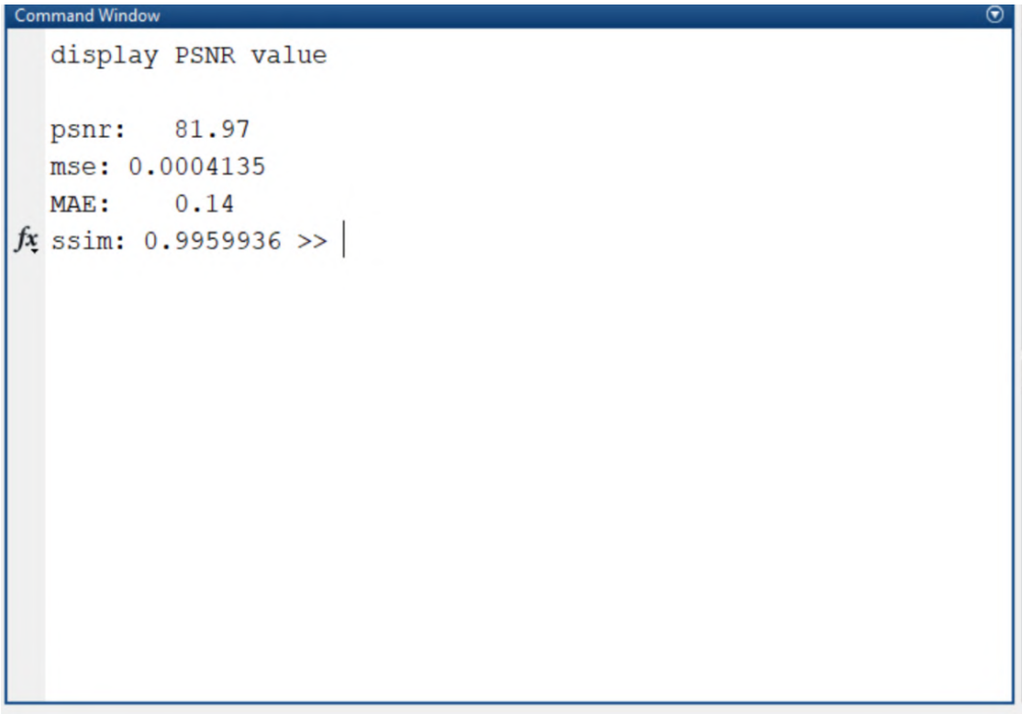


Figure 7.5 PSNR Values.

In the Command Window we will be having the niose values and PSNR values of our output image.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

CONCLUSION

We have proposed the use of flow-based model as a mathematically justifiable and realistic prior for image denoising. We have conducted qualitative and quantitative experiments on the BSD68 dataset that reveals the competitive performance of our method. Motivated by our success, we conjecture that using a flow based model prior should be effective for solving other image restoration tasks such as image deblurring and super resolution in an unsupervised fashion.

FUTURE SCOPE

With the explosion in the number of digital images taken every day, the demand for more accurate and visually pleasing images is increasing. However, the images captured by modern cameras are inevitably degraded by noise, which leads to deteriorated visual image quality. Therefore, work is required to reduce noise without losing image features (edges, corners, and other sharp structures.

PUBLICATION

Submitted Paper in the Conference ICSMEC-21 with Paper ID (ICSMEC21-0026) and got Acceptance for the Paper.

REFERENCES

[1] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," IEEE Transactions on Image Processing, vol. 16, no. 8, pp. 2080–2095, Aug 2007.

[2] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li, "Nonlocally centralized sparse representation for image restoration," IEEE transactions on Image Processing, vol. 22, no. 4, pp. 1620–1630, 2012.

[3] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng, "Weighted nuclear norm minimization with application to image denoising," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 2862–2869.

[4] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang, "Learning deep cnn denoiser prior for imagerestoration," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3929–3938.

[5] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3142–3155, July 2017.

[6] Stamatios Lefkimmiatis, "Non-local color image denoising with convolutional neural networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3587–3596.

[7] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila, "Noise2Noise: Learning image restoration without clean data," in Proceedings of the 35th International Conference on Machine Learning, Jennifer Dy and Andreas Krause, Eds., Stockholmsmassan, Stock- ¨ holm Sweden, 10–15 Jul 2018, vol. 80 of Proceedings of Machine Learning Research, pp. 2965–2974, PMLR.

[8] A. Krull, T. Buchholz, and F. Jug, "Noise2void - learning denoising from single noisy images," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019, pp. 2124–2132.

[9] Stefan Roth and Michael J Black, "Fields of experts," International Journal of Computer Vision, vol. 82, no. 2, pp. 205, 2009.

[10] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, “Deep image prior,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9446–9454.

[11] J. Chen, J. Chen, H. Chao, and M. Yang, “Image blind denoising with generative adversarial network based noise modeling,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2018, pp. 3155–3164.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative adversarial nets,” in Advances in neural information processing systems, 2014, pp. 2672–2680.

[13] Laurent Dinh, David Krueger, and Yoshua Bengio, “NICE: non-linear independent components estimation,” in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings, Yoshua Bengio and Yann LeCun, Eds., 2015.

A

MAJOR PROJECT REPORT

On

DEEP LEARNING UNDERWATER IMAGE COLOUR CORRECTION AND CONTRAST ENHANCEMENT BASED ON HUE PRESERVATION

Submitted by

Mr. K. Kiran Kalyan (17K81A0427)

Mr. Rishabh Singh (17K81A0442)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATIONS ENGINEERING

Under The Guidance of

Mr. K. Naga Venkateshwara Rao

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500100

NBA &NAAC A+ Accredited

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE
UGC AUTONOMOUS
Dhulapally, Secunderabad-500 100
NBA &NAAC A+ Accredited

BONAFIDE CERTIFICATE

This is to certify that the project entitled "Deep Learning Underwater Image Color Correction And Contrast Enhancement Based On Hue Preservation", is being submitted by **1.Mr. K. Kiran Kalyan (17K81A0427), 2.Mr.Rishabh Singh(17K81A0442)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report has been verified and found satisfactory.

Internal Guide

Mr. K. Naga Venkateshwara Rao

Department of Electronics and Communication Engineering

Head of the Department

Dr. Hari Krishna

Department of Electronics and Communication Engineering

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **K. KIRAN KALYAN** WITH ROLL NO.**17K81A0427, RISHABH SINGH** WITH ROLL NO.**17K81A0442**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**DEEP LEARNING UNDERWATER IMAGE COLOUR CORRECTION AND CONTRAST ENHANCEMENT BASED ON HUE PRESERVATION**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. ★Kompally-500014.★

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of "**BACHELOR OF TECHNOLOGY IN DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING"**, session: (2017 – 2021), St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled '**Deep Learning Underwater Image Color Correction And Contrast Enhancement Based On Hue Preservation'** is the outcome of our bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted to any university for the award of any degree.

K. Kiran Kalyan	**Rishabh Singh**
(17K81A0427)	**(17K81A0442)**

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P. Santosh Kumar Patra,** our beloved **Principal**, **St. Martin's Engineering College** for his invaluable encouragement, suggestions, and support from an early stage of this research and for providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr. B Hari Krishna**, **Head &Professor**, the Dept. of Information Technology for his/her advice, supervision, and vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our dept. **Project coordinator Dr. A. Anand** for his/her involvement with originality has triggered and nourished the intellectual maturity that will help me for a long time to come.

We sincerely thank our **Class Project Coordinator, Mrs. A. Soumya,** Department of ECE and our project **Internal guide, Mr. K. Naga Venkateshwara Rao, Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college-level Project Coordinators **Dr.Maalyadri, Professor in IT, and Dr.R.Santhosh Kumar, Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

1. K. Kiran Kalyan
2. Rishabh Singh

ABSTRACT

Underwater Image suffers from serious color distortion and low contrast problems because of complex light propagation in the ocean. Given the computing constraints of underwater vehicles, we propose a high-efficiency deep-learning framework based on hue preservation. The framework contains three convolutional neural networks for underwater image color restoration. At first, we use the first CNN to convert the input underwater image into the grayscale image. Next, we enhance the grayscale underwater image with the second CNN. And then, we perform the color correction to the input underwater image by the third CNN. At last, we can obtain the color-corrected image by integrating the outputs of three CNNs based on hue preservation. In our framework, that CNN's specialize in each work can be able to simplify each architecture and improve the regression quality to achieve low computing cost and high efficiency. However, the problem with the underwater CNNs is that the underwater training data is too few and without the corresponding ground truth. Thus, we use the unsupervised learning method CycleGAN to train the underwater CNNs. We design a training method as the combination of three CycleGANs that can train the three CNNs at the same time to share the regression status. This training method may let the three CNNs of our proposed framework support each other to avoid the training overfitting and without constraint. By the proposed framework and training method, our method can process the underwater images with high quality and low computing cost. The experimental results have demonstrated the correct colors and high image quality of the proposed method's results, compared with other related approaches.

TABLE OF CONTENTS

LIST OF EQUATIONS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

- Underwater Image suffers from serious color distortion and low contrast problems because of complex light propagation in the ocean.
- We propose a high-efficiency deep-learning framework based on hue preservation.
- The framework contains three convolutional neural networks for underwater image color restoration.
- We can obtain the color-corrected image by integrating the outputs of three CNNs based on the hue preservation.
- Thus, we use the unsupervised learning method Cycle-GAN to train the underwater CNNs. Our method can process the underwater images with high quality and low computing cost.

1.2 OBJECTIVES OF THE STUDY

The main objective of the project is to perform the following four stages

- The underwater image grayscale stage
- The grayscale underwater image details enhancement stage
- The underwater image color restoration stage
- The generation stage

1.3 SCOPE OF STUDY

- MATLAB Software
- Operating System – Windows 10
- 64bit (Intel or AMD) Processor

- GPU Acceleration using parallel computing toolbox

1.4 INTRODUCTION OF PROJECT

The earth is the associate aquatic planet and the maximum amount as eightieth of its surface is roofed by water. Moreover, there is a strong interest in knowing what lies underwater. In present days, an image of deep waters has scope to large investigation to explore the underwater for seafloor expedition and navigation. The enthusiasm of underwater imaging includes the inspection of plants, seabed exploration, the search for wrecks up, and the exploration of natural resources. There were several issues faced by the human underwater if he dives deep into the ocean and stays there for a long time to perform experimentation. [1]. Due to the above reasons, unmanned remote vehicles are used for seafloor exploration.

A. Historical Development

Underwater image quality improvement approaches present a path to magnify the object recognition in the underwater surrounding. A heap of research started for the up-gradation of image visual quality, but a little amount of work has been carried out in this area. In the deep waters, image quality is degraded due to poor illumination conditions and the light properties differ in water compared to air.[2]. Several parameters decrease the quality of an image in underground waters. So to remove all these effects

several techniques have been implemented and practiced.

B. Need for Preprocess

Initially, the processing is necessary for deep water images due to their poor quality during acquisition. A necessity for pre-processing of deepwater images are discussed below:

(i) Quality of images taken from deep water is deteriorated due to light ray attributes like scattering and absorption of light.
(ii) Specificity of surroundings such as lighting inequalities, water torridness, and blue complexion is more or less influential when vehicles move.

(iii) Video or image captured from deep waters like unknown rigid scene, and the depth of the scene and low light sensitivity due to Marine snow, etc.

Figure 1.1 Example of Underwater Image

1.5 INTRODUCTION TO IMAGE ENHANCEMENT AND RESTORATION

The goal of underwater image processing is to enhance visibility and rectify the color deviation. In general, underwater image processing techniques can be divided into two categories, namely, underwater image enhancement and restoration. The underwater image enhancement methods, such as histogram equalization, white balance, pixel stretching, retinex-based methods, and fusion-based methods, generally improve image visual effect by modifying image pixels without concerning the physical degradation mechanism of underwater images. These enhancement-based methods do not exploit physical imaging models and consequently, they are often inadequate for restoring original scene features, especially color features.

On the other hand, underwater image restoration methods aim to recover clear images by exploiting the optical imaging model. The most important task for restoration is to estimate two key model parameters, i.e., transmission and ambient light, which are usually estimated either by prior-based approaches or by learning-based approaches. The prior-based approaches heavily depend on the reliability of certain prior information, such as dark channel prior, red channel prior haze-line prior, and so on. However, priors generally have their respective limitations, and may not adapt to some conditions. Thus, a mismatch between the adopted prior and the target scene may incur significant estimation error, and consequently, recover distorted results.

By contrast, the learning-based approaches aim to obtain more robust and accurate estimation by exploring the relations between the underwater images and the corresponding parameters in a data-driven manner, such as. To this end, it is essential to have a suitable training dataset and an efficient neural network that can be trained to learn such relations. Unfortunately, the existing networks are not capable of estimating the parameters accurately enough and the resulting restored images often suffer from various artifacts; it is also difficult to create a dataset capturing complex and varying underwater environments. Based on the well-known model for underwater physical imaging, we propose a deep convolutional neural network (CNN) model with two parallel branches for underwater image restoration, accompanied by a training dataset in this paper. By exploring the inherent relations between the degraded underwater images and the associated blue channel transmission map as well as global ambient light in a data-driven manner, the proposed CNN can perform more accurate and robust parameter estimation; as a consequence, the restored images exhibit better visibility and more natural color compared to those produced by the state-of-the-art methods.

The contributions of this work are summarized as follows:

(1) We propose a new end-to-end underwater image restoration algorithm using a deep CNN model to improve the contrast and color cast of the recovered images. The whole network consists of two paralleled branches: a transmission estimation sub-network (T-network) and a global ambient light estimation sub-network (A-network), by which the transmission map and the ambient light can be estimated simultaneously. Since in our case no prior information is used to estimate the transmission and ambient light, it avoids the aforementioned mismatch issue commonly encountered in the prior-based methods and helps to improve the accuracy and universality of the estimation method.

1.6 RELATED WORK

Numerous methods have been proposed to enhance underwater degraded images in the past decade. Roughly speaking, these methods can be classified into two categories, namely, enhancement-based methods and restoration-based methods. Enhancement-based methods improve the image quality in a specific sense by using an image enhancement technique designed for that purpose. This type of method can effectively improve the visual effect of the images.

However, because the physical degradation principle is not considered and the relationship between degradation degree and scene depth is neglected, the enhancement results can not reflect the true color features of the scene correctly. On the contrary, the restoration-based methods use the constructed underwater imaging model to reverse the degradation process, for which it is necessary to estimate the unknown parameters: transmission and ambient light. If the model is properly constructed and the parameter estimation is accurate, the restored image will be close to the real scene. Depending on how parameter estimation is performed, they can be further divided into prior-based restoration methods and deep-learning-based restoration methods. In the following subsections, we mainly review the related works belonging to these two kinds of restoration methods.

1.6.1 Prior-Based Restoration Methods

Several techniques are used very frequently for processing the image to improve the visual quality. Some of them are as follows:

(i) Contrast Stretching

(ii) Adaptive Histogram Equalization

1.6.1.1 Contrast stretching

The contrast stretching is a method to transform the high intense region of an image into a brighter and less intense region into darker by using a predefined transformation function T(r) [2]. Generally, the underwater images will have fewer grey values. There are 256 grey values. ‘0’ indicates black and ‘255’ indicates white. In this method, the current grey value of the image is stretched towards 255 i.e., from black to white, pixel by pixel. That means the contrast of the image is stretched so that the quality of the image is improved for better vision.

For example:

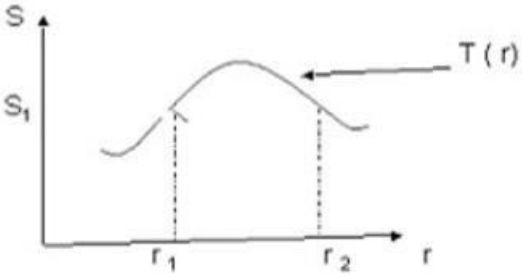


Figure 1.2 Two different gray levels look the same

Here two different thresholds are considered for the entire image and the values between them are stretched to the maximum extent so that the contrast increases. And more over by this method the entire global image contrast is enhanced.

Figure 1.3 Raw image

Figure 1.4 Enhanced image

1.6.1.2 Adaptive histogram equalization

Adaptive histogram equalization is a PC-based image processing technique that is used to improve the quality of image properties like the contrast. It is similar to the contrast stretching method but with a slight difference. It computes several intensities of a specific gray value, each corresponding to a distinct portion of an image, and with the help of them, intensities are rearranged by applying a suitable transformation function. For example, a simple transformation function such as each pixel transformed based on the histogram of a square surrounding the pixel [3]. Existing values will be mapped to new values keeping an actual number of intensities in the resulting image equal or less than the original number of intensities. The transformation function applied on the histogram is proportional to the cumulative distributive function (CDF) of pixel values in the neighborhood. Therefore it suits for enhancing the local details and enhancing the edge information of each region of an image.

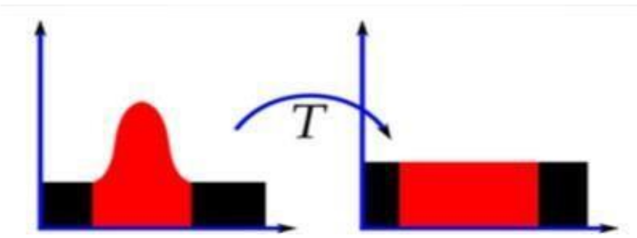


Figure 1.5 Histograms of an image before and after histogram equalization

Histogram equalization is a technique for changing the overall pixel intensities based on the transformation function and contrast of an image. Histogram equalization is an effective technique that will benefit the images with extreme contrast values. The limitation of this technique highlights the unwanted noise present in the background of an image and leads to a loss in the information signal. It results in undesired effects in the resultant images.

Figure 1.6 Raw image

Figure 1.7 Enhanced image

The ocean has a large storage of resources, but the riches of the sea have yet to be developed. The main problem is that the underwater environment is hostile to humans. Thus, automatic technology plays a major role in underwater engineering. The automatic technology includes automatic inspection, process control, and robot guidance. These applications are all based on machine vision to provide imaging-based inspection and analysis. However, machine vision is not effective enough in an underwater environment. Due to the complex light propagation in an underwater environment, the acquired images suffer from serious color distortion and poor visual quality. It would reduce the accuracy and efficiency of machine vision. Therefore, researchers have attempted restoring the underwater image to the correct color and high image quality. In this paper, we focus on underwater image color restoration from a single image.

1.7 DEEP-LEARNING-BASED RESTORATION METHODS

Deep learning techniques have gained increasing popularity in image processing, and have been used for image transmission map estimation in particular. Deep-learning-based methods rely on neural networks, trained by a large amount of data, to learn the relationship between an image and its associated transmission map. They can avoid, to a certain extent, the estimation errors incurred by invalid priors as seen in the prior-based methods. However, the neural networks that have been proposed so far are all for single-channel transmission estimation, with the implicit assumption that three color channels have the same transmission map. As such, they are only able to correct blurring and low contrast caused by scattering, but not color deviation induced by the discrepancies between the transmission maps of three channels commonly seen in underwater images.

Inspired by DehazeNet, Shin et al. proposed an underwater restoration method based on deep learning, which estimates global ambient light and local transmission of underwater images using the same network, and then leverages the estimates to restore the clear image via the optical model. This method uses synthetic data for training. Specifically, the underwater image patches are simulated by adding different color deviations to the clear image patches, and the training label of each image patch is the corresponding transmission value or the ambient light value. Although this method achieves promising results, there are several aspects to be improved. In particular, since local image patches are used as the training data (which lack global information) and that the discrepancies between the transmission maps of three channels are ignored in recovery, this method is often not able to estimate the parameters with sufficient accuracy and the restored images tend to suffer from color distortion and low clarify. A similar method was developed by Barbosa et al, which estimates the transmission map and the ambient light respectively using a retrained DehazeNet and a prior proposed by Drews et al. It is well known that the performances of deep-learning-based methods depend critically on the quality of the training data. The training datasets used to consist of haze images with no color deviation. The work by Shin et al uses a large number of training image patches with color deviation not specific to underwater environments, which are known to be mostly blue/green. For transmission estimation, Barbosa et al. rendered two 3D scenes and obtained a large set of images by positioning the camera in different locations, which are nevertheless still not sufficient to simulate diverse underwater environments. The lack of suitable training datasets has become a roadblock for the deep-learning-based approach to underwater image restoration.

1.8 INTRODUCTION TO IMAGE PROCESSING

1.8.1 Image

An image is a two-dimensional picture, which has a similar appearance to some subject, usually a physical object or a person.

Image is two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Figure 1.8 General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

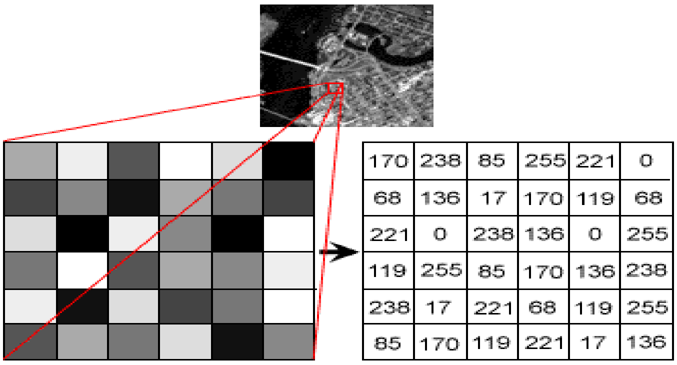


Figure 1.9 image pixel

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

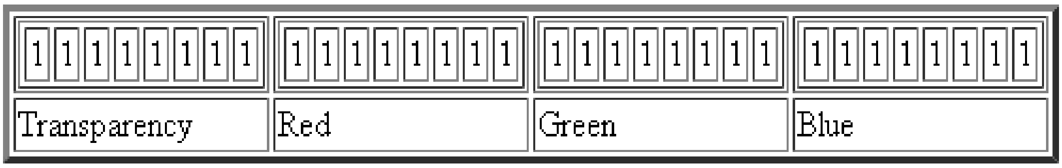


Figure 1.10 Transparency image

1.8.2 Image file sizes

The image file size is expressed as the number of bytes that increase with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High-resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

1.8.3 Image file formats

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

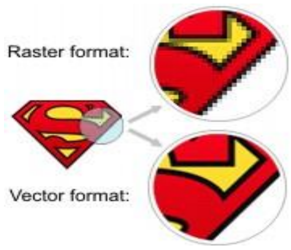


Figure 1.11 Resolution image

In addition to straight image formats, Metafile formats are portable formats that can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their native format.

1.8.4 Raster formats

These formats store images as bitmaps (also known as pixmaps)

- **JPEG/JFIF:**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is a lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

- **EXIF:**

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata is recorded for individual images and includes such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of the camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

- **TIFF:**

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

- **PNG:**

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

- **GIF:**

GIF (Graphics Interchange Format) is limited to an 8-bit palette or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos, and cartoon-style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

- **BMP:**

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

1.8.5 Vector formats

As opposed to the raster image formats above (where the data describes the characteristics of each pixel), vector image formats contain a geometric description that can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows, and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM:**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

- **SVG:**

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable, and all-purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

1.8.6 Image processing

Digital image processing, the manipulation of images by computer, is a relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of image with varying degrees of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields suffers from myths, dis-connect ions, misunderstandings, and misinformation. It is the vast umbrella under which fall diverse aspects of optics, electronics, mathematics, photography graphics, and computer technology. It is a truly multidisciplinary endeavor plowed with imprecise jargon.

Several factors combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low-cost microprocessors, and the use of charge-coupled devices (CCDs) for digitizing, storage during processing and display, and large low cost of image storage arrays.

1.8.7 Fundamental steps in digital image processing

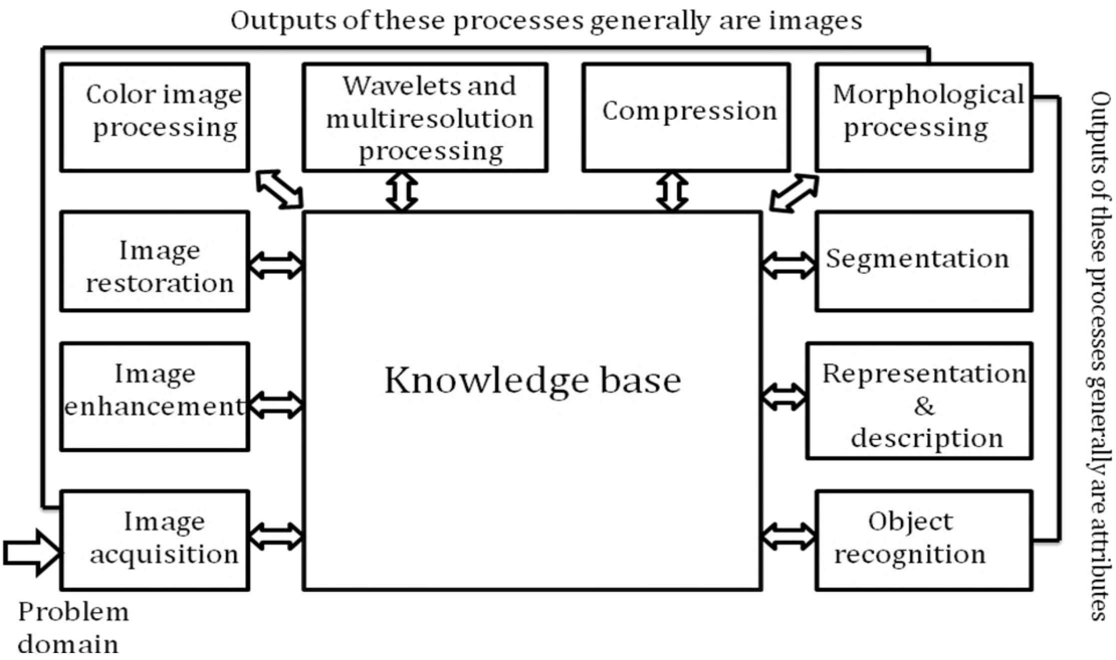


Figure 1.12 Image fundamental

1.8.8 Image Acquisition

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Figure 1.13 Digital camera image

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Figure 1.14 Digital camera cells

1.8.9 Image Enhancement

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of an interesting image. A familiar example of enhancement is when we increase the contrast of an image because “it looks better.” It is important to keep in mind that enhancement is a very subjective area of image processing.

Figure 1.15 Image enhancement

1.8.10 Image restoration

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Figure 1.16 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, whereas removal of image blur by applying a deblurring function is considered a restoration technique.

1.8.11 Color image processing

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Figure 1.17 Color & grayscale image

1.8.12 Wavelets and multiresolution processing

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform based image processing since the late1950’s, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

Figure 1.18 RGB histogram image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called Multiresolution theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

1.8.13 Compression

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

1.8.14 Morphological processing

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Figure 1.19 Blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

1.8.15 Segmentation

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

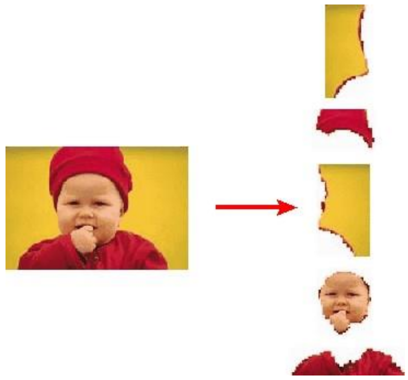


Figure 1.20 Image segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

1.8.16 Representation and description

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

1.8.17 Object recognition

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

1.8.18 Knowledgebase

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an inter related to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

1.8.19 Components of an image processing system

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

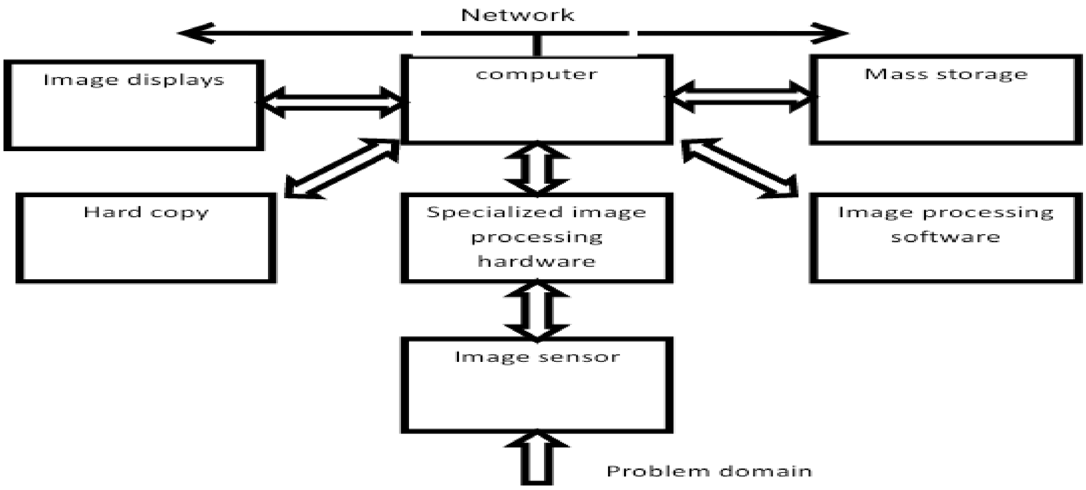


Figure 1.21 Components of image processing

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 1.24 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

- **Image sensors:**

With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

- **Specialized image processing hardware:**

Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

- **Computer:**

The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

- **Image processing software:**

Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

- **Mass storage:**

Mass storage capability is a must in image processing applications. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage forimage processing applications fall into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), Gbytes (meaning giga, or one billion, bytes), and Tbytes (meaning tera, or one trillion, bytes)

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig. 1.24. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in “jukeboxes” are the usual media for archival applications.

- **Image displays:**

Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are there requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

- **Hardcopy:**

Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

- **Network:**

Networking is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of optical fiber and other broadband technologies.

Color and texture are two low-level features widely used for image classification, indexing and retrieval. Color is usually represented as a histogram, which is a first order statistical measure that captures global distribution of color in an image One of the main drawbacks of the histogram- based approaches is that the spatial distribution and local variations in color are ignored. Local spatial variation of pixel intensity is commonly used to capture texture information in an image. Grayscale Co-occurrence Matrix (GCM) is a well-known method for texture extraction in the spatial domain. A GCM stores the number of pixel neighborhoods in an image that have a particular grayscale combination. Let I be an image and let p and Np respectively denote any arbitrary pixel and its neighbor in a given direction. If GL denotes the total number of quantized gray levels and gl denotes the individual gray levels, where, gl $\in$ {0, . . .,GL _ 1}, then each component of GCM can be written as follows:

$$\mathrm{gcm}(i,j) = \Pr((\mathrm{gl}_{\mathrm{p}}, \mathrm{gl}_{N_{\mathrm{p}}}) = (i,j))$$

Equation 1 Each component of GCM

gcm(i, j) is the number of times the gray level of a pixel p denoted by glp equals i, and the gray level of its neighbor Np denoted by glNp equals j, as a fraction of the total number of pixels in the image. Thus, it estimates the probability that the gray level of an arbitrary pixel in an image is i, and that of its neighbor is j. One GCM matrix is generated for each possible neighborhood direction, namely, 0, 45, 90 and 135.Average and range of 14 features like Angular Second Moment, Contrast, Correlation, etc., are generated by combining all the four matrices to get a total of 28 features. In the GCM approach for texture extraction, color information is completely lost since only pixel gray levels are considered.

To incorporate spatial information along with the color of image pixels, a feature called color correlogram has recently been proposed. It is a three dimensional matrix that represents the probability of finding pixels of any two given colors at a distance ‘d’ apart Auto correlogram is a variation of correlogram, which represents the probability of finding two pixels with the same color at a distance ‘d’ apart. This atproach can effectively represent color distribution in an image. However, correlogram features do not capture intensity variation Many image databases often contain both color as well as gray scale images. The color correlogram method does not constitute a good descriptor in such databases.

Another method called Color Co-occurrence Matrix (CCM) has been proposed to capture color variation in an image. CCM is represented as a three-dimensional matrix, where color pair of the pixels p and Np are captured in the first two dimensions of the matrix and the spatial distance ‘d’ between these two pixels is captured in the third dimension. This approach is a generalization of the color correlogram and reduces to the pure color correlogram for d = 1. CCM is generated using only the Hue plane of the HSV (Hue, Saturation and Intensity Value) color space. The Hue axis is quantized into HL number of levels. If individual hue values are denoted by hl, where

$$\mathrm{hl} \in \{0, \ldots, \mathrm{HL} - 1\}$$

Equation 2 Hue values set

then each component of CCM can be written as follows:

$$\mathrm{ccm}(i, j) = \mathrm{Pr}((\mathrm{hl}_{\mathrm{p}}, \mathrm{hl}_{N_{\mathrm{p}}}) = (i, j))$$

Equation 3 Each component of CCM

Four matrices representing neighbors at angles 0, 90, 180 and 270 are considered. This approach was further extended by separating the diagonal and the non-diagonal components of CCM to generate a Modified Color Co-occurrence Matrix (MCCM). MCCM, thus, may be written as follows: MCCM = (CCMD;CCMND)

Here, CCMD and CCMND correspond to the diagonal and off-diagonal components of CCM. The main drawback of this approach is that, like correlogram, it also captures only color information and intensity information is completely ignored.

An alternative approach is to capture intensity variation as a texture feature from an image and combine it with color features like histograms using suitable weights . One of the challenges of this approach is to determine suitable weights since these are highly application-dependent. In certain applications like Content-based Image Retrieval (CBIR), weights are often estimated from relevance feedback given by users.

While relevance feedback is sometimes effective, it makes the process of image retrieval user-dependent and iterative. There is also no guarantee on the convergence of the weight-learning algorithms. In order to overcome these problems, researchers have tried to combine color and texture features together during extraction proposed two approaches for capturing color and intensity variations from an image using the LUV color space. In the Single-channel Co-occurrence Matrix (SCM), variations for each color channel, namely, L, U and V are considered independently. In the Multi channel Co-occurrence Matrix (MCM), variations are captured taking two channels at a time – UV, LU and LV. Since the LUV color space separates out chrominance (L and U) from luminance (V), SCM in effect generates one GCM and two CCMs from each image independently. As a result, correlation between the color channels is lost

However, in MCM, the count of pairwise occurrences of the values of different channels of the color space is captured. Thus, each component of MCM can be written as follows:

$$mcmUV(i; j) = Pr((up; vNp) = (i; j))$$

$$mcmLU(i; j) = Pr((lp; uNp) = (i; j))$$

$$mcmLV(i; j) = Pr((lp; vNp) = (i; j))$$

Here, mcmUV(i, j) is the number of times the U chromaticity value of a pixel p denoted by up equals i, and the V chromaticity value of its neighbor Np denoted by vNp equals j, as a fraction of the total number of pixels in the image. Similarly, mcmLU(i,

j) and mcmLV(i, j) are defined. One MCM matrix is generated for each of the four neighborhood directions, namely, 0, 45, 90 and 135.

Deng and Manjunath (2001) proposed a two-stage method called JSEG, which combines color and texture after image segmentation. In the first stage, colors are quantized to the required levels for differentiating between various regions of an image. Pixel values of the regions are then replaced by their quantized color levels to form a color map. Spatial variation of color levels between different regions in the map is viewed as a type of texture composition of the image.

Yu et al. (2002) suggested the use of color texture moments to represent both color and texture of an image. This approach is based on the calculation of

Local Fourier Transformation (LFT) coefficients. Eight templates equivalent to LFT are operated over an image to generate a characteristic map of the image. Each template is a 3 · 3 filter that considers eight neighbors of the current pixel for LFT calculation. First and second order moments of the characteristic map are then used to generate a set of features.

In this paper, we propose an integrated approach for capturing spatial variation of both color and intensity levels in the neighborhood of each pixel using the HSV color space. In contrast to the other methods, for each pixel and its neighbor, the amount of color and intensity variation between them is estimated using a weight function. Suitable constraints are satisfied while choosing the weight function for effectively relating visual perception of color and the HSV color space properties. The color and intensity variations are represented in a single composite feature known as Integrated Color and Intensity Co-occurrence Matrix (ICICM). While the existing schemes generally treat color and intensity separately, the proposed method provides a composite view to both color and intensity variations in the same feature. The main advantage of using ICICM is that it avoids the use of weights to combine individual color and texture features. We use ICICM feature in an image retrieval application from large image databases.

Early result on this work was reported in (Vadivel et al., 2004a). In the next section, we describe the proposed feature extraction technique after introducing some of the

properties of the HSV color space. Choice of quantization levels for color and intensity axes, selection of parameter values and a brief overview of the image retrieval application

1.8.20 Integrated color and intensity co-occurrence matrix

We propose to capture color and intensity variation around each pixel in a two-dimensional matrix called Integrated Color and Intensity Co-occurrence Matrix (ICICM). This is a generalization of the Grayscale Co-occurrence Matrix and the Color Co-occurrence Matrix techniques. For each pair of neighboring pixels, we consider their contribution to both color perception as well as gray level perception to the human eye. Some of the useful properties of the HSV color space and their relationship to human color perception are utilized for extracting this feature. In the next sub-section, we briefly explain relevant properties of the HSV color space. In the subsequent subsection, we describe how the properties can be effectively used for generating ICICM.

1.8.21 HSV color space

HSV Color space: Basically there are three properties or three dimensions of color that being hue, saturation and value HSV means Hue, Saturation and Value. It is important to look at because it describes the color based on three properties. It can create the full spectrum of colors by editing the HSV values. The first dimension is the Hue. Hue is the other name for the color or the complicated variation in the color. The quality of color as determined by its dominant wavelength. This Hue is broadly classified into three categories. They are primary Hue, Secondary Hue and Teritiary hue. The first and the foremost is the primary Hue it consists of three colors they are red, yellow and blue. The secondary Hue is formed by the combination of the equal amount of colors of the primary Hue and the colors of the secondary Hue which was formed by the primary Hue are Orange, Green and violet. The remaining one is the teritiary Hue is formed by the combination of the primary Hue and the secondary Hue. The limitless number of colors are produced by mixing the colors of the primary Hue in different amounts. Saturation is the degree or the purity of color. Then the second dimension is the saturation. Saturation just gives the intensity to the colors. The saturation and intensity drops just by mixing the colors or by adding black to the color. By adding the white to the color in spite of more intense the color becomes lighter. Then finally the third dimension is the Value. The value is the brightness of the color. When the value is zero the color space is totally black with the increase in the color there is also increase in the brightness and shows the various colors. The value describes the contrast of the color. That means it describes the lightness and darkness of the color. As similar to the saturation this value consists of the tints and shades. Tints are the colors with the added white and shades are the colors with the added black.

1.8.22 Properties of the HSV color space

Sensing light from an image in the layers of the human retina is a complex process with rod cells contributing to the scotopic or dim-light vision and cone cells to photopic or bright-light vision (Gonzalez and Woods, 2002). At low levels of illumination, only the rod cells are excited so that only gray shades are perceived. As the illumination level increases, more and more cone cells are excited, resulting in increased color perception. Various color spaces have been introduced to represent and specify colors in a way suitable for storage, processing, or transmission of color information in images. Out of these, HSV is one of the models that separate the luminance component (Intensity) of a pixel color from its chrominance components (Hue and Saturation). Hue represents pure color, which is perceived when the incident light is of sufficient illumination and contains a single wavelength. Saturation gives a measure of the degree to which a pure color is diluted by white light. For light with low illumination, the corresponding intensity value in the HSV color space is also low.

The HSV color space can be represented as a Hexa cone, with the central vertical axis denoting the luminance component, I (often denoted by V for Intensity Value). Hue is a chrominance component defined as an angle in the range [0,2p] relative to the red axis with red at angle 0, green at 2p/3, blue at 4p/3, and red again at 2p. Saturation, S, is the other chrominance component, measured as a radial distance from the central axis of the hexacore with a value between 0 at the center to 1 at the outer surface. For zero saturation, as the intensity is increased, we move from black to white through various shades of gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the purest form of the color represented by its hue. When saturation is near 0, all the pixels in an image look alike even though their hue values are different.

As we increase saturation towards 1, the colors get separated and are visually perceived as the true colors represented by their hues. Low saturation implies the presence of a large number of spectral components in the incident light, causing loss of color information even though the illumination level is sufficiently high. Thus, for low values of saturation or intensity, we can approximate a pixel color by a gray level while for higher saturation and intensity, the pixel color can be approximated by its hue. For low intensities, even for high saturation, a pixel color is close to its gray value. Similarly, for low saturation even for a high value of intensity, a pixel is perceived as gray. We use these properties to estimate the degree to which a pixel contributes to color perception and gray level perception.

One possible way of capturing the color perception of a pixel is to choose suitable thresholds on the intensity and saturation. If the saturation and the intensity are above their respective thresholds, we may consider the pixel to have color dominance; else, it has gray level dominance. However, such hard thresholding does not properly capture color perception near the threshold values. This is because there is no fixed level of illumination above which the cone cells get excited. Instead, there is a gradual transition from scotopic to photopic vision. Similarly, there is no fixed threshold for the saturation of cone cells that leads to loss of chromatic information at higher levels of illumination caused by color dilution. We, therefore, use suitable weights that vary smoothly with saturation and intensity to represent both color and gray scale perception for each pixel.

1.8.23 Non interval quantization

Due to the large range for each component by directly calculating the characteristics for the retrieval then the computation will be very difficult to ensure rapid retrieval. It is essential to quantify the HSV space component to reduce computation and improve efficiency. At the same time, because the human eye to distinguish colors is limited, do not need to calculate all segments. Unequal interval quantization according to the human color perception has been applied on H, S, V components.

Based on the color model of substantial analysis, we divide the color into eight parts. Saturation and intensity are divided into three parts separately following the human eyes to distinguish. Per the different colors and subjective color perception quantification, quantified hue(H), saturation(S), and value(V)

Following the quantization level above, the H, S, V three-dimensional feature vector for different values with different weights to form one dimensional feature vector and is given by the following equation:

$$G = Qs*Qv*H+Qv*s+V$$

Where Qs is the quantized series of S and Qv is the quantized series of V. And now by setting Qs = Qv = 3, Then G = 9H+3S+V

In this way three-component vector of the HSV from a one-dimensional vector, which quantizes the whole color space for the 72 kinds of the main colors. So we can handle 72 bins of a one-dimensional histogram. This qualification is effective in reducing the images by the effect of the light intensity, but also reducing the computational time and complexity.

1.8.24 Image retrieval

Image retrieval is nothing but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords, or the descriptions to the images so that the retrieval can be performed. Manual image annotation is time-consuming, expensive, and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc..,

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others. one can today generate volumes of images with content as diverse as family get-togethers and national park visits. Low-cost storage and easy Web hosting has fueled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchable image data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real-world image search system designers.

As far as technological advances are concerned, growth in Content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real world implications, applications, and constraints of the technology. Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future.

An image retrieval system designed to serve a personal collection should focus on features such as personalization, flexibility of browsing, and display methodology. For example, Google's Picasa system [Picasa 2004] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods. The notable few applications include an FPGA implementation of a color-histogram-based image retrieval system [Kotoulas and Andreadis 2003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 2003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 2002]

Discussion. Regardless of the nature of the collection, as the expected user-base grows, factors such as concurrent query support, efficient caching, and parallel and distributed processing of requests become critical. For future real-world image retrieval systems, both software and hardware approaches to address these issues are essential.

More realistically, dedicated specialized servers, optimized memory and storage support, and highly parallelizable image search algorithms to exploit cluster computing powers are where the future of large-scale image search hardware support lies.

1.8.25 Overview of texture

We all know about the term Texture but for defining it is a hard time. One can differentiate the two different Textures by recognizing the similarities and differences. Commonly there are three ways for the usage of the Textures:

Based on the Textures the images can be segmented To differentiate between already segmented regions or to classify them. We can reproduce Textures by producing the descriptions.

1.9 ORGANIZATION OF CHAPTERS

1.9.1 Introduction

This section provides information about the project such as overview, objective and scope of the project. Along with that, this chapter also includes information about the techniques that have been used or considered while working on the project.

1.9.2 Literature Survey

This section provides the information about the existing methods on 'underwater image color correction and contrast enhancement' with the help of which this project was modified.

1.9.3 Project Design

This section provides the information about the methodology that is applied in this project i.e., the design is discussed and also the software used to implement it.

1.9.4 Project Implementation

This section provides the information about how the project is implemented. It also contains the results of the project.

1.9.5 Conclusion and Future Enhancement

This section concludes the project along with the applications of it. This project can be enhanced further for better results and wide utility and the future enhancement chances are mentioned here.

CHAPTER 2: LITERATURE SURVEY

Single image underwater image restoration is an ill-posed problem. To restore the information of underwater image, the underwater image transmission model need to be estimated. The image processing theories are the traditional method to solve this estimation problem. C. Li et al. [1] proposed a restoration method by blue-green channels dehazing and red channel color correction. It restores the red channel of underwater image with the Gray-World assumption theory. Blue and green channels are processed with the modification of Dark Channel. To prevent the result become too dark or too bright, the restored image would be adjusted by the adaptive exposure map at last. D. Berman et al. [2] proposed a underwater image color restoration method that consider wavelength-dependent attenuation and take into account various water types. In this method, the underwater image transmission are estimated by the attenuation ratios of the blue-red and blue-green color channels. The value of attenuation ratios are defined by each water type. The underwater image would evaluate the all possible water types and the one that matches the Gray-World assumption best is chosen as the result. Besides the image processing method, the deep learning is also the popular method for image restoration. However, the underwater datasets are scarce and without corresponding ground truth. The common deep supervised learning needs pairs of training data for training, so the unsupervised learning is used to generate the fake underwater image pairs as the training dataset. J. Li et al. [3] proposed an unsupervised pipeline called WaterGAN. WaterGAN is based on generative adversarial network(GAN) [4] to generate realistic underwater image by the in-air image and corresponding depth map.

2.1 CONCLUSION ON REVIEWS

The disadvantage of the existing methods is that the transformation function is not unique. Depending on the application the suitable transformation function is chosen. Also the noise in relatively homogeneous regions of the image is amplified which results in poor SNR. Only the local objects of the image are enhanced and the background is left unenhanced.

So, this paper proposed an end-to-end convolutional neural network for underwater image color restoration. This convolutional neural network architecture contains two stage. The first stage is relative depth estimation for the underwater image. The estimated depth map and underwater image are the input of the second stage to restore the correct color. The dataset that generated by WaterGAN is used to train this endto-end network to restore the underwater image

CHAPTER 3: PROJECT DESIGN

3.1 FRAMEWORK OF PROPOSED UNDERWATER IMAGE ENHANCEMENT

the proposed deep learning-based underwater image restoration framework consists of the four stages: (1) the underwater image grayscale stage relying on CNN to convert the underwater image to the best gray-channel image; (2) the grayscale underwater image details enhancement stage also relying on CNN to remove the noise and enhance the image quality; (3) the underwater image color restoration stage via end-to-end CNN; and (4) the generation stage of the final high image quality and correct color underwater image by integrating the outputs of the other three stages. The details of the four stage is in the following four subsection, respectively.

3.1.1 Underwater Image Grayscale Transformer

Based on hue preservation enhancement method, the first stage is converting the input underwater image to the grayscale image. Different from the in-air image, the red channel and the green channel of underwater image is attenuated by light propagation. We have to analysis the degree of light attenuation to evaluate the best ratio among three channels. To accelerate the algorithm computing, we use the convolutional neural network to predict the ratios. As shown in Fig. 2, the proposed underwater image grayscale transform CNN aims at transforming an input underwater RGB image to the three coefficients that are used to combine RGB three channel to the corresponding grayscale image. The CNN consists of four convolutional layers (denoted by Conv in Fig. 2). In the network architecture, the concatenation layer (denoted by Concat in Fig. 2) is used to stack the outputs from two convolutional layers with different filter size. It would be useful to condense multi-scale features to learn the prediction method. The activation functions of all convolutional layer in this architecture are the rectified linear unit (ReLU) [5]. ReLU is mainly utilized for nonlinearity, which has been shown to allow for faster and more correct regression than other non-linear separators. We obtains the coefficients from the mean of all elements in each feature map from the output of the final convolutional layer. This network architecture benefits the advantages that enable to extract multi-scale image feature, which would be helpful for grayscale ratio prediction.

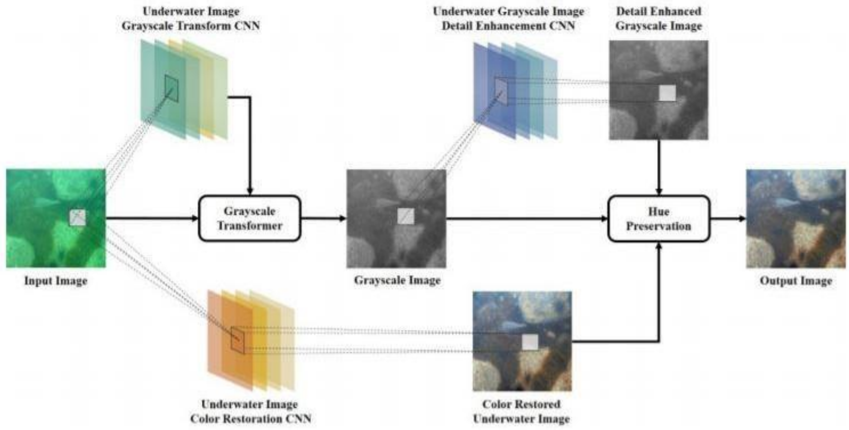


Figure 3.1 The diagram of the proposed deep learning-based underwater image color restoration framework

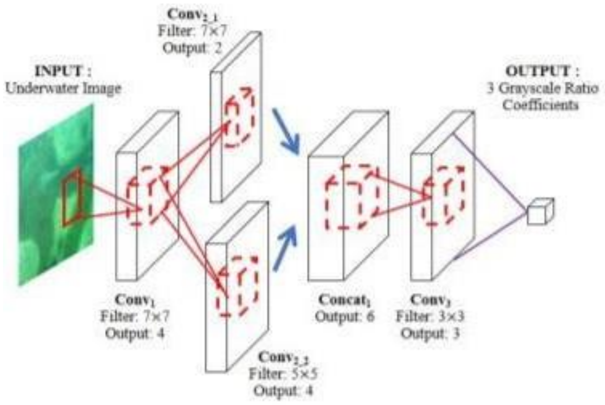


Figure 3.2 The proposed underwater image transform convolutional neural network architecture for gray ratio prediction

As a result, the grayscale image is obtained from the input underwater image and the three coefficients from underwater image grayscale transform CNN as:

$$I_{Gray} = \frac{p_R}{p_R+p_G+p_B} I_R + \frac{p_G}{p_R+p_G+p_B} I_G + \frac{p_B}{p_R+p_G+p_B} I_B, \quad (1)$$

where I_R , I_G , I_B ,denote the R, G, B channel of the input underwater image and P_R ,P_G , P_B are the gray channel ratio coefficients predicted by underwater image grayscale transform CNN. By this method, we can obtain the underwater grayscale image with the most image complexity and information.

3.1.2 Underwater Grayscale Image Detail Enhancement

The end-to-end convolutional neural network method is a great way to solve the image processing problems with low computing cost and high quality. However, the deep learning methods still contain some problems. Because of slight deviation in image regression, the processed image may contains noise and blurry. Otherwise, the convolutional neural network with the light architecture cannot burden to do the underwater image color correction and the underwater image denoise at the same time. Thus, based on hue preservation, we proposed the underwater grayscale image detail enhancement CNN for the underwater image denoise and CNN processed image detail correction. Inspired by the Google Inception V3 Net, the architecture of underwater grayscale image detail enhancement CNN is shown in Fig. 3. This CNN aims at transforming an input underwater grayscale image to the enhancement transmission map. The architecture consists of seven convolutional layers. In this architecture, we choose Parametric Rectified Linear Unit (PReLU) as the activation function. PReLU is better than ReLU at learning the transmission map about image texture modification and enhancement. At last, the enhanced underwater grayscale image is obtained by adding the enhancement transmission map to the input grayscale image.

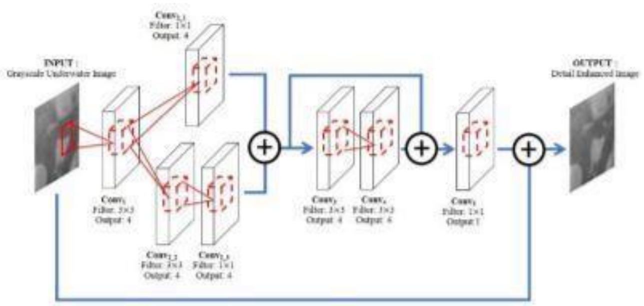


Figure 3.3 The proposed underwater grayscale image detail enhancement convolutional neural network architecture

3.1.3 Underwater Image Color Restoration

The underwater image color restoration is the main stage of the framework. We use the end-to-end convolutional neural network to correct the color loss in underwater image. Underwater color correction is a complex problem. A high image quality in-air images have to contain high color saturation, right color balance and appropriate image contrast. Thus, we design a loss function which restricts many conditions for the underwater image color restoration CNN regression. The content details of the loss function present in Sec. III.

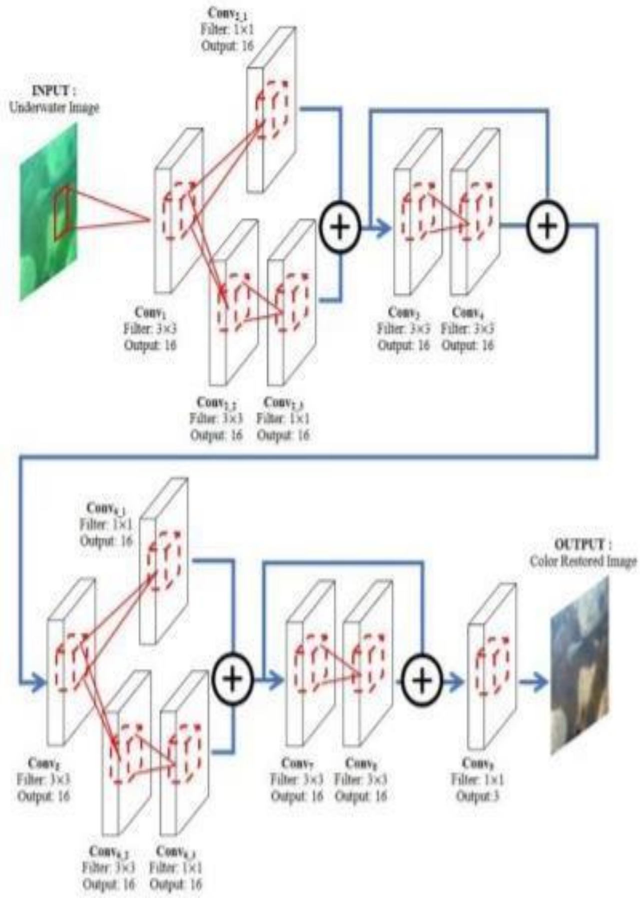


Figure 3.4 The proposed underwater image color restoration convolutional neural network architecture

3.1.4 Hue Preservation Enhancement

Hue preservation is necessary for color image enhancement. In this stage, we use the hue preservation enhancement method from Tian et al. [8]. Tian et al. proposed a global contrast adaptive enhancement method based on hue preservation enhancement. It is a great method to integrate the results of color correction and detail enhancement. The output image $I_R(x)$ is generated by integrating the outputsf other three stages as:

$$I_R(x) = \begin{cases} \frac{I_{GE}(x)}{I_G(x)} I_C(x), & if\ \frac{I_{GE}(x)}{I_G(x)} \leq 1 \\ \frac{255-G_E(x)}{255-G(x)} (I_C(x) - I_G(x)) + I_{GE}(x), & if\ \frac{I_{GE}(x)}{I_G(x)} > 1 \end{cases} \quad (2)$$

Where $I_C(x)$ represents the hue preservation enhanced color image, $I_G(x)$ is the grayscale image of the input underwater image generated by the underwater image grayscale transformer stage, $I_{GE}(x)$ is the output of the grayscale image detail enhancement stage, Hence it's the color restored underwater image from the underwater image color restoration stage and x denotes the positions of pixels on the image.

3.1.5 Proposed Unsupervised Training Method

The main problem for deep learning-based underwater image restoration method is the lack of dataset. The underwater image datasets are rare and have no ground truth. To solve this problem, we use the unsupervised learning to train the proposed convolutional neural networks. GAN is the popular unsupervised learning method which have achieved many impressive results in image generation and image editing. Based on the algorithm of GAN, CycleGAN is proposed. As shown in Fig. 5, the architecture contains backpropagation to restore the images that generate by another generator. Using the cycle consistency loss and adversary loss, It can also guide the generator network to perform better in reconstructing image

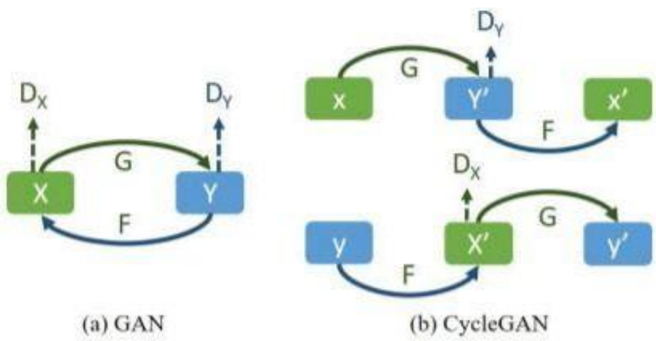


Figure 3.5 The architecture of GAN and CycleGAN

We combine three CycleGANs as our training method to train the three CNNs of the proposed framework at the same time. We distinguish these three CycleGANs by their generators. In the first part, we train the underwater grayscale image detail enhancement CNN as the generator. The two domains of the input training data are the gray underwater images and the gray in-air underwater image. The underwater image color restoration CNN are trained as the generator of the second part. The second CycleGAN learning the mapping between the color underwater image and the color in-air image. The generator of the last part is the whole framework. The training data of the last CycleGAN is same as the second ones. With these three parts, we can training all CNNs in the whole framework at the same time. However, CycleGAN alone cannot perfectly train the underwater image restoration framework. Overfitting and falling into a cycle of greedy optimization are the problems of CycleGAN that would affect the training results. Thus, we additional set some conditions in loss function to restrict the CNN regression. These conditions also share the training information from other CNNs to make the CNNs learn the mapping more comprehensively.

3.2 DEFINE THE MODULES

3.2.1 Introduction to MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modelling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In the industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available to include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

3.2.2 The MATLAB system

The MATLAB system consists of five main parts

- **Development Environment:**

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library:**

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language:**

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both ―programming in the small‖ to rapidly create quick and dirty throw-away programs, and ―programming in the large‖ to create large and complex application programs.

- **Graphics:**

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using the IMAGE PROCESSING toolbox.

3.2.3 Graphical User Interface (GUI)

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG file contains binary data that does not need to be parsed when the associated GUI-based M function is executed.

● A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited and callback functions that are executed when a user interacts with GUI objects for example when a button is pushed.

To launch GUIDE from the MATLAB command window, type :

guide filename

Where filename is the name of an existing FIG-file on the current path. If the filename is omitted,

GUIDE opens a new (i.e., blank) window.

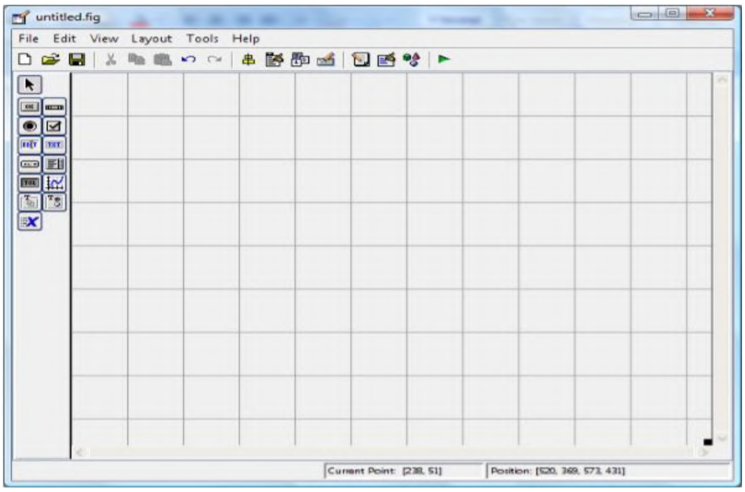


Figure 3.6 MATLAB procedure

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, pushbuttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

3.2.4 Getting Started to Matlab

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

3.2.5 Introduction.

Describes the components of the MATLAB system.

3.2.6 Development environment

Introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

3.2.7 Manipulating matrices

Introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

3.2.8 Graphics

Introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images. 2.4.5 Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

3.2.9 Development Environment

- **Introduction**

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

- **Starting MATLAB**

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop. On a UNIX platform, to start MATLAB, type MATLAB

At the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup and reduce startup time in some situations.

- **Quitting MATLAB**

To end your MATLAB session, select Exit MATLAB from the File menu on the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

- **MATLAB Desktop**

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB. The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

- **Desktop Tools**

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacsmagik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the Help button in the toolbar, or type help browser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favourite tab - View a list of documents you previously designated as favourites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term on the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find Mfiles and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces. To view the workspace and information about each variable, use the Workspace browser, or use the functions who and who's. To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function. The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT file, which has a .mat extension. There are options for saving to different formats. To read in a MAT file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to run MATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop.

3.2.10 Digital image processing background:

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem.

An important characteristic underlying the design of image processing systems is the significant level of testing & experimentation that normally is required before arriving at an acceptable solution. This characteristic implies that the ability to formulate approaches &quickly prototype candidate solutions generally plays a major role in reducing the cost & time required to arrive at a viable system implementation.

3.2.11 What is Digital Image Processing

An image may be defined as a two-dimensional function f(x, y), where x & y are spatial coordinates, & the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital images by means of a digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensors, so it is not surprising that images play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis& computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting & somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low-level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low- level process is characterized by the fact that both its inputs & outputs are images.

Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images. Finally higher- level processing involves "Making sense" of an ensemble of recognized objects, as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

3.2.12 What is an image:

An image is represented as a two dimensional function f(x, y) where x and y are spatial co-ordinates and the amplitude of 'f' at any pair of coordinates (x, y) is called the intensity of the image at that point.

3.2.13 Gray scale image:

A grayscale image is a function I (xylem) of the two spatial coordinates of the image plane.

I(x, y) is the intensity of the image at the point (x, y) on the image plane.

I (xylem) takes non-negative values assume the image is bounded by a rectangle [0, a] ·[0, b]I: [0, a] · [0, b] → [0, info)

3.2.14 Color image:

It can be represented by three functions, R (xylem) for red, G (xylem) for green *and* B (xylem) for blue.

An image may be continuous with respect to the x and y coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate's values is called sampling. Digitizing the amplitude values is called quantization.

3.2.15 Coordinate convention:

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image f(x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size M X N. The values of the coordinates (xylem) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates.

In many image processing books, the image origin is defined to be at (xylem)=(0,0).The next coordinate values along the first row of the image are (xylem)=(0,1).It is important to keep in mind that the notation (0,1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Following figure shows the coordinate convention. Note that x ranges from 0 to M-1 and y from 0 to N-1 in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (xylem) the toolbox uses the notation (race) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate topples, (alb), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentation refers to the coordinates. Less frequently the toolbox also employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refers to rows. This is the opposite of our use of variables x and y.

3.2.16 Image as Matrices:

The preceding discussion leads to the following representation for a digitized image function:

$$
f\,(xylem) = \begin{array}{cccc}
f\,(0,0) & f(0,1) & \ldots\ldots\ldots.. & f(0,N-1) \\
f\,(1,0) & f(1,1) & \ldots\ldots\ldots\ldots & f(1,N-1) \\
. & . & & . \\
f\,(M-1,0) & f(M-1,1) & \ldots\ldots\ldots\ldots\ldots & f(M-1,N-1)
\end{array}
$$

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements. A digital image can be represented naturally as a MATLAB matrix:

$$f = \begin{matrix} f(1,1) & f(1,2)\ldots\ldots\ldots\ldots f(1,N) \\ f(2,1) & f(2,2)\ldots\ldots\ldots\ldots f(2,N) \\ . & . \qquad\qquad . \\ . & . \qquad\qquad . \\ f(M,1) & f(M,2)\ldots\ldots\ldots f(M,N) \end{matrix}$$

Where f (1,1) = f(0,0) (note the use of a monoscope font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation f(p ,q) denotes the element located in row p and the column q. For example f(6,2) is the element in the sixth row and second column of the matrix f. Typically we use the letters M and N respectively to denote the number of rows and columns in a matrix. A 1xN matrix is called a row vector whereas an Mx1 matrix is called a column vector. A 1x1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using mono-scope characters. We use conventional Roman, italic notation such as f(x ,y), for mathematical expressions

3.2.17 Reading Images:

Images are read into the MATLAB environment using function imread whose syntax " Imread ('filename

Table 3.1 Different Formats Of Images

FORMAT NAME	DESCRIPTION	RECOGNIZED EXTENSION
TIFF	Tagged Image File Format	.tif, .tiff

JPEG	Joint Photographic Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	.xwd

Here filename is a spring containing the complete of the image file(including any applicable extension).For example the command line

>> f = imread ('8. jpg')

Table 3.2 DataTypes

NAME	DESCRIPTION
DOUBLE	Double_precision, floating_point numbers the approximate
uint8	unsigned 8_bit integers in the range [0,255] (1byte per element).
unit16	unsigned 16_bit integers in the range [0, 65535] (2byte per element).
unit32	unsigned 32_bit integers in the range [0, 4294967295](4 bytes per element).
int8	signed 8_bit integers in the range [-128,127] 1 byte per element).

int16	signed 16_byte integers in the range [32768, 32767] (2 bytes per element).
int32	Signed 32_byte integers in the range [-2147483648, 21474833647] (4 byte per element).
Single	single _precision floating _point numbers with values in the approximate range (4 bytes per element).
Char	characters (2 bytes per element).
Logical	values are 0 to 1 (1byte per element).

Reads the JPEG (above table) image chest x ray into image array f. Note the use of single quotes (‘) to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output If a semicolon is not included. MATLAB displays the results of the operation(s) specified in that line. The prompt symbol (>>) designates the beginning of a command line, as it appears in the MATLAB command window.

3.2.18 Data Classes:

Although we work with integers coordinates the values of pixels themselves are not restricted to be integers in MATLAB. Table above list various data classes supported by MATLAB and IPT are representing pixels values. The first eight entries in the table are refers to as numeric data classes. The ninth entry is the char class and, as shown, the last entry is referred to as logical data class.

Images are most common representations found in practice. These two data classes, classes logical, and, to a lesser degree, class unit 16 constitute the primary data classes on which we focus. Many ipt functions however support all the data classes listed in table. Data class double requires 8 bytes to represent a number uint8 and int 8 require one byte each, uint16 and int16 requires 2bytes and unit 32.

Int 32 and single required 4 bytes each. The char data class holds characters in Unicode representation. A character string is merely a 1*n array of characters logical array contains only the values 0 to 1,with each element being stored in memory using function logic or by using relational operators.

3.2.19 Image Types:
The toolbox supports four types of images:

1 .Intensity images;

2. Binary images;

3. Indexed images;

4. R G B images.

Most monochrome image processing operations are carried out using binary or intensity images, so our initial focus is on these two image types. Indexed and RGB color images.

3.2.20 Intensity Images:

An intensity image is a data matrix whose values have been scaled to represent intentions. When the elements of an intensity image are of class unit8, or class unit 16, they have integer values in the range [0,255] and [0, 65535], respectively. If the image is of class double, the values are floating point numbers. Values of scaled, double intensity images are in the range [0, 1] by convention.

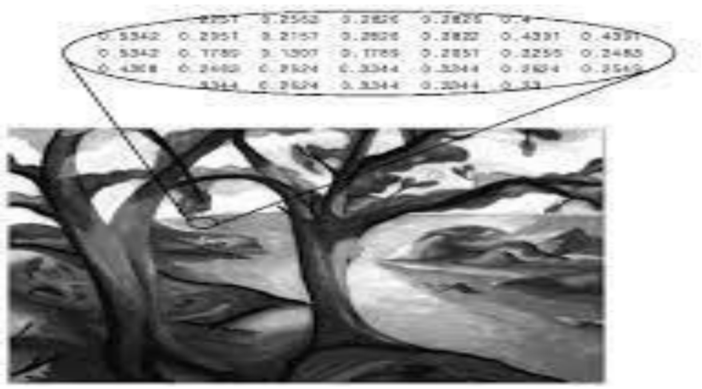


Figure 3.7 Intensity image example

3.2.21 Binary Images:

Binary images have a very specific meaning in MATLAB.A binary image is a logical array 0s and1s.Thus, an array of 0s and 1s whose values are of data class, say unit8, is not considered as a binary image in MATLAB. A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, we create an array B using the statement.

B=logical (A)

If A contains elements other than 0s and 1s.Use of the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.

Using relational and logical operators also creates logical arrays. To test if an array is logical, we use the I logical function: is logical(c).

If c is a logical array, this function returns a 1.Otherwise returns a 0. Logical array can be converted to numeric arrays using the data class conversion functions.

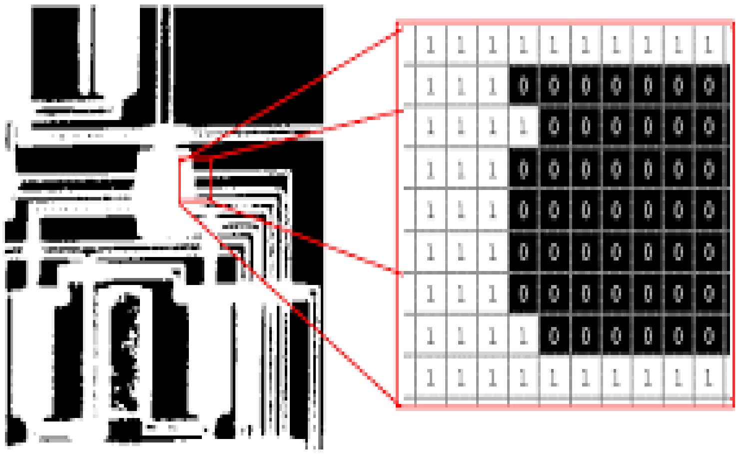


Figure 3.8 Binary image example

3.2.22 Indexed Images:

An indexed image has two components: 1. A data matrix integer, x 2. A color map matrix, map

Matrix map is an m*3 arrays of class double containing floating point values in the range [0, 1].The length m of the map are equal to the number of colors it defines. Each row of map specifies the red, green and blue components of a single color. An indexed images uses “direct mapping” of pixel intensity values color map values. The color of each pixel is determined by using the corresponding value the integer matrix x as a pointer in to map. If x is of class double ,then all of its components with values less than or equal to 1 point to the first row in map, all components with value 2 point to the second row and so on. If x is of class units or unit 16, then all components value 0 point to the first row in map, all components with value 1 point to the second and so on.

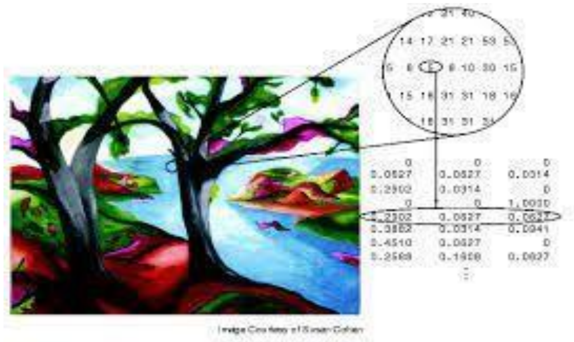


Figure 3.9 Indexed image example

3.2.23 RGB Image:

An RGB color image is an M*N*3 array of color pixels where each color pixel is triplet corresponding to the red, green and blue components of an RGB image, at a specific spatial location. An RGB image may be viewed as “stack” of three gray scale images that when fed in to the red, green and blue inputs of a color monitor

Produce a color image on the screen. Convention the three images forming an RGB color image are referred to as the red, green and blue components images. The data class of the components images determines their range of values. If an RGB image is of class double the range of values is [0, 1].

Similarly the range of values is [0,255] or [0, 65535].For RGB images of class units or unit 16 respectively. The number of bits use to represents the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8bit image, the corresponding RGB image is said to be 24 bits deep. Generally, the number of bits in all component images is the same. In this case the number of possible color in an RGB image is (2^b) ^3, where b is a number of bits in each component image. For the 8bit case the number is 16,777,216 colors.

Figure 3.10 RGB image example

CHAPTER 4: PROJECT IMPLEMENTATION

4.1 IMPLEMENTATION STAGES

4.1.1 Block Diagram:

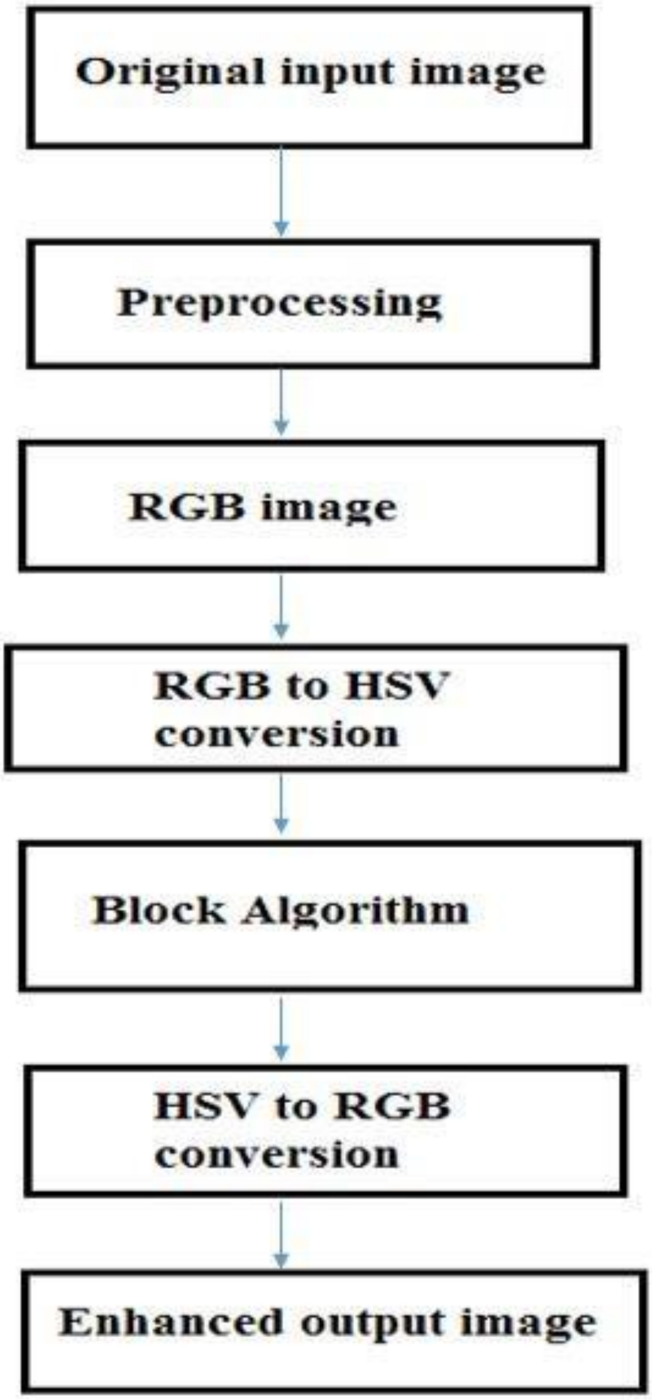


Figure 4.1 Block Diagram

a) **ORIGINAL IMAGE:** In the initial stage we give an original image as an input image, the input image is generally in a distorted condition

b) **PRE-PROCESSING:** This block is responsible for the pre-processes i.e., converting an RGB image into a Grayscale image for easier and effective results and also to make sure that all the images are of the same dimensions.

c) **RGB TO HSV CONVERSION:** Usually, objects in images have distinct colors (hues) and luminosities, so that these features can be used to separate different areas of the image. In the RGB representation the hue and the luminosity are expressed as a linear combination of the R, G, B channels, whereas they correspond to single channels of the HSV image (the Hue and the Value channels).

d) **RGB IMAGE:** An RGB image, sometimes referred to as a true color image, is stored in MATLAB as an m-by-n-by-3 data array that defines red, green, and blue colour components for each individual pixel. RGB images do not use a palette.

e) **BLOCK ALGORITHM:** Application of block algorithm is nothing but dividing the image into required number of blocks (or frames) for effective search.

f) **HSV TO RGB CONVERSION:** RGB = hsv2rgb(HSV) converts the hue, saturation, and value (HSV) values of an HSV image to red, green, and blue values of an RGB image. Rgb map = hsv2rgb(hsv map) converts an HSV colormap to an RGB colormap.

g) **ENHANCED OUTPUT IMAGE:** The final output Image we get is a completely enhanced image

4.1.2 Flow of Program:

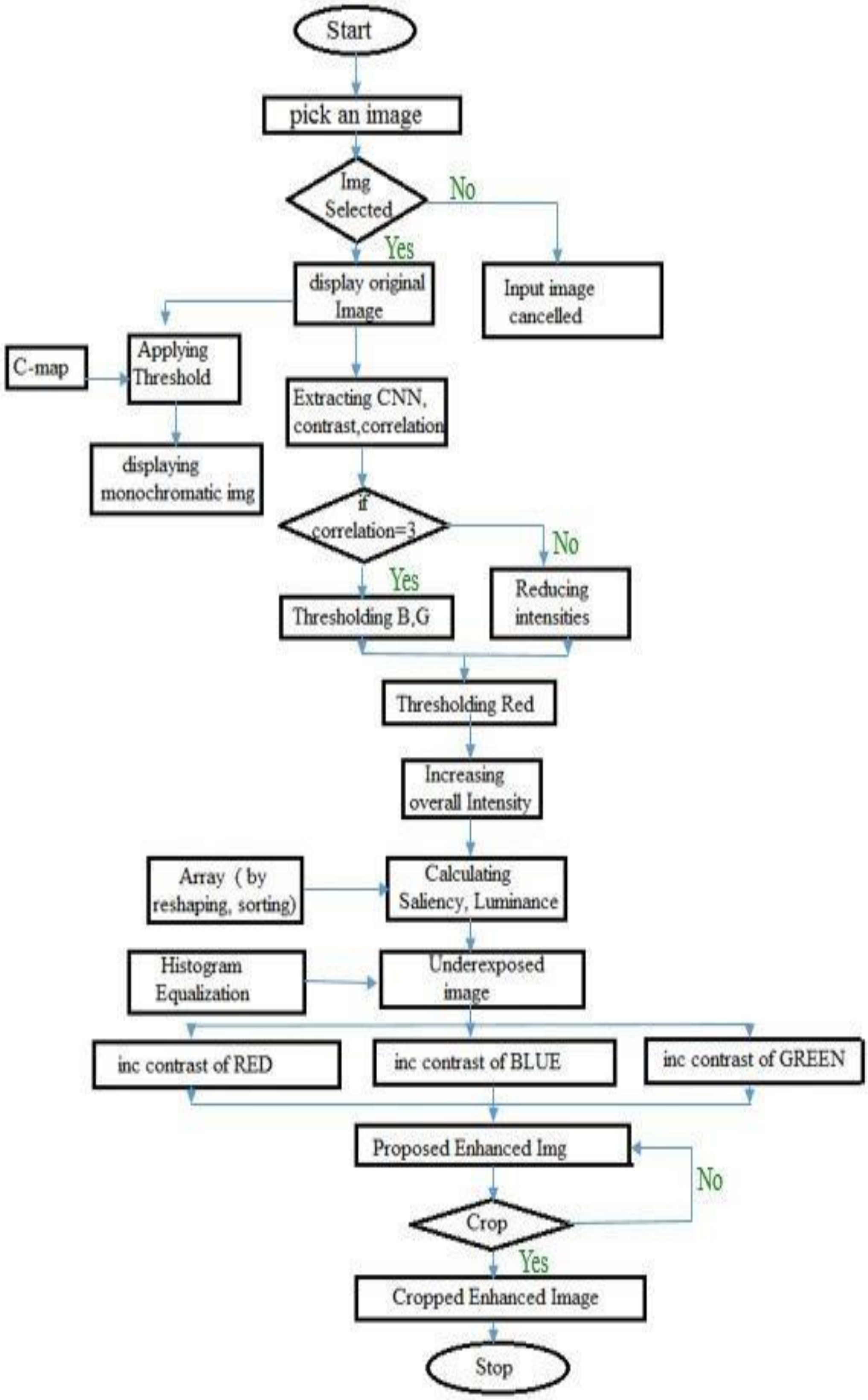


Figure 4.2 Flow Chart

Step1:

Pick an underwater image from the directory. As we have given the condition that, if the filename equal to 0 then no image is taken and it redirects a “help dialogue box” stating “Image input canceled”. If the filename is not equal to 0 then the image is read and stored in the variable.

Step2:

Applying a threshold to show a monochromatic image where monochromatic is nothing but ‘RGB differentiated colors’ where red, blue, green components of the image are separated.

Step3:

Extracting several components from the original image and applying a threshold to only blue and green channels of the image as these channels have higher intensities.

Step4:

Thresholding low-intensity components i.e, red channel of the original image

Step5:

After successfully limiting all the components, we will increase the overall intensity of the original image to get the perfect saliency and luminance values.

Step6:

Calculation of saliency and luminance values to obtain underexposed images.

Step7:

Increasing the contrast of each color using a function called adaptive histogram equalization.

Step8:

Concatenation of all the three enhanced contrast parts to get the “Proposed enhanced underwater image”.

Step9:

By cropping the particular part of the image to be enhanced, we get the "Proposed enhanced cropped underwater image"

4.2 RESULTS

- **Original Underwater frame:**

  When we run the code in MATLAB software it'll ask us to select a certain file that need to be enhanced from the file directory which is basically an underwater image as shown below in figure 4.3

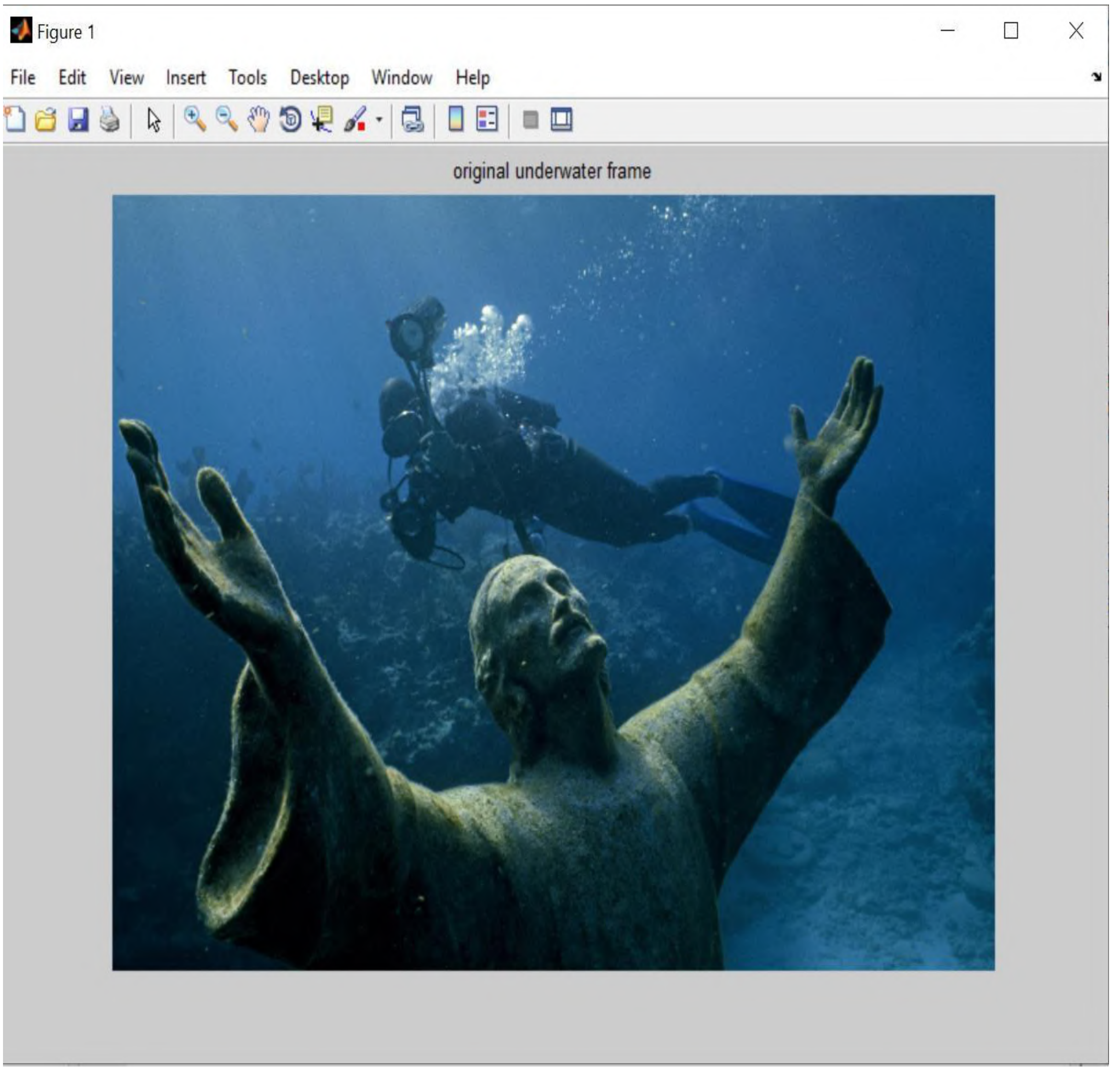


Figure 4.3 Input Underwater Image

- **Channelizing original image using CNN:**

  After applying a threshold to all the 3 channels that's (RED, GREEN, BLUE) of the image and using colormap(c map) we obtain a monochromatic image also known as RGB differentiated colors. The output of this process is shown below

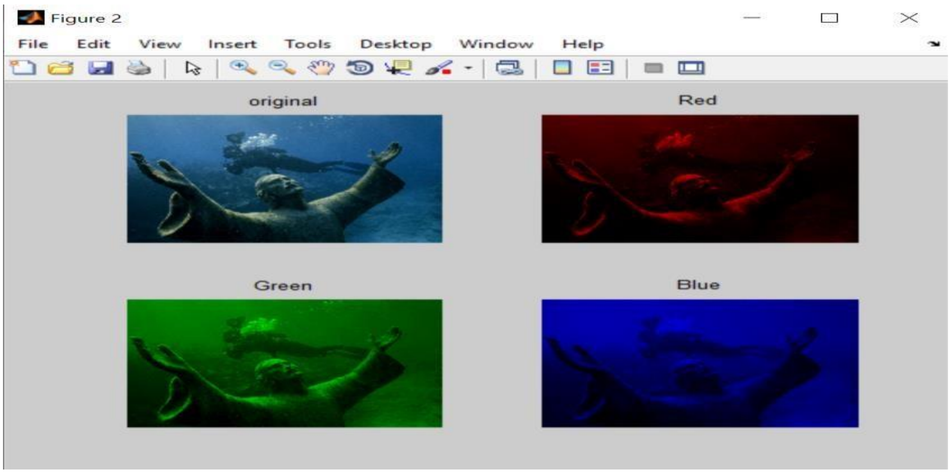


Figure 4.4 Channelized Underwater Image

- **Underexposed Image Enhancement:**

  Figure 4.5 below shows the enhanced underexposed image which is obtained by calculating the saliency and luminance values for the channelized underwater image

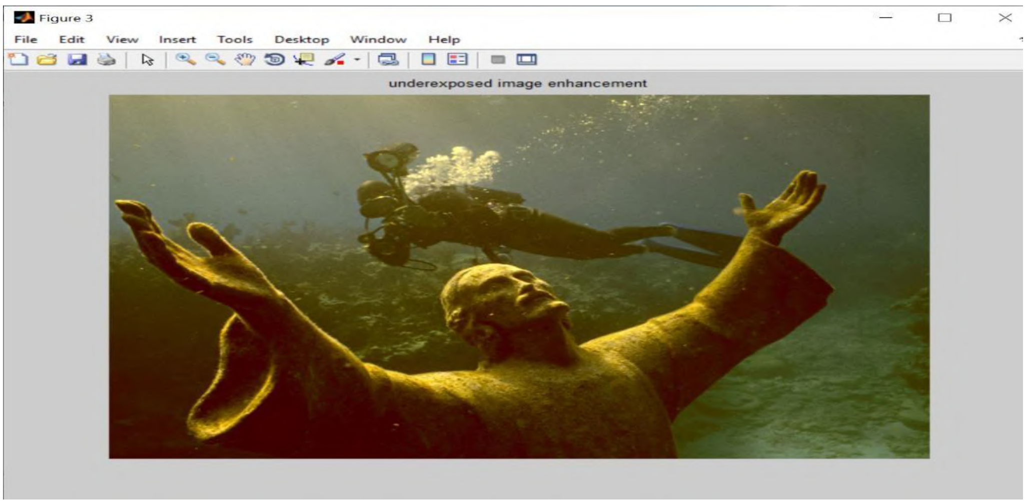


Figure 4.5 Underexposed Image Enhancement

- **Proposed Enhanced Image:** After Increasing the contrast of all the 3 channels using Histogram Equalization and by the concatenation of all the 3 channels, we obtain the figure below

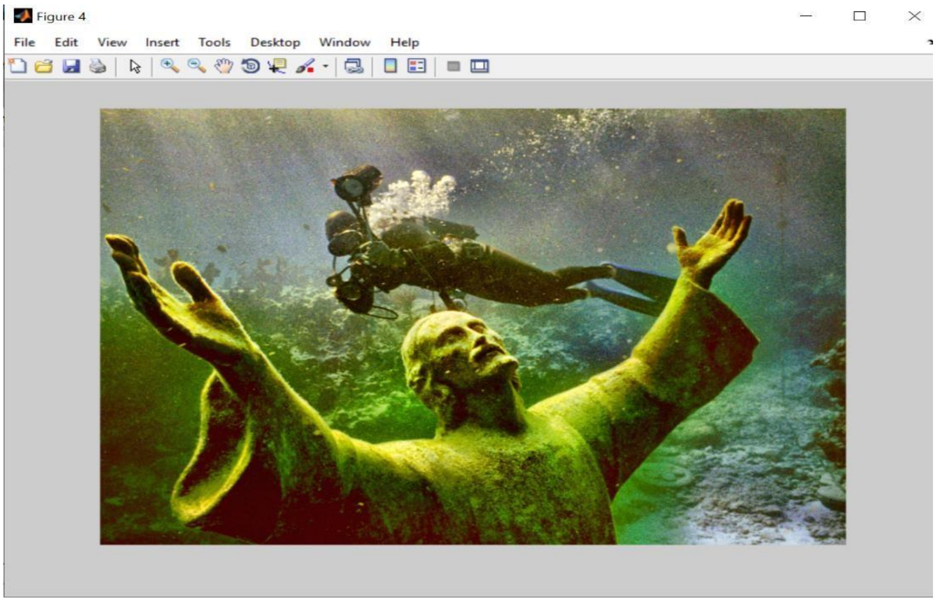


Figure 4.6 Proposed Enhanced Image

- **Particular Part of an Image:** In addition to this proposed method, we have an additional feature where we can crop a desired part of the image and we'll get an enhanced version of that cropped out part as shown in the above figure

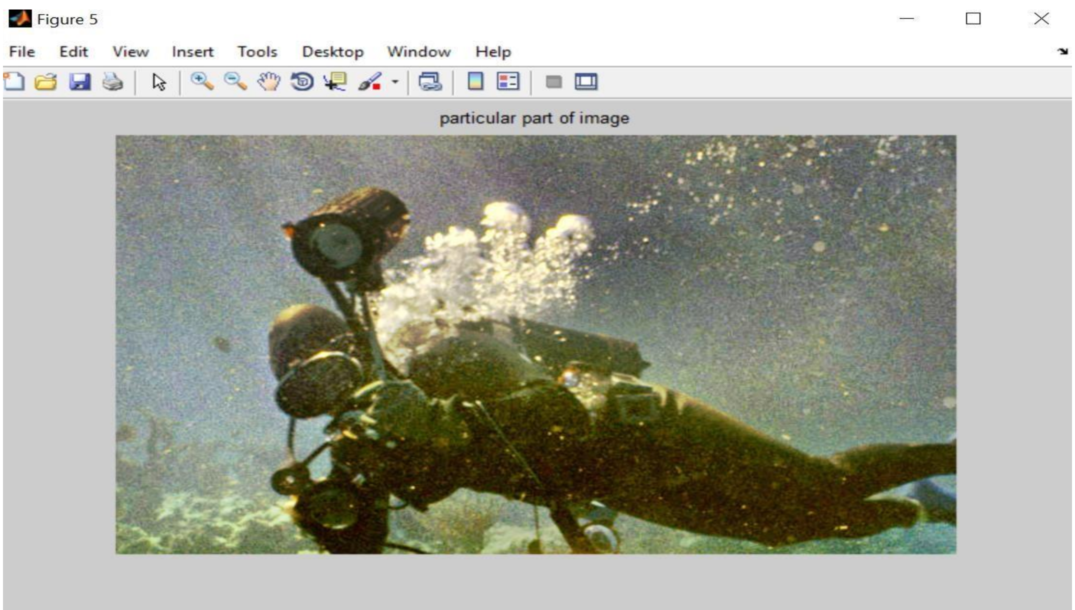


Figure 4.7 Particular Part of an Image

CHAPTER 5: CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

In this paper, we propose a CNN framework for underwater image color restoration. Our method converts an input underwater image to the grayscale image by using CNN to estimate the grayscale coefficients. We also use CNN to restore the color of the input underwater image. To remove the noise in the underwater environment and reserve the detail that may affect by CNN, we propose a grayscale detail enhancement CNN. At last, we integrate all works by the hue preservation enhancement. Our experimental results show that the proposed method achieves better underwater image restoration performance than other methods.

5.1.1 APPLICATIONS:

- This technology can be used to verify the authenticity of an image. It can be used in situations where images are legal evidence and also can be used in forensic investigations.
- It's used extensively in medical imaging, astronomical imaging, and forensic science
- It's ergonomically used in Underwater imaging especially in conditions where the visibility level is low and the water is denser

5.2 FUTURE SCOPE

- ➢ It also may concentrate on recognizing profoundly compounded pictures. This method can further be developed for recognizing phony videos for investigation
- ➢ Better and robust algorithms which are camera independent.
- ➢ These algorithms, when incorporated into these robots will not only aid in procuring better enhanced underwater exploration images but also in bringing down the cost of these robots significantly.

PUBLICATION

ic smec ece

to me

Mon, May 24, 3:43 PM

Dear Author(s),

Warm Greetings from St.Martins Engineering College.

With heartiest congratulations we are pleased to inform you that based on the recommendations of the expert reviewers, your paper has been accepted for journal publication and oral presentation in the online mega International Conference on "Smart Modernistic in Electronics and Communication" (ICSMEC-21).

Paper ID:ICSMEC21-0013

Paper Title: Deep Learning Underwater Image Colour Correction and Contrast Enhancement Based on Hue Preservation

Paper Accepted.

We have converted our project into paper with the title "**Deep learning underwater Image Colour Correction and Contrast Enhancement Based on Hue Preservation**" in the online mega International on "**Smart Modernistic in Electronics and Communication"(ICSMEC-21)** and it has been accepted with reference number ICSMEC21-0013

REFERENCES

[1] C. Li, J. Quo, Y. Pang, S. Chen & J. Wang, "Single underwater image restoration by blue-green channels dehazing and red channel correction," In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1731-1735, 2016.

[2] D. Berman, T. Treibitz & S. Avidan, "Diving into hazelines: Color restoration of underwater images," In Proc. British Machine Vision Conference (BMVC), vol. 1, no. 2, 2017.

[3] J. Li, K. A. Skinner, R. M. Eustice & M. Johnson-Roberson, "WaterGAN: Unsupervised generative network to enable real-time color correction of monocular underwater images," IEEE Robotics and Automation Letters, vol. 3, no. 1, pp. 387-394, 2018.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair & Y. Bengio, "Generative adversarial nets," In Advances in neural information processing systems, pp. 2672-2680, 2014

[5] V. Nair & G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," Proc. Int. Conf. Machine Learning, Haifa, Israel, June 2010, pp. 807–814.

[6] K. He, X. Zhang, S. Ren & J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," In Proceedings of the IEEE international conference on computer vision, pp. 1026-1034, 2015.

[7] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens & Z. Wojna, "Rethinking the inception architecture for computer vision," In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818-2826, 2016.

[8] H. Zhao, O. Gallo, I. Frosio & J. Kautz, "Loss functions for image restoration with neural networks," IEEE Transactions on Computational Imaging, vol. 3, no. 1, pp. 47-57, 2017.

[9] J. Y. Zhu, T. Park, P. Isola & A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” In Proceedings of the IEEE International Conference on Computer Vision, pp. 2223-2232, 2017.

[10] DALLMYD youtube channel, https://www.youtube.com/user/DALLMYD.

[11] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The Cityscapes Dataset for Semantic Urban Scene Understanding,” in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016

[12] S. Wang, K. Ma, H. Yeganeh, Z. Wang & W. Lin, “A patch-structure representation method for quality assessment of contrast changed images,” IEEE Signal Processing Letters, vol. 22, no. 12, pp. 2387- 2390, 2015. [13] K. Panetta, C. Gao & S. Agaian, “Human-visual-system-inspired underwater image quality measures,” IEEE Journal of Oceanic Engineering, vol. 41, no. 3, pp. 541-551, 2016.

APPENDICES

```
%

disp('Deep Learning Underwater Image Color Correctionand Contrast Enhancement Based on Hue Preservation')
clc;
close all;
clear all;
warning off
%%%%%under water image enhncement

  %%%%%%%%%%%%%%%%%%                    Image                    Aquisition
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[filename, pathname] = uigetfile({'*.jpg';'*.png'}, 'pick an image');
if isequal(filename, 0) || isequal(pathname, 0)
  helpdlg('Image input canceled.');

else
  X=imread(fullfile(pathname, filename));
end

figure,imshow(X);title('original underwater frame');
%%%%%%%%%%%%%%devide color space images
 N = 256;
  A = im2uint8(X);
  fig = figure;

  subplot(2,2,1);
  imshow(A);
  title('original');
  ColorList = { 'Red' 'Green' 'Blue' };
  gr = 0:1/(N-1):1;

  for k = 1:3

    % color map:
    cMap = zeros(N,3);
    cMap(:,k) = gr;

    % Display monochromatic image:
    subplot(2,2,k+1);
    imshow(ind2rgb(A(:,:,k),cMap));
    title(ColorList{k});

  end
```

```
%%% deep learning based Image Color Correction and Contrast
normal_thr_limit=0.5;
low_limt=0.002;
up_limit=0.999;
%.........................................................................................
under_image=X;
[CONTRAST CNN Correction]=size(X);
%.........................................................................................
if Correction==3
   inc_pixel_limit=0.04;dec_pixel_limt=-0.04;
   max_chromatic=rgb2ntsc(under_image); % analysis of grouping pixels data
   mean_adjustment=inc_pixel_limit-mean(mean(max_chromatic(:,:,2)));
   max_chromatic(:,:,2)=max_chromatic(:,:,2)+mean_adjustment*(0.989-
max_chromatic(:,:,2));
   mean_adjustment=dec_pixel_limt-mean(mean(max_chromatic(:,:,3)));
   max_chromatic(:,:,3)=max_chromatic(:,:,3)+mean_adjustment*(0.898-
max_chromatic(:,:,3));
else
   max_chromatic=double(under_image)./255;
end
%.........................................................................................
mean_adjustment=normal_thr_limit-mean(mean(max_chromatic(:,:,1)));
max_chromatic(:,:,1)=max_chromatic(:,:,1)+mean_adjustment*(0.958-max_chromatic(:,:,1));
if Correction==3
   max_chromatic=ntsc2rgb(max_chromatic);
end
%.........................................................................................
under_image=max_chromatic.*255;
%--------------------caliculate the max to min pixels---------------------
for k=1:Correction
   arr=sort(reshape(under_image(:,:,k),CONTRAST*CNN,1));
   saliency_min(k)=arr(ceil(low_limt*CONTRAST*CNN));
   luminance_max(k)=arr(ceil(up_limit*CONTRAST*CNN));
end
%.........................................................................................
if Correction==3
   saliency_min=rgb2ntsc(saliency_min);
   luminance_max=rgb2ntsc(luminance_max);
end
%.........................................................................................
under_image=(under_image-saliency_min(1))/(luminance_max(1)-saliency_min(1));
figure,imshow(under_image);title('underexposed image enhancement');


% % %%%%%%%%%%%%%%%temporal correlation pixels separation
red_color_correlation = adapthisteq(under_image(:,:,1));
         green_adapthisteq_green = adapthisteq(under_image(:,:,2));
         blue_adapthisteq_blue = adapthisteq(under_image(:,:,3));
         fusion_enhanced                                                      =
cat(3,red_color_correlation,green_adapthisteq_green,blue_adapthisteq_blue);
```

```
figure,imshow(mat2gray(fusion_enhanced));title('proposed enhanced output image');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
END%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% single part extraction

single_partr=imcrop(fusion_enhanced);
    figure,imshow(mat2gray(single_partr));title('particular part of image');
```

A

PROJECT REPORT

On

IOT COVID PATIENT HEALTH MONITOR IN QUARANTINE

Submitted by

1) Ms. V.V.S Naveena (17K81A0458)
2) Ms. J. Sahithya (17K81A0421)
3) Ms. K. Bhavana (17K81A0424)

in partial fulfillment for the award of the

degreeof

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

B.Shiny Sucharitha

M.Tech,

Assisant professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

<Fo

ST.MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

St.MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the project entitled entitled **IOT COVID PATIENT HEALTH MONITOR IN QUARANTINE**, is being submitted by **1.Ms. V.V.S Naveena (17K81A0458), 2.Ms. J.Sahithya (17K81A0421), 3.Ms.K.Bhavana (17K81A0424)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

< Signature>
B.Shiny Sucharitha
Department of ECE

Head of the Department
Dr. B.HARI KRISHNA
Department of ECE

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **V.VS NAVEENA** WITH ROLL NO.**17K81A0458, J. SAHITHYA** WITH ROLL NO.**17K81A0421**, **K.BHAVANA** WITH ROLL NO.**17K81A0424**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED **"IOT COVID PATIENT HEALTH MONITOR IN QUARANTINE"** AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. ★ Kompally-500014 ★

ORUGANTI VENKAT

DIRECTOR
TRAININGS & PLACEMENTS LASYA IT SOLUTIONS
PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of **Bachelor of Technology** in Department of Electronics and Communication Engineering', session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled **'IOT COVID PATIENT HEALTH MONITOR IN QUARANTINE '** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

V.V.S.Naveena (17K81A0458)
J.Sahithya (17k81A0421)
K.Bhavana (17K81A0424)

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would b incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna**, **Head &Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our Dept. Project coordinator Dr.A.Anand** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide, **Ms. B. Shiny Sucharitha, Assistant Professor**, Department of Electronics and Communication Engineering and our class project coordinator **Mrs. A. Soumya,** Department of ECE for guidance and encouragement in carrying out the project work.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. V.V.S.Naveena
2. J.Sahithya
3. K.Bhavana

ABSTRACT

In times of COVID we have special Covid 19 Quarantine centres setup in order to treat covid patients. Since covid is highly infectious it is very important to quarantine covid patients but at the same time doctors need to monitor health of covid patients too. With the increasing number of cases, it is becoming difficult to keep a track on the health conditions of s many quarantined patients.

The problems here are:

• Doctors need to regularly monitor patient health.

• There is increasing number of patients for the doctors to monitor.

• The doctors are at risk of infection just for monitoring purpose.

To solve this issue, we here design a remote IOT based health monitor system that allows for remotely monitoring of multiple covid patients over the internet. The system monitors patient heartbeat, temperature and blood pressure using a heartbeat sensor, temperature sensor respectively.

The system then transmits this data over the internet using wi-fi transmission by connecting to Wi-Fi internet connection. The data is transmitted and received over IOT-by-IOT Gecko platform to display data of patient remotely. The entire system is run by a microcontroller-based circuitry. If any anomaly is detected in patient health pr if the patient presses the emergency help button on IOT device, an alert is sent over IOT remotely.

This System allows:

• Doctors to monitor patients remotely without risk of infection

• A single doctor over 500 patients at a time.

• Doctor gets instant alert in case of health fluctuations of emergency.

The system is mounted at patient bedside and constantly transmits patient health data over the internet so that doctors can monitor multiple patients remotely and attend the desired patient urgently when needed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY OF COMMONLY USED TERMS

IOT: The Internet of Things (IoT) describes the network of physical objects—"things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticated industrial tools.

Covid-19: Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus.Most people infected with the COVID-19 virus will experience mild to moderate respiratory illness and recover without requiring special treatment. Older people, and those with underlying medical problems like cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to develop serious illness.

Sensors: A sensor is a device that detects the change in the environment and responds to some output on the other system. A sensor converts a physical phenomenon into a measurable analog voltage (or sometimes a digital signal) converted into a human-readable display or transmitted for reading or further processing.

Wi-Fi Module: The ESP8266 Wi-Fi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

CHAPTER-1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

Health is always a major concern in every growth the human race is advancing in terms of technology. Like the recent corona virus attack that has ruined the economy of China to an extent is an example how health care has become of major importance. In such areas where the epidemic is spread, it is always a better idea to monitor these patients using remote health monitoring technology. So, Internet of Things (IoT) based health monitoring system is the current solution for it [1]. Remote Patient Monitoring arrangement empowers observation of patients outside of customary clinical settings (e.g., at home), which expands access to human services offices at bring down expenses [2].

The core objective of this project is the design and implementation of a smart patient health tracking system that uses Sensors to track patient health and uses internet to inform their loved ones in case of any issues. The objective of developing monitoring systems is to reduce health care costs by reducingSMS based patient flourishing viewing and IOT based patient checking framework. In IOT based framework, subtle parts of the patient flourishing can be seen by different clients [4]. The explanation behind this is the information should be checked by passing by a site or URL.

While, in GSM based patient viewing, the flourishing parameters are sent utilizing GSM by strategies for SMS. In most of the rural areas, the medical facility would not be in a hand reach distance for the natives [5]. So normally the peoplephysician office visits, hospitalizations, and diagnostic testing procedure [3]. Each of our bodies utilizes temperature and also pulse acknowledging perusing understanding wellbeing. The sensors are linked to a microcontroller to track the status which is thus interfaced to an LCD screen and additionally remote association with have the capacity to exchange alarms.

If framework finds any sudden changes in understanding heart beat or body temperature, the framework consequently alarms the client about the patient's status over IOT and furthermore indicates subtle elements of pulse and temperature of patient live in the web. In this manner IOT set up tolerant wellbeing following framework viably utilizes web to screen quiet wellbeing measurements and spare persists time. There is a significant capability betweenneglect any kind of minor health issues which is shown in early stages by variation of vital elements like body temperature, heartbeat rate etc.

1.2 OBJECTIVES OF THE STUDY

The main objective of the this project is to design and implement an efficient monitoring system through which the required parameters are monitored remotely using internet and the data gathered from the sensors are stored in the cloud and to project the estimated trend on the web browser. Monitoring and control of infection diseases, including Covid-19, have attracted the researchers, to adopt many solutions to control different types of infections such as using health monitoring mobile applications. A mobile based android system for infectious diseases, the system is limited but do not cover the cough system detection and temperature, although the device is not wearable. A PCB Board has been used to enable portable, reconfigurable, multichannel amperometry data acquisition to measure the current-input signals from biosensor. The electronic read-out circuit can provide constant biasing voltages to the amperometry sensor. However, there is no scope for data analysis or data storing. The work has focused on finding, analysing the geographic location of the patients without providing health information system. The paper in is good proposal to send infection index information for analysis, but again, no hardware part application. Mobile phone enabled social community extraction for controlling of disease propagation in healthcare is designed and tested to follow the disease, in other hand, the project is mainly focus on following the propagation pattern only through mobiles and without giving any health data analysis. Modelling and evaluation of disease propagation factors has been studied theoretically on, but still there is no real time implementation. In addition to the previous discussion, countries have been implemented drones, Mobile phones, and thermal cameras to follow infections; however, this can be used only for on sport checking. That is representing a temporary solution as they had no physical contact. Recently a wearable IOT design framework has been suggested on which can be considered as a good technical background for the proposed system implementations if it it’s implemented on its modified version.

1.3 SCOPE OF THE STUDY

In recent years, the healthcare industry has shown rapid growth and has been a major contributor to revenue and employment .A few years ago, the diagnosis of diseases and abnormality in the human body was only being possible after having a physical analysis in the hospital. Most of the patients had to stay in the hospital throughout their treatment period. This resulted in an increased healthcare cost and also strained the healthcare facility at rural and remote locations. The technological advancement that has been achieved through these years has now allowed the diagnosis of various diseases and health monitoring using miniaturized devices like smartwatches. Moreover, technology has transformed a hospital-centric healthcare system into a patient-centric system .For example, several clinical analyses (such as measuring blood pressure, blood glucose level, pO2, level, and so on) can be performed at home without the help of a healthcare professional. Further, the clinical data can be communicated to healthcare centers from remote areas with the help of advanced telecommunication services. The use of such communication services in conjunction with the rapidly growing technologies (e.g., machine learning, big data analysis, Internet of things (IoT), wireless sensing, mobile computing, and cloud computing) has improved the accessibility of the healthcare facilities.

1.4 MATERIAL REQUIREMENT

1.4.1 HARDWARE REQUIREMENT:

1. Arduino UNO
2. Lcd
3. Power Supply Unit
4. LM35
5. Heart beat Sensor
6. Wi-Fi Module
7. UV Lights
8. Slide Switch

1. Arduino UNO

Arduino is open source software. Arduino Uno is a microcontroller, ATmega328Pcontain in it. It has 14 digital i/o pins and 6 analog input pin, a 16 MHz quartz crystal, a USB connection, a power jack and a reset button. Simply connect to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. Embedded C program is used for coding.

2. LCD

The LCD is used to display whenever there is a gas leakage or fire detected in the area. Whenever there is a gas leakage the Arduino that is connected to the sensor and LCD sends a message to LCD to display ‘Gas leakage’. When there is fire outbreak the Arduino again sends a message to LCD to display ‘Fire Detected’ notifying us if there is a gas leakage or fire detected in the vicinity.

3. Power Supply Unit

Every electrical and electronic device that we use in our day-to-day life will require a power supply. In general, we use an AC supply of 230V 50Hz, but this power has to be changed into the required form with required values or voltage range for providing power supply to different types of devices. There are various types of power electronic converters such as step-down converter, step-up converter, voltage 31 stabilizer, AC to DC converter, DC 25 to DC converter, DC to AC converter, and so on. For example, consider the microcontrollers that are used frequently for developing many embedded systems based projects and kits used in real-time applications. These microcontrollers require a 5V DC supply.

4. LM35

LM35 is a precession Integrated circuit Temperature sensor, whose output voltage varies, based on the temperature around it. It is a small and cheap IC which can be used to measure temperature anywhere between -55°C to 150°C. There will be rise of 0.01V (10mV) for every degree Celsius rise in temperature.

5. Heart beat Sensor

An optical heart rate sensor measures pulse waves, which are changes in the volume of a blood vessel that occur when the heart pumps blood. Pulse waves are detected by measuring the change in volume using an optical sensor and green LED.

6. ESP-01 (Wi-Fi Module)

The ESP8266 Wi-Fi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much Wi-Fi-ability as a Wi-Fi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely costeffective board with a huge, and ever growing, community

7. UV Lights

UV light (ultraviolet light) is having a wavelength between 10 and 400 nm. UV light is used as disinfection i.e., kills virus and bacteria.

8. Slide switch

Slide switches as the name suggest are mechanical devices which have a slider on the top of switch which slides from one position to another. These switches have a slider that moves (slides) from one position to another. Based on the actuator type, the handle can be either flush or raised.

1.4.2 SOFTWARE REQUIREMENT:

1. **ARDUINO IDE**: Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs.

2.**PROTEUS SOFTWARE:** Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.It is a proprietry software tool suite used primarily for electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

1.5 PROCUMENT OF EQUIPMENT

We brought all the required hardware components in an electronic manufacturing shop and before assembling and interfacing hardware components to Arduino UNO micro-controller, we connected individual sensors to Arduino in Proteus Software and understood working of every individual sensor in Proteus Software. Now, we assembled all sensors and interfaced to Arduino UNO micro-controller using connecting wires through bread board and we wrote program to work all the hardware sensors in Arduino IDE, we understood the working of all sensors in a practical way. We measured al the heartbeat and temperature values of patient and all the information is sent to the online server.

.

CHAPTER-2

LITERATRE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH AREA:

There is considerable work in the literature regarding the use of the Internet of Things (IoT) to deliver health services. Usak et al. conducted a systematic literature review of the use of IoT in health care systems. That work also included a discussion of the main challenges of using IoT to deliver health services, and a classification of the reviewed work in the literature. Wu et al. proposed a hybrid IoT safety and health monitoring system. The goal was to improve outdoor safety. The system consists of two layers: one is used to collect user data, and the other to aggregate the collected data over the Internet. Wearable devices were used to collect safety indicators from the surrounding environment, and health signs from the user. Hamidi studied authentication of IoT smart health data to ensure privacy and security of health information. The work proposed a biometric-based authentication technology. Rath and Pattanayak proposed a smart healthcare hospital in urban areas using IoT devices, inspired by the literature. Issues such as safety, security and timely treatment of patients in VANET zone were discussed. Evaluation of the proposed system was conducted using simulators such as NS2 and Net Sim. Darwish et al. proposed a Cloud IoT-Health paradigm, which integrates cloud computing with IoT in the health area, based on the relevant literature. The paper presented the challenges of integration, as well as new trends in Cloud IoT-Health. These challenges are classified at three levels: technology, communication and networking, and intelligence.Zhong and Li studied the monitoring of college students during their physical activities. The paper focused on a Physical Activity Recognition and Monitoring (PARM) model, which involves data pre-processing. Several classifiers, such as decision tree, neural networks, and SVM, were tested and discussed. Din and Paul proposed an IoT-based smart health monitoring and management architecture. The architecture is composed of three layers: (1) data generation from battery-operated medical sensors and processing, (2) Hadoop processing, and (3) application layers. Because of the limited capacity of batteries to power the sensors, the work employed an energy-harvesting approach using piezoelectric devices attached to the human body. Otoom et al. developed an IoT-based prototype for real-time blood sugar control. ARIMA and Markov-based statistical models were used to determine the appropriate insulin dose. Alshraideh et al. proposed an

IoT-based system for Cardiovascular Disease detection. Several machine learning algorithms were used for CVD detection.

Nguyen presented a survey of Artificial intelligence (AI) methods being used in the research of COVID-19. This work classified these methods into several categories, including the use of IoT. Maghdid proposed the use of sensors available on smartphones to collect health data, such as temperature. Rao and Vazquez proposed the use of machine learning algorithms to identify possible COVID-19 cases. The learning is done on collected data from the user through web survey accessed from smartphones. Allam and Jones discussed the need to develop standard protocols to share information between smart cities in pandemics, motivated by the outbreak of COVID-19. For instance, AI methods can be applied to data collected from thermal cameras installed in smart cities, to identify possible COVID-19 cases. Fatima et al. proposed an IoT-based approach to identify coronavirus cases. The approach is based on a fuzzy inference system. Peeri et al. conducted a comparison between MERS, SARS, and COVID-19, using the available literature. They suggested the use of IoT in mapping the spread of the infection.To our knowledge, no one has developed a complete framework for using IoT technology for the identification and monitoring of COVID-19.

2.2 REVIEW ON RELATED LITERATURE:

On December 31, 2019, the China Health Authority alerted the World Health Organization (WHO) to several cases of pneumonia of unknown aetiology in Wuhan City in Hubei Province in central China. The cases had been reported since December 8, 2019, and many patients worked at or lived around the local Huanan Seafood Wholesale Market although other early cases had no exposure to this market . On January 7, a novel coronavirus, originally abbreviated as 2019-nCoV by WHO, was identified from the throat swab sample of a patient . This pathogen was later renamed as severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) by the Coronavirus Study Group [3] and the diseasewas named coronavirus disease 2019 (COVID-19) by theWHO. As of January 30, 7736 confirmed and 12,167 suspected cases had been reported in China and 82 confirmed cases had been detected in 18 other countries . In the same day, WHO declared the SARS-CoV-2 outbreak as a Public Health Emergency of International Concern (PHEIC)According to the National Health Commission of China, the mortality rate among confirmed cased in China was 2.1% as of February 4 and the mortality rate was 0.2% among cases outside China . Among patients admitted to hospitals, the mortality rate ranged between 11% and 15% . COVID-19 is moderately infectious with a relatively high mortality rate, but the information available in public reports and published literature is rapidly increasing. The aim of this review is to summarize the current understanding of COVID-19 including causative agent, pathogenesis of the disease, diagnosis and treatment of the cases, as well as control and prevention strategies.

2.3 CONCLUSION ON REVIEWS:

Hence as the technology is advancing day by day and we have to follow the new technologies so in the proposed system we used internet of things for makings things smarter. Here in the proposed system it consists of two units – in the first unit , we connect all the sensors to the Arduino Uno that the sensors sense the heartbeat and temperature of patient and sends to Arduino Uno. In the second unit, this transmitted information is displayed on the lcd and simultaneously sends to the online server. Later UV LIGHT is switched “ON” to kill the viruses.

CHAPTER-3

PROJECT DESIGN

3.1 OVERVIEW OF THE DESIGN:

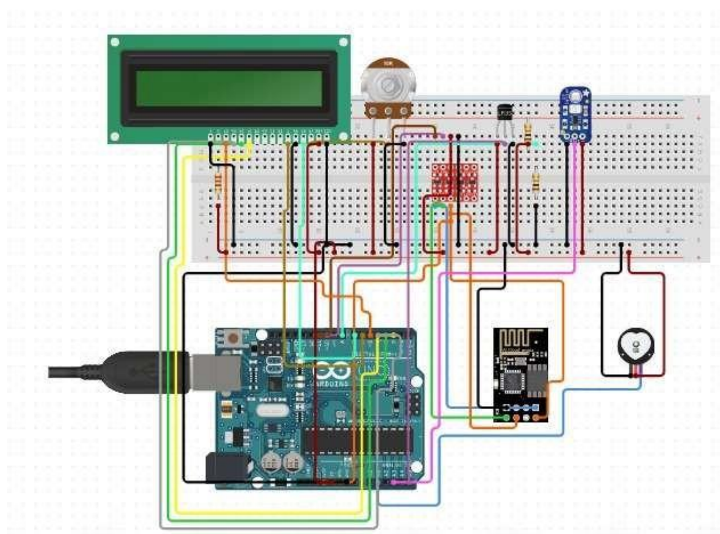


Fig 3.1 Overview of the Design

Arduino UNO has 6 analog pins from A0 to A6 which are generally input pins. It also has 14 digital pins from 0-13 which are the output pins. The analog pin A4 is connected to the temperature sensor, heartbeat sensor at pin 4 and buzzer at pin 7. The digital pins 8-13 are connected to the LCD to display the output values. Here pin 2 and 3 is used for serial communication. These pins are used for the esp8266 wi-fi module to transmit the information to the online server.

3.2 EQUIPMENT ANALYSIS:

3.2.1 ARDUINO:

3.2.1.1 INTRODUCTION

It is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

- The ATmega8 microcontroller contains 32 general purpose working registers. As shown in the below figure these registers are directly connected to ALU. Two registers can carry one single instruction consequently in one clock cycle.
- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

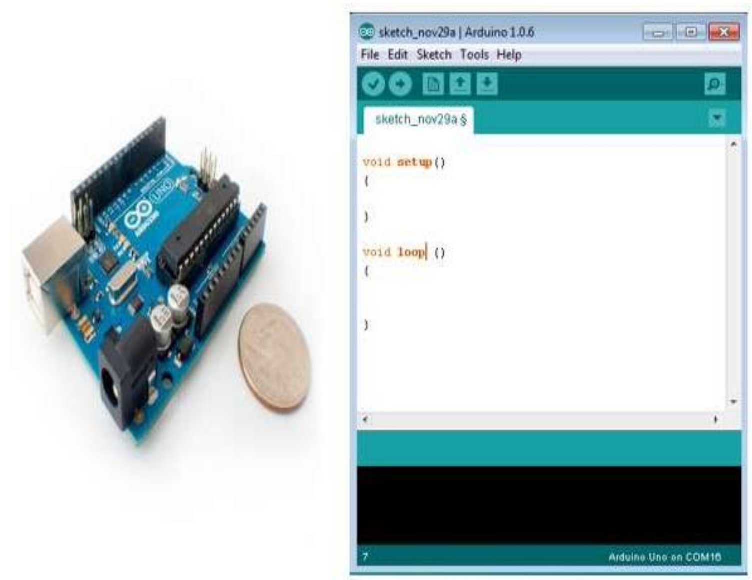


Fig.3.2 Arduino UNO

3.2.1.2 BOARD TYPES

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE. The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V. Here is a list of different Arduino boards available.

Table 3.1 Arduino Boards based on ATMEGA328 Microcontroller

Board Name	Operating Volt	lock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming nterface

Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header

Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.2 Arduino Boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 3.3 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 3.4 Arduino Boards based on ATMEGA2560 Microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table 3.5 Arduino Boards based on AT91SAM3X8E microcontroller

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit.

	• Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

3.2.1.3 Board Description

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most of them remain the same.

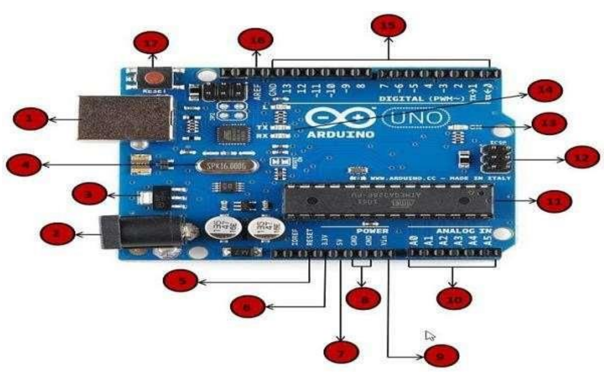


Fig.3.3 Arduino UNO Pin Description

Table 3.6 Arduino UNO Pin Description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).

3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.

13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

3.3.2 Liquid Crystal Display

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCDs connected to the contollers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Description Of 16x2:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all-Parallel Ports. It however doesn't show the use of the Status Port as an input. So what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

Schematic Diagram

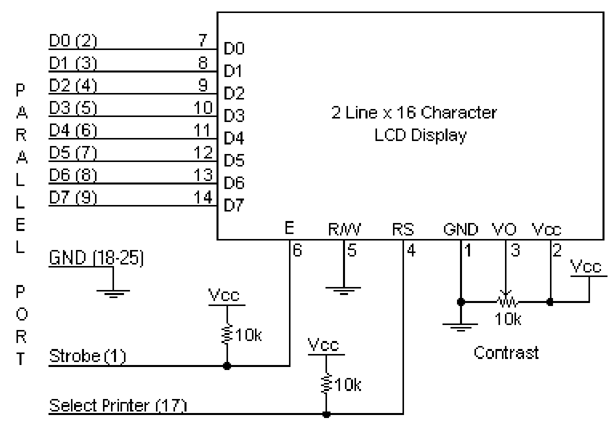


Fig: 3.4 Schematic Diagram

- Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.
- We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.
- The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.

16 x 2 Alphanumeric LCD Module Features:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

Table: 3.7 16 x 2 Alphanumeric LCD Module Specifications:

Pin	Symbol	Level	Function
1	V_{SS}	-	Power, GND
2	V_{DD}	-	Power, 5V
3	Vo	-	Power, for LCD Drive
4	RS	H/L	Register Select Signal H: Data Input L: Instruction Input
5	R/W	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)
6	E	H,H->L	Enable
7-14	DB0-DB7	H/L	Data Bus; Software selectable 4- or 8-bit mode
15	NC	-	**NOT CONNECTED**
16	NC	-	**NOT CONNECTED**

FEATURES:

• 5 x 8 dots with cursor

• Built-in controller (KS 0066 or Equivalent)

• + 5V power supply (Also available for + 3V)

• 1/16 duty cycle

• B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)

• N.V. optional for + 3V power supply

Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are:

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

Figure Address locations for a 1x16 line LCD

Even limited to character based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8, 16,20,24,32 and 40 characters are all standard, in one, two and four line versions..

PIN DESCRIPTION:

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections).

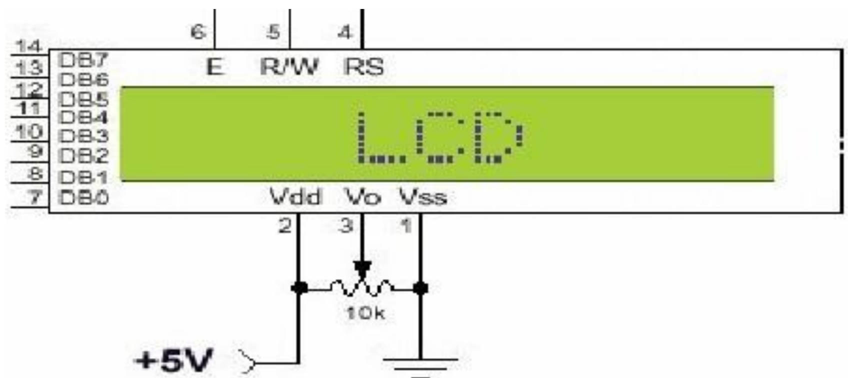


Figure 3.5 Pin diagram of 1x16 lines LCD

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Table 3.8 Pin specifications

CONTROL LINES:

EN: Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS: Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW: Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands, so RW will almost always be low.Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Logic status on control lines:

• E - 0 Access to LCD disabled

- 1 Access to LCD enabled

• R/W - 0 Writing data to LCD

- 1 Reading data from LCD

• RS - 0 Instructions

- 1 Character

Writing data to the LCD:

1) Set R/W bit to low

2) Set RS bit to logic 0 or 1 (instruction or character)

3) Set data to data lines (if it is writing)

4) Set E line to high

5) Set E line to low

Read data from data lines (if it is reading) on LCD:

1) Set R/W bit to high

2) Set RS bit to logic 0 or 1 (instruction or character)

3) Set data to data lines (if it is writing)

4) Set E line to high

5) Set E line to low

Entering Text:

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the d.i.l. switch pack with a of re-wiring is necessary.

LCD Commands:

There are some present commands instructions in LCD, which we need to send to LCD through some microcontroller. Some important command instructions are given below:

Table 3.9 Commands

Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left

0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

3.3.3 TEMPERATURE SENSOR (LM35)

General Description:

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in °Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1/4°C at room temperature and ±3/4°Cover a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package (Fig 3.2.1).

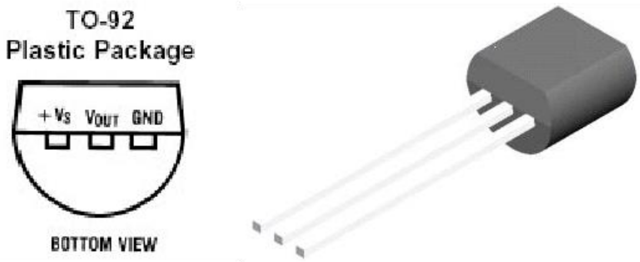
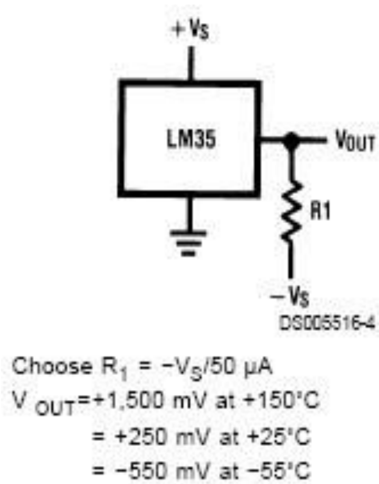


Fig: 3.6 Bottom View

Applications:

The LM35 can be applied easily in the same way as other integrated-circuit temperature sensors. It can be glued or cemented to a surface and its temperature will be within about 0.01°C of the surface temperature. This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an intermediate temperature between the surface temperature and the air temperature. This is especially true for the TO-92 plastic package, where the copper leads are the principal thermal path to carry heat into the device, so its temperature might be closer to the air temperature than to the surface temperature. To minimize this problem, be sure that the wiring to the LM35, as it leaves the device, is held at the same temperature as the surface of interest. The easiest way to do this is to cover up these wires with a bead of epoxy which will insure that the leads and wires are all at the same temperature as the surface, and that the LM35 die’s temperature will not be affected by the air temperature. The TO-46 metal package can also be soldered to a metal surface or pipe without damage. Of course, in that case the V− terminal of the circuit will be grounded to that metal. Alternatively, the LM35 can be mounted inside a sealed-end metal tube, and can then be dipped into a bath or screwed into a threaded hole in a tank. As with any IC, the LM35 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion.



Features:

Calibrated directly in ° Celsius (Centigrade)

Linear + 10.0 mV/°C scale factor

0.5°C accuracy guaranteeable (at +25°C)

Rated for full −55° to +150°C range

Suitable for remote applications

Low cost due to wafer-level trimming

Operates from 4 to 30 volts

Less than 60 µA current drain

Low self-heating, 0.08°C in still air

Nonlinearity only ±1/4°C typical

Low impedance output, 0.1 W for 1 mA load

Testing of Temperature sensor LM35 using OP-AMP Circuit:

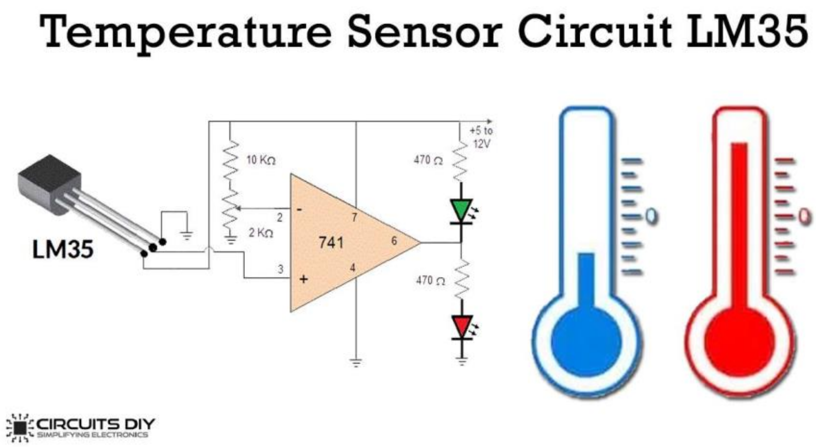


Fig: - 3.7 LM35 testing circuit

Gain calculation:

Gain A = 1+ R_f / R_1

If R_f =4k & R_1=1k Then Gain A=5

Output of LM35:

		Before Amplification,	After amplification With gain of 5
0^0 C	→	0V,	0V
1^0C	→	10mV,	50mV
25^0C	→	250mV,	1.25V
100^0C	→	1000mV,	5V

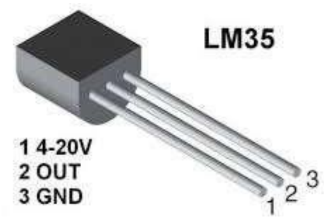


Fig: 3.8 LM35

3.3.4 Pulse Sensor

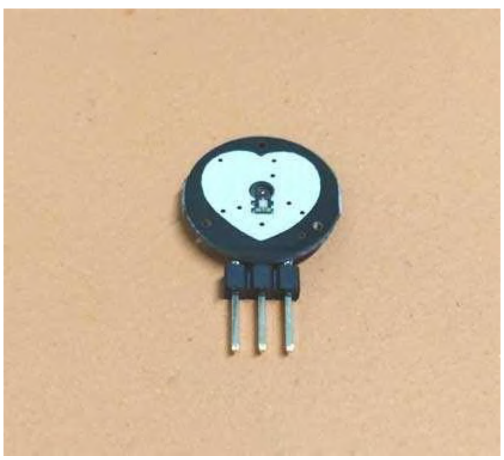


Fig: 3.9 Pulse sensor

Features

- Biometric Pulse Rate or Heart Rate detecting sensor
- Plug and Play type sensor
- Operating Voltage: +5V or +3.3V
- Current Consumption: 4mA
- Inbuilt Amplification and Noise cancellation circuit.
- Diameter: 0.625”
- Thickness: 0.125” Thick

Warning: This sensor is not medical or FDA approved. It is purely intended for hobby projects/demos and should not be use for health critical applications.

Pin Configuration

Pin Number	Pin Name	Wire Colour	Description
1	Ground	Black	Connected to the ground of the system
2	Vcc	Red	Connect to +5V or +3.3V supply voltage
3	Signal	Purple	Pulsating output signal.

How Pulse sensor works

The working of the **Pulse/Heart beat sensor** is very simple. The sensor has two sides, on one side the LED is placed along with an ambient light sensor and on the other side we have some circuitry. This circuitry is responsible for the amplification and noise cancellation work. The LED on the front side of the sensor is placed over a vein in our human body. This can either be your Finger tip or you ear tips, but it should be placed directly on top of a vein.

Now the LED emits light which will fall on the vein directly. The veins will have blood flow inside them only when the heart is pumping, so if we monitor the flow of blood we can monitor the heart beats as well. If the flow of blood is detected then the ambient light sensor will pick up more light since they will be reflect ted by the blood, this minor change in received light is analysed over time to determine our heart beats.

How to use Pulse sensor

Using the pulse sensor is straight forward, but positioning it in the right way matters. Since all the electronics on the sensor are directly exposed it is also recommended to cover the sensor with hot glue, vinyl tape or other non conductive materials. Also it is not recommended to handle these sensors with wet hands. The flat side of the sensor should be placed on top of the vein and a slight presser should be applied on top of it, normally clips or Velcro tapes are used to attain this pressure.

To use the sensor simply power it using the Vcc and ground pins, the sensor can operate both at +5V or 3.3V system. Once powered connect the Signal pin to the ADC pin of the microcontroller to monitor the change in output voltage. If you are using a development board like Arduino then you can use the readily available code which will make things a lot easier. Refer the datasheet at the bottom of the page for more information on how to interface the sensor with Arduino and how to mount it.

Applications

- Sleep Tracking
- Anxiety monitoring
- Remote patient monitoring/alarm system

- Health bands
- Advanced gaming consoles

3.3.5 WIFI Module:

The ESP8266 is a low-cost Wi-Fi module that can be integrated easily into IoT devices. We've featured several projects using this module, such as How To Make Smart Home Electronics: A Smart Mailbox and How To Read Your Arduino's Mind: Building A Childproof Lock. This tutorial will walk you through setting up ESP8266 Wi-fi module which can be used with Arduino. The ESP8266 comes in many models with different functionalities. We'll be focusing on the ESP8266 ESP-01 module, the most common and basic one available.

What is ESP8266?

The ESP8266 is a small Wi-Fi module built around the ESP8266 chip that can connect your microcontroller to the internet wirelessly for a very small cost. It can be a great option for Internet of Things (IoT) projects, but can be difficult to work with for beginner hobbyists who do not have prior experience with the module. In this tutorial, we hope to show you how to interface the ESP8266 with an Arduino and perform some basic functions like connecting it to a Wi-Fi network.

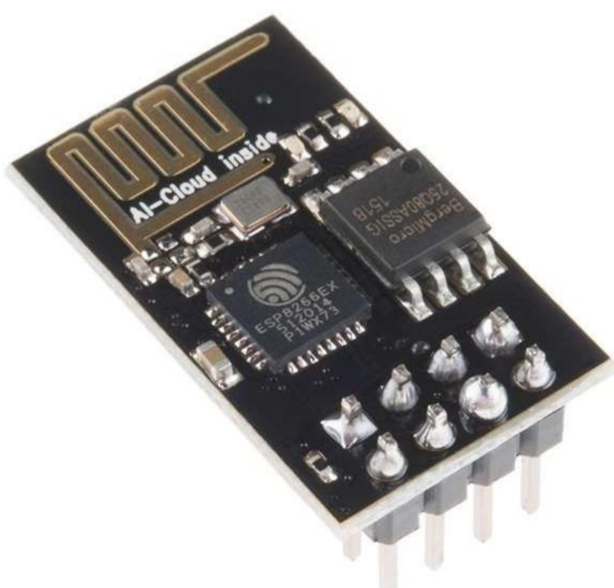


Figure 3.10 ESP8266 ESP-01 module

ESP-01 Features – Sparkfun**:**

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 1MB Flash Memory
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)

The first feature to notice about the ESP8266 is its awkwardly spaced header pins. The module has 8 pins that serve different functions, but they are packed in a 4×2 arrangement that makes plugging the module into a breadboard impossible. This means that to prototype projects on a breadboard, you'll need male-female jumper wires to connect the pins on the ESP8266 to rows on the breadboard. If you'd like to make your prototyping more compact, you can also purchase breadboard breakouts for the ESP8266 such as this one. For prototyping, I chose to just use jumper wires.

The pinout for the ESP8266's pins are according to the following diagram:

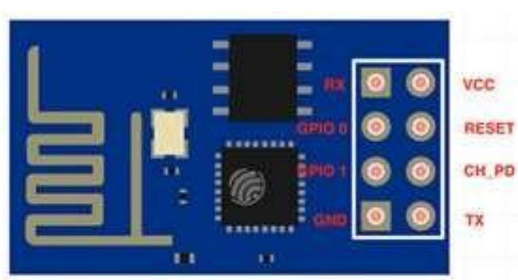


Figure 3.11 ESP8266 Pinout

RX: UART serial communication receive pin

GPIO 0: GPIO pin (unused in this project)

GPIO 1: GPIO pin (unused in this project)

GND: Connection to Ground

VCC: Connection to 3.3V Vcc (Vcc cannot exceed 3.3V!)

RESET: Reset pin (pull down to reset)

CH_PD: Chip enable and power down pin

TX: UART serial communication transmit pin

Note that the maximum voltage input for the ESP8266 is 3.3V. Any input voltage greater than 3.3V will damage the module! To program settings on the ESP8266, we'll first need to connect it to a serial terminal on a computer through which we can send it special commands. Settings that we'll have to program include, for example, the SSID and password for the wifi network the module will be connected to. To connect the ESP8266 to a computer and configure its settings, we'll need a USB to serial adapter with 3.3V logic, along with a serial terminal program. Fortunately for us, we have the Arduino and the Arduino IDE's serial monitor! This means that we'll just have to connect the ESP8266 module to the Arduino and upload a custom sketch to the Arduino.

Use the following diagram to connect the ESP8266 module to the Arduino:

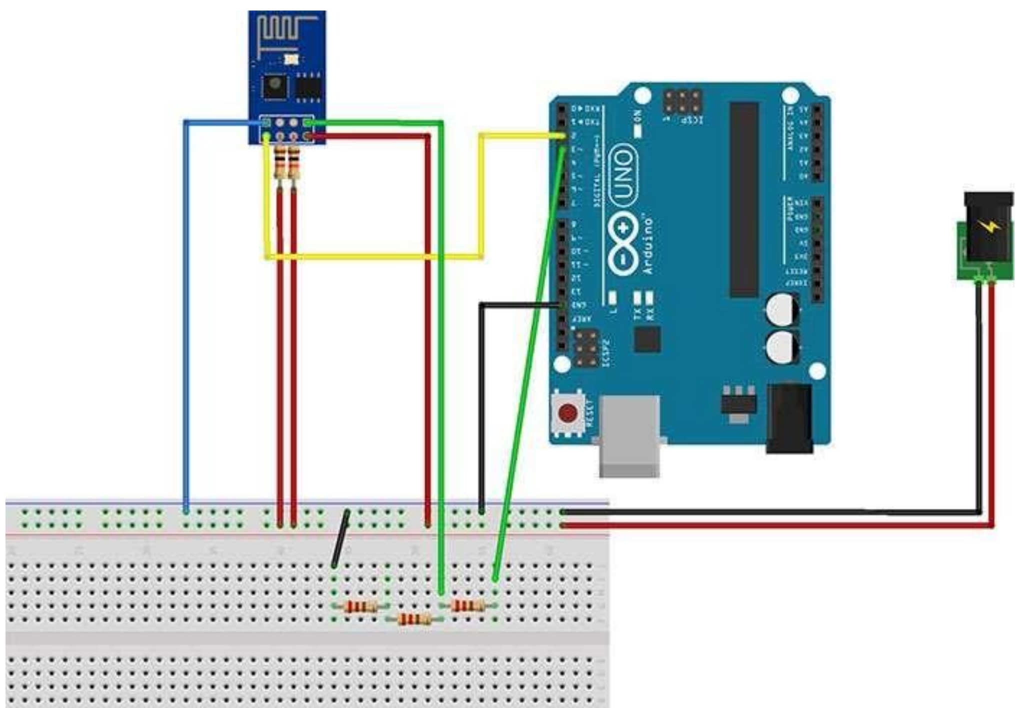


Figure 3.12 Connection between ESP8266 and Arduino Uno

You will need the following parts:

- Arduino Uno or similar module
- ESP8266 ESP-01 module
- 2 x 1kΩ resistors for CH_PD and RESET pull-up
- 3 x 220Ω resistors for serial line voltage divider
- 3.3V regulated power supply for ESP8266 (you cannot plug the power line into the Arduino's 3.3V out pin because the ESP8266 draws a lot of current!)
  - You can use something like this: https://www.sparkfun.com/products/114
- Breadboard and jumper wires

3.3.6 Switches

Introduction

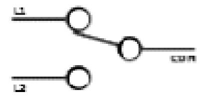
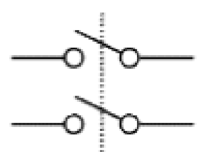
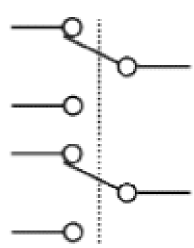
A switch is an electrical component that can break an electrical circuit, interrupting the current or diverting it from one conductor to another. The most familiar form of switch is a manually operated electromechanical device with one or more sets of electrical contacts. Each set of contacts can be in one of two states: either 'closed' meaning the contacts are touching and electricity can flow between them, or 'open', meaning the contacts are separated and non-conducting.

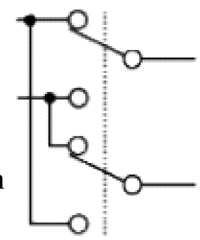
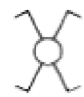
Figure: 3.13 switches in a row

Construction

Keys are placed for giving instructions to the particular part. For example if need to operate a robot .keys are placed and each key is giving an instruction through which the robot moves. If we need to operate the

Electronics specification and abbreviation	Expansion of abbreviation	British mains wiring name	American electrical wiring name	Description	Symbol	IEC60617
SPST	Single pole, single throw	One-way	Two-way	A simple on-off switch: The two terminals are either connected together or disconnected		

				from each other. An example is a light switch.		
SPDT	Single pole, double throw	Two-way	Three-way	A simple changeover switch: C (COM, Common) is connected to L1 or to L2.		
SPCO SPTT, c.o.	Single pole changeover *or* Single pole, centre off *or* Single Pole, Triple Throw			Similar to *SPDT*. Some suppliers use *SPCO/SPTT* for switches with a stable off position in the centre and *SPDT* for those without.[citation needed]		
DPST	Double pole, single throw	Double pole	Double pole	Equivalent to two *SPST* switches controlled by a single mechanism		
DPDT	Double pole, double throw			Equivalent to two *SPDT* switches controlled by a single mechanism: A is connected to B and D to E, or A is connected to C and D to F.		
DPCO	Double pole changeover *or* Double pole, centre off			Equivalent to *DPDT*. Some suppliers use *DPCO* for switches with a stable off position in the centre and *DPDT*for		

		those without.		
Intermediate switch	Four-way switch	*DPDT* switch internally wired for polarity-reversal applications: only four rather than six wires are brought outside the switch housing; with the above, B is connected to F and C to E; hence A is connected to B and D to C, or A is connected to C and D to B.		

Switches with larger numbers of poles or throws can be described by replacing the "S" or "D" with a number or in some cases the letter "T" (for "triple"). In the rest of this article the terms SPST, SPDT and intermediate will be used to avoid the ambiguity in the use of the word "way".

Working

A pair of contacts is said to be "closed" when current can flow from one to the other. When the contacts are separated by an insulating air gap, an air space, they are said to be "open", and no current can flow at typical voltages.

Switches are classified according to the arrangement of their contacts in electronics. Electricians installing building wiring use different nomenclature, such as "one-way", "two-way", "three-way" and "four-way" switches

In a push-button type switch, in which the contacts remain in one state unless actuated, the contacts can either be normally open (abbreviated "n.o." or "no") until closed by operation of the switch, or normally closed ("n.c. or "nc") and opened by the switch action. A switch with both types of contact is called a changeover switch. These may be "make-before-break" which momentarily connect both circuits, or may be "break-before-make" which interrupts one circuit before closing the other

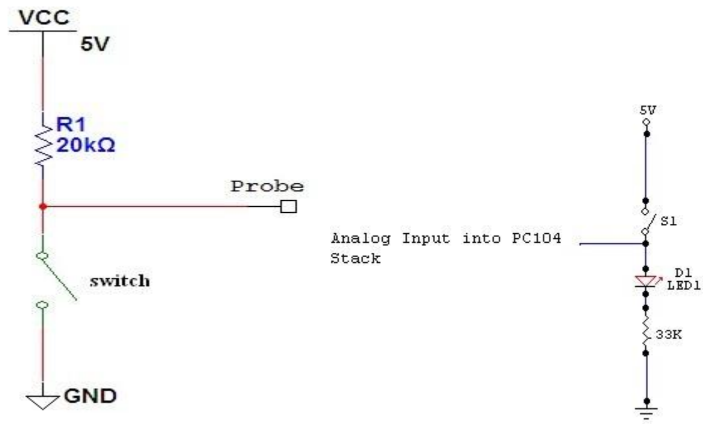


Figure 3.14: circuits of the switch

3.3 SOFTWARE COMPONENTS

- Proteus simulation
- Arduino software
- Programming language

3.3.1 Arduino Software

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs.

3.3.1.2 Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

1) Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50

2) Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to windows.

3) Simple, clear programming environment - The Arduino Software (IDE) is easy- to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

4) Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

5) Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers c an make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the

breadboard version of the module in order to understand how it works and save money

6) Getting Started with Arduino and Genuino products:- Install the Arduino Software (IDE) on Windows PCs This document explains how to install the Arduino Software (IDE) on Windows machines.

7) Download the Arduino Software(IDE).

➢ How to Download the Arduino Software (IDE)

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

3.3.1.3 INSTALLATION

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.3.15 cable

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig.3.16 Mini-B Cable

Step 2 − Download Arduino IDE Software

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

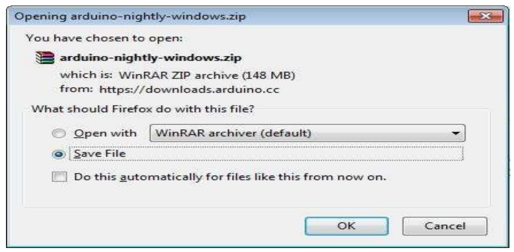


Fig.3.17 Arduino IDE Software

Step 3 − Power Up Your Board

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port. Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE

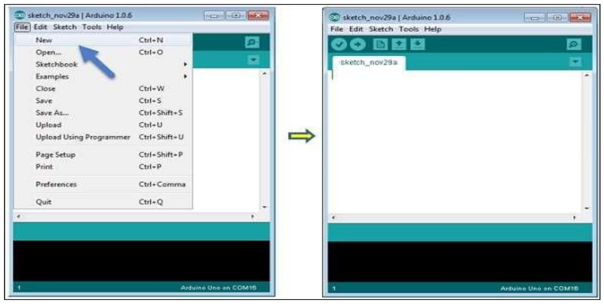
After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

Step 5 − Open Your First Project

Once the software starts, you have two options −

Create a new project.

Open an existing project example.



.

Fig.3.18 Creating New File

Step 6 − Select Your Arduino Board

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

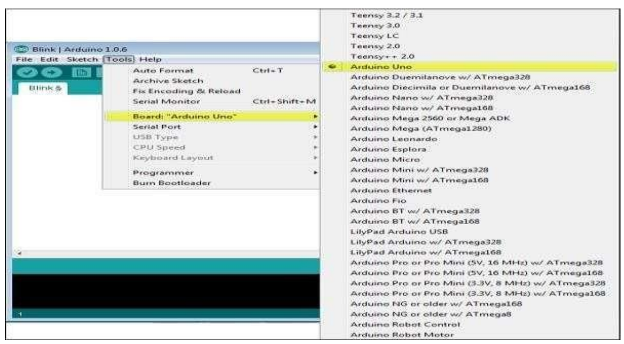


Fig.3.19 Tools Board

Step 7 − Select Your Serial Port

Select the serial device of the Arduino board. Go to Tools → Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

Step 8 − Upload The Program To Your Board

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

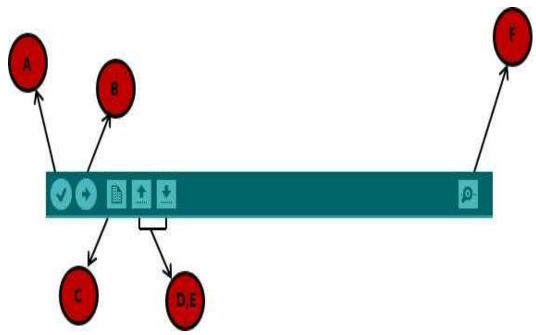


Fig.3.20 Uploading Program

A − Used to check if there is any compilation error. **B** − Used to upload a program to the Arduino board. **C** − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

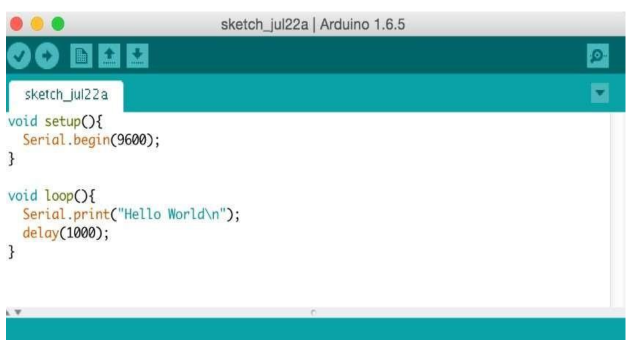


Fig. 3.21 Arduino

3.3.2 PROTEUS*:*

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

3.3.2.1 About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

3.3.2.2 Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc. ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

3.3.2.3 Starting New Design

Step 1: ***Open ISIS software and select New design in File menu***

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

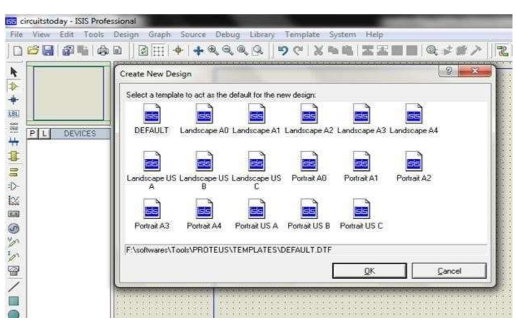


Fig.3.22 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

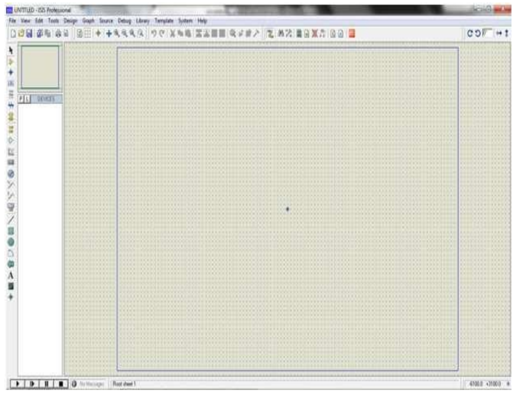


Fig.3.23 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

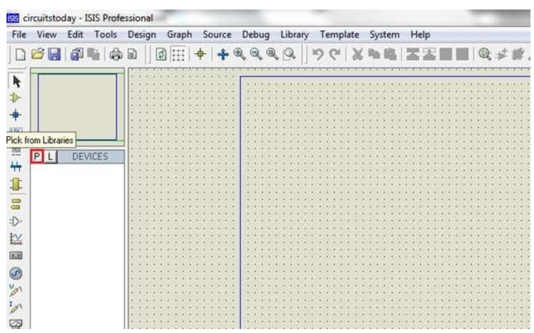


Fig.3.24 Pick From Libraries

Step 6: Select the components from categories or type the part name keyword.

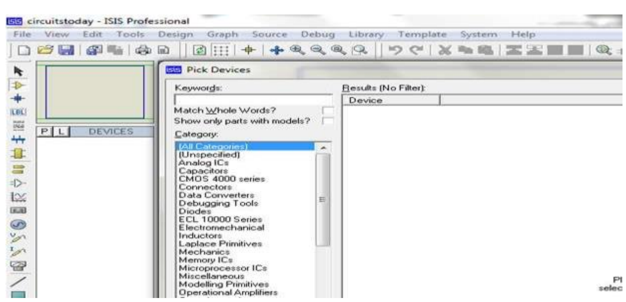


Fig.3.25 Key Word Textbox

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click. Place all the required components and route the wires i.e., make connections. Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively. Double click on the component to edit the properties of the components and click on Ok.

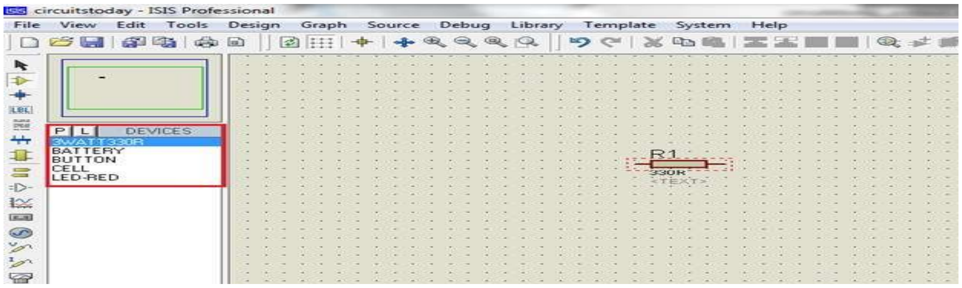


Fig.3.26 Component Selection

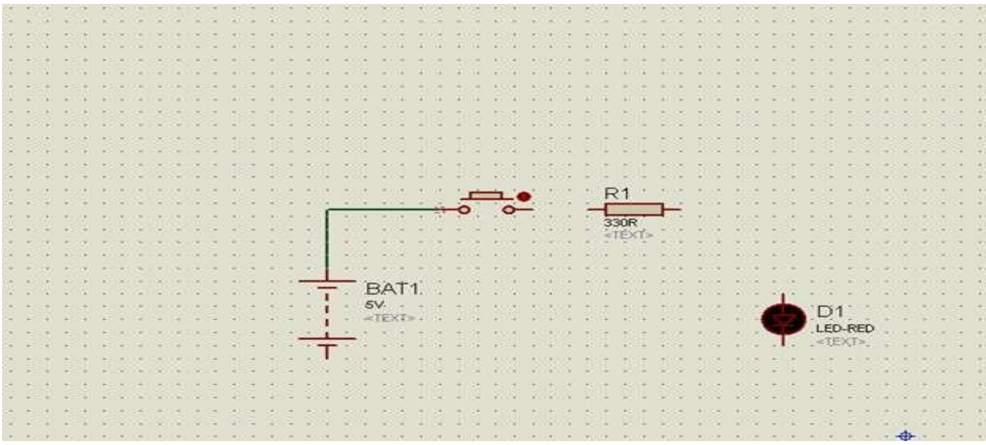


Fig.3.27 Simulation

Step 8: After connecting the circuit, click on the play button to run the simulation. Simulation can be stepped, paused or stopped at any time. In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

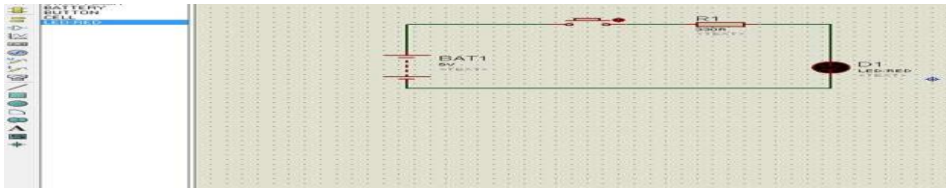


Fig.3.28 Simulation Run

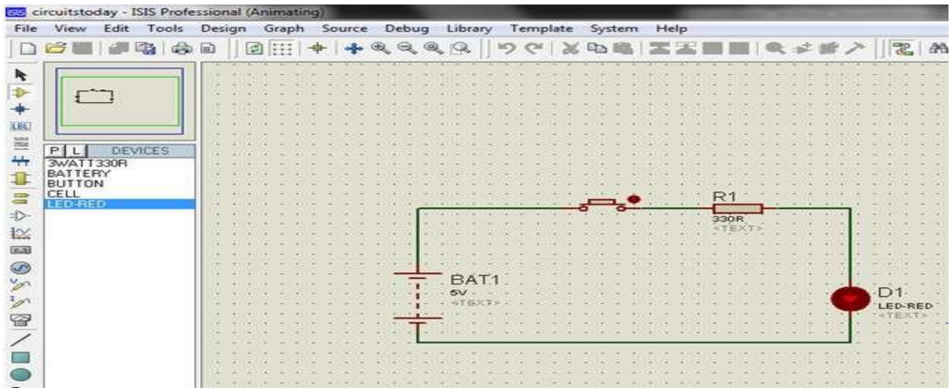


Fig .3.29 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

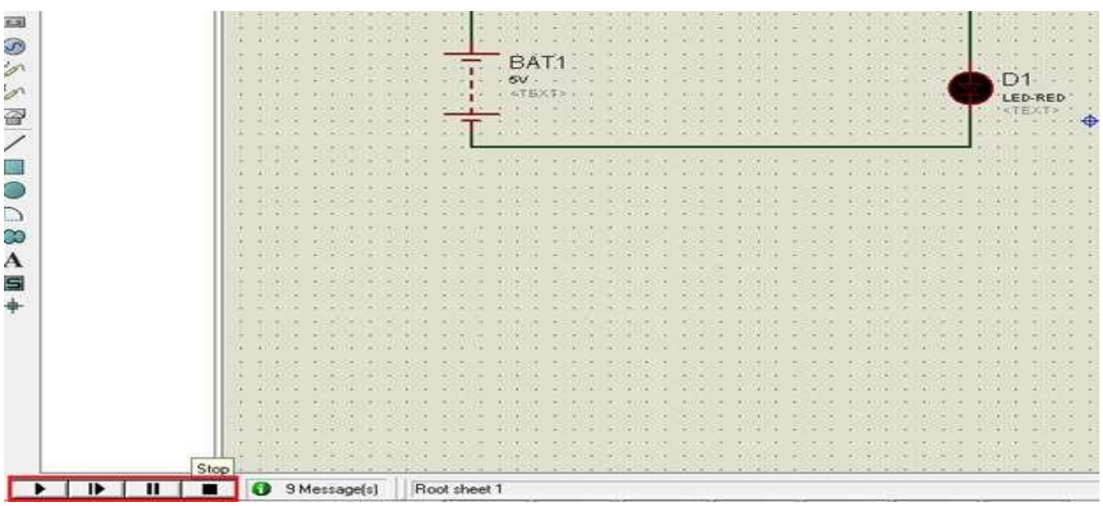


Fig. 3.30 Simulation Step-Pause-Stop Buttons

3.4 DEFINE THE MODULES:

3.4.1 WIFI MODULE

The ESP8266 is a low-cost WiFi module that can be integrated easily into IoT devices. The ESP8266 is a small WiFi module built around the ESP8266 chip that can connect your microcontroller to the internet wirelessly for a very small cost. It can be a great option for Internet of Things (IoT) projects, but can be difficult to work with for beginner hobbyists who do not have prior experience with the module.

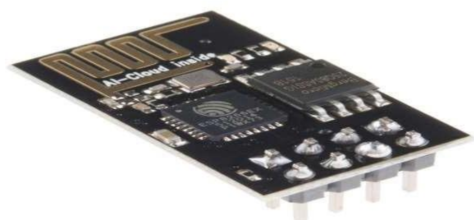


Fig.3.31 ESP8266 ESP-01 Module

3.4.2 ESP-01 Features

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 1MB Flash Memory
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)

The first feature to notice about the ESP8266 is its awkwardly spaced header pins. The module has 8 pins that serve different functions, but they are packed in a 4×2 arrangement that makes plugging the module into a breadboard impossible. This means that to prototype projects on a breadboard, you’ll need male-female jumper wires to connect the pins on the ESP8266 to rows on the breadboard. If you’d like to make your prototyping more compact, you can also purchase breadboard breakouts for the ESP8266 such as this one. For prototyping, I chose to just use jumper wires.

The pinout for the ESP8266’s pins are according to the following diagram:

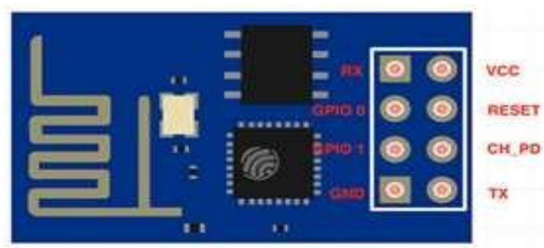


Fig.3.32 ESP8266 Pinout

RX: UART serial communication receive pin

GPIO 0: GPIO pin (unused in this project)

GPIO 1: GPIO pin (unused in this project)

GND: Connection to Ground

VCC: Connection to 3.3V Vcc (Vcc cannot exceed 3.3V!)

RESET: Reset pin (pull down to reset)

CH_PD: Chip enable and power down pin

TX: UART serial communication transmit pin

Note that the maximum voltage input for the ESP8266 is 3.3V. Any input voltage greater than 3.3V will damage the module! To program settings on the ESP8266, we’ll first need to connect it to a serial terminal on a computer through which we can send it special commands. Settings that we’ll have to program include, for example, the SSID and password for the wifi network the module will be connected to. This means that we’ll just have to connect the ESP8266 module to the Arduino and upload a custom sketch to the Arduino. Use the following diagram to connect the ESP8266 module to the Arduino:

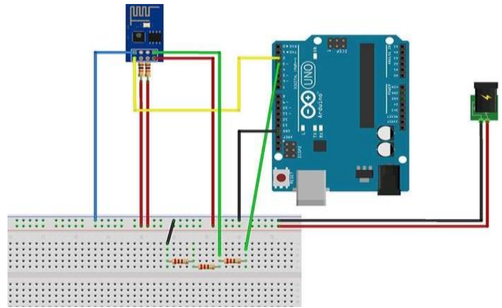


Fig.3.33 Connection Between ESP8266 and Arduino Uno

- Arduino Uno or similar module
- ESP8266 ESP-01 module
- 2 x 1kΩ resistors for CH_PD and RESET pull-up
- 3 x 220Ω resistors for serial line voltage divider
- 3.3V regulated power supply for ESP8266 (you cannot plug the power line into the Arduino’s 3.3V out pin because the ESP8266 draws a lot of current!)
- Breadboard and jumper wires

First, the ESP8266 module is powered by a 3.3V regulated power supply. As described earlier, the ESP8266 has a maximum voltage input rating of 3.3 volts. Using a power supply rated higher than this recommended 3.3 volts could fry the module! When connecting the ESP8266 to this power supply, you should not only connect the Vcc and Ground pins to the power supply lines, but also connect the power supply’s ground line to the Arduino’s ground pin. Since the Arduino will be powered by the USB connection to the laptop, creating a common ground essentially creates a common reference (you can think of it as a baseline) to compare voltages and thereby interpret digital high and low signals.

Second, the receive (RX) line for the ESP8266 module is connected to the output of a resistor voltage-divider circuit. We do this to shift the serial communication logic level (the highs and lows of the digital signals that make up the serial communications) from a logic high of 5 volts on the Arduino to a logic high of 3.3 volts on the ESP8266 module..

Finally, there are two pull-up resistors on the ESP8266’s CH_PD and RESET pins. These are specified by the module’s creators to ensure that the device functions properly while it is being used. To reset the ESP8266 or disable it, these pins must be pulled down, but since we do not want those processes to occur, we will instead pull these pins high through some 10kΩ pull-up resistors. Settings that we’ll have to program include, for example, the SSID and password for the wifi network the module will be connected to. This means that we’ll just have to connect the ESP8266 module to the Arduino and upload a custom sketch to the Arduino. Use the following diagram to connect the ESP8266 module to the Arduino

3.4.3 STRUCTURE AND CONFIGURATION:

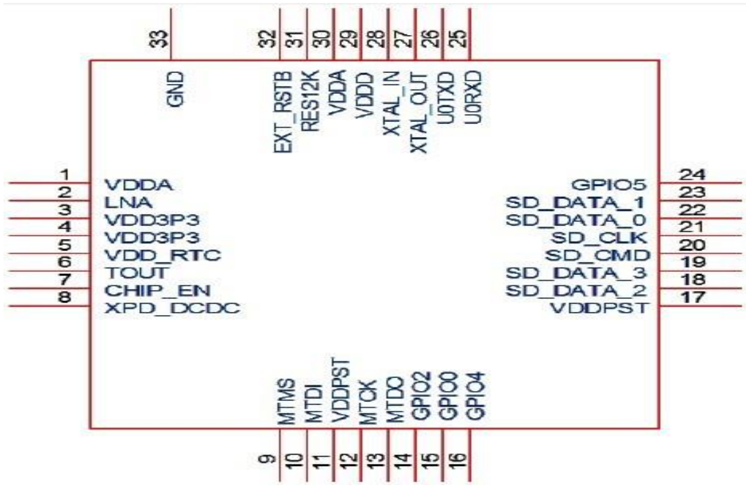


Fig.3.34 Structure and configuration

3.5 MODULE FUNCTIONALITIES:

3.5.1 ESP8266 WIFI MODULE

The ESP8266 WiFi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either

hosting an application or offloading all WiFi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community. This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existance interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts. There is an almost limitless fountain of information available for the

ESP8266, all of which has been provided by amazing community support. In the Documents section below you will find many resources to aid you in using the ESP8266, even instructions on how to transform this module into an IoT (Internet of Things) solution.

CHAPTER-4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM

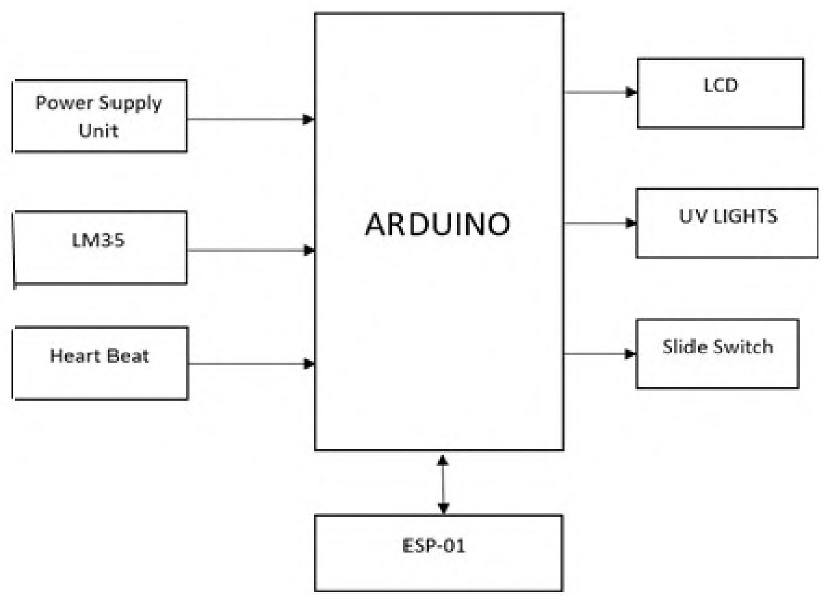


Fig.4.1 Block Diagram

The system consists of two input sensors to detect temperature, and heartbeat sensor, a LCD to display ,Wi-Fi module and slide switch to turn on UV light. At the same time ,data regarding these parameters are sent to IOT module .WiFi module is used to establish connection between microcontroller and the WiFi network connections and sends the data to the IOT.

4.2 IMPLEMENTATION STAGES

Stepwise implementation of IOT covid patient health monitor system is given under:

Step 1: Switch on the kit by giving power supply.

Step 2: Keep the ssid: project, password: project1235 in mobile and turn on Hotspot

Step 3: Kit gets connected to Hotspot using Wi-Fi module.

Step 4: The temperature and heartbeat of patient is measured using LM35 and heartbeat sensors respectively

Step 5: Based on conditions given in code i.e., hbval> 78 and 10< hbval < 50, td > 35.Sensor senses and updates to sever according to conditions.

Step 6: “Thingspeak” is the server where the results are updated.

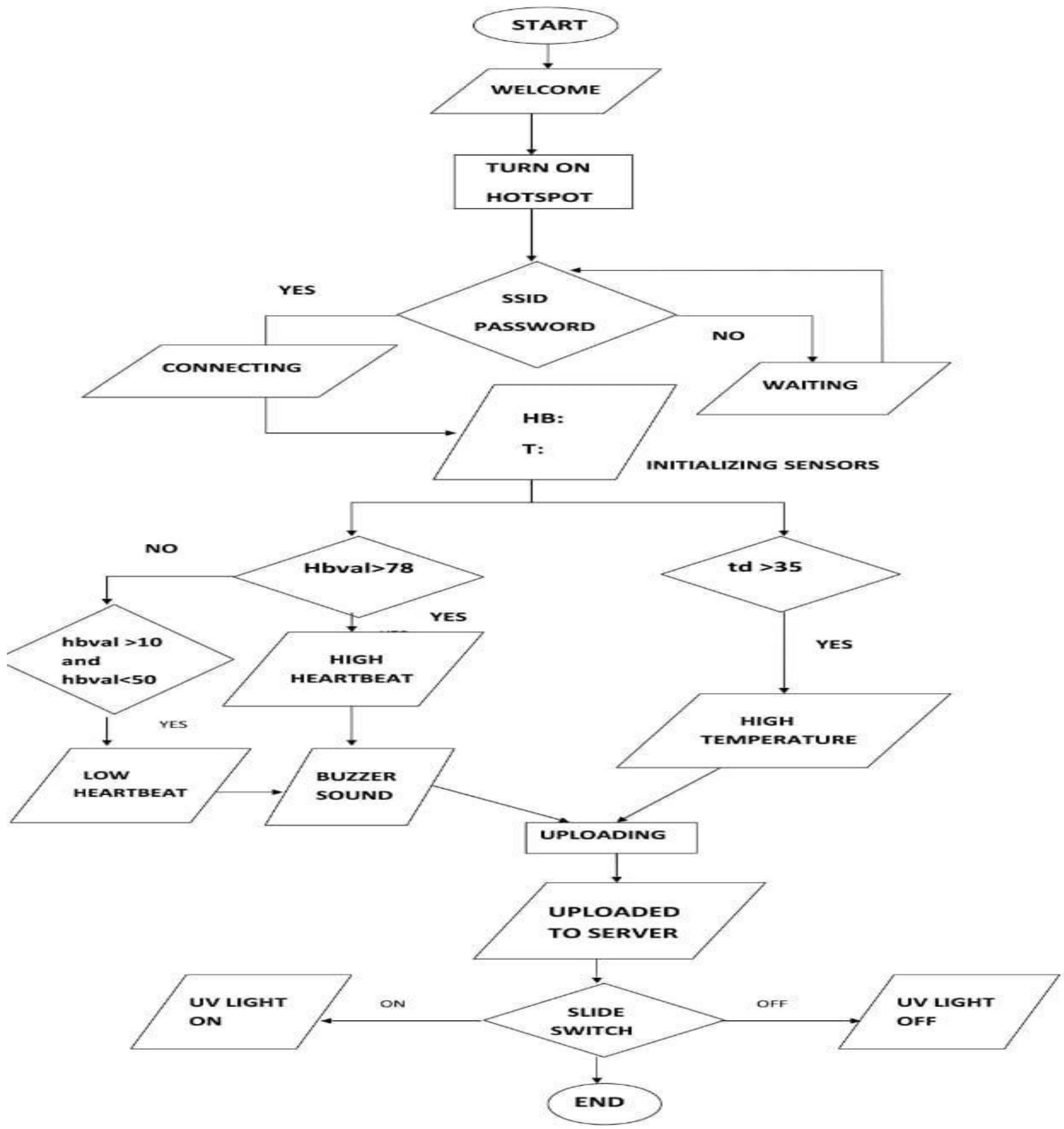


Fig 4.2 Flow Chart

4.3 RESULTS

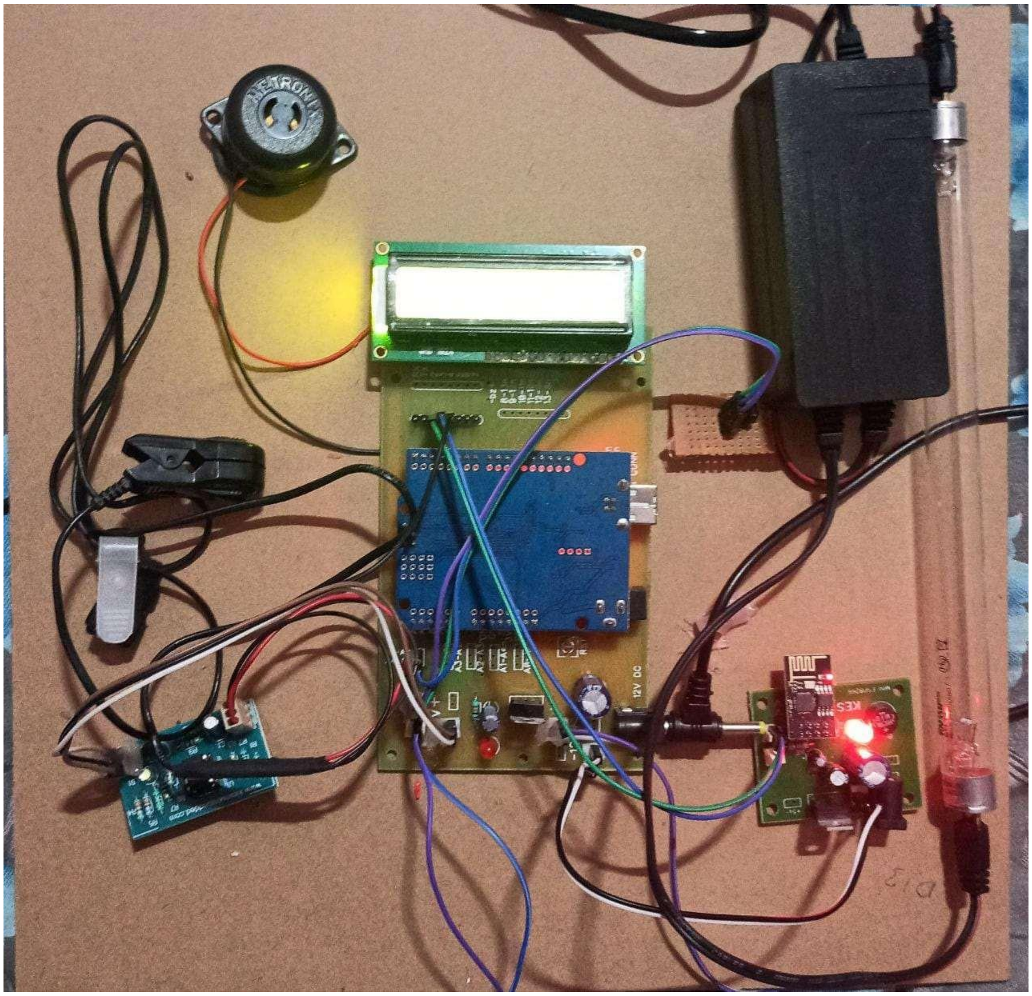


Fig: 4.3 Overall system

Fig:4.4 Displays Commands in LCD

Fig 4.5 Displayed on LCD as HIGH HEARTBEAT (hbval>78)

Fig:4.6 Displayed on LCD as HIGH TEMPERATURE (td>35)

Fig: 4.7 UV light ON for Disinfection

CHAPTER-5

PROJECT TESTING

5.1 OVERVIEW OF TESTING METHODS

5.1.1 SOFTWARE TESTING

Software Testing is a process of executing the application with intent to find any software bugs. It is used to check whether the application met its expectations and all the functionalities of the application is working. The final goal of testing is to check whether the application is behaving in the way it is supposed to under specified conditions. All aspects of the code are examined to check the quality of application. The primary purpose of testing is to detect software failures so that defects may be uncovered and corrected. The test cases are designed in such way that scope of finding the bugs is maximum.

5.1.2 TESTING LEVELS

There are various testing levels based on the specificity of test.

• Unit testing: Unit testing refers to tests conducted on a section of code in order to verify the functionality of that piece of code. This is done at the function level.

• Integration Testing: Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Its primary purpose is to expose the defects associated with the interfacing of modules.

• System Testing: System testing tests a completely integrated system to verify that the system meets its requirements.

• Acceptance testing: Acceptance testing tests the readiness of application, satisfying all requirements.

• Performance testing: Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or device such as response time or millions of instructions per second etc.

5.1.3 SYSTEM TEST CASES

A test case is a set of test data, preconditions, expected results and post conditions, developed for a test scenario to verify compliance against a specific requirement. We have designed and executed a few test cases to check if the project meets the functional requirement.

CHAPTER-6

CONCLUSION AND FUTURE ENHANCEMENT

The Internet of Things is considered now as one of the feasible solutions for any remote value tracking especially in the field of health monitoring. The proposed system provides best way to monitor covid patients without any life risk. Moreover, the health can be monitored by any doctor at any distance.

The user (patient) can perform the following basic functions on the system after successful registration:

• Control home devices

• Check for medication and prescription

• Send symptoms of the virus to the doctor while under isolation

• Chat with the doctor or leave a message

• Enter physiological values of measured parameters manually

• Book appointment with the doctor

• View medical data.

In the next level of project other sensors, devices like low cost portable ventilators; ECG can also be connected to this device to monitor respiration rate and blood pressure.Humidity of the room can also be measured in patient's room IOT enabled smart pill boxes keep track of patients medicine schedule .It is used to intimate the patient to take medicine at certain time. An emergency button is kept when a patient is in need of any kind of medical assistance

.

PUBLICATION

Submitted Paper in the Conference ICISECT - 21 with Paper ID (ICISECT21-0027) and got Acceptance for the Paper.

REFERENCES

[1]. S.H. Almotiri, M. A. Khan, and M. A. Alghamdi. Mobile health (m- health) system in the context of iot. In 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), pages 39–42, Aug 2016.

[2]. Gulraiz J. Joyia, Rao M. Liaqat, Aftab Farooq, and Saad Rehman, Internet of Medical Things (IOMT): Applications, Benefits and Future Challenges in Healthcare Domain, Journal of Communications Vol. 12, No. 4, April 2017.

[3]. Shubham Banka, Isha Madan and S.S. Saranya, Smart Healthcare Monitoring using IoT. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 15, pp. 11984-11989, 2018.

[4]. K. Perumal, M. Manohar, A Survey on Internet of Things: Case Studies, Applications, and Future Directions, In Internet of Things: Novel Advances and Envisioned Applications, Springer International Publishing, (2017) 281-297.

[5]. S.M. Riazulislam, Daehankwak, M.H.K.M.H., Kwak, K.S.: The Internet of Things for Health Care: A Comprehensive Survey. In: IEEE Access (2015).

[6]. P. Rizwan, K. Suresh. Design and development of low investment smart hospital using Internet of things through innovative approaches, Biomedical Research. 28(11) (2017).

[7]. K.R. Darshan and K.R. Anandakumar, “A comprehensive review on usage of internet of things (IoT) in healthcare system,” in Proc. International Conference on Emerging Research in Electronics, Computer Science and Technology, 2015.

[8]. Internet of Things (IoT): Number of Connected Devices Worldwide From 2012 to 2020 (in billions). [Online]. Available: https://www.statista.com/statistics/471264/iotnumberof-connected-devices-worldwide.

[9]. Anna Odone, Nicola Clementi, "The efficacy of UV light-emitting technology against coronaviruses: a systematic review”, 2021.

BIBILOGRAPHY

International Journal of Engineering Research & Technology (IJERT). An IoT based Health Monitoring System using Arduino Uno. Available from: https://www.ijert.org/an-iot-based-health-monitoring-system-using-arduino-uno

Smart healthcare support for remote patient monitoring during covid-19 quarantine. . Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7490242/

[Anna Odone, Nicola Clementi], "The efficacy of UV light-emitting technology against coronaviruses: a systematic review", 2021.

Internet of Things (IoT): Number of Connected Devices Worldwide From 2012 to 2020 (in billions). [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

APPENDIX

```
#include <LiquidCrystal.h>
LiquidCrystallcd(13,12,11,10,9,8);
#include <SoftwareSerial.h>
SoftwareSerialmySerial(2,3);
//adc
unsigned int hbpin = 4;
int hcount = 0, hbval = 0;
int temp = A4,buz = 7;
char res[130];
//pins
void serialFlush(){
while(Serial.available() > 0) {
char t = Serial.read();
```

```
}}
void myserialFlush(){
while(mySerial.available() > 0) {
char t = mySerial.read();
}}
char check(char* ex,int timeout)
{
int i=0;
int j = 0,k=0;
while (1)
{
sl:
if(mySerial.available() > 0)
{
res[i] = mySerial.read();
if(res[i] == 0x0a || res[i]=='>' || i == 100)
{
i++;
res[i] = 0;break;
}
i++;
}
j++;
if(j == 30000)
{
k++;
//Serial.println("kk");
```

```
j = 0;
}
if(k > timeout)
{
// Serial.println("timeout");
return 1;
}
}//while 1
if(!strncmp(ex,res,strlen(ex)))
{
//Serial.println("ok..");
return 0;
}
else
{
// Serial.print("Wrong ");
// Serial.println(res);
i=0;
gotosl;
}}
char buff[200],k=0;
int phvalue=0;
void upload1();
const char* ssid = "project";
const char* password = "project1235";
int T;
int tt;
```

```
void setup() {

int i=0;

char ret;

pinMode(temp,INPUT);

pinMode(buz,OUTPUT);

digitalWrite(buz,HIGH);

delay(500);

Serial.begin(9600);

mySerial.begin(115200);

lcd.begin(16,2);

lcd.clear();lcd.setCursor(0, 0);lcd.print("WELCOME");

delay(1000);

// serialFlush();

//gsm

st:

mySerial.println("ATE0");

Serial.println("ATE0");

ret = check((char*)"OK",50);

mySerial.println("AT");

Serial.println("AT");

ret = check((char*)"OK",50);

if(ret != 0)

{

delay(1000);

gotost;

}
```

```
lcd.clear();lcd.setCursor(0, 0);lcd.print("CONNECTING");

mySerial.println("AT+CWMODE=1");

Serial.println("AT+CWMODE=1");

ret = check((char*)"OK",50);

cagain:


myserialFlush();

Serial.print("AT+CWJAP=\"");

mySerial.print("AT+CWJAP=\"");

mySerial.print(ssid);

Serial.print(ssid);

mySerial.print("\",\"");

Serial.print("\",\"");

mySerial.print(password);

Serial.print(password);

mySerial.println("\"");

Serial.println("\"");

if(check((char*)"OK",300))gotocagain;

mySerial.println("AT+CIPMUX=1");

Serial.println("AT+CIPMUX=1");

delay(1000);


lcd.clear();lcd.setCursor(0,  0);lcd.print("WAITING");

}

unsigned long int duration = 0;

int hbeat = 0;

void loop() {
```

```
duration = pulseIn(hbpin, LOW, 5000000) / 1000;
if (duration == 0)
hbeat = 0;
else
hbeat = 64 + duration % 18;
delay(1000);
lcd.clear();
lcd.setCursor(0, 1);
hbval = hbeat;
Serial.println(hbval);
int td = analogRead(temp)/4;


Serial.print(td);
Serial.print("\r\n");
Serial.print(wd);
Serial.print("\r\n");
lcd.setCursor(0, 0);lcd.print("T:");lcd.print(td);lcd.print("      ");
lcd.setCursor(0, 1); lcd.print("HB:"); lcd.print(hbval); lcd.print("  ");
delay(1000);
///////////////// ldr1 start //////////
if(td > 35 )
{
lcd.clear();
lcd.setCursor(0, 0);lcd.print("HIGH TEMPRATURE");
delay(3000);
upload1(td,hbval);
```

```
delay(500);
}

if (hbval>78 )
{
lcd.setCursor(0, 1); lcd.print("HIGH HEARTBEAT  ");
//sendmsg(number,"HIGH HEART RATE:");
delay(200);
//      sendmsg1(number1,"HIGH HEART RATE:");
// delay(3000);
upload1(td,hbval);
digitalWrite(buz,LOW);
delay(500);
}
if (hbval>10 &&hbval<50 )
{
lcd.setCursor(0, 1); lcd.print("LOW HEARTBEAT ");
upload1(td,hbval);
digitalWrite(buz,LOW);
delay(500);
delay(3000);
}
digitalWrite(buz,LOW);
delay(500);
}//loop
char bf2[100];
void upload1(unsigned char *chr ,unsigned char *chr1,unsigned char *chr2,unsigned char *chr3)
```

```
{
delay(2000);
lcd.clear();lcd.setCursor(0, 1);lcd.print("UPLOADING");
myserialFlush();
mySerial.println("AT+CIPSTART=4,\"TCP\",\"api.thingspeak.com\",80");
// Serial.println("AT+CIPSTART=4,\"TCP\",\"embeddedspot.top\",80");
delay(8000);
sprintf(buff,"GET
https://api.thingspeak.com/update?api_key=ZHR6DIPXM6G1SBYN&field1=0=%u&field2=%u\r
\n\r\n",chr,chr1,chr2,chr3);
//sprintf(buff,"GET
http://embeddedspot.top/iot/storedata.php?name=sensors010&s1=%u\r\n\r\n",chr);
myserialFlush();
sprintf(bf2,"AT+CIPSEND=4,%u",strlen(buff));
mySerial.println(bf2)
delay(5000)
myserialFlush();
mySerial.print(buff);
Serial.print(buff);
delay(2000);
mySerial.println("AT+CIPCLOSE");
Serial.println("AT+CIPCLOSE");
lcd.setCursor(0, 1);lcd.print("UPLOADDED");    lcd.clear();
}
```

A

MAJOR PROJECT REPORT

On

FINGER PRINT BASED SECURITY SYSTEM

Submitted by

Ms. V.V.S Vasavi (17K81A0459)
Ms. A. Kranthi (17K81A0403)
Ms. T. Meena (17K81A0454)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
in

ELECTRONICS AND COMMUNICATION ENGINEERING

Under the Guidance of

Mr. V.V. Ramana Rao, M.Tech

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN’S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

BONAFIDE CERTIFICATE

This is to certify that the project entitled "**Finger Print Based Security System**", is being submitted by **Ms.V.V.S Vasavi(17K81A0459), Ms.A.Kranthi(17K81A0403), Ms.T.Meena(17K81A0454)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in Electronics and Communications Engineering is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Internal Guide	**Head of the Department**
Mr.V.V .Ramana Rao, M.Tech Associate Professor	Dr.B.Hari Krishna, Ph.D Professor

External Examiner

Place: Secunderabad

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT WITH ROLL **VURAGAYALA VENKATA SAI VASAVI** NO.**17K81A0459, ALLURI KRANTHI** WITH ROLL NO.**17K81A0403**, **T.MEENA** WITH ROLL NO.**17K81A0454**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**FINGER PRINT BASED SECURITY SYSTEM**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014 Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of Bachelor of Technology in Department of 'ECE', session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled **'Finger Print Based Security System'** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

Ms. V.V.S Vasavi	17K81A0459
Ms. A. Kranthi	17K81A0403
Ms. T. Meena	17K81A0454

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, our beloved **Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision ateach and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B. Hari Krishna**, **HoD & Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our department **Project Coordinator, Dr. A. Anand Professor** for his involvement with originality has triggered and nourished my intellectualmaturity that will help me for a long time to come.

We sincerely thank our **Class Project Coordinator, Mrs. A. Soumya,** Department of ECE andour project **Internal guide, Mr.V.V.Ramana Rao, Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level **Project Coordinators Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for theirsupport and co- operation that is difficult to express in words.

1. V.V.S.Vasavi
2. A.Kranthi
3. T.Meena

ABSTRACT

Fingerprint-based security system to authenticate users from entering particular premises. The system is useful for secure sites to provide access only to authorized users automatically. This ensures safety and security at secure sites/premises like military, navy, government as well as corporate premises. For this purpose we here use a Atmega 32 microcontroller circuit. The circuit consists of atmega microcontroller that is interfaced to fingerprint sensor, LCD display and motors to operate a door. Users are allowed to register into the system first. After registration/enrolment the system allows to start monitoring. In monitoring mode, the system monitors for fingerprints. Now if a fingerprint is detected, the system scans the fingerprint against stored ones. If a match is found, the system operates the motors to open the door for those users, else the system does not open the door. Thus we ensure a secure fingerprint authorized security system.

Finger print comparison is one of the tough task, mainly comparing latent finger prints like palm, foot prints will be most difficult part in finger print science. Finger print expert must be well trained and expert to perfectly analyze finger prints.

The comparison of fingerprints, especially latent fingerprints (palm prints, footprints), can be the single most difficult task in all the fingerprint science. No matter how skillful an individual may be in all the related areas, it is the additional ability to accurately compare latent prints against known prints that allows the examiner to be called a fingerprint expert. These utilize fingerprint recognition technology to allow access to only those whose fingerprints you choose. It contains all the necessary electronics to allow you to store, delete, and verify fingerprints with just the touch of a button.

Stored fingerprints are retained even in the event of complete power failure or battery drain. These eliminates the need for keeping track of keys or remembering a combination password, or PIN. It can only be opened when an authorized user is present, since there are no keys or combinations to be copied or stolen, or locks that can be picked.

The main aim of this project is develop a security lock system based on fingerprint scanning. In this project we are using microcontroller for opening and closing lock based on finger print which is stored in microcontroller itself so that only authorized person will access the security lock.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY OF MOST COMMONLY USED TERMS

FINGERPRINT: A fingerprint is an impression left by the friction ridges of a human finger.There are four groups of whorls: plain (concentric circles), central pocket loop (a loop with a whorl at the end), double loop (two loops that create an S-like pattern) and accidental loop (irregular shaped).

SECURITY: The security which we provide through this system is by using the fingerprints of the user In this system we can enroll our fingerprints and if we attach this to our door so the motor of the system will operate your door based on commands given by Arduino board of this system.

ARDUINO UNO BOARD: This is a microcontroller for this system of version ATMEGA328,this board will control entire operations of the system and act as like heart of the system.

AUTHORIZED USERS: The persons who have enrolled their fingerprints into that system are been considered as authorized users. These people will only be allowed by the system to enter.

UNAUTHORISED USERS: The persons who have not enrolled their fingerprints into that system are been considered as unauthorized users. These people will not be allowed by the system to enter

CHAPTER 1
INTRODUCTION

1.0 OVERVEIW OF THE PROJECT:

In this project at the present age, safety has become a necessary issue for most of the people mainly in the rural and urban areas. Some people are more concern about their safety for them at a expensive thing like jewellery money etc. So, the bank lockers are the safest place to accumulate them but the conventional security system is not providing the higher security because in conventional security system a user can open the lockers using keys. Sometimes the keys could be stolen. Then the user will apply for original keys but the time period is longer to get new keys so in its place of using this security system we have implemented. In this present age, safety has become a necessary issue for biometric based security system which provide more security than conventional system. In this paper we have implemented security of the money in the bank locker, house, and workplace (treasury) by using Fingerprint and GSM technology which will be more protected than other systems. We use only one Fingerprint module scan for no. of only authorized persons to open the separate bank locker. In this project we have designed and implemented a bank locker safety system based on Fingerprint technology. In this system only authentic person can be recovered money from bank locker with password protection method.

The purpose of this project is to increase the security for lockers because conventional banks lockers are not secure so to rectify the problem which we discuss above we are implementing this project. In this project each locker has separate fingerprint module to opening the locker. Users scan their fingerprint. After scanning process complicated user enters their password with the help of keypad then their particular locker is open. After the work has been completed if key is pressed again with help of keypad the locker door will be closed again. If an unauthorized person tries to check his fingerprint image, then a signal will be given by a buzzer which is interfaced to the controller and also if incorrect password is entered by the user again indication will be given by the buzzer.

1.1 OBJECTIVES OF STUDY:

Fingerprint based security system is the most secured system as compared to other systems, Reason is that RFID card or Keys of lock can be stolen, password may be leaked. However, thumbnail of every human being is unique, so lock will not open unless the same person is present to give the impression of fingerprint.

The basic function of every type of scanner is to obtain an image of a person's fingerprint and find a match for it in its database. The measure of the fingerprint image quality is in dots per inch (DPI). Capacitive or CMOS scanners use capacitors and thus electrical current to form an image of the fingerprint. This system was been designed to control thefts in private areas and to provide more security and authenticity to the users, as we are using finger prints for identifying the persons this was one of the most efficient way to identify people because the fingerprints will acts as the unique IDs for persons.

No two person will not be matched with their fingerprints so based on this fact this system was been designed for providing the security for example if a person enrolls his/her fingerprints with the finger print scanner provided in this system then he/her will only be allowed by this system to enter into system protected area. Here we can place this system to operate the door to your private room so as to make the door to open or close according to the system commands, humans cannot open this door by their own because that door will not be provided with any lock or key.

By implementing this project in the public areas like banks and in educational institutes and at offices thefts will be reduced to most extent because the person who have access to that room will only be allowed into the room if any unauthorized person comes and places his finger print and if that finger print was not enrolled in the system, then the door will not open and it will generate a message as a unauthorized person.

1.2 SCOPE OF THE STUDY:

Finger print based security system can be used at many places like Industries, Offices, and Colleges or even at our home. This project is a fine combination of “Biometrics technology” and “Embedded system technology”. ... It makes use of Biometric sensor to detect fingerprint. It is also called as Biometric sensor.

1. We can send this data to a remote location using mobile or internet
2. We can use non-contact fingerprint sensor. Which is also called as touchless 3D fingerprint
3. We can implement other related modules like fire sensor, GSM modem.

Fingerprint based security system is the most secured system as compared to other systems. Reason is that RFID card or Keys of lock can be stolen, password may be leaked. However, thumbnail of every human being is unique, so lock will not open unless the same person is present to give the impression of fingerprint., No need to carry the keys to open the lock. Or even there is no need to remember the password or any Pin number. One of the main advantages is that this system remembers the stored password even if the power supply is turned off. Scientific research and studies have proved that fingerprints do not change as you grow up. Using Fingerprint saves time to gain access as compared to other methods like RFID card, Password or Key.

Since 2000 electronic fingerprint readers have been introduced as consumer electronics security applications. Fingerprint sensors could be used for login authentication and the identification of computer users. However, some less sophisticated sensors have been discovered to be vulnerable to quite simple methods of deception, such as fake fingerprints cast in gels. In 2006, fingerprint sensors gained popularity in the laptop market. Built-in sensors in laptops, such as Think Pads, VAIO, HP Pavilion and EliteBook laptops, and others also double as motion detectors for document scrolling, like the scroll wheel.The use of fingerprints in crime fiction has, of course, kept pace with its use in real-life detection. Sir Arthur Conan Doyle wrote a short story about his celebrated sleuth Sherlock Holmes which features a fingerprint: "The Norwood Builder" is a 1903 short story set in 1894 and involves the discovery of a bloody fingerprint which helps Holmes to expose the real criminal and free his client. The British detective writer R. Austin Freeman's first Thorndyke novel The Red Thumb-Mark was published in 1907 and features a bloody fingerprint left on a piece of paper together with a parcel of diamonds inside a safe-box. These become the center of a medico-legal investigation.

1.3 MATERIALS REQUIREMENT:

1.3.1 HARDWARE COMPONENTS:

- ARDUINO UNO (ATMEGA328) MICROCONTROLLER BOARD

Fig:1.1 Arduino Uno board

The above shown figure is Arduino Uno board.

- FINGER PRINT SCANNER ALONG WITH 3 PUSHBUTTONS

Fig:1.2 Fingerprint Scanner

The above shown figure is fingerprint Scanner.

- LCD DISPLAY (16*2)

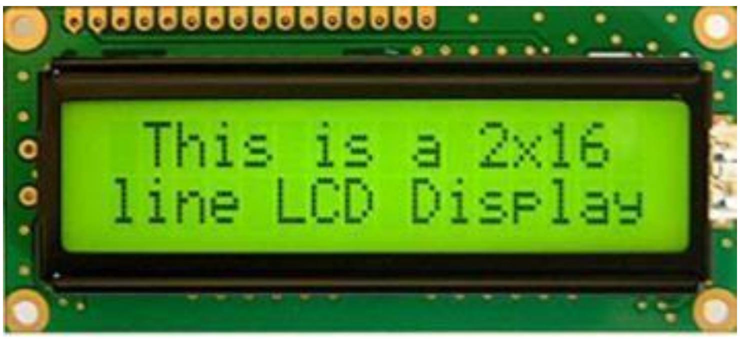


Fig:1.3 LCD Display

The above shown figure is 16x2 LCD Display.

➢ L293D DRIVER IC

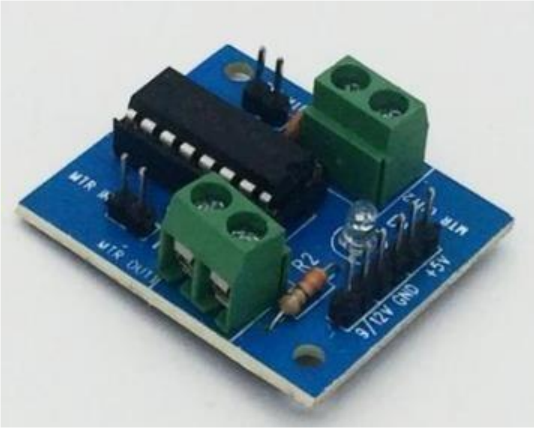


Fig:1.4 Driver IC

The above shown figure is Driver IC

➢ DC MOTOR (5V SUPPLY)

Fig:1.5 DC Motor

The above shown figure is DC Motor

➢ MB27L-12V PIEZO BUZZER

Fig:1.6 Piezo Buzzer

The above shown figure is Piezo Buzzer.

1.3.2 SOFTWARE REQUIREMENTS:

1.3.2.1 ARDUNO IDE SOFTWARE: This software is basically used for giving commands to ARDUNO Microcontroller**.**

STEPS FOR INSTALLING ARDUNO IDE SOFTWARE:

1.Download Arduino IDE Software in your PC.

2. When the download finishes, proceed with the installation.

3. Allow the driver installation process when you get a warning from the operating system.

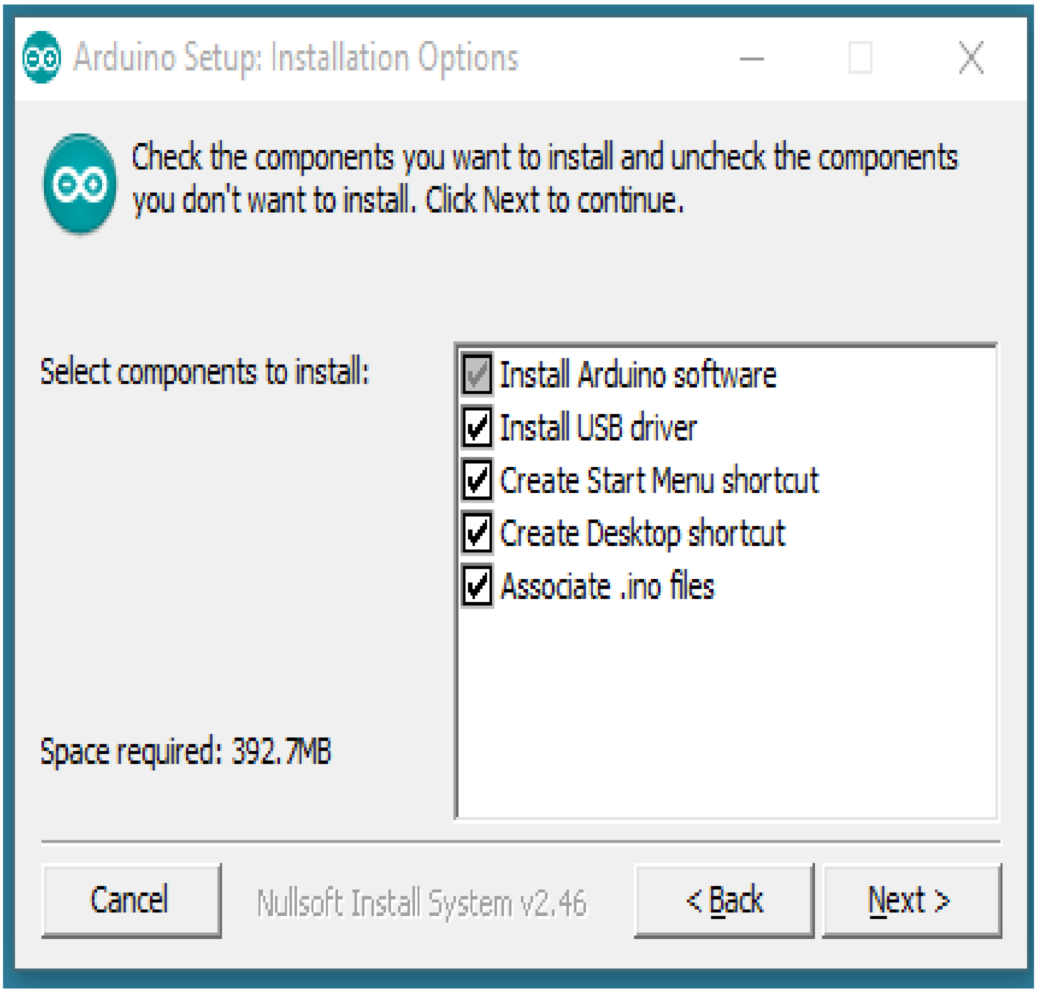


Fig:1.7 Installation process of Driver

4. Choose the components to install

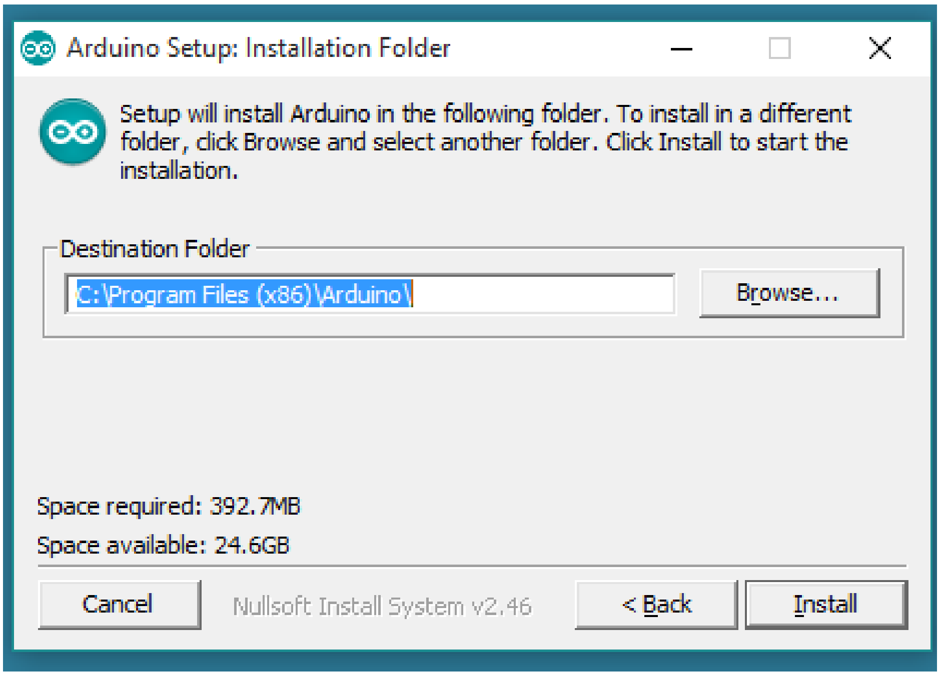


Fig 1.8 Destination folder

5. Choose the installation directory (we suggest to keep the default one)

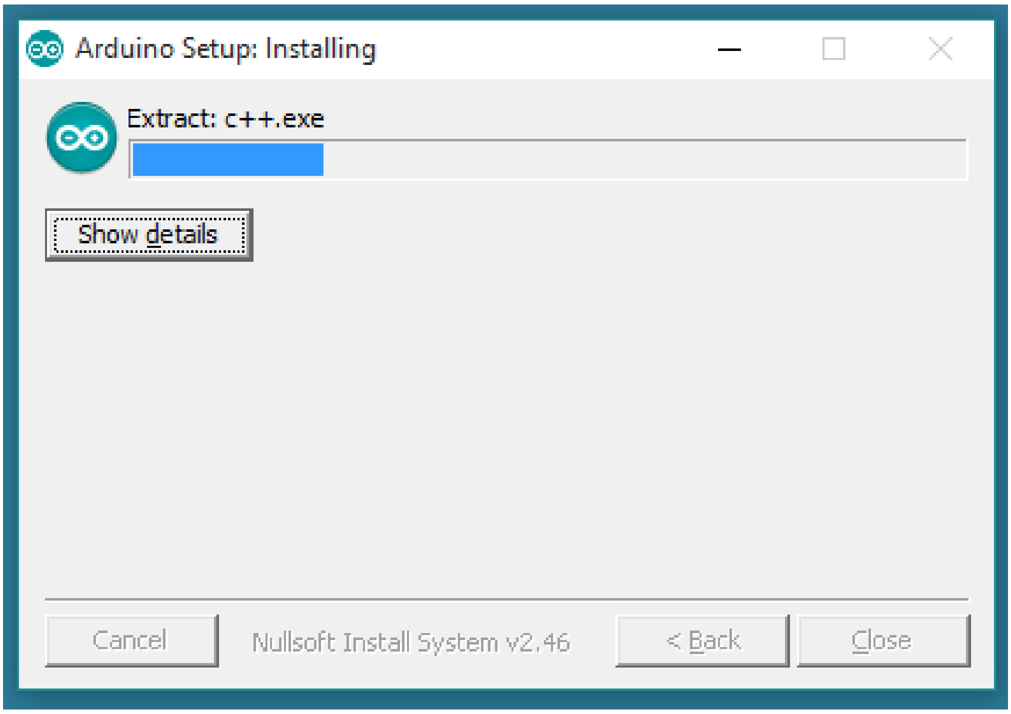


Fig:1.9 Installing directory

6. Successfully completed Installation of Software

7. Writing code on IDE platform

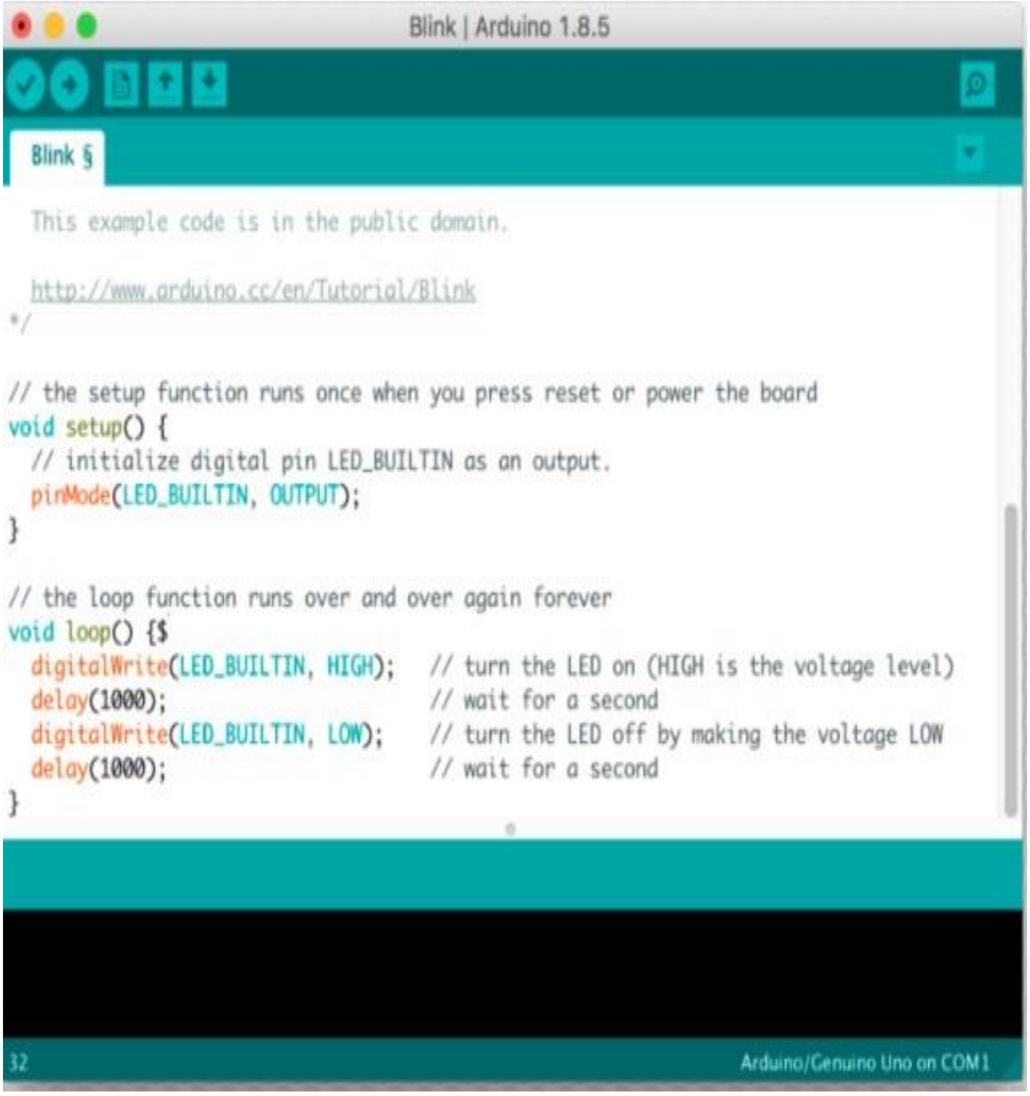


Fig:1.10 Source code written on IDE platform

1.3.2.2 PROTEUS SOFTWARE:

Through this software Schematic implementation of the embedded system will be done.

STEPS FOR DESIGNING SYSTEM ON PROTEUS DESIGN SUITE:

Step 1: Open ISIS software and select new design in File menu

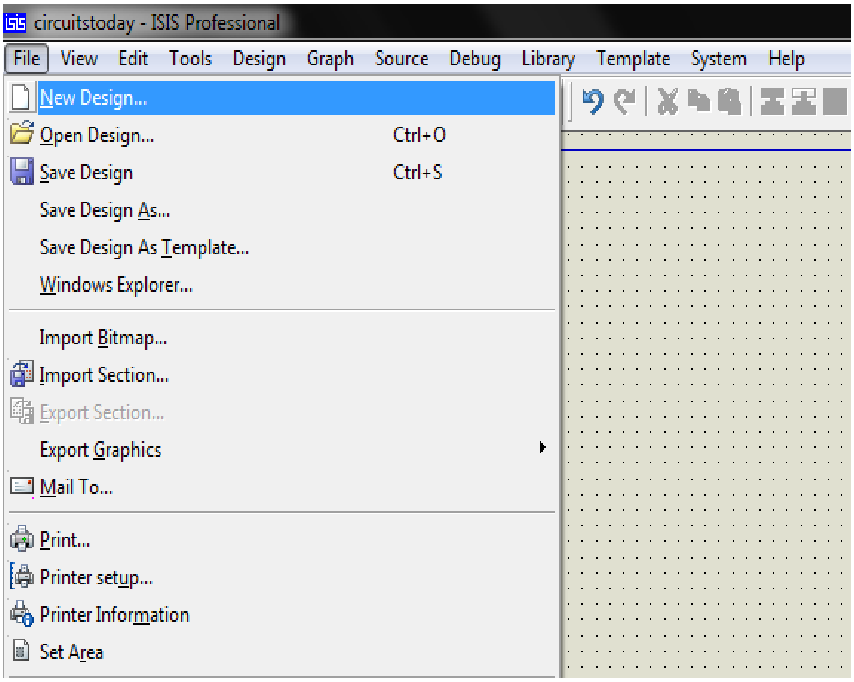


Fig:1.11 Simulation Procedure 1

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

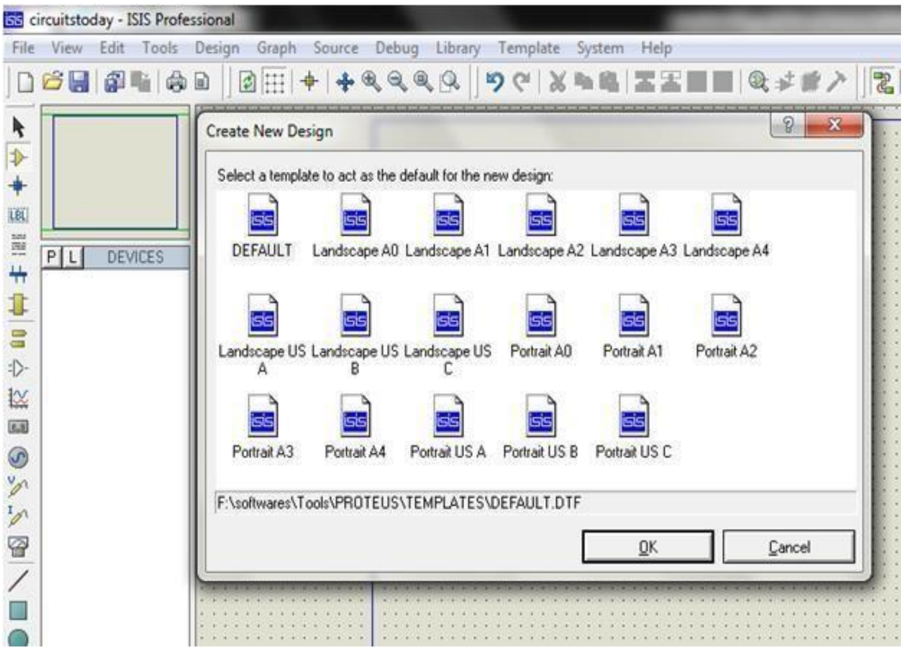


Fig:1.12 Simulation Procedure 2

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create new folder for every layout as it generates other files supporting your design.

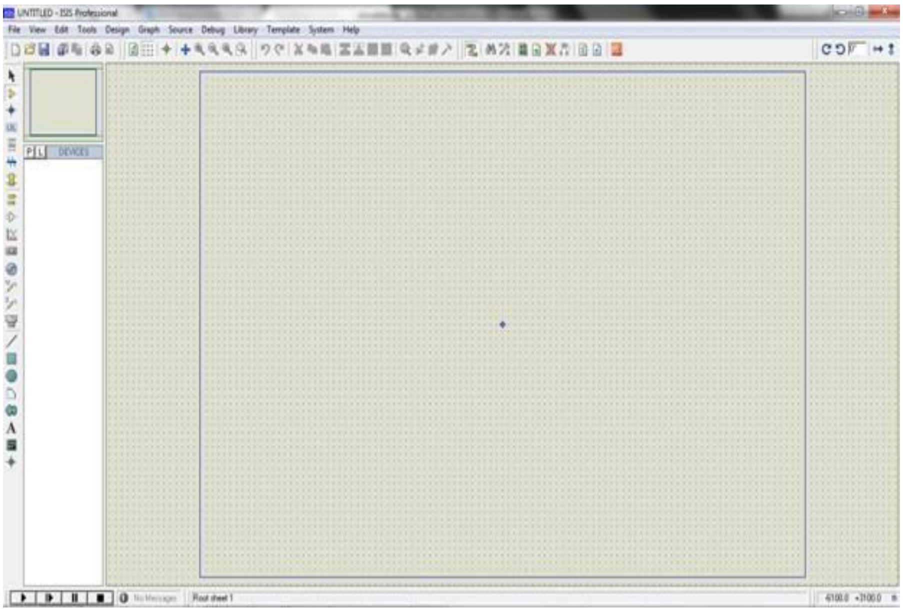


Fig:1.13 Simulation Procedure 3

Step 4: To Select components, Click on the component mode button.

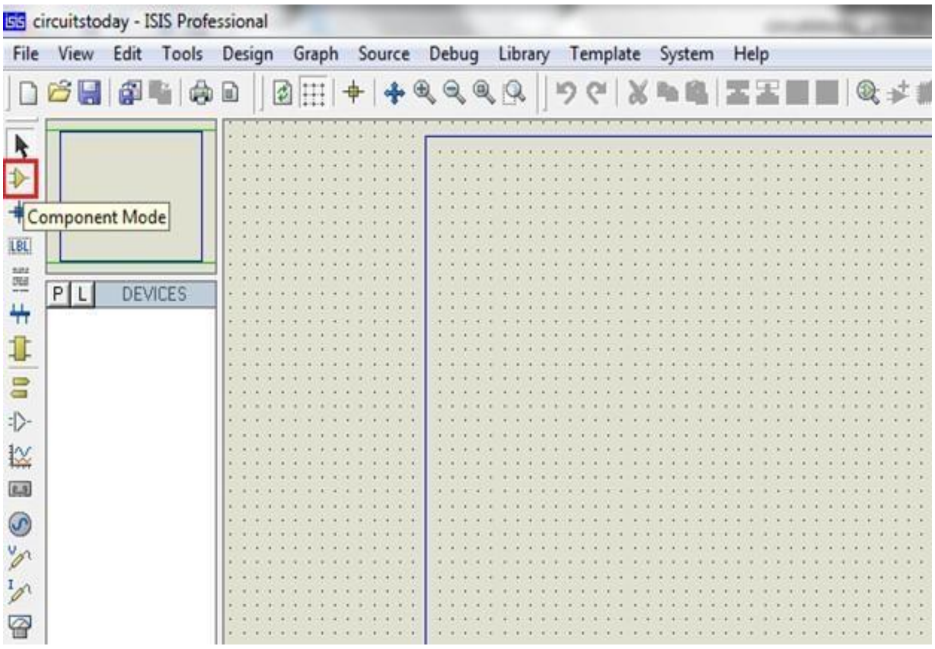


Fig:1.14 Simulation Procedure 4

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

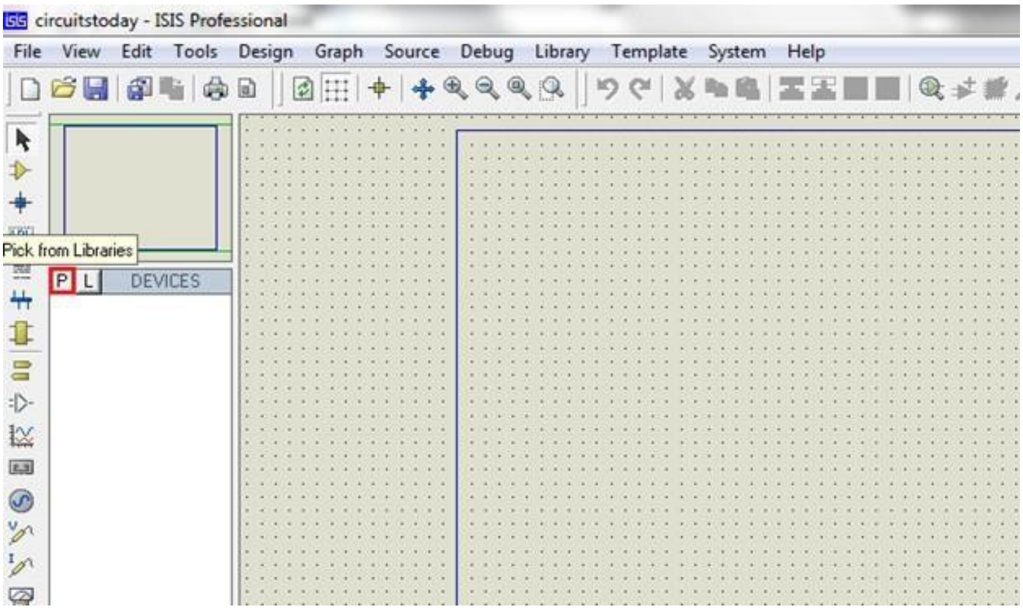


Fig:1.15 Simulation Procedure 5

Step 6: Select the components from categories or type the part name in Keyword's text box.

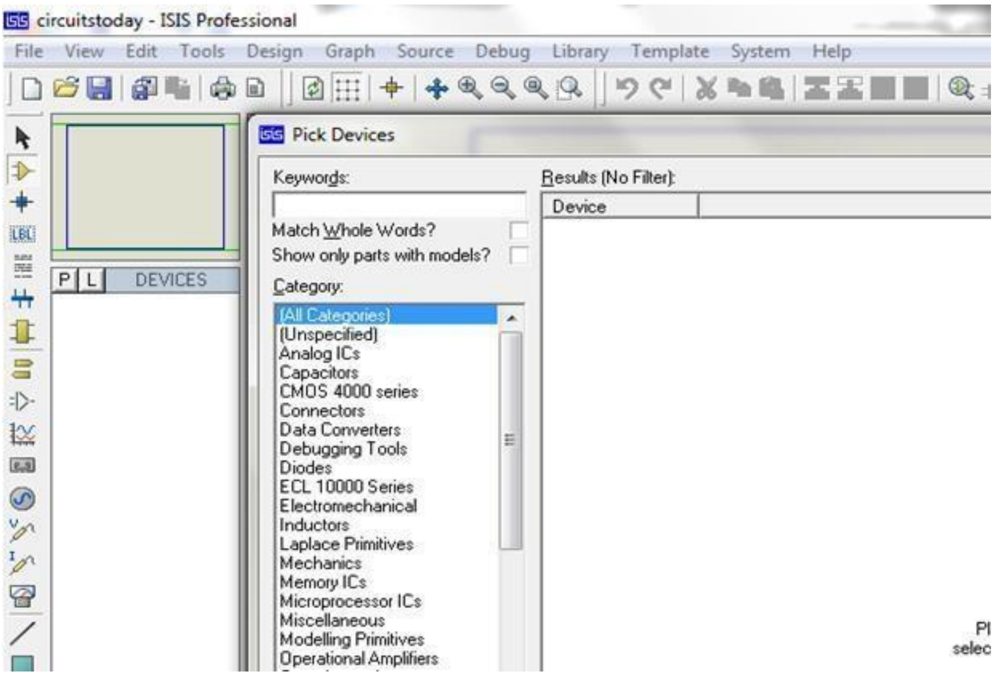


Fig:1.16 Simulation Procedure 6

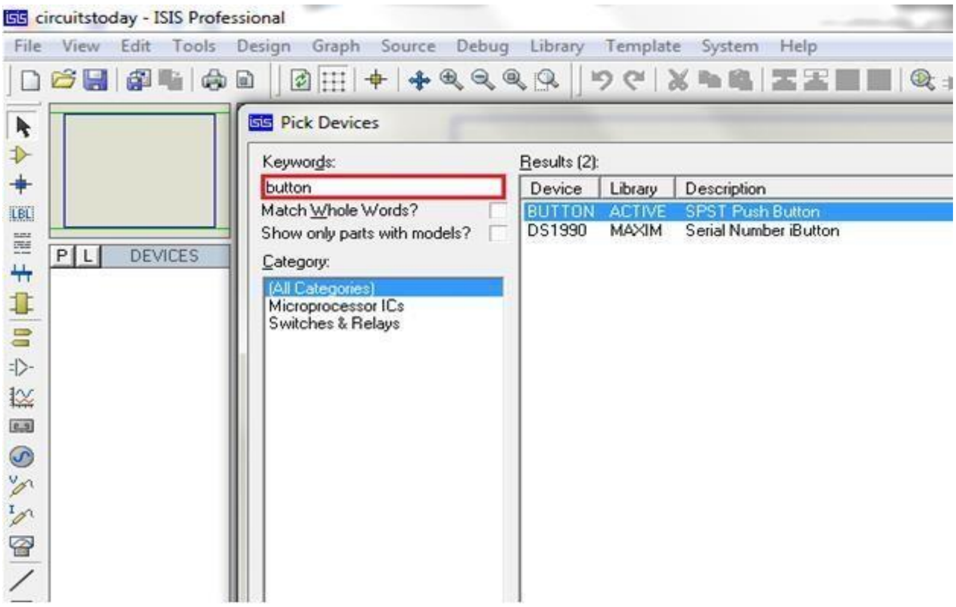


Fig:1.17 Simulation Procedure 7

Step 7**:** The selected components will appear in the devices list. Select the component and place it in the design sheet by left-cl.

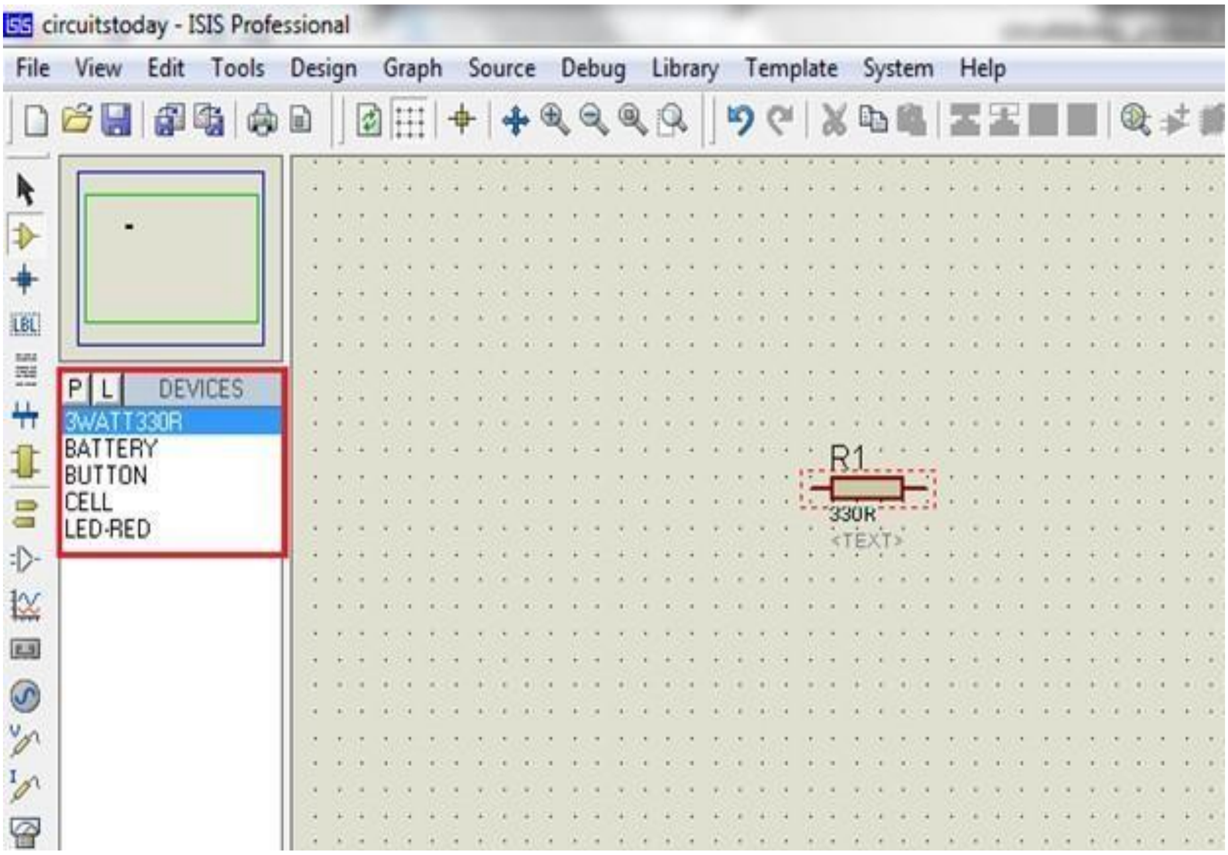


Fig:1.18 Simulation Procedure 8

Step 8: Place all the required components and route the wires i.e., make connections, either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

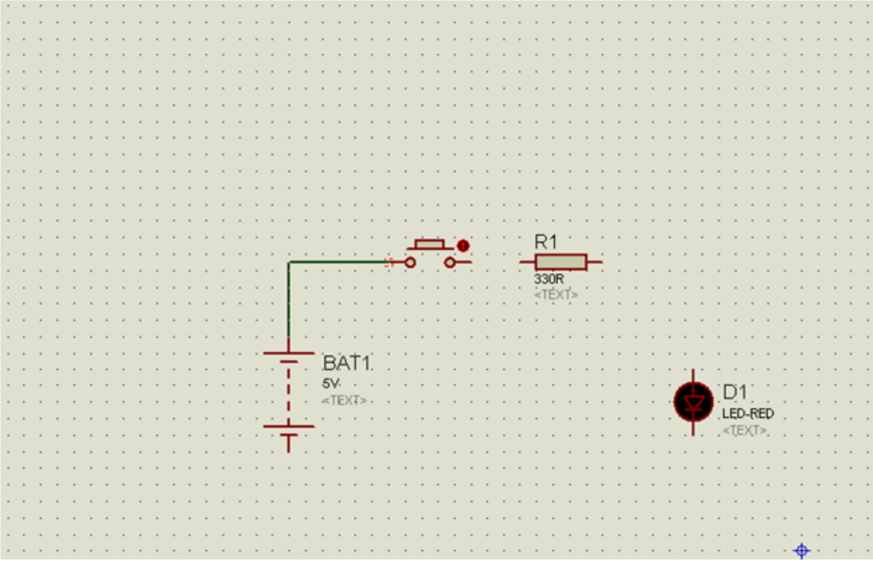


Fig:1.19 Simulation Procedure 9

Step 9: Double click on the component to edit the properties of the components and click on Ok.

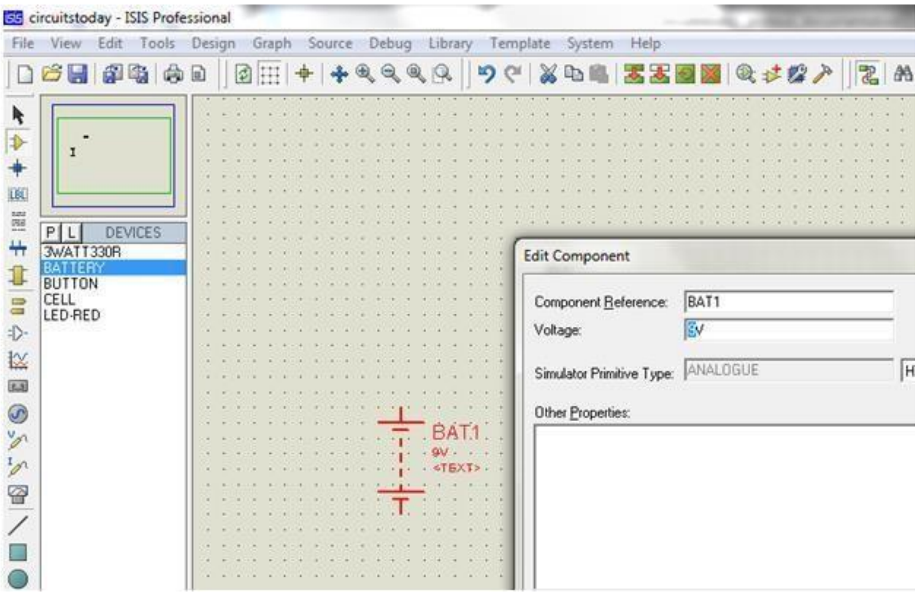


Fig.1.20 Simulation Procedure 10

Step 10: In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

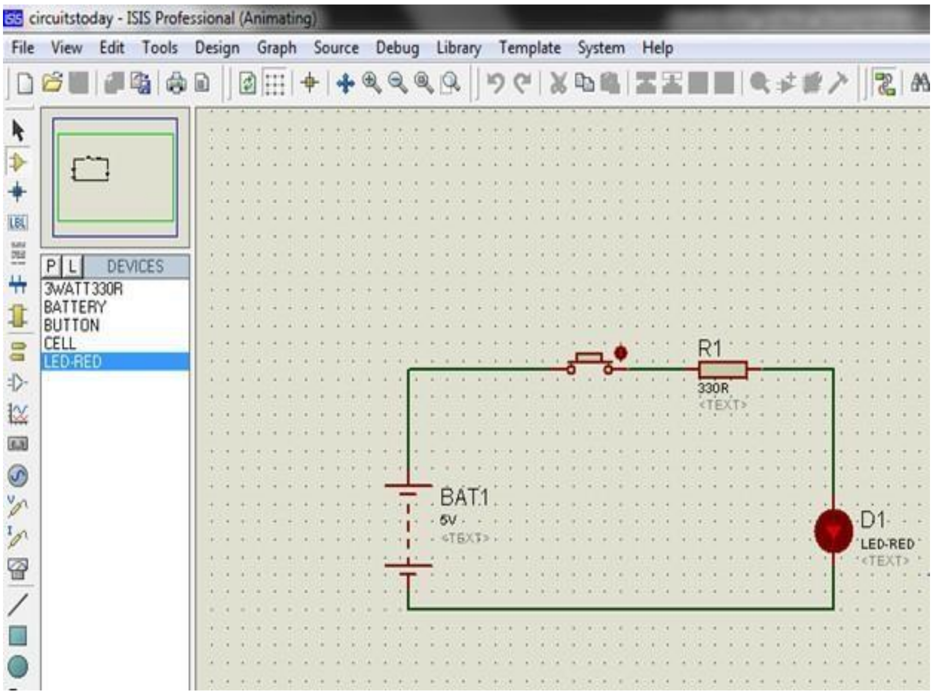


Fig:1.21 Simulation Procedure 11

Step 11: Simulation can be stepped, paused or stopped at any time.

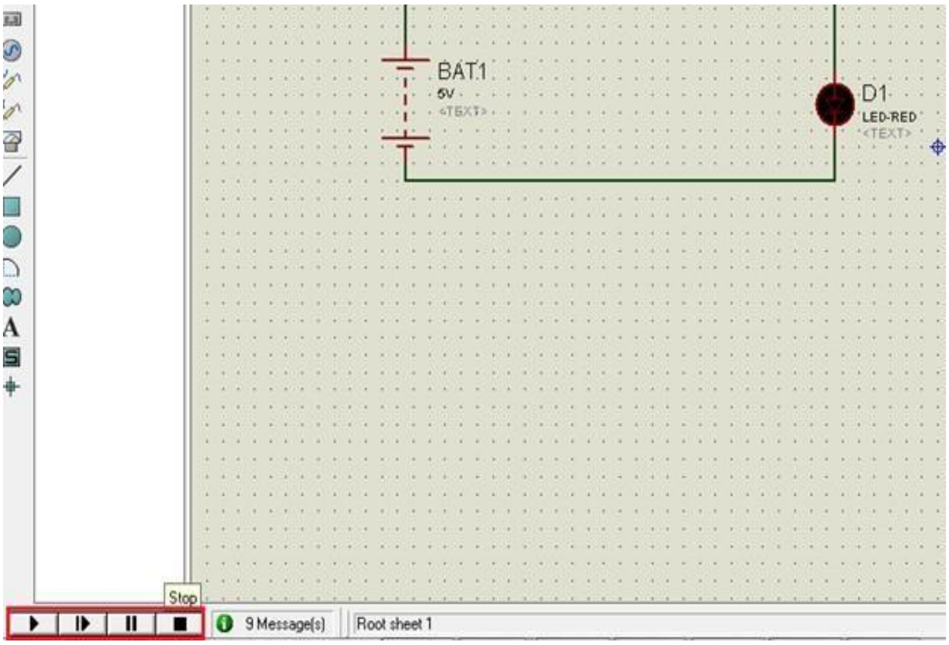


Fig:1.22 Simulation Procedure 12

1.4 PROCURMENT OF EQUIPMENT:

Fingerprints—the small ridges of skin on the fingertips which may enhance texture perception and grip friction—are unique to each person and have been used in identification for more than 100 years. Identification, the process of comparing a set of prints or a single print to others in a database, is used in many aspects of law enforcement and counterterrorism operations and investigations. Fingerprints may be compared to local and national databases for criminal history or to watch lists for border security. Latent prints from crime scenes are used in forensic investigations. Fingerprints can also be used to identify victims and for access control. Each application involves different processes for fingerprint capture and database searching and matching.

1.4.1 Components of Fingerprint Identification Systems

Fig:1.23 Finger print capture device

Fingerprint Capture the first step in fingerprint identification is obtaining an image of prints interest. Fingerprints deliberately collected from an individual are called exemplar prints, and may include simultaneous four finger images or rolled prints, which capture an image of each side the fingernail to the other. Exemplar prints have historically been obtained using paper and ink, which must then be optically scanned for database input. Live scanners electronically capture and store a digital image, eliminating paper and ink. Digitized prints are transmitted electronically for AFIS searches. Most current live scanners are based on optical sensors. Emerging technologies include thermal, capacitive, and ultrasound sensors, and some have to use a striped pattern projected onto the fingertip to obtain a three- dimensional image.

Minutiae Encoding

The second step in fingerprint identification is to extract the fingerprints which are identifible characteristics to be used in a database search. Fingerprints can be sorted into general classification by pattern type, such as arch, loop, or whorl, but minute details such as the ridge endings or burifications are unique to each individual Software identifies.

Recognition Software

An AFIS search involves the interaction of various databases where the minutiae are stored. Depending on the application, proprietary algorithms search segments of the database. For example, software may search the minutiae of two fingerprints of each subject, such as two index fingers, two thumbs, or a combination of finger and thumb. In other cases it may be necessary to search a database of all ten fingers. The simplest type of search is an identity verification used for access control or credentialing applications. Another identifier, such as a username, is submitted with a single fingerprint to specify who is attempting the authentication.

Certifications and Evaluations

A wide variety of fingerprint capture equipment and proprietary matching algorithms are commercially available. Interoperability is critical to the identification process and is achieved through standards and certification. The FBI has established image quality specifications for fingerprint images as well as data compression requirements for storage and transmission. The FBI certifies equipment compatible with the IAFIS, including live scanners, card scanners, and printers are listed.

Standards

Two standards are currently used in FBI certification. The image quality specifications for interoperability with IAFIS for fingerprint scanners and printers are contained in Appendix F of IAFIS- DOC-01078-9.3 Criminal Justice Information Services Electronic Biometric Transmission Specification. Fingerprint verification using 1:1 matching is the focus of the standard PIV-071006.

Development of national and international standards for fingerprint technology is ongoing, and many other standards are available. For example, ANSI/INCITS 381-2004 specifies data compression for image storage and transmission, and ANSI/INCITS 378-2009 defines a fingerprint standard minutiae template.

1.5 ORGANISATION OF THE CHAPTERS

The Overall Overview of Documentation is:

1.5.1 INTRODUCTION

1.5.2 LITERATURE SURVEY

1.5.3 PROJECT DESIGN

1.5.4 PROJECT IMPLEMENTATION

1.5.5 PROJECT TESTING

1.5.6 CONCLUSION AND FUTURE ENHANCEMENT

CHAPTER 2
LITERATURE SURVEY

This chapter gives a brief overview of previous works done related to Fingerprint system.

2.1 INTRODUCTION:

The main idea of our project is to secure the places by using fingerprint sensor used to scan the finger of n the persons, the authorized people have to register their fingerprints to the system by using fingerprint sensor and each person is provided a particular ID which is stored in the data, Authorized people are the only ones who can open the door, while the remaining people can't. For more security, the project is done along with buzzer indicator and GSM system.

2.2 RELATED WORK:

Biometrics is a way used to recognize a person based on his physical nature. The fingerprint, iris, face, voice, etc. are the mainly used biometrics to recognize a person. There are two key functions for biometrics, first is one to one matching and other is one too many matchings. In one to many matching the biometric sample is compared with the already stored samples. In one to one matching, it compares with the previously stored sample. Biometric method results in a faster security, and more convenient method for user verification. Biometric method is better than password security. Fingerprint is unique for each individual so it can be used as a mark of signature, verification and authentication. Fingerprint is the biometric which is used in this project. Finger-print will be different for each individual. In this project, fingerprint is used for the authentication of the user and allows him to cast vote based on his fingerprint image. Finger-print matching can be divided into three types: correlationbased matching, minutiae-based matching, pattern-based (or image-based) matching. In correlationbased matching, two fingerprint images are superimposed and therefore the correlation between corresponding pixels is computed for various alignments. In minutiae- based matching, minutiae from the two fingerprints are extracted and stored in a two-dimensional plane as a set. This matching method consists of finding the alignment between the template and the input minutiae sets that result in the maximum number of minutiae pairings. In pattern-based (or image-based) matching method it compares with stored template and the candidate's fingerprint. This requires that the images to be aligned within the same orientation. To do this, the algorithm finds a central point within the fingerprint image and centers there on. In a pattern-based algorithm, the template contains the sort, size, and orientation of patterns within the aligned fingerprint image. Almost all the sectors are storing data digitally. To create digital India, most of the tasks are made through on-line. When the voting is made on-line, it helps the voters to vote from anywhere in the world. Thingspeak is one of the ways which helps in making voting on-line. Obtaining online result makes the system faster. Traditionally voting was done by marking with stamp casting vote for the corresponding candidate and then dropping the paper to a ballot box. To calculate the number of vote

each vote must be calculated in each ballot box and then sum all the votes for each candidate and candidate who secured largest vote will be selected as the winner.

2.3 EXISTING METHODOLOGY:

In the existing system, most doors are controlled by persons with the use of keys, security cards, password or pattern to open the door. The aim of this project is to help users for improvement of the door security. With the advancement in the technology, Wireless controlling technique is proposed with a fine combination of new technologies and embedded system.

2.4 PROPOSED METHODOLOGY:

In the proposed system, we are introducing finger print module to access door lock. Here we have used MEMS (micro electro mechanical system, this signal is then sent to the Arduino. In this project Arduino act as a master and the remaining acts as slaves. Information from the Arduino is displayed on the LCD. The lock device is connected to the circuit door will be opened or closed depending on the command given. If any unauthorized people try to open the door it gives buzzing sound.

CHAPTER 3

PROJECT DESIGN

3.0 OVERVEIW OF THE SYSTEM DESIGN

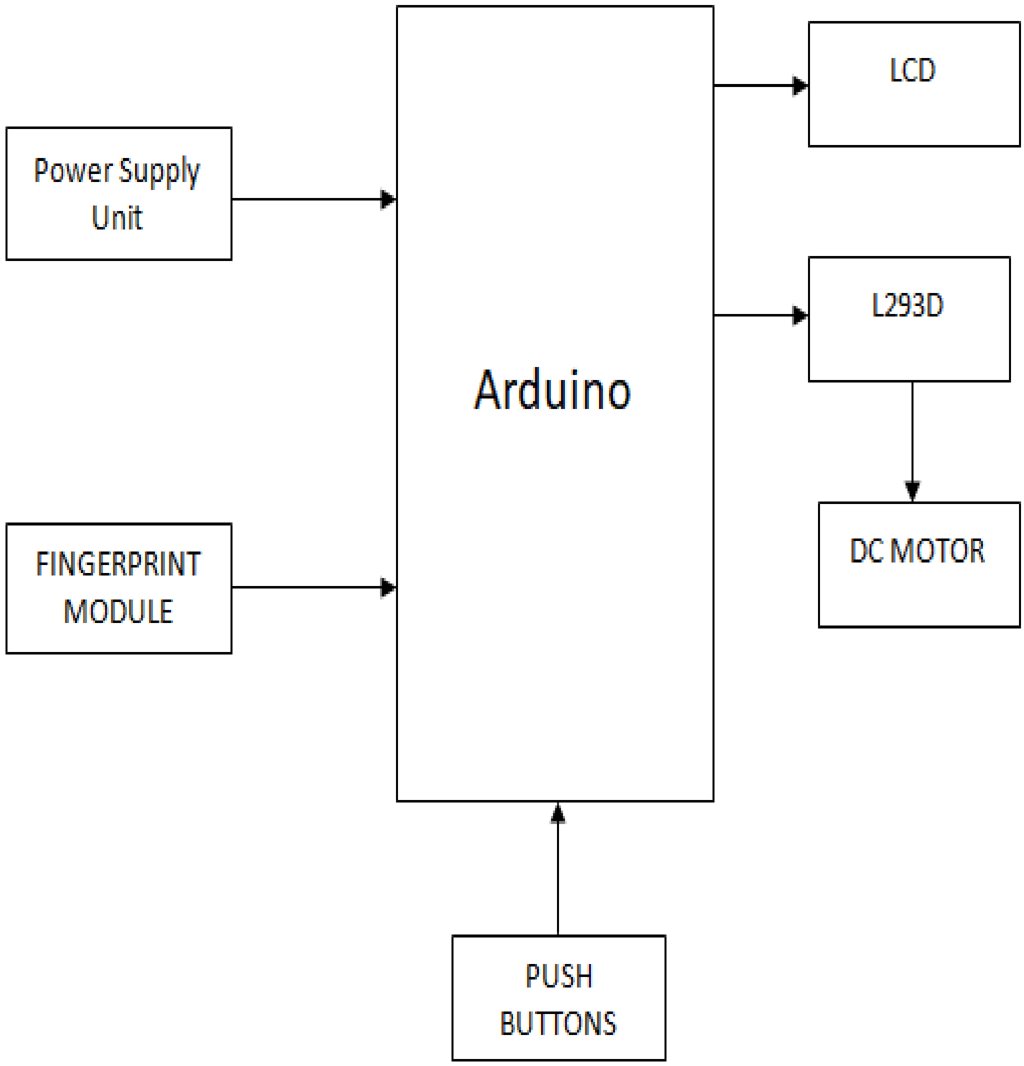


Fig:3.1 Block Diagram of finger print based security system

The block diagram of finger print based security is shown above which consists of fingerprint module, push buttons, Arduino UNO, power supply unit ,L293D IC,DC Motor and power supply unit.

3.1 SCHEMATIC DIAGRAM:

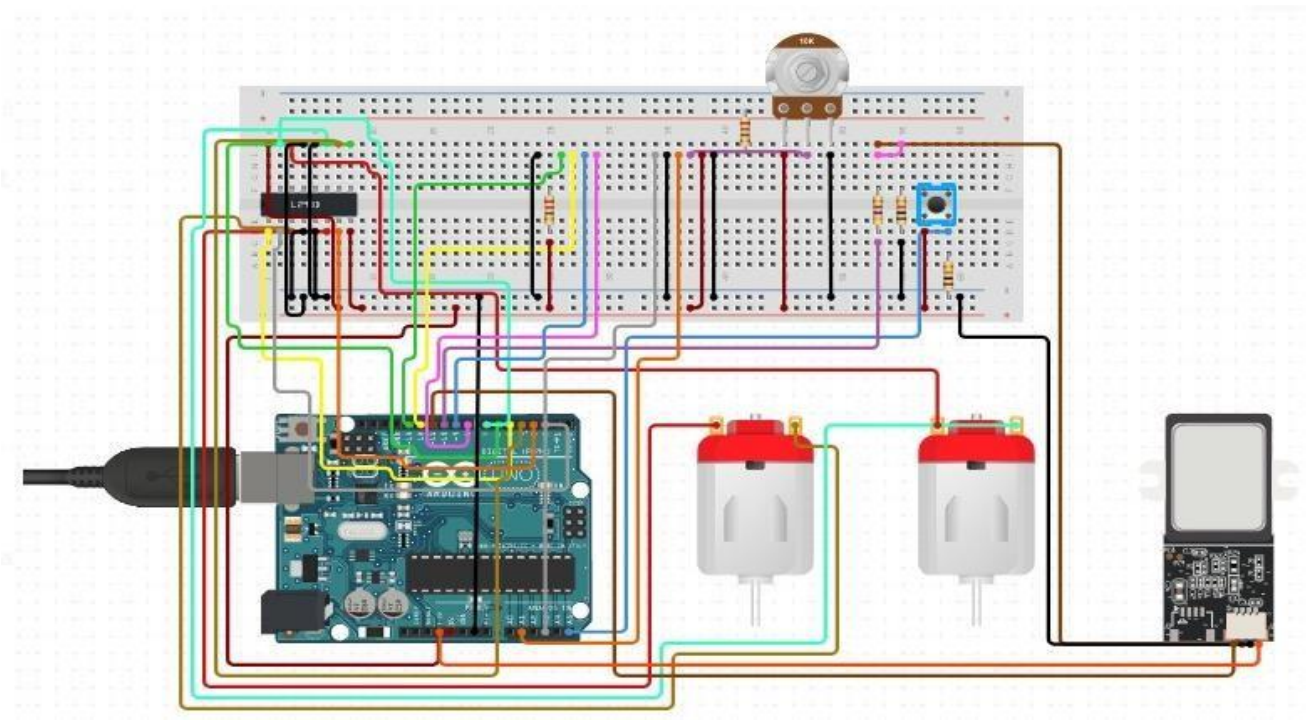


Fig:3.2 Simulation layout

The above schematic diagram of Finger Print Based Security System explains the interfacing section of each component with Arduino uno and input output modules.

3.2 MODULES FUNCTIONALITIES:

3.2.1 Arduino

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board. The key features are :

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

3.2.1.1 Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header

Table:3.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro-3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
Lily pad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table:3.2 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table:3.3 Arduino boards based on ATMEGA2560 microcontroller

Operating Volt	**Board Name**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**
3.3V	Arduino Mega 2560 R3	84MHz	54	12	12	4	USB native

Table:3.4 Arduino boards based on AT91SAM3X8E microcontroller

3.2.1.2 Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common

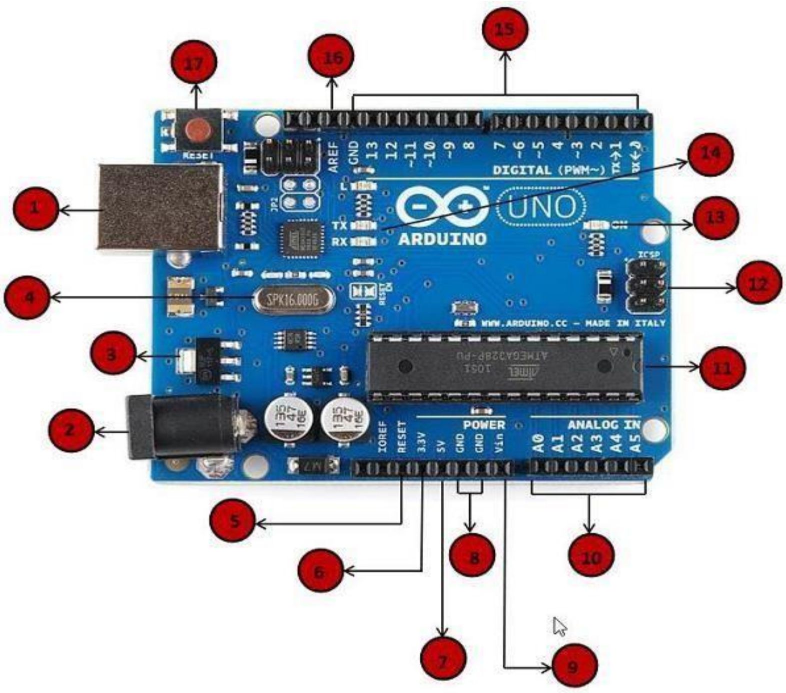


Fig:3.3 Board description of Arduino uno

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt Most of the components used with Arduino board works fine with 3.3 volt and 5 volts. GND (8) (Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.

10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

3.3 Liquid Crystal Display in LCD

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to a controller is an LCD display. Some of the most common LCDs connected to the controllers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines (RS RW D7 D6 D5 D6).

When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller

3.3.1 Description of 16x2 LCD:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all-Parallel Ports. It however doesn't show the use of the Status Port as an input. So, what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

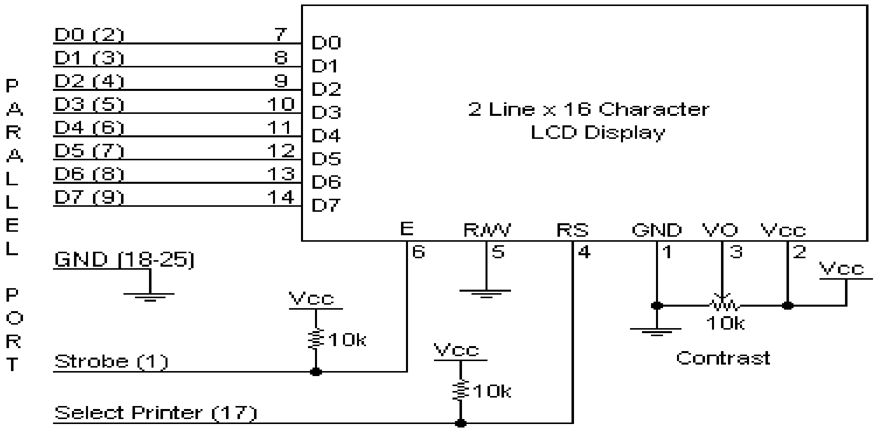


Fig 3.4: Schematic diagram of LCD

- Above is the quite simple schematic. The LCD panel's *Enable* and *Register Select* is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore, by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.
- We make no effort to place the Data bus into reverse direction. Therefore, we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result, we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.
- The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. You can use a bench power supply set to 5v or use an onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.

3.3.2 16 x 2 Alphanumeric LCD Module Features:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area.
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line.
- Can display 224 different symbols.
- Low power consumption (1 mA typical).
- Powerful command set and user-produced characters.
- TTL and CMOS compatible.
- Connector for standard 0.1-pitch pin headers.

Pin	Symbol	Level	Function
1	V_{SS}	-	Power, GND
2	V_{DD}	-	Power, 5V
3	Vo	-	Power, for LCD Drive
4	RS	H/L	Register Select Signal H: Data Input L: Instruction Input
5	R/W	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)
6	E	H, H->L	Enable

Table:3.5 16x2 LCD Features

3.3.3 FEATURES:

- 5 x 8 dots with cursor
- Built-in controller (KS 0066 or Equivalent)
- + 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)
- N.V. optional for + 3V power supply

Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are:

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

Table:3.6 Data Location in LCD

3.3.4 Figure Address locations for a 1x16 line LCD

Even limited to character-based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8,16,20,24,32 and 40 characters are all standard, in one-, two- and four-line versions.

Several different LC technologies exists. “supertwist” types, for example, offer Improved contrast and viewing angle over the older “twisted nematic” types. Some modules are available with back lighting, so that they can be viewed in dimly-lit conditions. The back lighting may be either “electro-luminescent”, requiring a high voltage inverter circuit, or simple LED illumination.

3.3.5 Pin Description In LCD:

Most LCDs with 1 controller have 14 Pins and LCDs with 2 controllers has 16 Pins (two pins are extra in both for back-light LED connections).

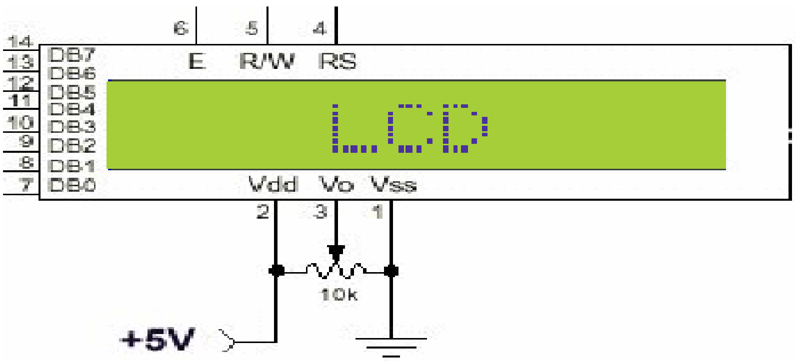


Fig:3.5 Pin Description of 1x16 lines LCD

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Table:3.7 Pin specifications

3.3.6 CONTROL LINES:

EN: Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS: Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW: Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands, so RW almost always be low. Finally, the data bus consists of 4 or 8 lines (depending on the mode of the selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0-DB8.

Logic status on control lines

- E - 0 Access to LCD disabled.
- 1 Access to LCD enabled.
- R/W - 0 Writing data to LCD.
- 1 Reading data from LCD.
- RS - 0 Instructions.
- 1 Character.

Writing data to the LCD

- Set R/W bit to low.
- Set RS bit to logic 0 or 1 (instruction or character).
- Set data to data lines (if it is writing).
- Set E line to high.
- Set E line to low.

Read data from data lines (if it is reading) on LCD

- Set R/W bit to high.
- Set RS bit to logic 0 or 1 (instruction or character).
- Set data to data lines (if it is writing).
- Set E line to high.
- Set E line to low.

Entering Text

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the d.i.l. switch pack with a of re-wiring is necessary.

LCD Commands

There are some present commands instructions in LCD, which we need to send to LCD through some microcontroller. Some important command instructions are given below

Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left
0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

Table:3.8 LCD Commands

3.4 Fingerprint Sensor

The fingerprint sensor can read different fingerprints and store in its own flash memory. The sensor can perform three functions namely Add (Enroll), Empty Database or search database and return the ID of stored fingerprint. Any of three functions can be called simply by making the pin low of the sensor or pressing onboard three switches. The response is either error or ok which is indicated by onboard LED. The response is also returned as single serial data byte.

The return byte is a valid ID or error code. The response byte is a single byte at 9600 bps thus making whole sensor very easy to use. We have provided indicating LEDs and function switch already so it’s ready to use when you receive it. Just give power and start using the sensor using onboard switches. Then you can move on making external application using these functions.

3.4.1 Inputs and Outputs of Sensor

Input: Two ways to trigger the function of fingerprint sensor

- Onboard switch: Add, Empty or Search.
- Make pin low from external microcontroller for 5ms as per function required to be executed.

Output: Two ways to monitor output response after a function is executed

- Onboard LEDs: ERROR or OK.
- Read byte after executing function.

3.4.2 Types of Sensor Function

There are namely three functions you can call for the fingerprint sensor. We will see each in brief.

Add (Enroll) Function: Adds a fingerprint to database and return a byte of newly added ID. Return values are from 0x00 to 0xFE. In case of error like no finger placed, return code is 0xFF. Here 0xFF means error executing function

Search Function: When a finger is put and search function is called, it returns a matching ID if found in its existing memory. Return values are from 0x00 to 0xFE. In case of error like no finger placed, return code is 0xFF. Here 0xFF means error executing function.

Empty Function: When you wish to empty all fingerprint, data stored on sensor you can use this function. After executing this function, you will get 0xCC as OK or 0xFF in case of error.

3.4.3 Fingerprint Recognition Sensor Module

The sensor is a solid-state fingerprint sensor that reliably captures fingerprint information. It is designed to integrate into devices for improved security and convenience. The sensor provides a reliable, quick and user-friendly alternative to passwords, PIN's and other forms of user authentication.

This fingerprint scanner is capable of gathering and storing unique finger prints. Simply hold your finger on the optical scanner, query the device over serial, and you will be issued a unique ID. Use that ID within your embedded system to determine access levels, time clocks, door locks, etc. Unit includes 4 pin connector cable to connect and read to controller. The outputs are TTL level serial data.

A biometric sensor, fingerprint sensor to be specific, also known as the fingerprint reader, is a fingerprint image capture device, the very front end of the biometric fingerprint identification/verification module. The fingerprint sensor captures the fingerprint images, matches the uniqueness of each print read by the sensor and compares it to the one stored in its module or local system database.

It consists of optical fingerprint sensor, high performance DSP processor and Flash. It boasts of functions such as fingerprint enrollment, deletion,updation,searching,scanning.

3.4.4 Applications

- Computer peripherals – improves security and convenience
- Transportation systems – validation of operators, drivers and inspectors
- Medical equipment – authorization of operator or technician
- Physical access systems – approval for entry
- Kiosks and vending machines – confirmation of person receiving the selection
- Point of Sale terminals – authentication of tellers and cashiers

3.4.5 Features

- Rugged, solid-state optical fingerprint sensor
- High resolution 500 DPI imager
- Adapts to wet/dry fingers
- Simple Interfacing protocol

Fig:3.6 Fingerprint sensor

3.4.6 Specifications of Sensor

- Fingerprint enrollment time <250ms
- Fingerprint search time <1s（100 fingerprint, average value in test） Resolutions 500 DPI
- Security level 5, Adjustable Capacity 768 Templates
- FAR <0.0001%
- FRR <0.01%
- Power Supply 4.3V to 6V Working Current <80mA Peak Current <90mA Communication Interface TTL
- Communication Baud Rate 57600 bps Working Temperature -10 deg C to +40 deg C
- Working Humidity 40% RH to 85% TH (no dew) Module Dimensions 60x21x25 mm (LxWxH)

3.5 BUZZER:

3.5.1 Types of Buzzer

Magnetic Transducer

Magnetic transducers contain a magnetic circuit consisting of a iron core with a wound coil and a yoke plate, a permanent magnet and a vibrating diaphragm with a movable iron piece. The diaphragm is slightly pulled towards the top of the core by the magnet's magnetic field. When a positive AC signal is applied, the current flowing through the excitation coil produces a fluctuating magnetic field, which causes the diaphragm to vibrate up and down, thus vibrating air. Resonance amplifies vibration through resonator consisting of sound hole(s) and cavity and produces a loud sound.

Magnetic Buzzer (Sounder)

Buzzers like the TMB-series are magnetic audible signal devices with built-in oscillating circuits. The construction combines an oscillation circuit unit with a detection coil, a drive coil and a magnetic transducer. Transistors, resistors, diodes and other small devices act as circuit devices for driving sound generators. With the application of voltage, current flows to the drive coil on primary side and to the detection coil on the secondary side. The amplification circuit, including the transistor and the feedback circuit, causes vibration. The oscillation current excites the coil and the unit generates an AC magnetic field corresponding to an oscillation frequency. This AC magnetic field magnetizes the yoke comprising the magnetic circuit. The oscillation from the intermittent magnetization prompts the vibration diaphragm to vibrate up and down, generating buzzer sounds through the resonator.

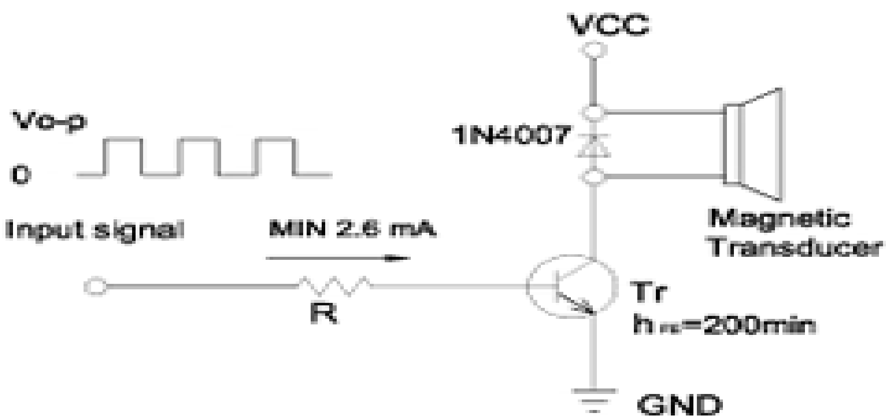


Fig:3.7 Driving Circuit for Magnetic Transducer

3.5.2 Specifications of buzzer:

- **Rated Voltage**: A magnetic buzzer is driven by 1/2 square waves (V o-p). **Operating Voltage**: For normal operating. But it is not guaranteed to make the minimum Sound Pressure Level (SPL) under the rated voltage.
- **Consumption Current**: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.
- **Direct Current Resistance**: The direct current resistance is measured by ammeter directly.
- **Sound Output**: The sound output is measured by decibel meter. Applying rated voltage and 1/2 square waves, and the distance of 10 cm.
- **Rated Frequency**: A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.
- **Operating Temp**: Keep working well between -30℃ and +70℃.
- **Driving methods**: AX series with built drive circuit will be the best choice when we cannot provide frequency signal to a buzzer, it only needs direct current.
- **Dimension**: Dimension affects frequency, small size result in high frequency.
- **Voltage**: Depend on V o-p (1/2 square waves)
- **Fixed methods**: From the highest cost to the lowest- DIP, wires/ connector, SMD.

3.5.3 How to choose a buzzer

There are many different kinds of buzzer to choose, first we need to know a few parameters, such as voltage, current, drive method, dimension, mounting type, and the most important thing is how much SPL and frequency we want.

Operating voltage: Normally, the operating voltage for a magnetic buzzer is from 1.5V to 24V, for a piezo buzzer is from 3V to 220V. However, in order to get enough SPL, we suggest giving at least 9V to drive a piezo buzzer.

Consumption current: According to the different voltage, the consumption current of a magnetic buzzer is from dozens to hundreds of mill amperes; oppositely, the piezo type much more electricity, only needs a few mill amperes, and consumes three times current when the buzzer starts to work.

Driving method: Both magnetic and piezo buzzer have self-drive type to choose. Because of the int set drive circuit, the self-drive buzzer can emit sound as long as connecting with the direct current. to the different work principle, the magnetic buzzer needs to be driven by 1/2 square waves, and the buzzer need square waves to get better sound output.

Dimension: The dimension of the buzzer affects its SPL and the frequency, the dimension of the magnetic buzzer is from 7 mm to 25 mm; the piezo buzzer is from 12 mm to 50 mm, or bigger.

Connecting way: Dip type, Wire type, SMD type, and screwed type for big piezo buzzer are usually seen.

Sound Pressure Level (SPL): Buzzer is usually tested the SPL at the distance of 10 cm, if distance double, the SPL will decay about 6 dB; oppositely, the SPL will increase 6 dB when the distance is shortened by one time. The SPL of the magnetic buzzer can reach to around 85 dB/ 10 cm; the piezo buzzer can be designed to emit very loud sound, for example, the common siren, are mostly made of silicon.

Fig:3.8 Buzzer

The above shown figure is buzzer

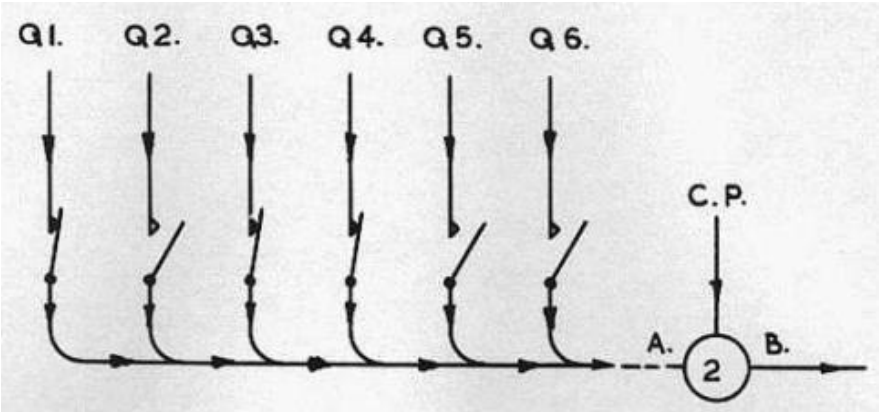


Fig:3.9 Pin Diagram of buzzer

The above shown figure is the diagram of a buzzer

3.6 Switches

A switch is an electrical component that can break an electrical circuit, interrupting the current or diverting it from one conductor to another. The most familiar form of switch is a manually operated electromechanical device with one or more sets of electrical contacts. Each set of contacts can be in one of two states: either 'closed' meaning the contacts are touching and electricity can flow between them, or 'open', meaning the contacts are separated and non-conducting.

Fig:3.10 Switches in a row

Switches with larger numbers of poles or throws can be described by replacing the "S" or "D" with a number or in some cases the letter "T" (for "triple"). In the rest of this article the terms *SPST*, SPDT and intermediate will be used to avoid the ambiguity in the use of the word "way".

3.6.1 Working of Switch

A pair of contacts is said to be "closed" when current can flow from one to the other. When the contacts are separated by an insulating air gap, an air space, they are said to be "open", and no current can flow at typical voltages.

Switches are classified according to the arrangement of their contacts in electronics. Electricians installing building wiring use different nomenclature, such as "one-way", "two- way", "three-way" and "four-way" switches

In a push-button type switch, in which the contacts remain in one state unless actuated, the contacts can either be normally open (abbreviated "no." or "no") until closed by operation of the switch, or normally closed ("n.c. or "nc") and opened by the switch action. A switch with both types of contact is called a changeover switch. These may be "make-before- break" which momentarily connect both circuits, or may be "break-before-make" which interrupts one circuit before closing the other

3.6.1 APPLICATIONS

- They are used for the given instructions to the required device like robot, driving the motor, etc.
- Switches can arrange in the matrix from or in the required from according to the application.
- Reset circuit of the controller, motor also has the switches for its particular functioning.

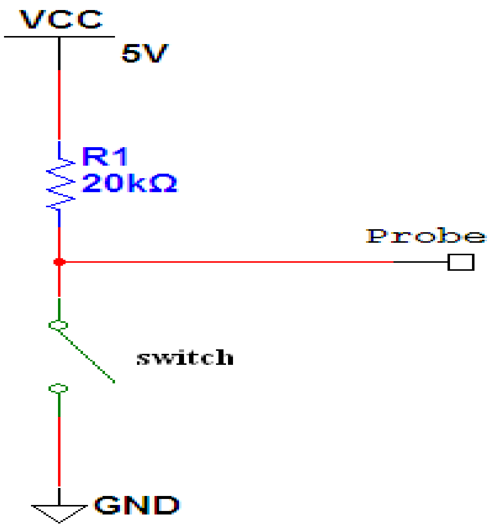


Fig:3.11 Circuit of Switch

From the above circuit it explains the working of the switch when supply is given to the circuit the resistor takes the current and oppose to the certain extend until the switch is pressed. These switches can be connected parallel or serial in required manner.

We can have resistor, capacitor, light resistor diode etc. as the combination of the switch. For example, w the reset circuit in which we the resistor and capacitor as the combination.

CHAPTER 4
PROJECT IMPLEMENTATION

4.0 IMPLEMENTATION STAGES:

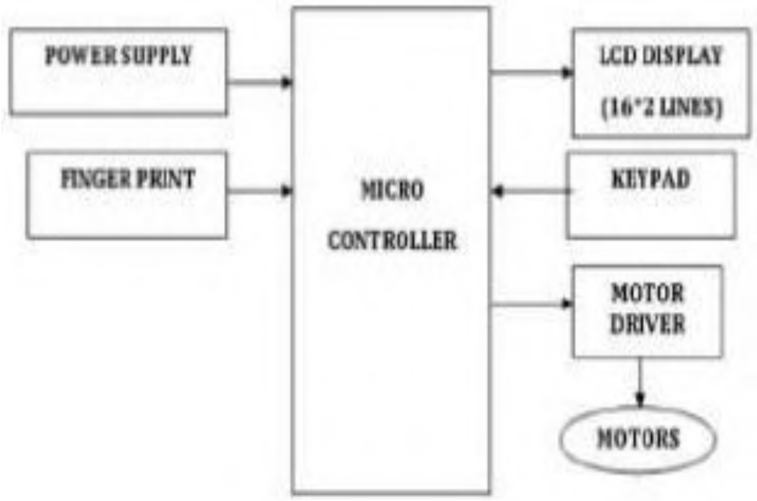


Fig:4.1 Finger print based security sensor

Here we use two microcontrollers to minimize the fee of the locker in view that these microcontrollers are very commercial so that any person can afford the locker. The foremost a part of this venture is the finger print module without which user can't open his/her locker. The fingerprint we used is an optical scanner; the heart of this scanner is a charged coupled gadget (CCD). The CCD has an array of light touchy diodes, referred to as picture sites. These photograph websites generate an electrical in response of light alerts. These alerts are saved in the type of dark and lights pixels for ridges and valleys respectively within the fingerprint module, these darkish and lightweight pixels are used to differentiate between specific fingerprints.

Stepwise implementation of finger print based security system is given under:
Step 1: Enter your password with the keypad.
Step 2: Now, scan your fingerprint on the fingerprint scanner. In case your fingerprint is not matched then snapshot seize by way of the digicam module and retailer in the pc system.
Step 3: If the password and fingerprint are of approved individual then the door attached to the dcmotor will open.
Step 4: Now, that you can entry to your locker.

4.1 WORKING FLOW OF THE PROJECT:

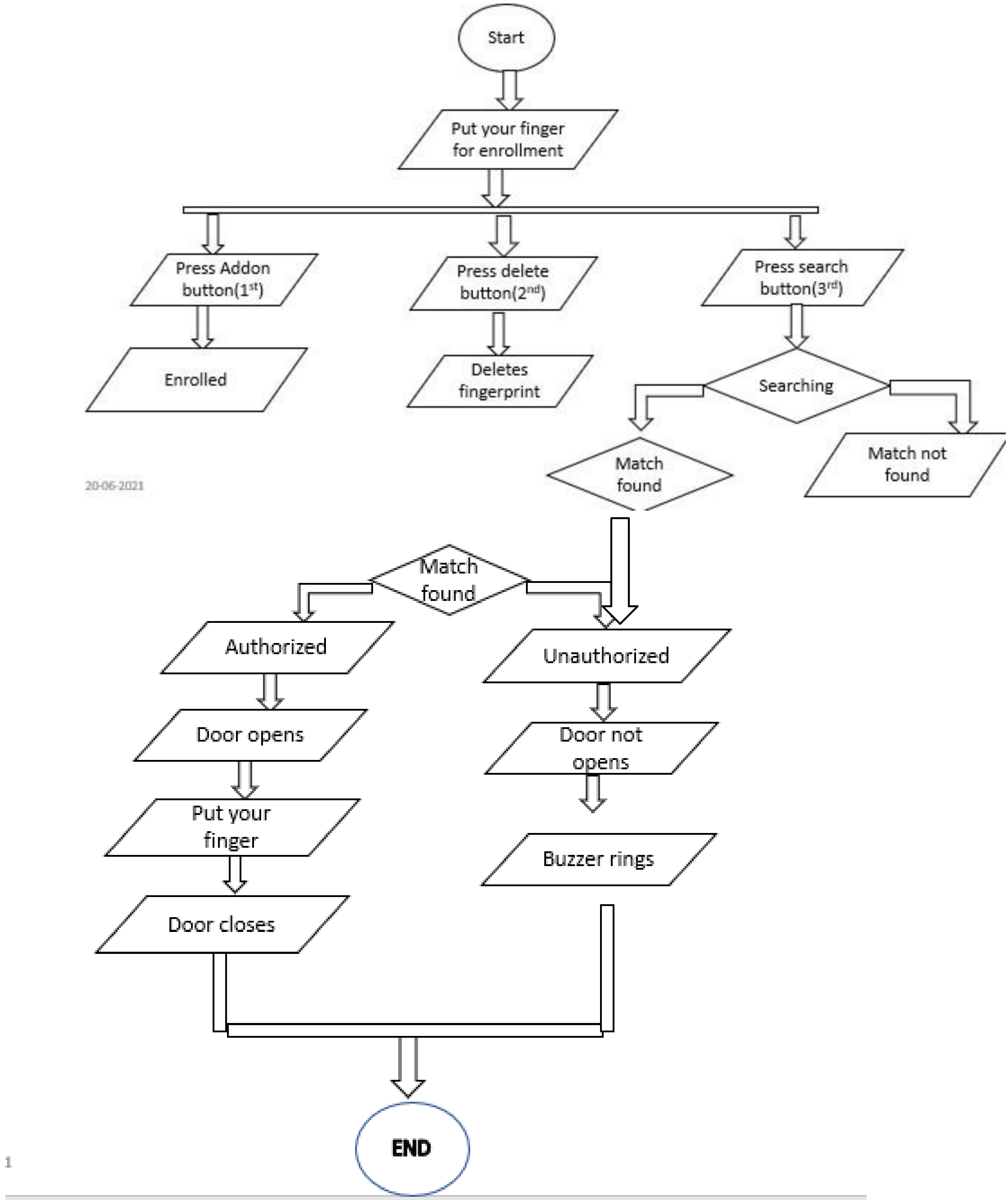


Fig:4.2 Working flow of Finger Print Based Security System

The above figure is the working flow of Finger print based security system.

4.2 EXPERIMENTAL RESULTS:

4.2.1 INTRODUCTION:

Security is a major concern in our day-to-day life, and digital locks have become an important part of these security systems. There are many types of security systems available to secure our place. Some examples are PIR based Security System, RFID based Security System, Digital Lock System, systems, Electronics Code lock. In this post, we will Interface a Fingerprint Sensor Module with Arduino and will build a Fingerprint based Biometric Security System with door locking. Fingerprint is considered one of the safest keys to lock or unlock any system as it can recognize any person uniquely and can't be copied easily. In the proposed system, we are introducing finger print module to access door lock. Here we have used MEMS (micro electro mechanical sensor), which is used to send message to user through GSM if any theft happens. This signal is then sent to the microcontroller. In this project microcontroller Arduino act as a master and the remaining acts as slaves. Information from the micro controller is displayed on the LCD. Fingerprint Sensor Module or Fingerprint Scanner is a module which captures finger's print image and then converts it into the equivalent template and saves them into its memory on selected ID (location) by Arduino.

This is a perfect/optimal solution for saving/protecting one from the hassle of stolen/lost key or an unauthorized entry. Fingerprint is a boon solution for these problems which provides high level of recognition accuracy. The skin on our palms and soles exhibits a flow like pattern of ridges called friction ridges. The pattern of friction ridges on each finger is unique and immutable. This makes fingerprint a unique identification for everyone.

Here all the process is commanded by Arduino like taking an image of fingerprint, convert it into templates and storing location etc. Our proposed system overcomes all the security problems in existing system and provides high security and efficiency.

4.2.2 PROJECT OUTPUT RESPONSE:

The general view of the projected is illustrated in the following figure, followed by the expected performance of the project. each figure summarizes one implementation step that has to take place in the project, each figure shows the output by using LCD, and all the comments appear in the console of LCD, those comments help the user to deal with system.

General view of the Project

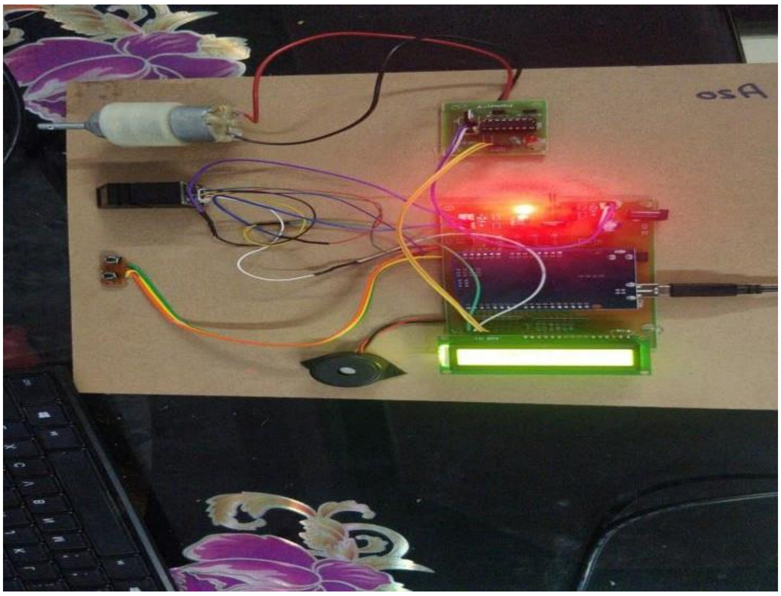


Fig:4.3 Overall system

The above figure is Overall System of Finger print based Security System

4.2.3 INTERPRETING EXPERIMENTAL RESULTS:

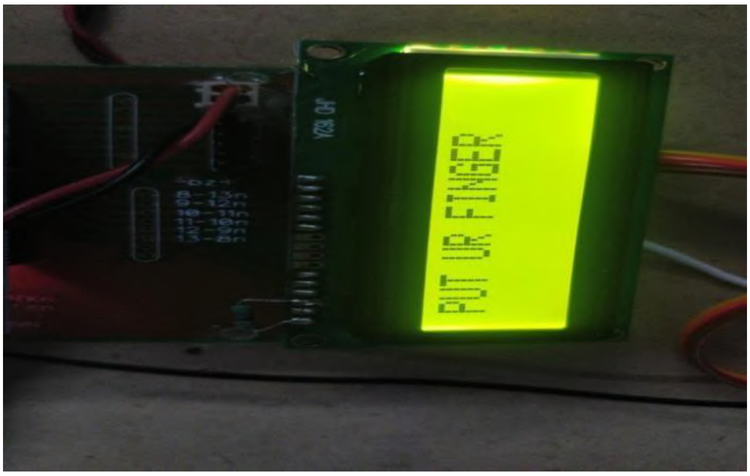


Fig:4.4 Displays Commands in LCD

The above figure displays commands in LCD

Fig 4.5 Finger print Scanner

The above shown figure is Finger Print Scanner

Fig:4.6 Displays on LCD to open the door

The above shown figure is it displays the open door Command in LCD.

Fig:4.7 Command on LCD to close the opened door

The above shown figure is it displays the close door Command in LCD.

CHAPTER 5
ADVANTAGES AND LIMITATIONS

5.1 ADVANTAGES

- Fingerprint based security system is the most secured system as compared to other systems. Reason is that RFID card or Keys of lock can be stolen, password may be leaked. However, thumbnail of every human being is unique, so lock will not open unless the same person is present to give the impression of fingerprint.
- No need to carry the keys to open the lock. Or even there is no need to remember the password or any Pin number.
- One of the main advantages is that this system remembers the stored password even if the power supply is turned off.
- Scientific research and studies have proved that fingerprints do not change as you grow up.
- Using Fingerprint saves time to gain access as compared to other methods like RFID card, Password or Key.

5.2 LIMITATIONS

- **System failures –** Scanners are subject to the same technical failures and limitations as all other electronic identification systems such as power outages, errors and environmental factors.
- **Cost** – it is true that fingerprint recognition systems are more cost effective than ever, but for smaller organisations the cost of implementation and maintenance can still be a barrier to implementation. This disadvantage is lessening as devices become more cost effective and affordable.
- **Exclusions** – while fingerprints remain relatively stable over a person's lifetime there are sections of the population that will be excluded from using the system. For example, older people with a history of manual work may struggle to register worn prints into a system or people who have suffered the loss of fingers or hands would be excluded.

5.3 APPLICATIONS

- **Industrial application:** "Fingerprint-based security system" project can be used by the employees, staff or workers in various industries like Automobile industries, manufacturing industries, Software development companies.

- **Home or domestic application:** This project can be used to automate the door locking process at our home, so the user needs not to carry the door lock keys along with him, he can just use his/her finger to open the door

- **Bank Lockers or security safes:** Many of the banks use key based or password-based locks for their lockers or safes. We can implement Fingerprint based bank locker system using this project.

5.4 FURTHER ENHANCEMENT OF THE PROJECT

- To this project we can add an GSM (global system monitoring) in sending alert messages to the owner if unauthorized person tries to enter the room.

- Touchless fingerprint scanner can be used, so that there is no need to touch the scanner with your finger which ensures some safety on virus attacks.

5.5 PROJECT CODE

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial (2,3); */

#include <LiquidCrystal.h>
LiquidCrystal lcd (13,12,11,10,9,8) ;//rs//en//d7, d8,d11,d12=datapins //from arduino txd connected to lcd rxd

char buff [200], k=0;
char res [130];


unsignedchar enroll[12]
={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X03,0X01,0X00,0X05};  // ok
unsignedchargenerate_ch[13]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X04,0x02,0X01,0X00,0X08};  //ok
unsignedchargenerate_ch1[13]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X04,0x02,0X02,0X00,0X09}; //ok
unsigned char un_cmd[12]={0xef,0x01,0xff,0xff,0xff,0xff,
0x01,0x00,0x03,0x05,0x00,0x09 };
unsigned char
store[12]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X06,0X06,0X02,0x00};
//ok
unsignedcharidentify[17]={0xef,0x01,0xff,0xff,0xff,0xff,0x01,0x00,0x08,0x1b,0x01,0x00,0x00,0x01,0x01,0x00,0x27};


void serialFlush(){//to read data from fingerprint scanner in form of 0 and 1s
  while(Serial.available() > 0) {
    char t = Serial.read();
  }
}
```

```
int fpenroll(char);//to enroll
int fpsearch();//to search for fingerprint

  int s1=5,s2=6,m1=2,m2=3,buz=7;
  void setup() {
  char ret;
  pinMode(s1, INPUT_PULLUP); //increase voltage level//s1,s2 switches//s1 for enrolling
//s2 for searching of enrolled one
  pinMode(s2, INPUT_PULLUP);
  pinMode(m1, OUTPUT);//motor connected to m1,m2
  pinMode(m2, OUTPUT);
  pinMode(buz, OUTPUT); //buzz at negative logic

    digitalWrite(buz, HIGH);
    digitalWrite(m1,LOW);
    digitalWrite(m2,LOW);
    Serial.begin(9600);//the bordrate of fingerprint scanner//the rate at which it can read
inputs//data transfered to arduino through scom with this rate
 // mySerial.begin(57600);//57600


  lcd.begin(16,2);
  lcd.clear();lcd.setCursor(0, 0);lcdPrint("FINGER PRINT BASED DOOR LOCK");


}
 int err =0;
 int idk = 0,eid=0;
void loop()
{
  //digitalWrite(buz,LOW);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("PUT UR FINGER"); delay(1000);
if(digitalRead(s1) == 0)//active low
{
  //Serial.println("Enrolling");
  lcd.clear();lcd.setCursor(0, 0);lcd.print("ENROLLING..");
  if(fpenroll(eid) == -1)//-1=null//if enrollment failed
```

```
  {
   // Serial.print("Enroll failed:");Serial.print(err);Serial.println("");
    err=0;
    lcd.clear();lcd.setCursor(0, 0);lcd.print("ENROLL FAILED");
    digitalWrite(buz,HIGH);delay(1000);digitalWrite(buz,LOW);
  }
  else
  {
    lcd.clear();lcd.setCursor(0, 0);lcd.print("ENROLLED:");lcd.print((int)eid);
    //Serial.print("Enroll Success to id:");Serial.print((int)eid);Serial.println("");
    //Serial.print("*E");Serial.print((int)eid);Serial.println("#");
    eid++;
  }
  delay(2000);
 // lcd.clear();lcd.setCursor(0, 0);lcd.print("SELECT OPTION");
}
if(digitalRead(s2) == 0)//identify//active low
{
  lcd.clear();lcd.setCursor(0, 0);lcd.print("SEARCHING..");
    idk = fpsearch();
    if(idk == -1)
    {
      err=0;lcd.clear();lcd.setCursor(0, 0);lcd.print("UNAUTHOISED ");
      lcd.clear();lcd.setCursor(0, 0);lcd.print("DOOR LOCKED");


        digitalWrite(buz,LOW);//on
        delay(400);
        digitalWrite(buz,HIGH);//of


    }
    else
    {
      lcd.clear();lcd.setCursor(0, 0);lcd.print("IDENTIFIED:");lcd.print((int)idk);

      if(idk == 0)//user 1
      {
```

```
  lcd.clear();lcd.setCursor(0, 0);lcd.print("AUTHORISED ");
  //lcd.clear();
  lcd.setCursor(0, 1);lcd.print("DOOR OPEN");
  digitalWrite(m1,HIGH);digitalWrite(m2,LOW); delay(2000);
  digitalWrite(m2,HIGH); digitalWrite(m1,LOW);delay(2000);
  digitalWrite(m1,LOW);digitalWrite(m2,LOW);

 }
 if(idk == 1)//user 2
 {
   lcd.clear();lcd.setCursor(0, 0);lcd.print("AUTHORISED ");
  //lcd.clear();
  lcd.setCursor(0, 1);lcd.print("DOOR OPEN");
  digitalWrite(m1,HIGH);digitalWrite(m2,LOW); delay(2000);
  digitalWrite(m2,HIGH); digitalWrite(m1,LOW);delay(2000);
  digitalWrite(m1,LOW);digitalWrite(m2,LOW);
 }
 if(idk == 2)//user 3
 {
  lcd.clear();lcd.setCursor(0, 0);lcd.print("AUTHORISED ");
  //lcd.clear();
  lcd.setCursor(0, 1);lcd.print("DOOR OPEN");
  digitalWrite(m1,HIGH);digitalWrite(m2,LOW); delay(2000);
  digitalWrite(m2,HIGH); digitalWrite(m1,LOW);delay(2000);
  digitalWrite(m1,LOW);digitalWrite(m2,LOW);
 }
 if(idk == 3)//user 4
 {
   lcd.clear();lcd.setCursor(0, 0);lcd.print("AUTHORISED ");
  //lcd.clear();
  lcd.setCursor(0, 1);lcd.print("DOOR OPEN");
  digitalWrite(m1,HIGH);digitalWrite(m2,LOW); delay(2000);
  digitalWrite(m2,HIGH); digitalWrite(m1,LOW);delay(2000);
  digitalWrite(m1,LOW);digitalWrite(m2,LOW);
 }
}
```

```
    delay(2000);
    lcd.clear();lcd.setCursor(0, 0);lcd.print("SELECT OPTION");
}

}

int ct=0;
char dummy=0x0f;
int fpenroll(char id)
{

  serialFlush();
  //enroll buffer send 12 bytes
  for(int i =0;i<12;i++)
     Serial.write(enroll[i]);
  res[9] = 1;//
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){}
  else{err=1;return -1;}

  //generate ch buffer
  for(int i =0;i<13;i++)
     Serial.write(generate_ch[i]);
  res[9] = 1;
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){}
  else{err=2;return -1;}

 //enroll buffer send 12 bytes
  for(int i =0;i<12;i++)
     Serial.write(enroll[i]);
  res[9] = 1;//
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
```

```
  if(res[9] == 0){}
  else{err=3;return -1;}

  //generate ch1 buffer
  for(int i =0;i<13;i++)
      Serial.write(generate_ch1[i]);
  res[9] = 1;
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){}
  else{err=4;return -1;}

  //uncmd buffer send 12 bytes
  for(int i =0;i<12;i++)
      Serial.write(un_cmd[i]);
  res[9] = 1;//
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){}
  else{err=5;return -1;}

  //store buffer send 12 bytes
  for(int i =0;i<12;i++)
      Serial.write(store[i]);
  dummy = 0x0f+id;
  Serial.write((uint8_t)id);
  Serial.write((uint8_t)0x00);
  Serial.write((uint8_t)dummy);
  res[9] = 1;//
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){return id;}
  else{err=6;return -1;}
}
int fpsearch()
{
```

```
  ct=0;
   serialFlush();
  //enroll buffer send 12 bytes
  for(int i =0;i<12;i++)
      Serial.write(enroll[i]);
  res[9] = 1;//
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){}
  else{err=1;return -1;}


   //generate ch buffer
  for(int i =0;i<13;i++)
      Serial.write(generate_ch[i]);
  res[9] = 1;
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){}
  else{err=2;return -1;}


  //enroll buffer send 12 bytes
  for(int i =0;i<17;i++)
      Serial.write(identify[i]);
  res[9] = 1;//
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){return (int)res[11];}
  else{err=1;return -1;}

}
```

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

- In our proposed project, the performance is summarized as the following:
- It provides more security compared to the currently used systems.
- It is easy to implement and easy to access by the user.
- It ensures the owner by the range of security provides.
- It Is far better when compared to RFIDs and pattern locking
- No need of remembering of passwords set to the door locks.
- The aim of the work is to design a fingerprint lock system that can be used to lock and unlock a door system but its use can be extended to other electronic locking systems such as vaults.
- Having realized the device and found it working properly according to its design specifications and couple with the facts that relatively cheap components were involved in its realization, the aim of the research can be said to be achieved. The system can be used as an effective security lock.
- Effective Communication with administrators.
- Quick response to accident.
- Quick alert to police and other security services.

6.2 FUTURE SCOPE

- In the future, fingerprint detection can be used for better improvement of home automation and it gives strong security. It can be widely used in the banks, offices and homes…etc.
- It is more interactive by adding a display to show basic information about the vehicle and provides alert system for reporting any kind of trouble occurrences.
- The system could be modified into a web based system.
- Hence, A Vehicle tracking system is becoming increasingly important in rural and urban areas and it is more secured than other systems.

PUBLICATION

Submitted Paper in the Conference ICISECT - 21 with Paper ID (ICSMEC21-0085) and got Acceptance for the Paper**.**

REFERENCES

1. A. Aditya Shankar, “Finger Print Based Door Locking System”, IJECS Volume 4 Issue 3 March, 2015 ISSN:2319-7242.
2. Omidiora E. O. (2011) “A Prototype of a Fingerprint Based Ignition Systems in Vehicles” Published in European Journal of Scientific Research ISSN 1450-216X Vol.62 No.2 (2011), pp. 164-171 © Euro Journals Publishing, Inc. 2011
3. Karthikeyan.a “Fingerprint Based Ignition System” Published in Karthikeyan. A, Sowndharya.j /International Journal of Computational Engineering Research / ISSN: 2250–3005.
4. Pavithra. b.c., “Fingerprint Based Bank Locker System Using Microcontroller”, Proceedings of IRF International Conference, 05th April-2014, Pondicherry, India, ISBN: 978-93-82702-71-
5. Crystalynne D. Cortez, “Development of Microcontroller-Based Biometric Locker System with Short Message Service” Lecture Notes on Software Engineering, Vol. 4, No. 2, May 2016. [6]. Smita s. Mudholkaretal, “Biometrics Authentication Technique for Intrusion Detection Systems Using Fingerprint Recognition”, International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), Vol.2, No.1, February 2012.
6. Sagar S. Palsodkar, “Bank Lockers Security System using Biometric and GSM Technology”, SSRG International Journal of Electronics and Communication Engineering (SSRG-IJECE) – Volume 2 Issue 4–April 2015.
7. Anil k. Jain, Ling Hong, Sharath Pankanti, Ruud Bolle “An Identity-Authentication System using Fingerprints” .IEEE Vol.85 No.9 September1997.
8. A. O. Oke, A. A. Adigun, A. S. Falohun, and F. O. Alamu, “Development of a programmable electronic digital code lock system,” International Journal of Computer and Information Technology, vol.2, issue 1, pp. 127-131, January 2013.
9. D. Matoni, D. Maio, A. K. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition, 2nd ed., London: Springer Verlay London Limited, 2009, ch. 2, pp. 57-58.
10. Anil K. Jain, Arun Ross, Salil Prabhakar, "An Introduction to Biometric Recognition", IEEE Transactions on Circuits and Systems for Video Technology Special Issue on Image and Video Ba

BIBILOGRAPHY

1. R. P. Wildes, "Iris recognition: an emerging biometric technology", Proceedings of the IEEE, vol. 85, no. 9, pp. 1348-1363, September, 1997.

2. Lourde R Mary, Khosla Dushyant, "Fingerprint Identification in Biometric Security Systems", International Journal of Computer and Electrical Engineering, vol. 2, no. 5, October 2010.

3. Zevdin Pala, Nihat Inanc, "Smart Parking Applications Using RFID Technology", 1st Annual RFID Eurasia Istanbul, pp. 1-3, 2007.

4. Raffaele Cappelli, Alessandra Lumini, Dario Maio, Davide Maltoni, "Fingerprint Image Reconstruction from Standard Templates", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 29, no. 9, pp. 1489-1503, September 2007.

5. Fernando L. Podio, "Personal authentication through biometric technologies", Proceedings 2002 IEEE 4th International Workshop on Networked Appliances (Cat. No.02EX525) Gaithersburg, pp. 57-66, 2002.

APPENDIX

The aim of the project is to design a fingerprint lock system that can be used to lock and unlock a door system but its use can be extended to other electronic locking systems such as vaults. Having realized the device and found it working properly according to its design specifications and couple with the facts that relatively cheap components were involved in its realization, the aim of the research can be said to be achieved. The system can be used as an effective security lock. It provides more security compared to the currently used systems. It is easy to implement and easy to access by the user. It ensures the owner by the range of security provides and it is far better when compared to RFIDs and pattern locking and there is no need of remembering of passwords set to the door locks.

A

PROJECT REPORT

On

RFID BASED SMART MASTER CARD FOR BUS TRAIN METRO TICKETING

Submitted by

1)R. Sai Krishna (18K85A0402)

2)M. Sai Kumar (18K85A0409)

3) P. Santhosh Kumar (18K85A0408)

in partial fulfillment for the award of the

degree

of

BACHELOR OF TECHNOLOGY

IN

ELCTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

N. Vishwanath

M.Tech, (Ph. D)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

BONAFIDE

This is to certify that the project entitled **“RFID Based smart master card for bus metro ticketing”** is being submitted by 1.**R.SaiKrishna(18K85A0402), 2.M.Saikumar (18K85A0409) 3.P.Santhosh kumar (18K85A0408).** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN** Electronics And Communication engineering is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

N. Vishwanath
Department of ECE

Head of the Department
Dr. B.Hari Krishna
Department of ECE

Internal Examiner

External Examiner

Place:

Date:

DECLARATION

We, the student of **Bachelor of Technology** in Department of ‘Electronics And Communication Engineering’ session: (2017 – 2021), St. Martin’s Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled “RFID Based Smart Master Card For Bus Train Metro Ticketing “is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

R. Sai Krishna (18K85A0402)
M. Sai kumar (18K85A0409)
P. Santhosh kumar (18K85A0408)

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr. B.Hari Krishna**, **Head &Professor**, the Dept. of Electronics And Communication for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept. project coordinator Dr.A.Anand kumar** for his/her involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide**, Mr. N. Vishwanath, Assistant professor**, Department of ECE and our class project coordinator **Ms. A. Soumya**, **Assistant professor**, Department of ECE for guidance and encouragement in carrying out this project work

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr. Malayadri, professor in IT,** and **Dr.R. Santhosh kumar Associate Professor in ECE** and for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. R. Sai Krishna
2. M.Sai kumar
3. P. Santhosh kumar

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES

LIST OF ACRONYMS AND DEFINITIONS

RFID: Radio frequency identification

RFID uses an electromagnetic field to identify and track tags attached to objects. An RFID system consists of a tiny radio transponder, a radio receiver and transmitter

ARDUINO: Arduino is an open-source platform used for building electronic projects. Arduino consists of a both physical programmable circuit board and a software used to programme it

EM18 READER: EM 18 reader is the one of the commonly used RFID reader to read 125khz tags.it features low cost, low power consumption, small form factor and easy to use.it can be directly interfaced with microcontrollers using UART and with pc using an RS232 converter

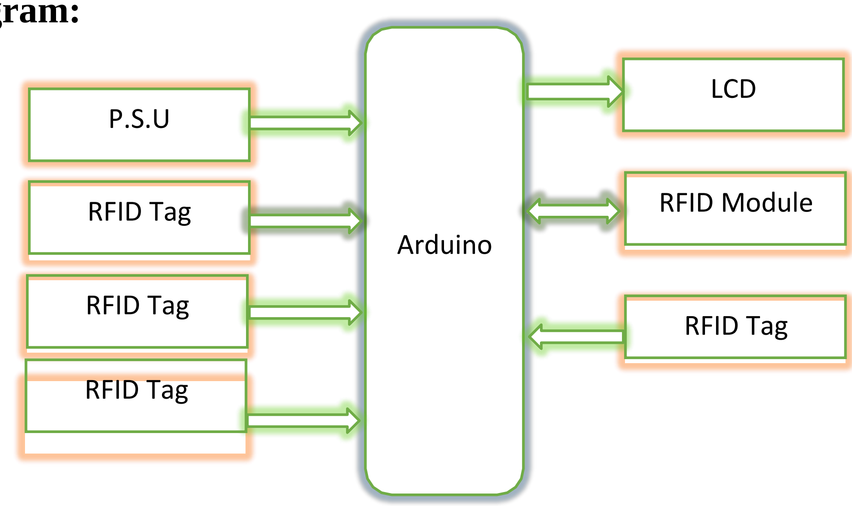
PSU: Power supply unit

RFID BASED SMART MASTER CARD FOR BUS TRAIN METRO TICKETING

Abstract:

Modern cities of today have developed multiple means of communication including Buses, trains, metros and private vehicles. Now each transport system has their own smart card and it becomes a hectic process for users to manage separate smart cards for every transport medium. Thus, we propose a smart master card approach that integrates all these systems together and allows for a single master card and a centralized system for all transportation mediums. To demonstrate this concept, we use three RFID scanners to demonstrate as bus train and metro train smart card scanners respectively. Now we use three smart cards that work particularly on each of the systems viz one RFID card for bus, one for train and one for metro respectively. Now we also provide one more card that is the master smart card that can work on all three scanners thus making it very easy for the user to use any transport as desired using the same card. The system also allows for source and destination selection and based on that deducts particular amount from the user master card.

Block Diagram:



CHAPTER 1
INTRODUCTION

Introduction:

As for the RFID application, it’s been a widespread tool for both tracking the transit transports and for the public ticketing system. It’s already been an outstanding achievement throughout the globe including big cities like London, Helsinki, Shanghai, Istanbul, Moscow, Porto and many more. The system can be implemented for subways, railways and public bus services for the sake of systematic operations in corresponding cases. In the megacities, the conventional system of public transport is based on project-based bus or railway tickets that ultimately lead to chaos among public, system loss, corruption and most of all traffic jam that is responsible for a huge wastage of time. No prior notification of the arrival and departure of the transports are available creating a lot of confusion among the passengers resulting in a rough argument between them and the bus supervisors or the operators.

Again, having no government authority to take control or keep an eye over the whole scenario, the private sectors are creating a monopoly, taking control over the public transport and autocratic raise in bus fare. The tracking and ticketing systems using RFID can be merged to solve the prevailing problems. Even though the GPS based system can be designed, we propose the RFID based tickets for its low cost, easy operation, portability, durability, reliability and being much more user friendly. Also, the high-speed RFID tags and detectors make the tracking system of a running bus merely a child’s play. Public carrying RFID based electronic tickets will have access to any bus service of the city only entering his current location and his destination on the keypad attached to every bus. The data will directly be transferred to the server main database and the equivalent credit will be stored in the corresponding bus account. Also, the screen at every bus stop will notify the passengers, the departure time of the last bus of any route. This automated system will save time, have a higher authoritative inspection and reduce chaos and confusion on the road.

CHAPTER 2

LITERATURE SURVEY

Charlie Fine et. al (2006): RFID technology has generated much hype in the last few years. The major driver for its development has been the tagging of physical objects – people, places, and things – with single chip radios so they can interface with computers. RFID technology is both hailed as the key to the ―Internet of Things,‖ and condemned as invasive surveillance technology, and in more extreme circles it is feared as the Mark of the Beast. An RFID system can be broken down into two key dimensions. The technical infrastructure includes the actual data capture technology comprised of tags, readers, and transmission medium. The logical infrastructure refers to the overall identification (ID) scheme used in representing objects.

Ahuja Sanjay and Potti Pavan (2010) : RFID is still in a developing phase and more is in the pipeline in terms of new applications. Among applications already developed, RFID tags are being used in clothing for billing and security purposes. RFID tags are embedded inside animals for tracking purposes. RFID tags embedded in uniforms can be used to know the number of hours an employee spends to complete a particular task. There are several associations that are pro-testing against the use of RFID to track people fearing the impact on people's social life and privacy.

Matija Bumbak (2005): Considering enormous scale on which RFID will be operating and considering the fact that it will be present everywhere in our lives it is absolutely necessary for it to be a secure system.Even though RFID is still not operational on large scale, some experimental shops, literature written about it is relatively extensive. there are three primary issues surrounding RFID and the need to protect proprietary information: Protecting data stored on the tag; Protecting the integrity of the tag (and thus the product); Securing data related to the serial number on a tag, which may be stored in a network database.

Kamaran Ahsan et. al (2010): Industries use RFID for various applications such as personal/vehicle access control, departmental store security, equipment tracking, baggage, fast food establishments, logistics, etc. The enhancement in RFID technology has brought advantages that are related to resource optimization, increased efficiency within business processes, and enhanced customer care, overall improvements in business operations and healthcare. Other applications includes automatic toll payments, departmental access control in large buildings. personal and vehicle control in a particular area, security of items which shouldn't leave the area, equipment tracking in engineering firms, hospital filing systems.

Srivastav Nandita (2006) : The ―Radio Frequency Identification (**RFID**) is an automatic identification system. RFID uses RF to identify ―tagged‖ items .This data is then collected and transmitted to a host system using an RF Reader. The data transmitted by the tag may provide identification or location information, or specifics about the product tagged, such as price, color, date of purchase; etc.In Bar code the scanner device directs a light beam at the bar code. The device contains a small sensory reading element. This sensor detects the light being reflected back from the bar code, and converts light energy into electrical energy. The result is an electrical signal that can be converted into data.

Daniel M Dobkin et. al (2005): Every form of communication must follow a set of protocols, dealing with such issues as providing access to the communications medium, structure and meaning of the data to be transmitted, and coding and modulation of the data into the transmitted signals. The protocols have been developed somewhat independently and even when standardized are generally mutually inoperable. For example, an ISO15693 tag doesn't detect or understand and ISO11784 reader, and neither can communicate with an EPCGlobal UHF system. **Daniel M Dobkin et. al (2005)**: Every RFID system consists of at least one interrogator, more commonly known as a reader, which uses a radio link to communicate with at least one transponder. The tag generally contains one or more integrated circuits, and a unique identifying number stored in non-volatile memory. The reader is often (though not always) integrated into a network in order to make efficient use of the identification data it collects. There are three key architectural parameters that determine the type of RFID system in use: the frequency (practically equivalent to the mode of coupling), the means of powering the tag, and the communications protocol employed

Juels Ari (2005): RFID raises two main privacy concerns for users: clandestine *tracking* and *inventorying*.RFID tags respond to reader interrogation without alerting their owners or bearers. Thus, where read range permits, clandestine scanning of tags is a plausible threat. Most RFID tags emit unique identifiers, even tags that protect data with cryptographic algorithms In consequence, a person carrying an RFID tag effectively broadcasts a fixed serial number to nearby readers, providing a ready vehicle for clandestine physical tracking. Such tracking is possible even if a fixed tag serial number is random and carries no intrinsic data. The threat to privacy grows when a tag serial number is combined with personal information.

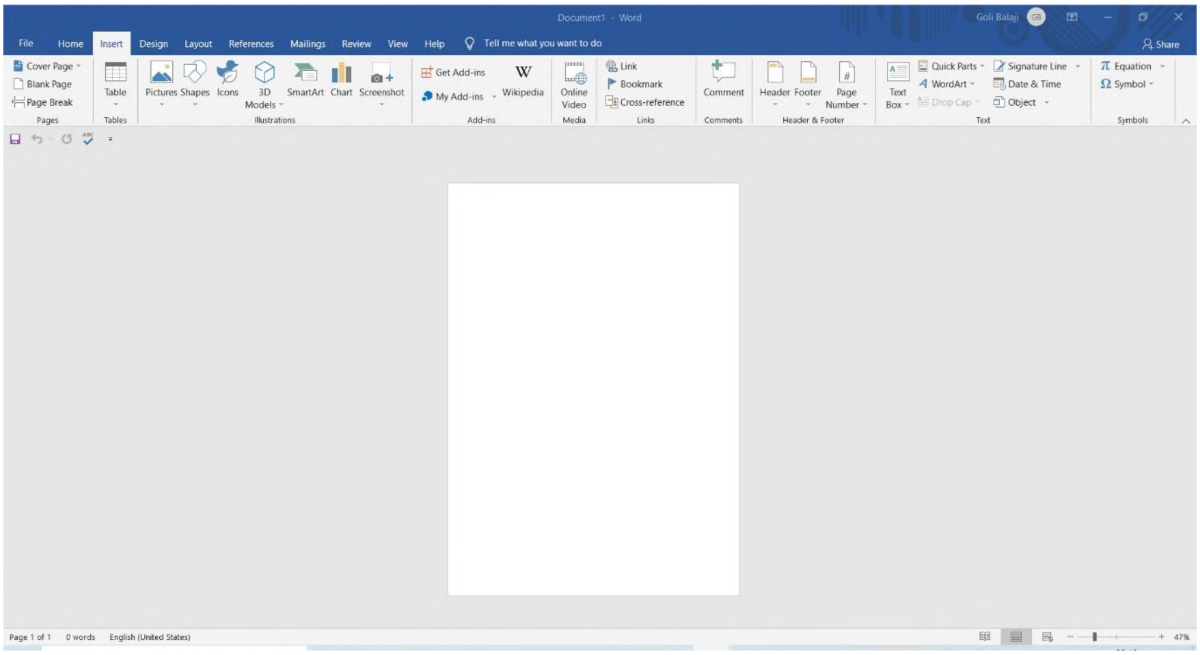
CHAPTER 3

EMBEDDED SYSTEMS

2.1 Embedded System

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.



The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes. The applications software on such processors is sometimes referred to as firmware

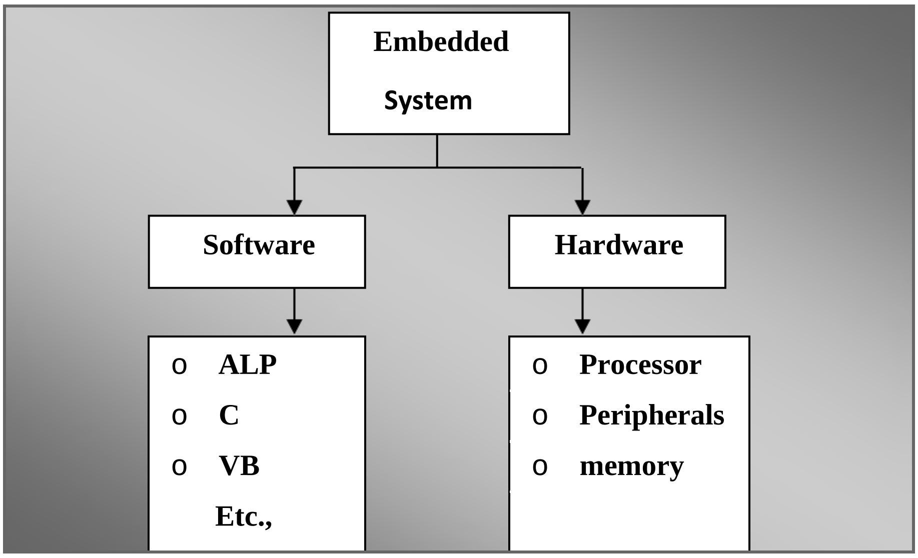


Figure 2.1 Block diagram of Embedded System

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

Memory: It is used to store data or address.

Peripherals: These are the external devices connected

Processor: It is an IC which is used to perform some task

Applications of embedded systems

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

Processors are classified into four types like:

- Micro Processor (µp)
- Micro controller (µc)
- Digital Signal Processor (DSP)
- Application Specific Integrated Circuits (ASIC)

Micro Processor (µp):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Three basic characteristics differentiate microprocessors:

- **Instruction set**: The set of instructions that the microprocessor can execute.
- **Bandwidth :** The number of bits processed in a single instruction.
- **Clock speed :** Given in megahertz (MHz), the clock speed determines how many instructions per second the processor can execute.

In both cases, the higher the value, the more powerful the CPU. For example, a 32-bit microprocessor that runs at 50MHz is more powerful than a 16-bit microprocessor that runs at 25MHz. In addition to bandwidth and clock speed, microprocessors are classified as being either RISC (reduced instruction set computer) or CISC (complex instruction set computer).

A microprocessor has three basic elements, as shown above. The ALU performs all arithmetic computations, such as addition, subtraction and logic operations (AND, OR, etc). It is controlled by the Control Unit and receives its data from the Register Array. The Register Array is a set of registers used for storing data. These registers can be accessed by the ALU very quickly. Some registers have specific functions - we will deal with these later. The Control Unit controls the entire process. It provides the timing and a control signal for getting data into and out of the registers and the ALU and it synchronizes the execution of instructions (we will deal with instruction execution at a later date).

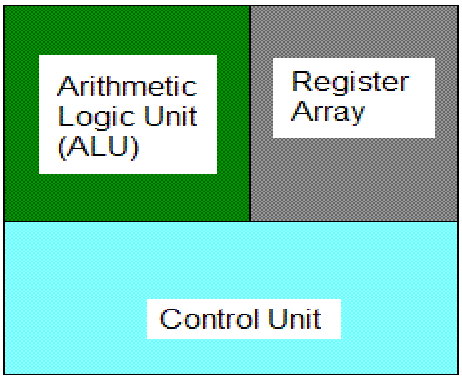


Fig : 2.1.1 Architecture of CPU

Three Basic Elements of a Microprocessor

Micro Controller (μc):

A microcontroller is a small computer on a single **integrated circuit** containing a processor core, memory, and programmable **input/output** peripherals. Program memory in the form of **NOR flash** or **OTP ROM** is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the **microprocessors** used in **personal computers** or other general purpose applications.

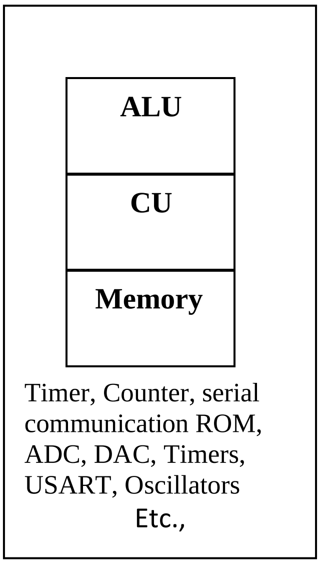


Figure 2.2 Block Diagram of Micro Controller (μc)

Digital Signal Processors (DSPs):

Digital Signal Processors is one which performs scientific and mathematical operation. Digital Signal Processor chips - specialized microprocessors with architectures designed specifically for the types of operations required in digital signal processing. Like a general-purpose microprocessor, a DSP is a programmable device, with its own native instruction code. DSP chips are capable of carrying out millions of floating-point operations per second, and like their better-known general-purpose cousins, faster and more powerful versions are continually being introduced. DSPs can also be embedded within complex "system-on-chip" devices, often containing both analog and digital circuitry.

Application Specific Integrated Circuit (ASIC)

ASIC is a combination of digital and analog circuits packed into an IC to achieve the desired control/computation function

ASIC typically contains

- CPU cores for computation and control
- Peripherals to control timing critical functions
- Memories to store data and program
- Analog circuits to provide clocks and interface to the real world which is analog in nature
- I/O s to connect to external components like LEDs, memories, monitors etc.

2.2 Computer Instruction Set

There are two different types of computer instruction set there are:

1. RISC (Reduced Instruction Set Computer) and
2. CISC (Complex Instruction Set computer)

2.2.1 Reduced Instruction Set Computer (RISC)

A RISC (reduced instruction set computer) is a microprocessor that is designed to perform a smaller number of types of computer instruction so that it can operate at a higher speed (perform more million instructions per second, or millions of instructions per second). Since each instruction type that a computer must perform requires additional transistors and circuitry, a larger list or set of computer instructions tends to make the microprocessor more complicated and slower in operation.

Besides performance improvement, some advantages of RISC and related design improvements are:

- A new microprocessor can be developed and tested more quickly if one of its aims is to be less complicated.
- Operating system and application programmers who use the microprocessor's instructions will find it easier to develop code with a smaller instruction set.
- The simplicity of RISC allows more freedom to choose how to use the space on a microprocessor.

Higher-level language compilers produce more efficient code than formerly because they have always tended to use the smaller set of instructions to be found in a RISC computer.

2.2.2 RISC characteristics

- **Simple instruction set:**

  In a RISC machine, the instruction set contains simple, basic instructions, from which more complex instructions can be composed.

- **Same length instructions.**

  Each instruction is the same length, so that it may be fetched in a single operation.

- **1machine-cycleinstructions.**

  Most instructions complete in one machine cycle, which allows the processor to handle several instructions at the same time. This pipelining is a key technique used to speed up RISC machines.

2.2.3 Complex Instruction Set Computer (CISC)

CISC, which stands for **Complex Instruction Set Computer,** is a philosophy for designing chips that are easy to program and which make efficient use of memory. Each instruction in a CISC instruction set might perform a series of operations inside the processor. This reduces the number of instructions required to implement a given program, and allows the programmer to learn a small but flexible set of instructions.

The advantages of CISC

At the time of their initial development, CISC machines used available technologies to optimize computer performance.

- Microprogramming is as easy as assembly language to implement, and much less expensive than hardwiring a control unit.

- The ease of micro-coding new instructions allowed designers to make CISC machines upwardly compatible: a new computer could run the same programs as earlier computers because the new computer would contain a superset of the instructions of the earlier computers.
- As each instruction became more capable, fewer instructions could be used to implement a given task. This made more efficient use of the relatively slow main memory.
- Because micro program instruction sets can be written to match the constructs of high-level languages, the compiler does not have to be as complicated.

The disadvantages of CISC

Still, designers soon realized that the CISC philosophy had its own problems, including:

- Earlier generations of a processor family generally were contained as a subset in every new version --- so instruction set & chip hardware become more complex with each generation of computers.
- So that as many instructions as possible could be stored in memory with the least possible wasted space, individual instructions could be of almost any length---this means that different instructions will take different amounts of clock time to execute, slowing down the overall performance of the machine.
- Many specialized instructions aren't used frequently enough to justify their existence --- approximately 20% of the available instructions are used in a typical program.
- CISC instructions typically set the condition codes as a side effect of the instruction. Not only does setting the condition codes take time, but programmers have to remember to examine the condition code bits before a subsequent instruction changes them.

2.3 Memory Architecture

There two different type's memory architectures there are:

- Harvard Architecture
- Von-Neumann Architecture

2.3.1 Harvard Architecture

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The **Harvard architecture** is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

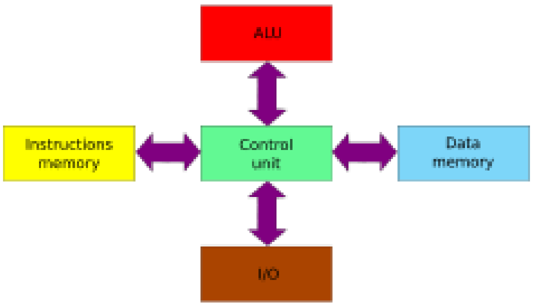


Figure 2.3 Harvard Architecture

Modern uses of the Harvard architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

- Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behavior must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.
- Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

2.3.2 Von-Neumann Architecture

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. The **von Neumann architecture** is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A **stored-program** digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

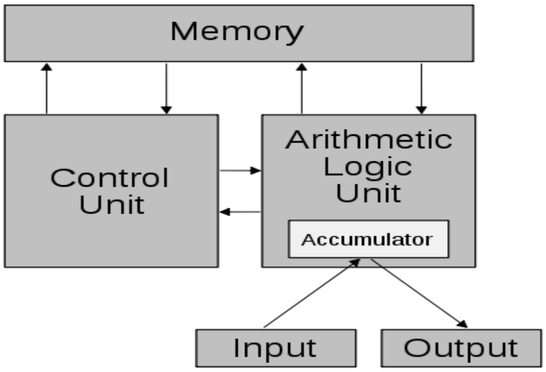


Figure 3.4 Schematic of the Von-Neumann Architecture.

Basic Difference between Harvard and Von-Neumann Architecture

- The primary difference between Harvard architecture and the Von Neumann architecture is in the Von Neumann architecture data and programs are stored in the same memory and managed by the same information handling system.
- Whereas the Harvard architecture stores data and programs in separate memory devices and they are handled by different subsystems.
- In a computer using the Von-Neumann architecture without cache; the central processing unit (CPU) can either be reading and instruction or writing/reading data to/from the memory. Both of these operations cannot occur simultaneously as the data and instructions use the same system bus.
- In a computer using the Harvard architecture the CPU can both read an instruction and access data memory at the same time without cache. This means that a computer with Harvard architecture can potentially be faster for a given circuit complexity because data access and instruction fetches do not contend for use of a single memory pathway.
- Today, the vast majority of computers are designed and built using the Von Neumann architecture template primarily because of the dynamic capabilities and efficiencies gained in designing, implementing,

operating one memory system as opposed to two. Von Neumann architecture may be somewhat slower than the contrasting Harvard Architecture for certain specific tasks, but it is much more flexible and allows for many concepts unavailable to Harvard architecture such as self-programming, word processing and so on.

CHAPTER 4

POWER SUPPLY

4.1 Block Diagram

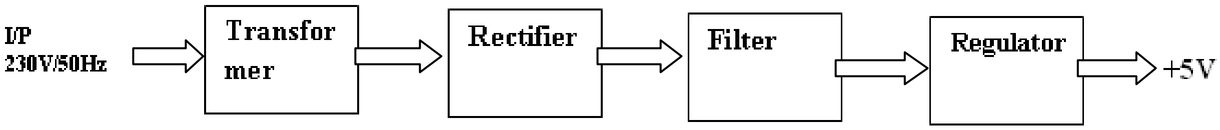


Figure 3.1 Power Supply

4.2 Circuit Diagram

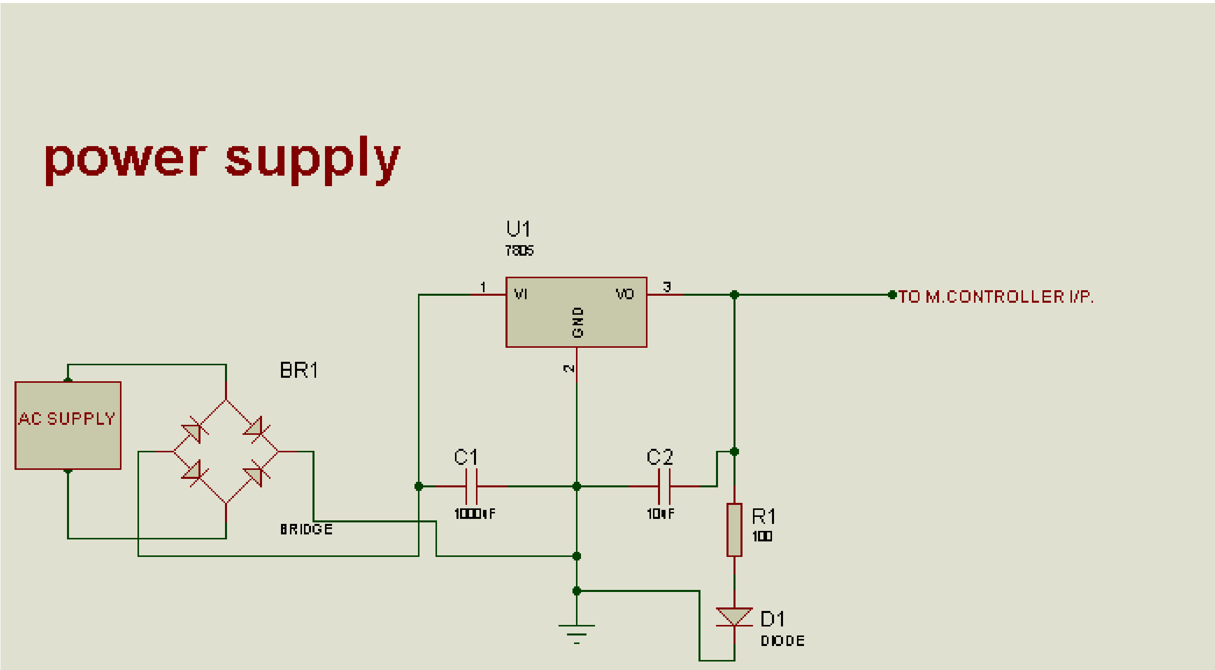


Fig : 4.2 Circuit Diagram

Description

4.2.1 Transformer

A **transformer** is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

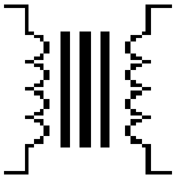


Figure4.2.1: Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

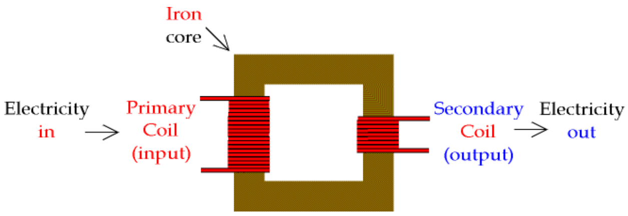


Figure 4.2.2 Transformer

Basic Principle:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

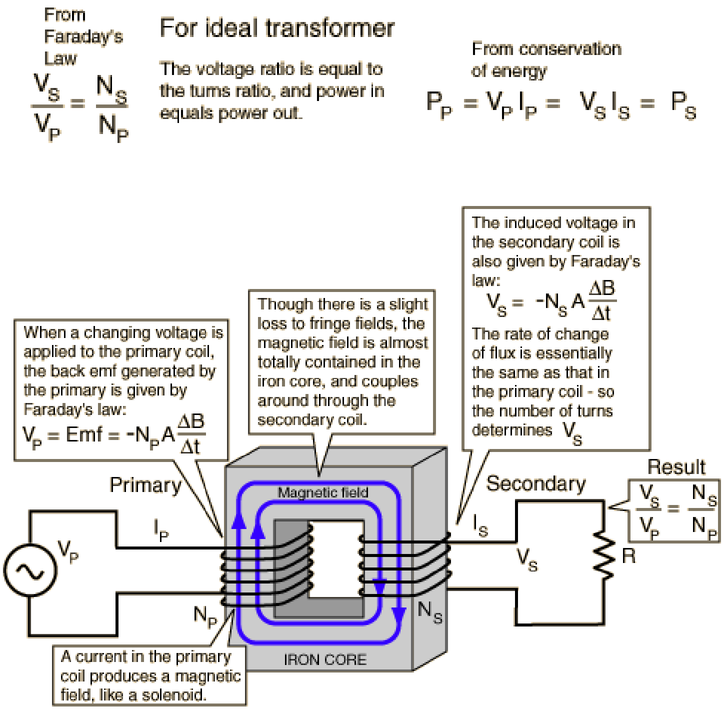


Figure 4.2.3: Basic Principle

Transformer Working

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

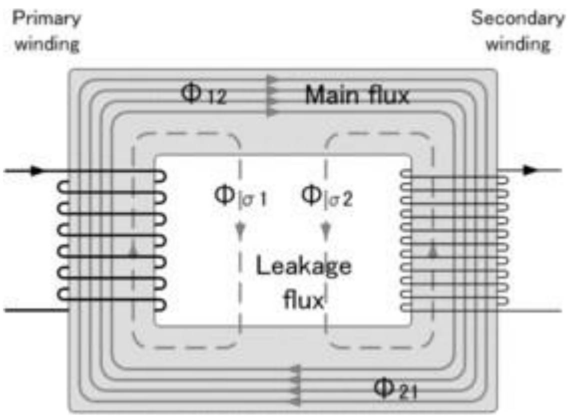


Figure 4.2.4: Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f.

The iron core is laminated to prevent 'eddy currents' flowing in the core. These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

Classification of Transformer:

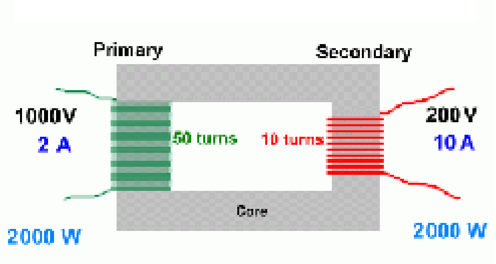
- Step-Up Transformer
- Step-Down Transformer

Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn's ratio of the two sets of windings determines the amount of voltage transformation.



Figur4.2.5: Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single-phase step-down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase step up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

Step-Up Transformer:

A step-up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply.

A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

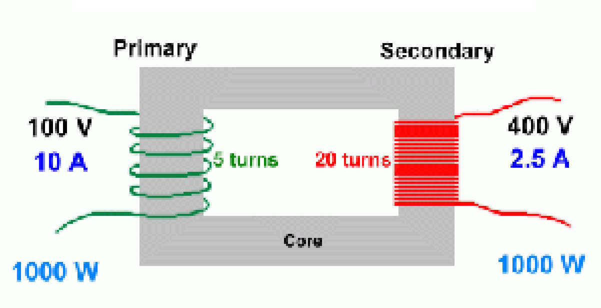


Figure4.2.6: Step-Up Transformer

Applications

Generally these **Step-Up Transformers** are used in industries applications only.

Turns Ratio and Voltage

The ratio of the number of turns on the primary and secondary coils determines the ratio of the voltages...

$$\frac{V_s}{V_p} = \frac{N_s}{N_p}$$

...where V_p is the primary (input) voltage, V_s is the secondary (output) voltage, N_p is the number of turns on the primary coil, and N_s is the number of turns on the secondary coil.

4.2.2 Diodes

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

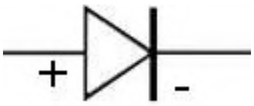


Figure 4.3 Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

3.2.3 Rectifier

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

The Half-wave Rectifier

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure .

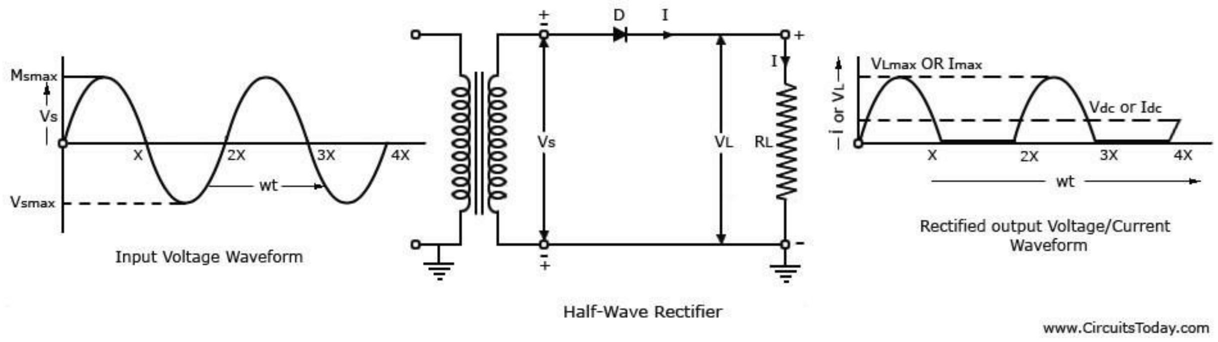


Figure 4.4 Half Wave Rectifier

Figure 2 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is 0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

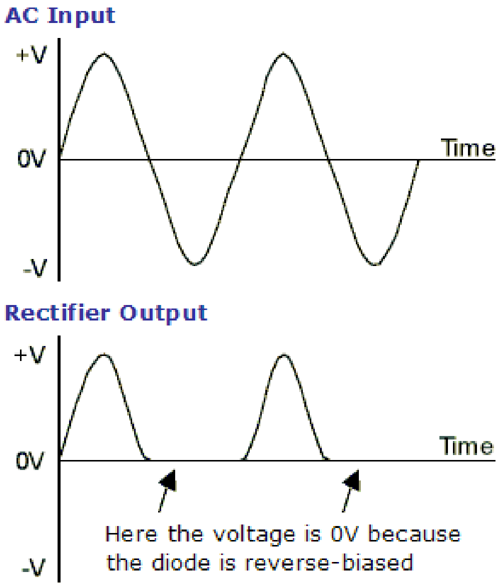


Figure 4.5 Half-Wave Rectification

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

The Full-wave Rectifier

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC. The resulting waveform is shown in figure 4.

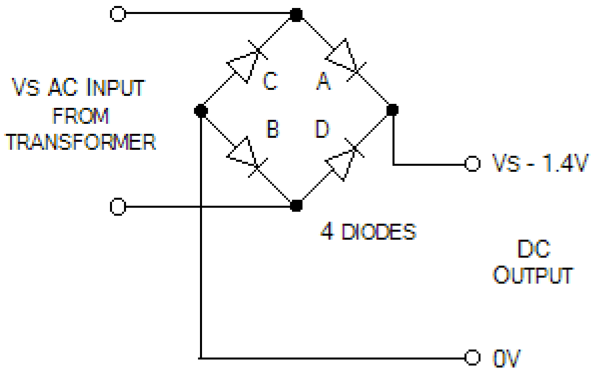


Figure 4.5.1: Full-Wave Rectifier

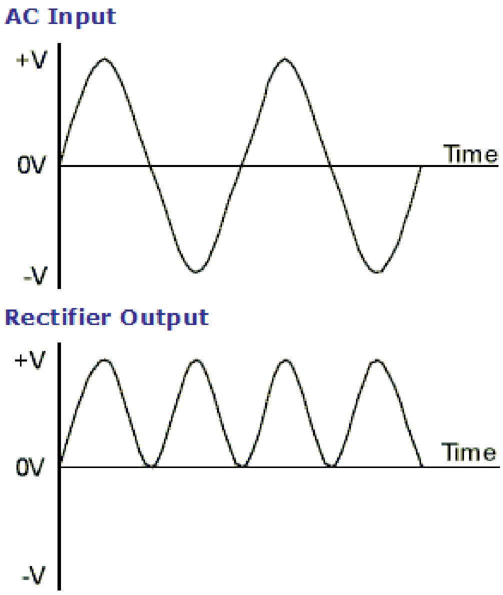


Figure : 4.5.2 : Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased.

While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

Capacitor Filter

The **capacitor-input filter**, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

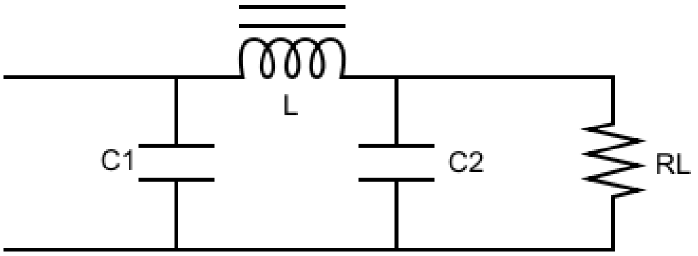


Figure 4.5.3: Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L

2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result only the DC component appears across the load RL.

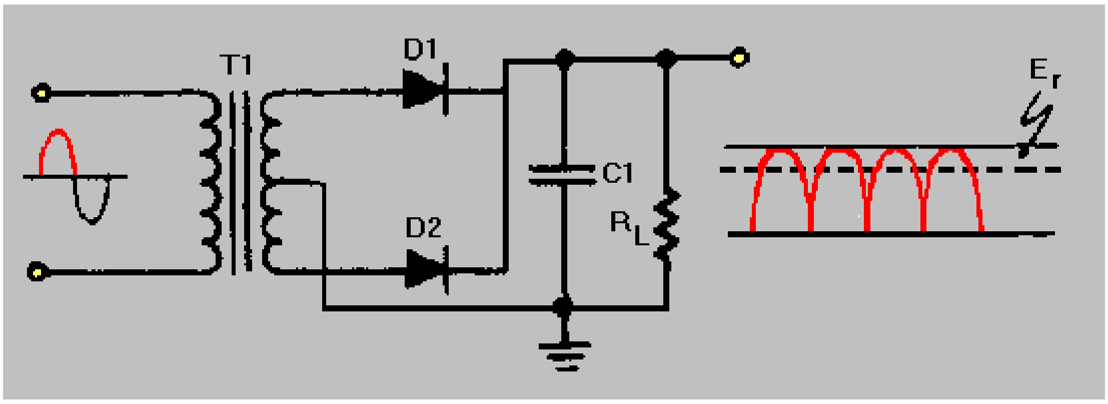


Figure 4.5.4: Centered Tapped Full-Wave Rectifier with a Capacitor Filter

4.2.4 Voltage Regulator:

A **voltage regulator** is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx'indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx'indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consists the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

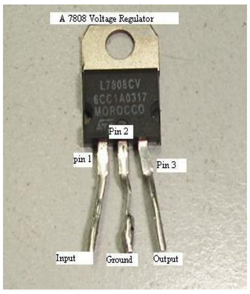


Figure 4.6 Regulator

CHAPTER 5
SOFTWARE

5.1 Proteus

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possesses 2D CAD drawing feature. It deserves to bear the tagline "From concept to completion".

5.1.1 About Proteus

It is a software suite containing schematic, simulation as well as PCB designing.
ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.
The designer can also develop 2D drawings for the product.

5.1.2 Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

5.2 Starting New Design

Step 1: Open ISIS software and select New design in File menu

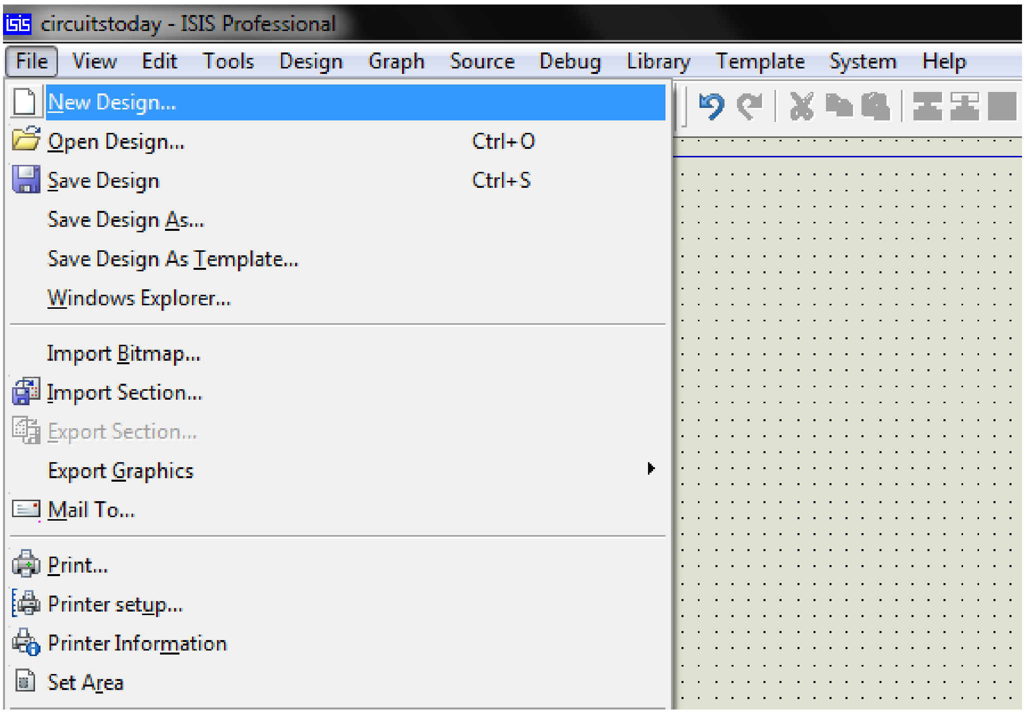


Figure 5.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

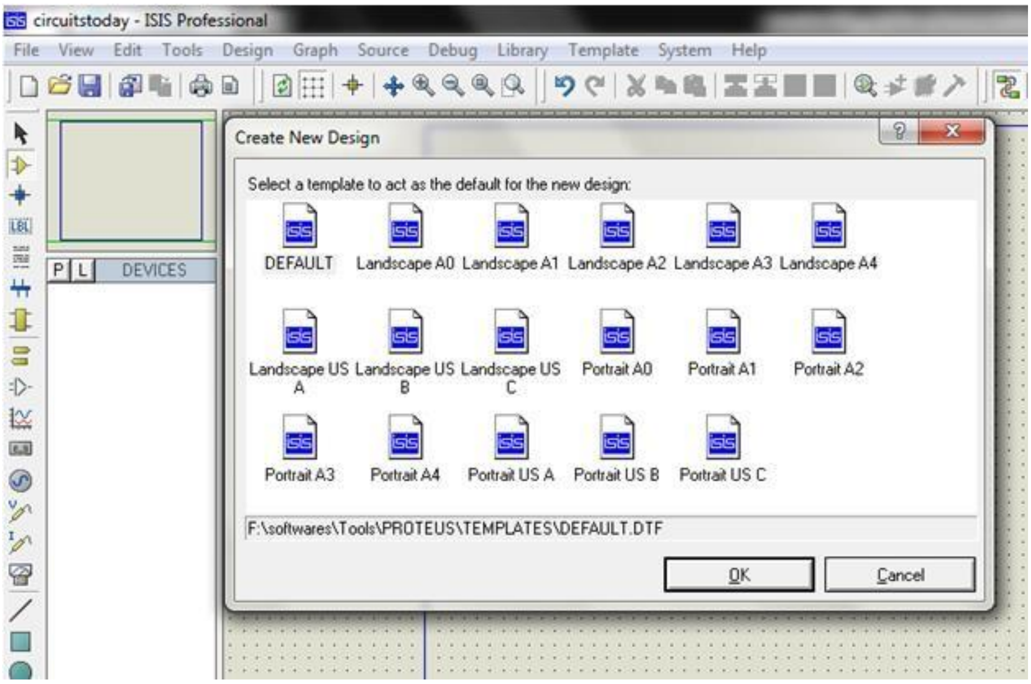


Figure 5.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

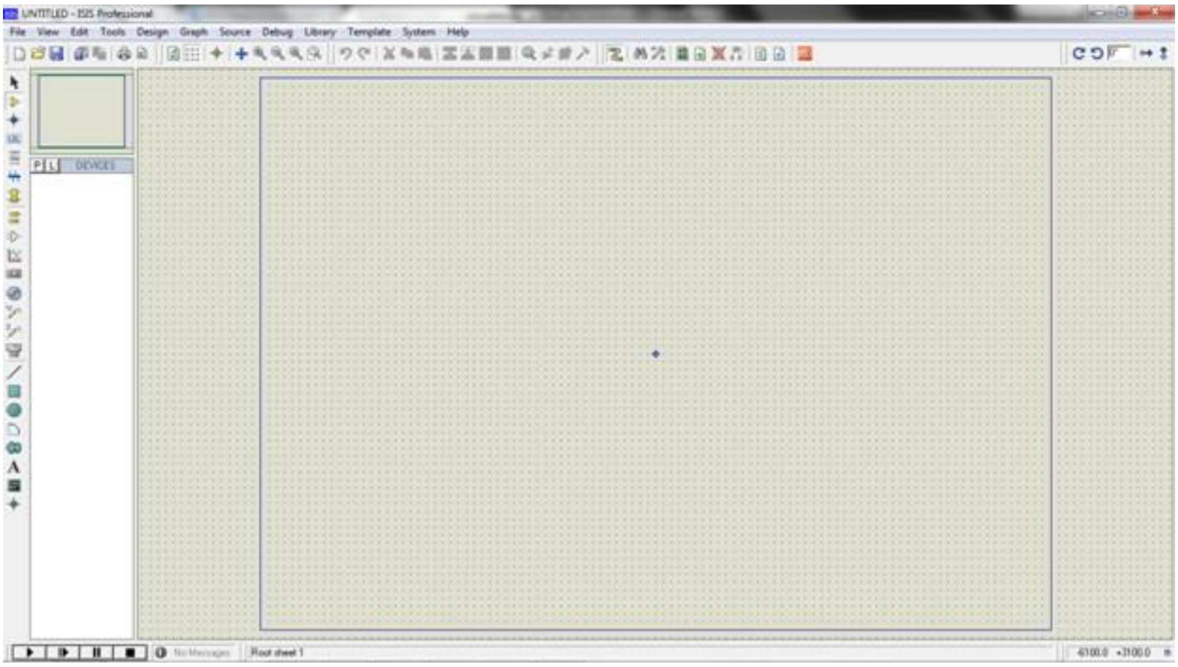


Figure 5.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

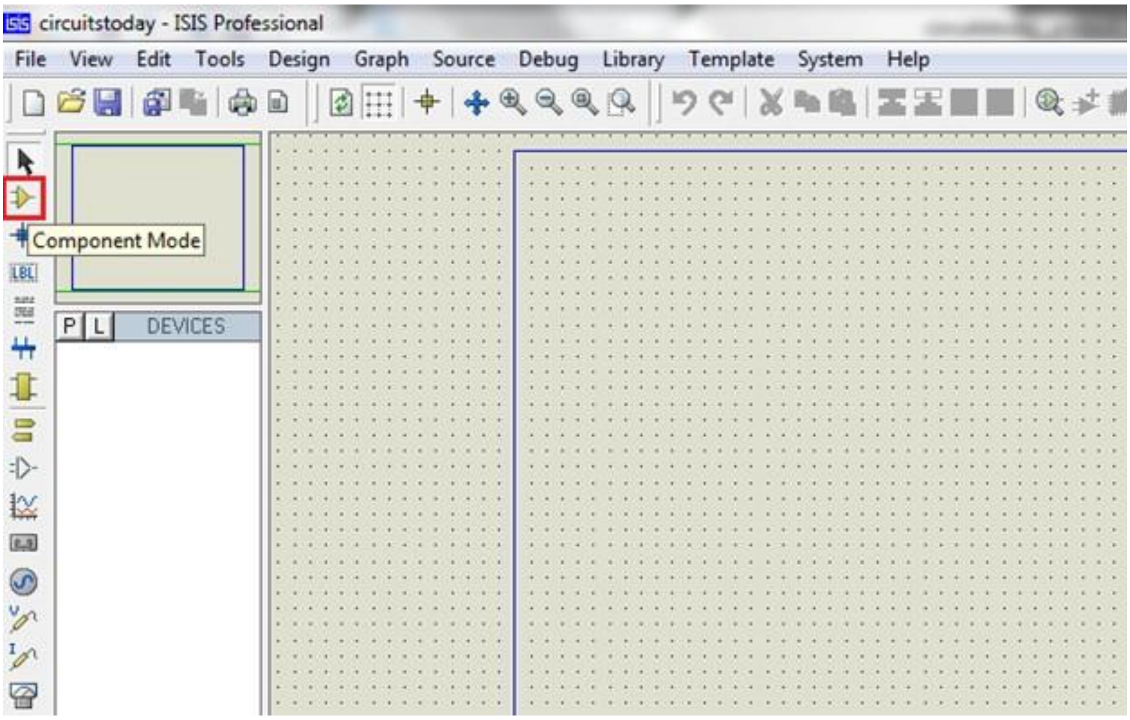


Figure 5.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

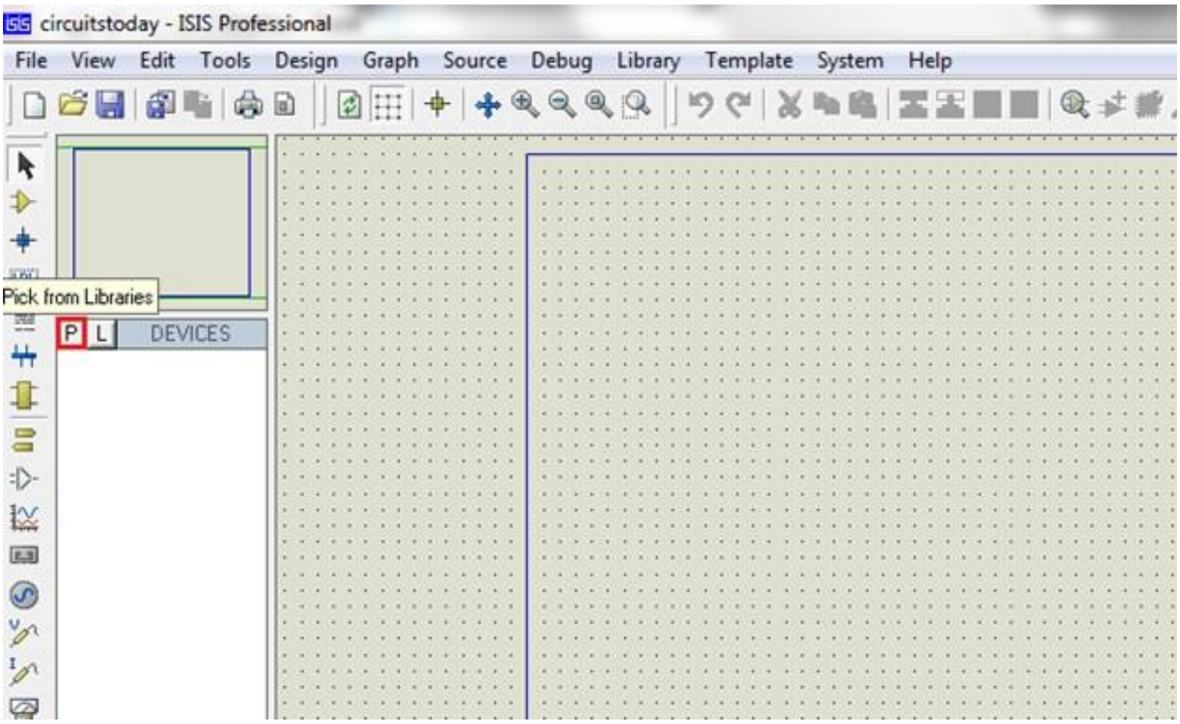


Figure 5.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

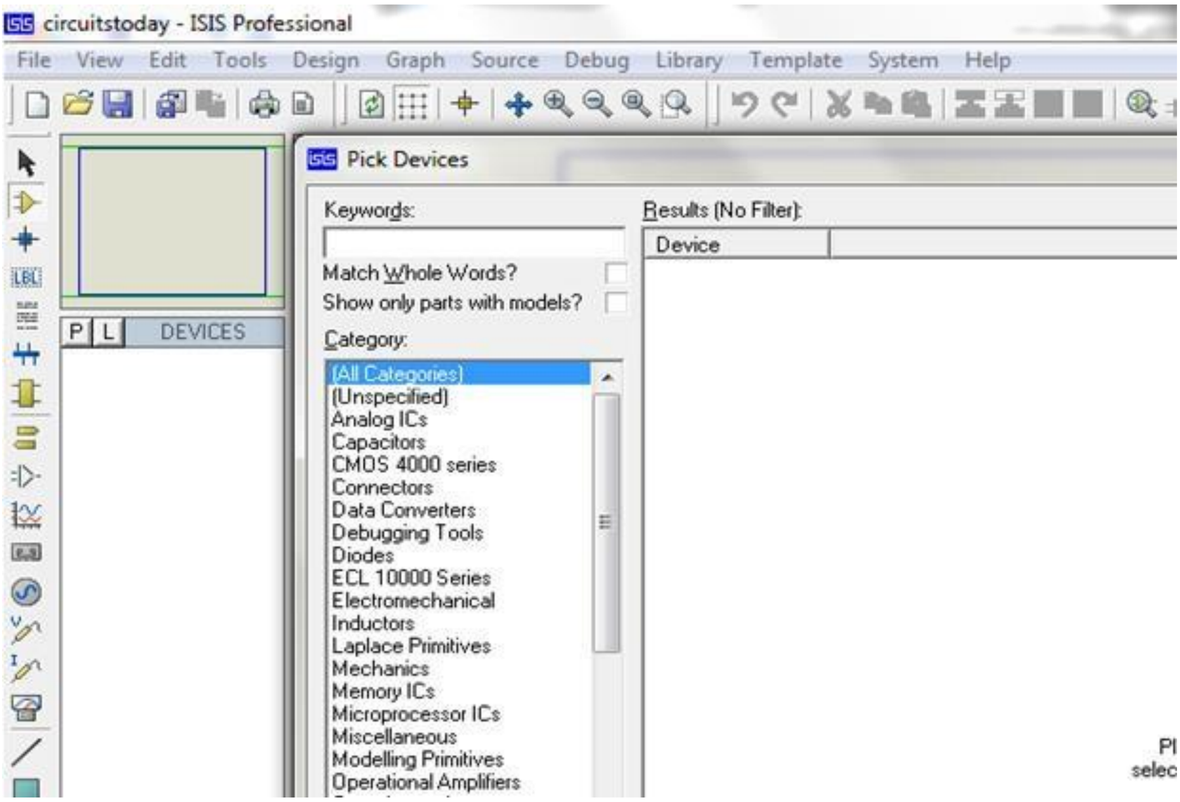


Figure 5.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

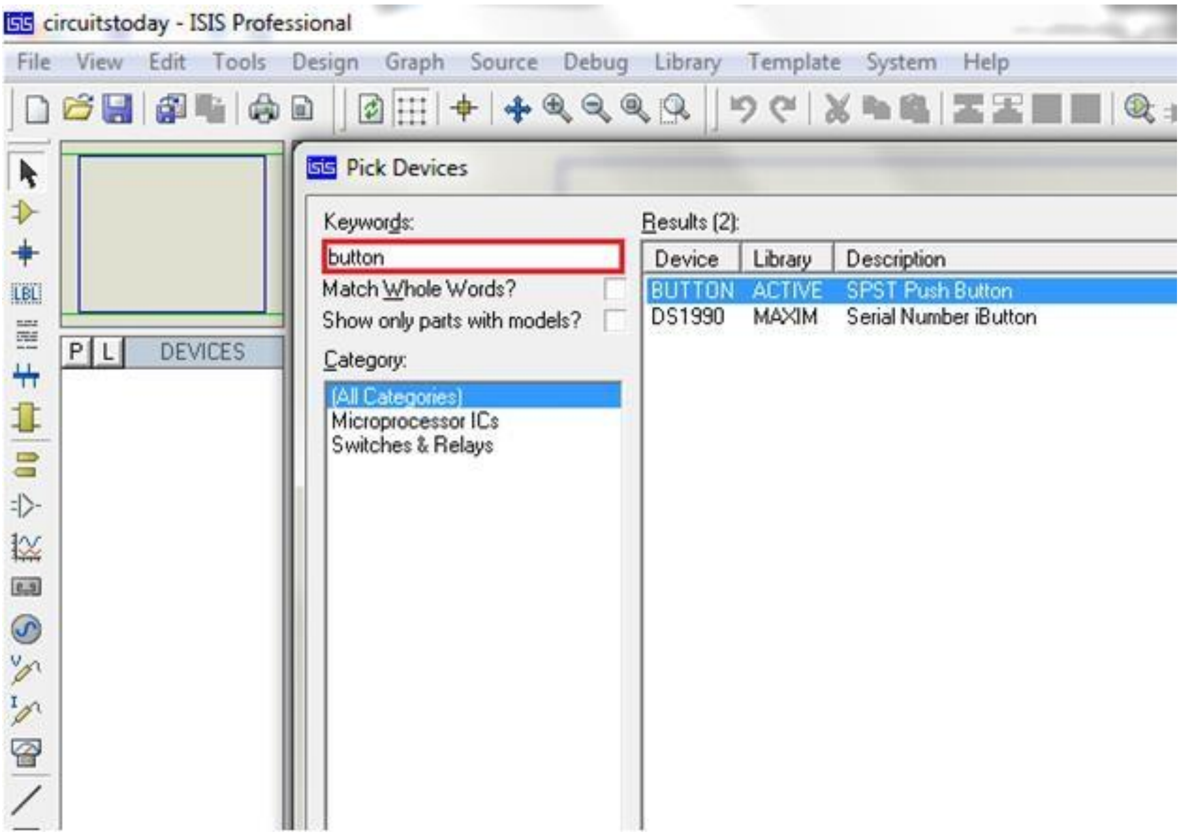


Figure 5.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

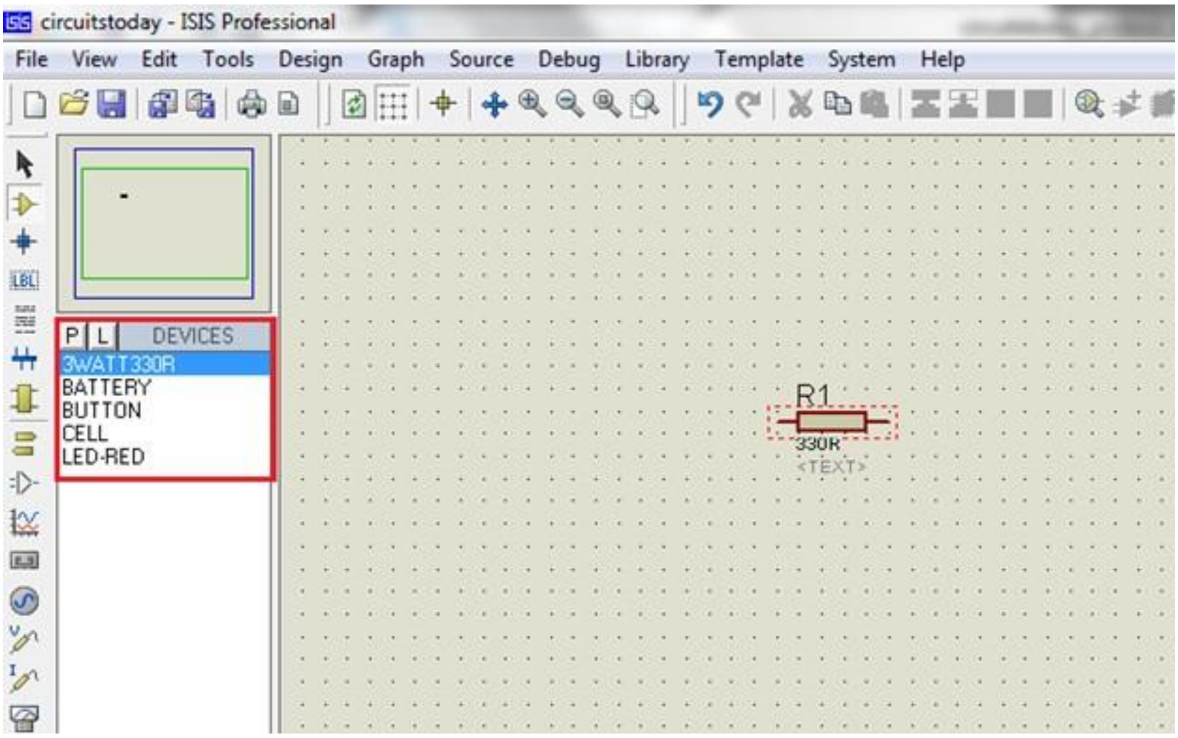


Figure 5.8 Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

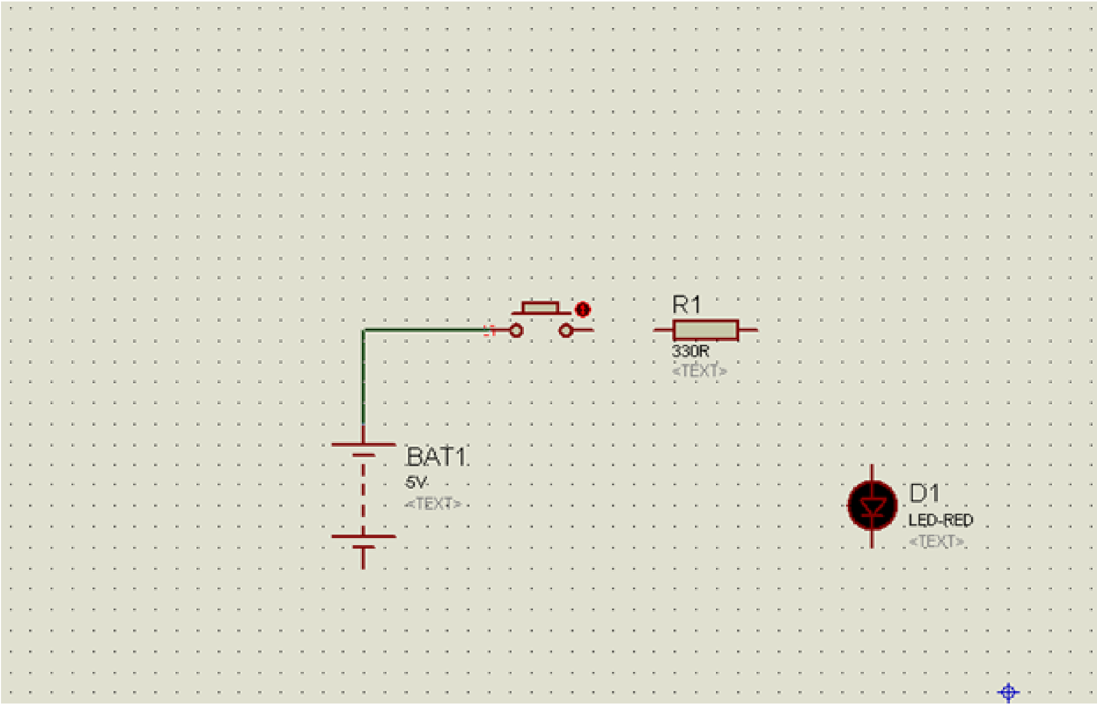


Figure 5.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

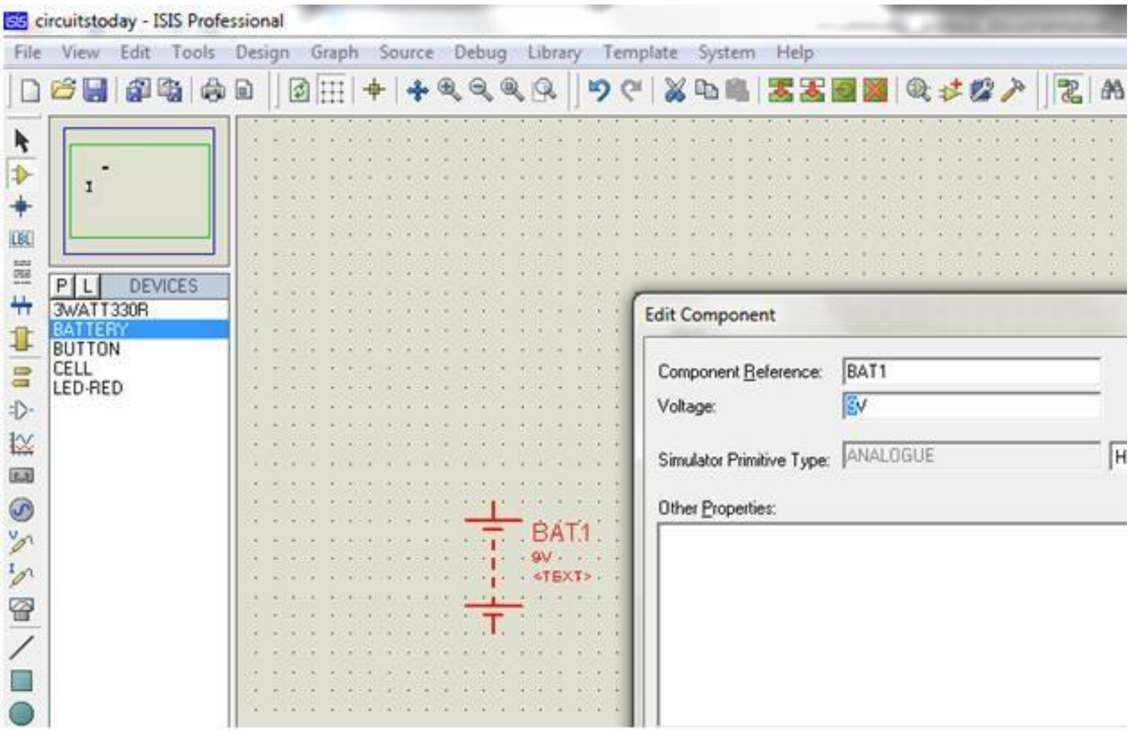


Figure 5.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

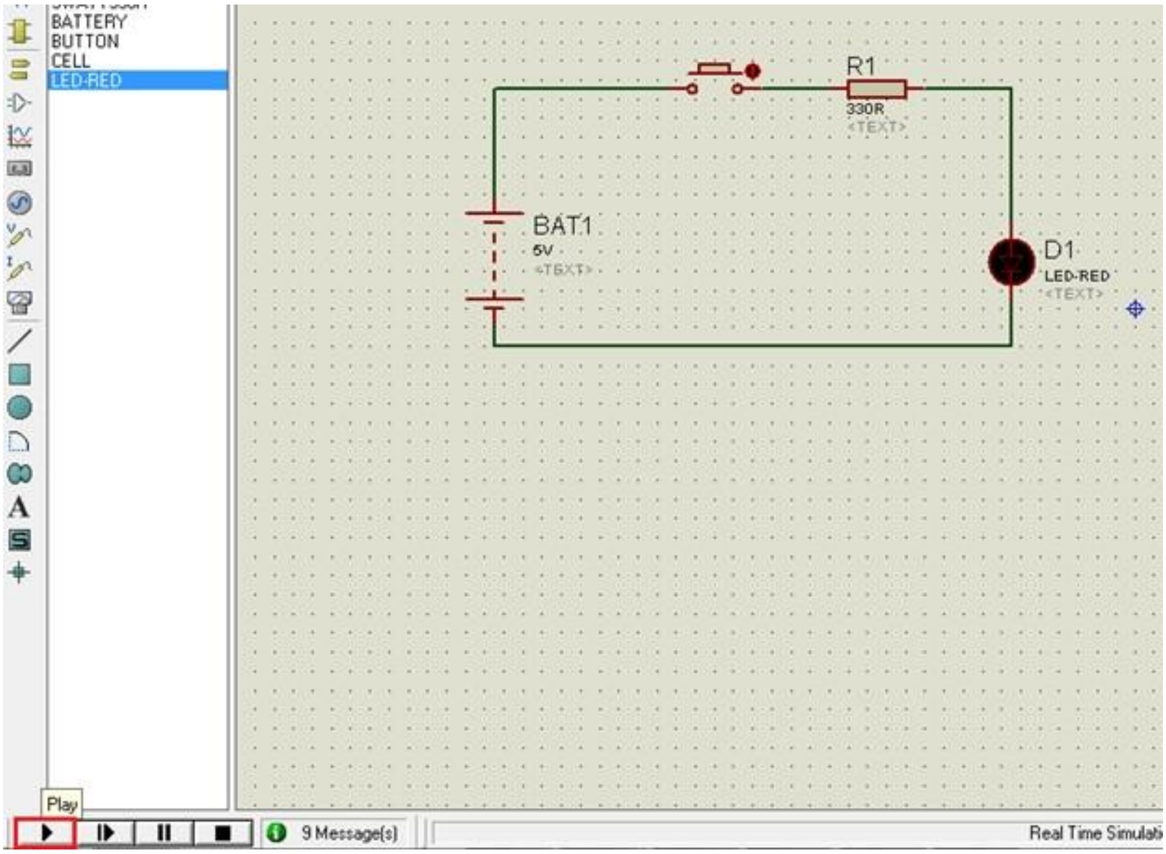


Figure 5.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

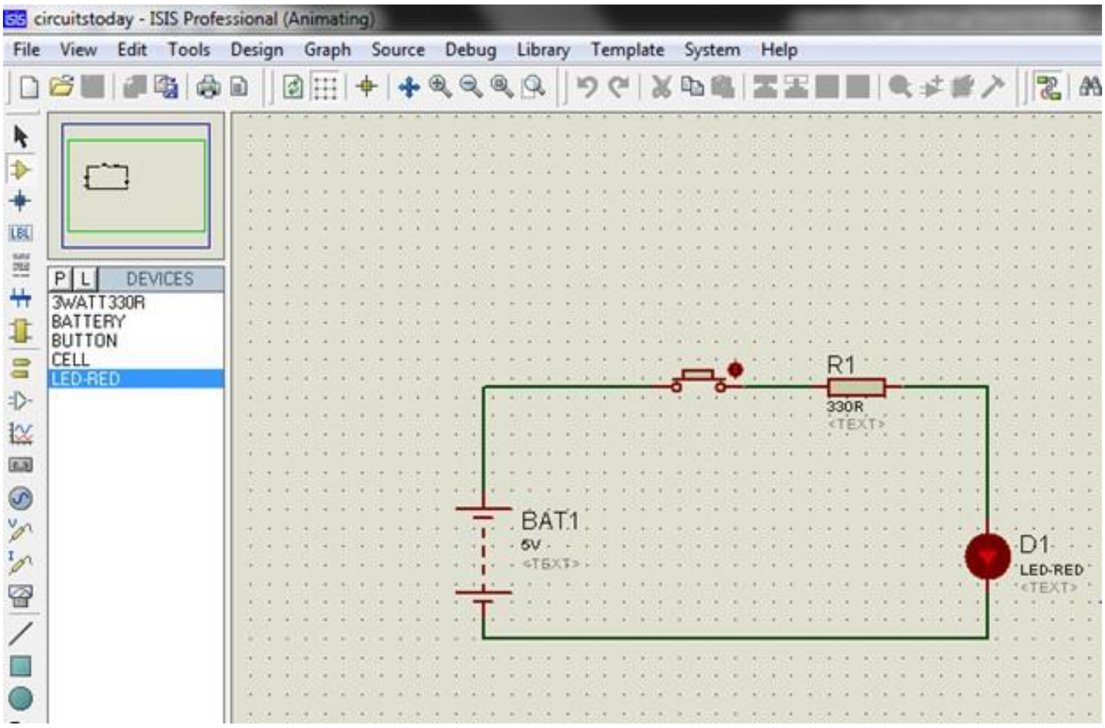


Figure 5.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

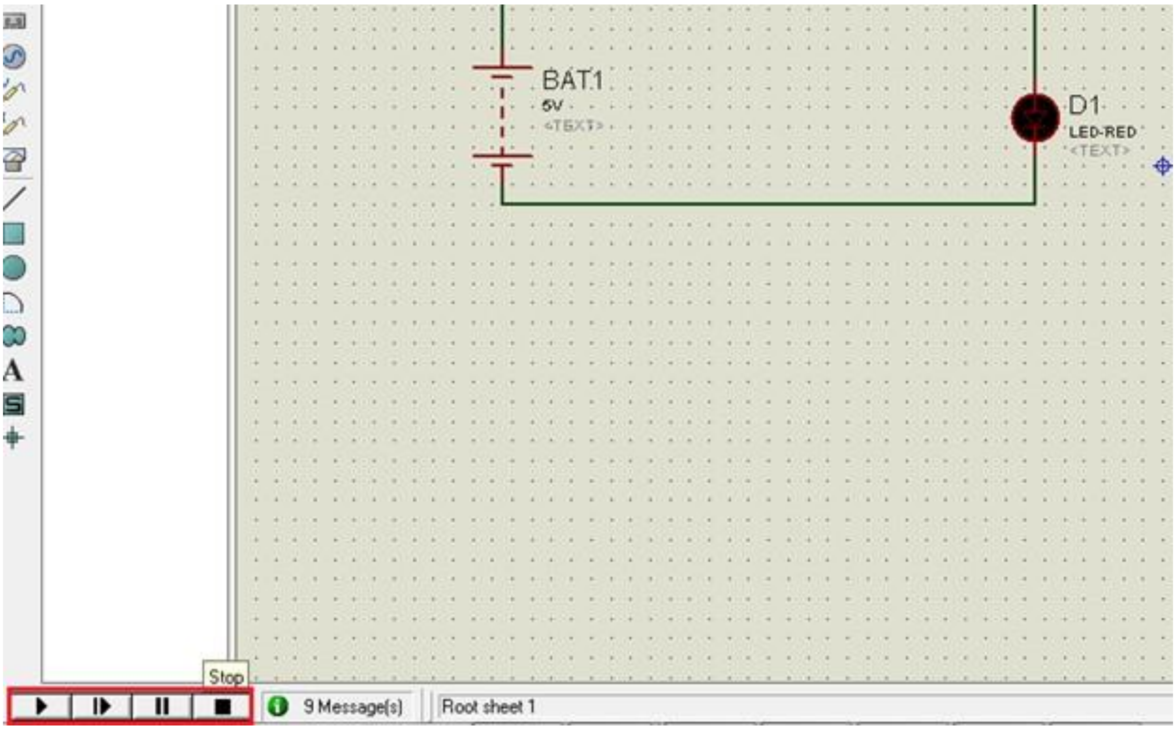


Figure 5.13 Simulation Step-Pause-Stop Buttons

5.3 Arduino IDE

The **Arduino Integrated Development Environment (IDE)** is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Arduino Pro IDE	
Developer(s)	Arduino Software
Preview release	v0.0.2 / 28 October 2019; 3 months ago[8]
Repository	github.com/Arduino/Arduino
Written in	C, C++
Operating system	Windows, macOS, Linux
Platform	IA-32, x86-64, ARM
Type	Integrated development environment
License	LGPL or GPL license
Website	blog.arduino.cc/2019/10/18/arduino-pro-ide-alpha-preview-with-advanced-features/

In October 2019 the Arduino organization began providing early access to a new Arduino Pro IDE with debugging[9] and other advanced features.

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

5.3.1 Installation

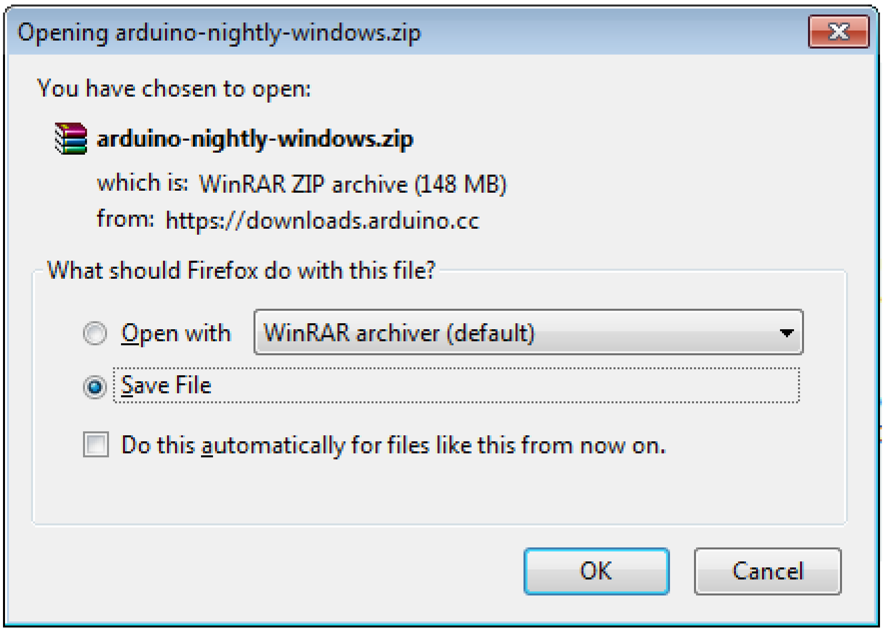
In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 – Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



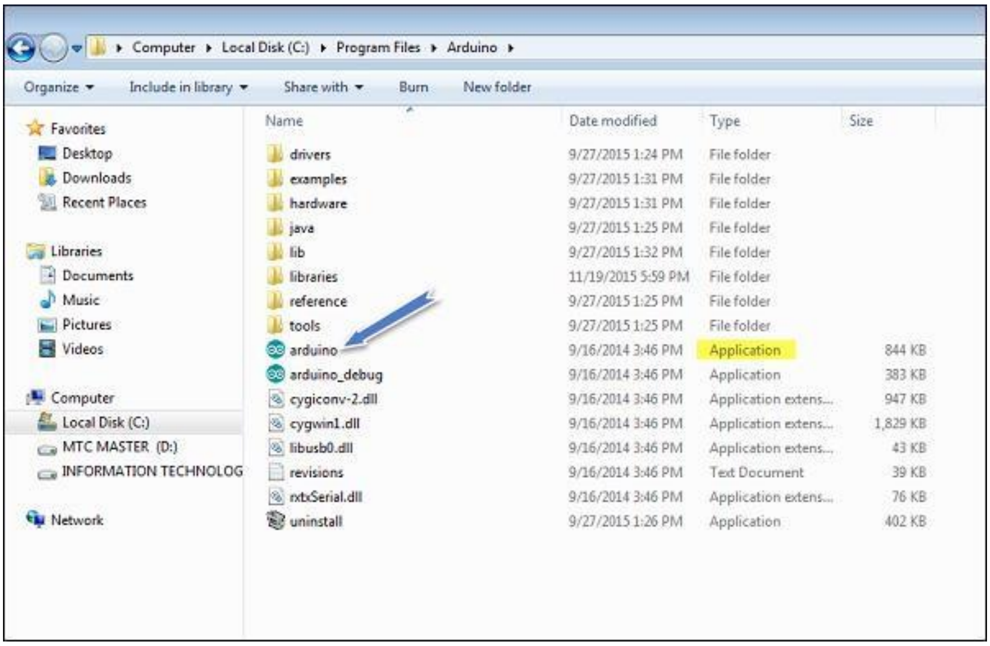
Step 3 – Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 – Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

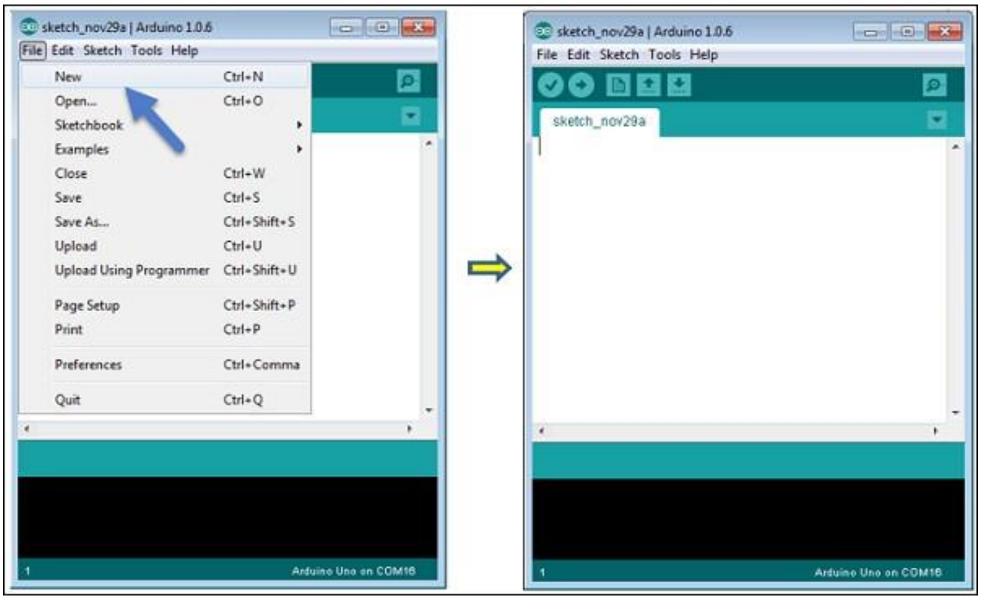


Step 5 − Open your first project.

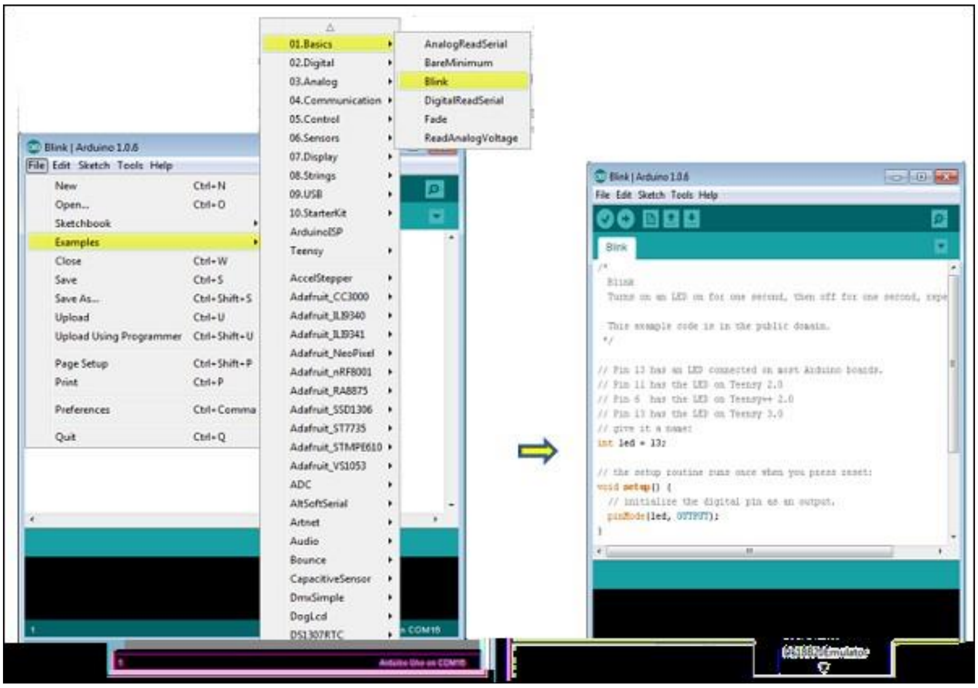
Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.



To open an existing project example, select File → Example → Basics → Blink.

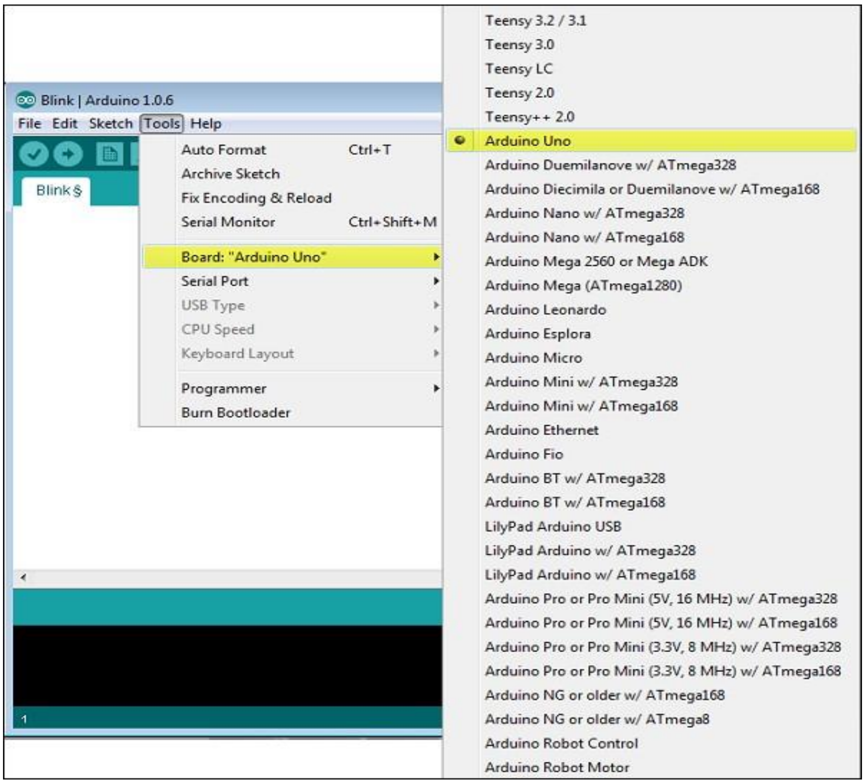


Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

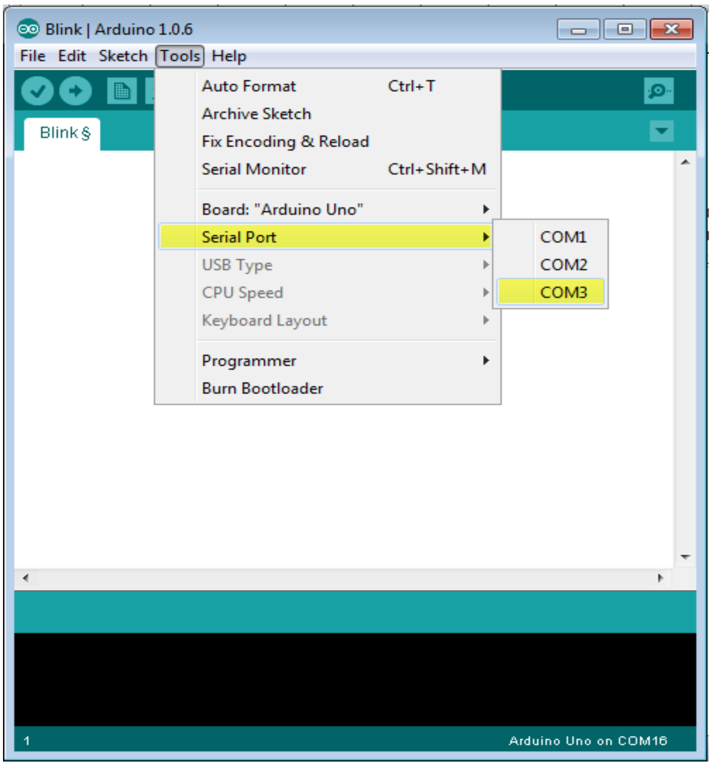
Go to Tools → Board and select your board.



Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

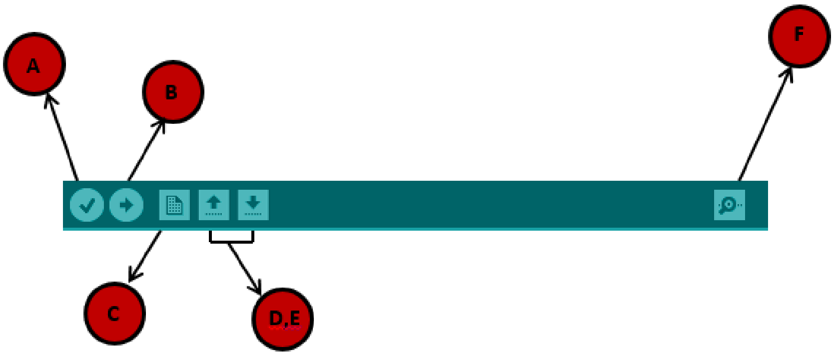
Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.



Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.



A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

CHAPTER 6
HARDWARE

6.1 Arduino

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

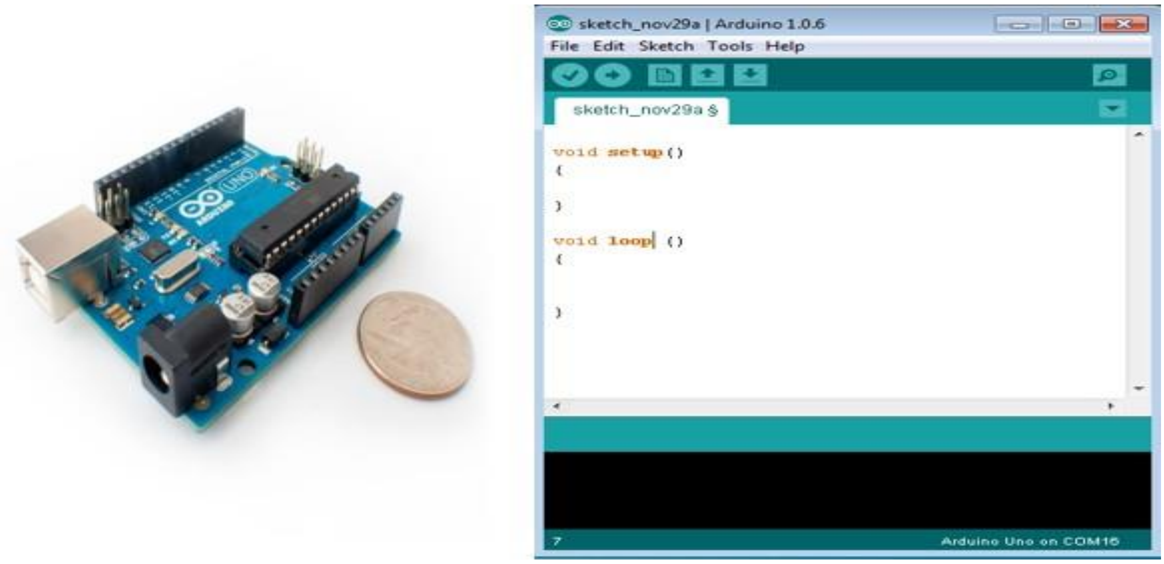


Figure : 6.1 Arduino Board and Dashboard

6.1.1 Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header

Table 6.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

6.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

6.4 Arduino boards based on AT91SAM3X8E microcontroller

Board Name	Operating Volt	Clock Speed	Digital I/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

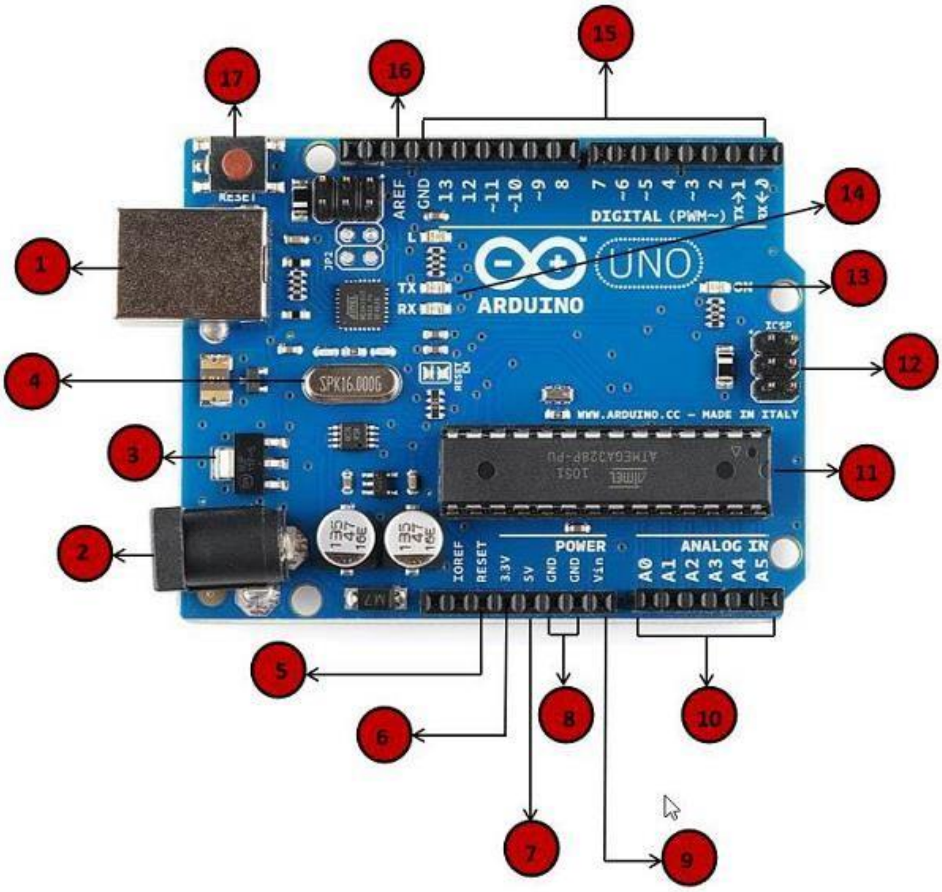


Fig : 6.2 Aurdino Parts

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC.

	For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

6.2 Liquid Cristal Display

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCDs connected to the contollers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines(RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

5.2.1 Description Of 16x2:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all-Parallel Ports. It however doesn't show the use of the Status Port as an input. So what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

Schematic Diagram

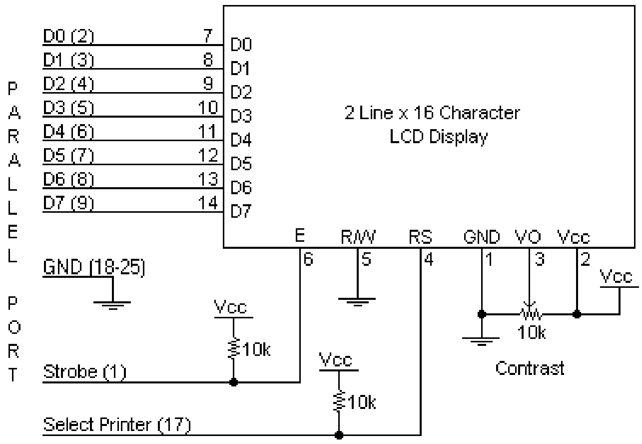


Fig : 6.3 : LCD Panel

- Above is the quite simple schematic. The LCD panel's *Enable* and *Register Select* is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.
- We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.
- The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.

16 x 2 Alphanumeric LCD Module Features:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

16 x 2 Alphanumeric LCD Module Specifications:

Pin	Symbol	Level	Function
1	V_{SS}	-	Power, GND
2	V_{DD}	-	Power, 5V
3	Vo	-	Power, for LCD Drive
4	RS	H/L	Register Select Signal H: Data Input L: Instruction Input
5	R/W	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)

6	E	H,H->L	Enable
7-14	DB0-DB7	H/L	Data Bus; Software selectable 4- or 8-bit mode
15	NC	-	**NOT CONNECTED**
16	NC	-	**NOT CONNECTED**

FEATURES:

• 5 x 8 dots with cursor

• Built-in controller (KS 0066 or Equivalent)

• + 5V power supply (Also available for + 3V)

• 1/16 duty cycle

• B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)

• N.V. optional for + 3V power supply

Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are:

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

Figure 6.4 Address locations for a 1x16 line LCD

Even limited to character based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8,16,20,24,32 and 40 characters are all standard, in one, two and four line versions.

Several different LC technologies exists. “supertwist” types, for example, offer Improved contrast and viewing angle over the older “twisted nematic” types. Some modules are available with back lighting, so that they can be viewed in dimly-lit conditions. The back lighting may be either “electro-luminescent”, requiring a high voltage inverter circuit, or simple LED illumination.

PIN DESCRIPTION:

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections).

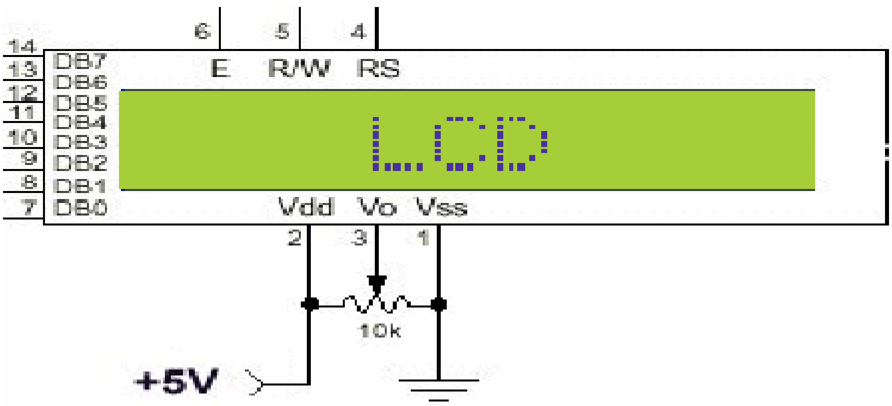


Figure 6.5 Pin diagram of 1x16 lines LCD

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Figure 6.6 Pin specifications

CONTROL LINES:

EN: Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS: Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW: Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands, so RW will almost always be low. Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Logic status on control lines:

• E - 0 Access to LCD disabled

- 1 Access to LCD enabled

• R/W - 0 Writing data to LCD

- 1 Reading data from LCD

• RS - 0 Instructions

- 1 Character

Writing data to the LCD:

1) Set R/W bit to low

2) Set RS bit to logic 0 or 1 (instruction or character)

3) Set data to data lines (if it is writing)

4) Set E line to high

5) Set E line to low

Read data from data lines (if it is reading) on LCD:

1) Set R/W bit to high

2) Set RS bit to logic 0 or 1 (instruction or character)

3) Set data to data lines (if it is writing)

4) Set E line to high

Entering Text:

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the d.i.l. switch pack with a of re-wiring is necessary.

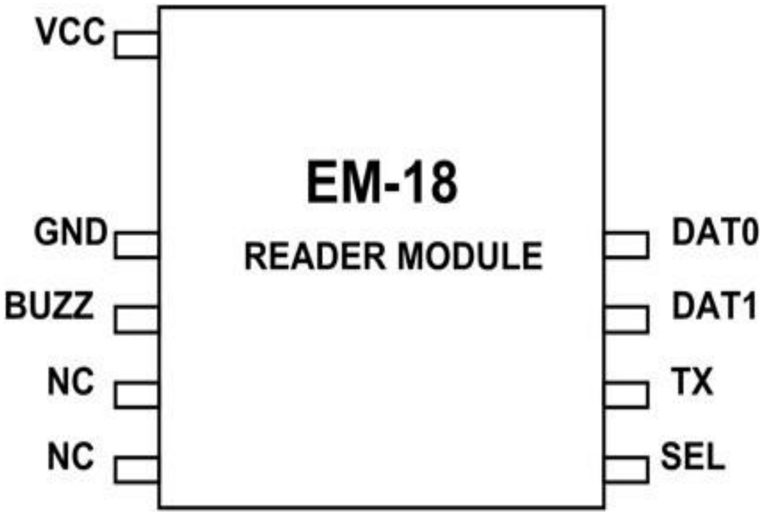
LCD Commands:

There are some present commands instructions in LCD, which we need to send to LCD through some microcontroller. Some important command instructions are given below:

Table 6.5 Commands

Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left
0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

EM18 - RFID Reader Module



EM18 - RFID Reader Module EM18 - RFID Reader Pinout

EM18 RFID Reader is a module which reads the ID information stored in **RFID TAGS**. This ID information is unique for every TAG which cannot be copied.

EM-18 Pin Configuration

EM-18 is a nine-pin device. Among nine pins, 2 pins are not connected, so we basically have to consider seven terminals.

Pin Number	**Description**
VCC	Should be connected to positive of power source.
GND	Should be connected to ground.
BUZZ	Should be connected to BUZZER
NC	No Connection
NC	No Connection
SEL	SEL=1 then o/p =RS232 SEL=0then o/p=WEIGAND
TX	DATA is given out through TX of RS232
DATA1	WEIGAND interface DATA HIGH pin
DATA0	WEIGAND interface DATA LOW pin

EM-18 Features and Specifications

- Operating voltage of EM-18: +4.5V to +5.5V
- Current consumption:50mA
- Can operate on LOW power
- Operating temperature: 0ºC to +80ºC
- Operating frequency:125KHz
- Communication parameter:9600bps
- Reading distance: 10cm, depending on TAG
- Integrated Antenna

How to Use EM-18 RFID Module

EM-18 is used like any other sensor module. First, we choose the mode of communication between MODULE and CONTROLLER. Next, we will program the controller to receive data from module to display. Next power the system. When a tag is brought near the MODULE it reads the ID and sends the information to controller. The controller receives the information and performs action programmed by us.

Step1: Establishing a mode of communication. EM-18 can provide ouput through two communication interfaces. One is RS232 and another is WEIGAND. The form of communication is selected by SEL pin. If SEL pin is selected HIGH then form of communication is RS232 and if SEL pin is pulled LOW then form of communication is WEIGAND. Usually, the RS232 is selected because it's popular so SEL pin is pulled HIGH.

Step2: The output of MODULE bit rate is 9600bps (bit per second). The controller should be programmed to receive information from MODULE at this rate. If bit rate of controller mismatches, then the system will not work correctly.

Now let us consider a **simple EM-18 circuit diagram**,

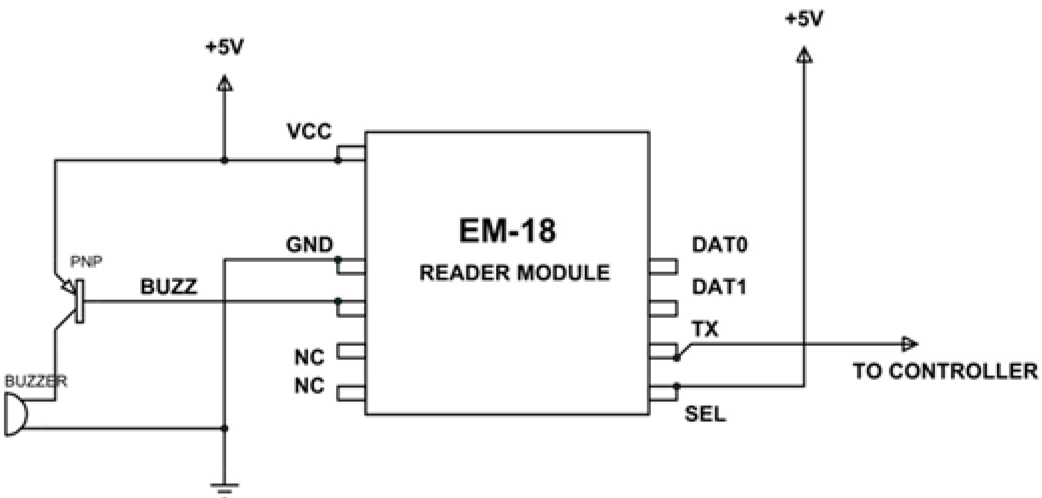


Fig: 6.9: Circuit diagram of EM18 reader module

In the circuit BUZZER is not compulsory. When a TAG is read the BUZZER turns ON. As given in circuit, TX is given to CONTROLLER which is to receive DATA.

Consider a TAG is brought near the MODULE. The MODULE reads the ID and sends the information to controller in 12 ASCII CHARACTERS. In them, 10CHARACTERS represent the TAG ID and 2 CHARACTERS are XOR of previous 10 CHARACTERS.

So, DATA sent = 10ASCII DATA (tag no.) + 2ASCII DATA (XOR result)

Once the Information is sent, the MODULES stop sending DATA. This serial DATA received by the controller though RX pin contains TAG information which is ready for processing. We can program the controller to save the DATA or process it to provide response immediately.

Applications

- Robotics
- Security systems
- Medical tags
- Computer Peripherals
- Package Identification
- Theft protection systems
- Data authorization
- Unique Identity
- Body implants

RFID Tag

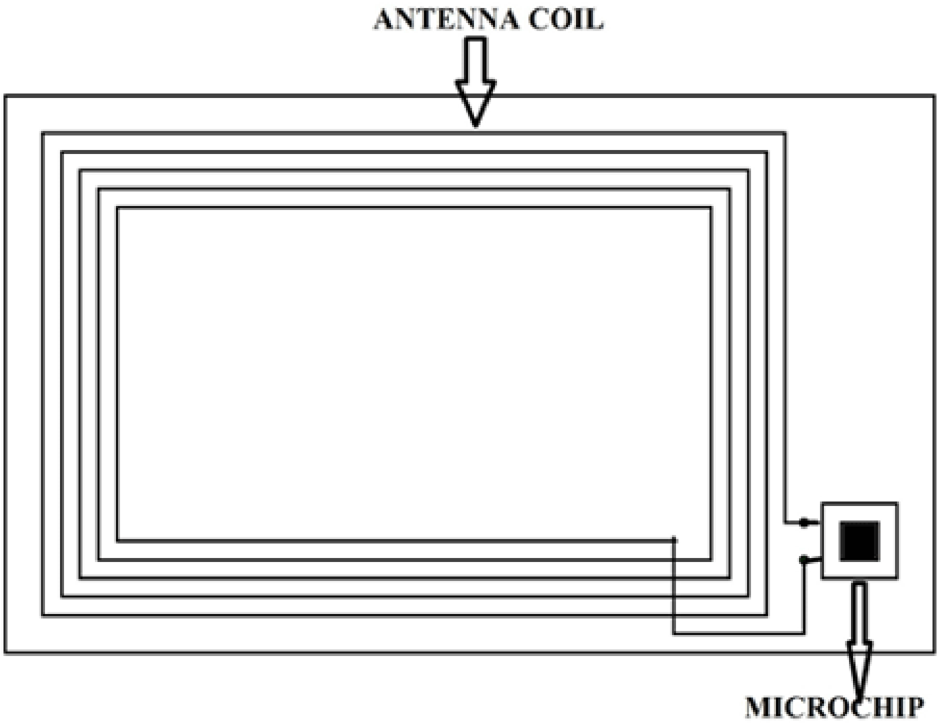


Fig:6.10: RFID Tag

RFID Tag

RFID TAGS are mainly divided in to two **types** - PASSIVE RFID TAGS and ACTIVE RFID TAGS. PASSIVE RFID TAGS are the TAGS with no internal power source. And ACTIVE RFID TAGS are TAGS with their own power source. We will discuss each of them briefly below.

PASSIVE RFID TAGS

In **PASSIVE RFID Tags** we have generally two main components. One is ANTENNA COIL and another is MICROCHIP. As you can see in the above image, there is no BATTERY or any other ACTIVE POWER here.

Working:

With no power source the usually TAG stays ideal. The whole process starts with RFID transmitter. The transmitter generates the electromagnetic radiation through its antenna. When tag gets in range of transmitter, the TAG ANTENNA COIL gets energized through ELECTROMAGNETIC INDUCTION. This energy acts as power source for the MICROCHIP in TAG. This MICROCHIP uses this energy to provide a feedback response for the transmitter. The transmitter receives this response which is unique to the tag and provides the corresponding output.

All PASSIVE TAGS works the same way.

Where PASSIVE TAGS Are Used?

There are various reasons to use PASSIVE TAGS; few are as below:

1. When you want cheaper TAGS. PASSIVE TAGS are very cheap to buy and are available massively. So replacing them will be easy.

2. When TAGS need to be slim. PASSIVE TAGS are very slim. There are available as thin as paper and as small as finger nail.

3. When you need high life. PASSIVE tags do not need any power, so once they are installed in a system they can work for a lifetime.

Types of PASSIVE TAGS

PASSIVE tags are further divided based on operating frequency. As it is known the functioning distance depends on operating frequency, with higher frequency the TAG will be recognized from farther distance. Also with higher frequency the system will be influenced less by other systems.

Low frequency: 125 KHz works up to 10cm.Usually used for ID cards.

Medium frequency: 13.5MHz works up to 1meters. Usually used at airport security.

High frequency: 900MHz works up to 30 meters. Usually used for File tracking and package tracking

How to Use PASSIVE RFID TAGS

Using PASSIVE tags is quite easy. All you need to do is bring the tag near the TRANSMITTERS effective range and all the work will be done by MICROCHIPS present in TRANSMITTER and TAG.

Applications of Passive RFID Tags

- Security systems
- Medical tags
- Package Identification
- Theft protection systems
- Data authorization
- Unique Identity
- Body implants

ACTIVE RFID TAGS

ACTIVE TAGS are tags with power sources fixed internally. So for them there will be usually three main components. One is ANTENNA COIL, second is MICROCHIP and third is power source. For ACTIVE TAGS power sources are usually non rechargeable batteries. These systems usually last for 2-5 years. Once the battery is dead it needs to be replaced for the tag to work again.

There is no standard model for ACTIVE RFID TAGS, they are normally custom made and are available in different sizes and shapes. Appropriate one is chosen depending on requirement. The active tags with battery source on the body are usually bulky.

Where to use ACTIVE TAGS

1. For higher range - Active tags can respond to distances many times higher than passive tags. So ACTIVE TAGS are used for high distance detection.

2. For accurate position - Active tags provide more accurate position then passive ones. Hence active systems are used when direction of system is important.

Types of ACTIVE RFID TAGS

ACTIVE TAGS are divided in to two types based on their working. They are namely Transponders and Beacons.

Transponders: These Tags acts as passive tags and sit ideal under any instance until they receive a signal from transmitter. Once they receive the signal the Transponders turns active and sends a response to the transmitter. By staying ideal these tags save power and usually last longer. These types of tags are used qat tool booths

Beacons: These tags will be ON 24x7 and will be providing specific information for every few seconds. These tags usually have a working range of few hundred meters. These types of tags are used in oil rigs and other industries.

Although they have differences both PASSIVE and ACTIVE TAGS are popular for their own reasons. And both are used extensively.

How to Use ACTIVE RFID TAGS

Using ACTIVE tags is similar to PASSIVE ones. All you need to do is bring the tag near the TRANSMITTER effective range and all the work is will be done by MICROCHIPS present in TRANSMITTER and TAG.

Applications of Active RFID Tags

- Security systems
- Toll booths
- Industrial systems
- Data authorization
- System locating

CHAPTER 7
CONCLUSION AND FUTURE ENHANCEMENT

Conclusion:

The system is expected to be fully automated, reliable, transparent and convenient. The whole system can also be used in vehicle on highways, their toll payment and in the railway ticketing system with small or no modification. The cards being reusable, they are much more convenient compared to the paper-based ticketing system. The card also can be used to be a universal travel pass card that will allow any transportation on any route. Any unwanted events can be avoided as all the person carrying RFID tickets are monitored every time they travel. Also, the possibilities of reducing traffic jams, chaos in the bus stoppage that we usually experienced in Dhaka city are immense.

Reference:

[1] Venugopal Prasanth, Hari Prasad R., K.P. Soman, "Ticketing Solutions for Indian Railways Using RFID Technology," act, pp.217-219, 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, 2009

[2] http://rfid.bemrosebooth.com/benefits_of_rfid_tickets.p hp

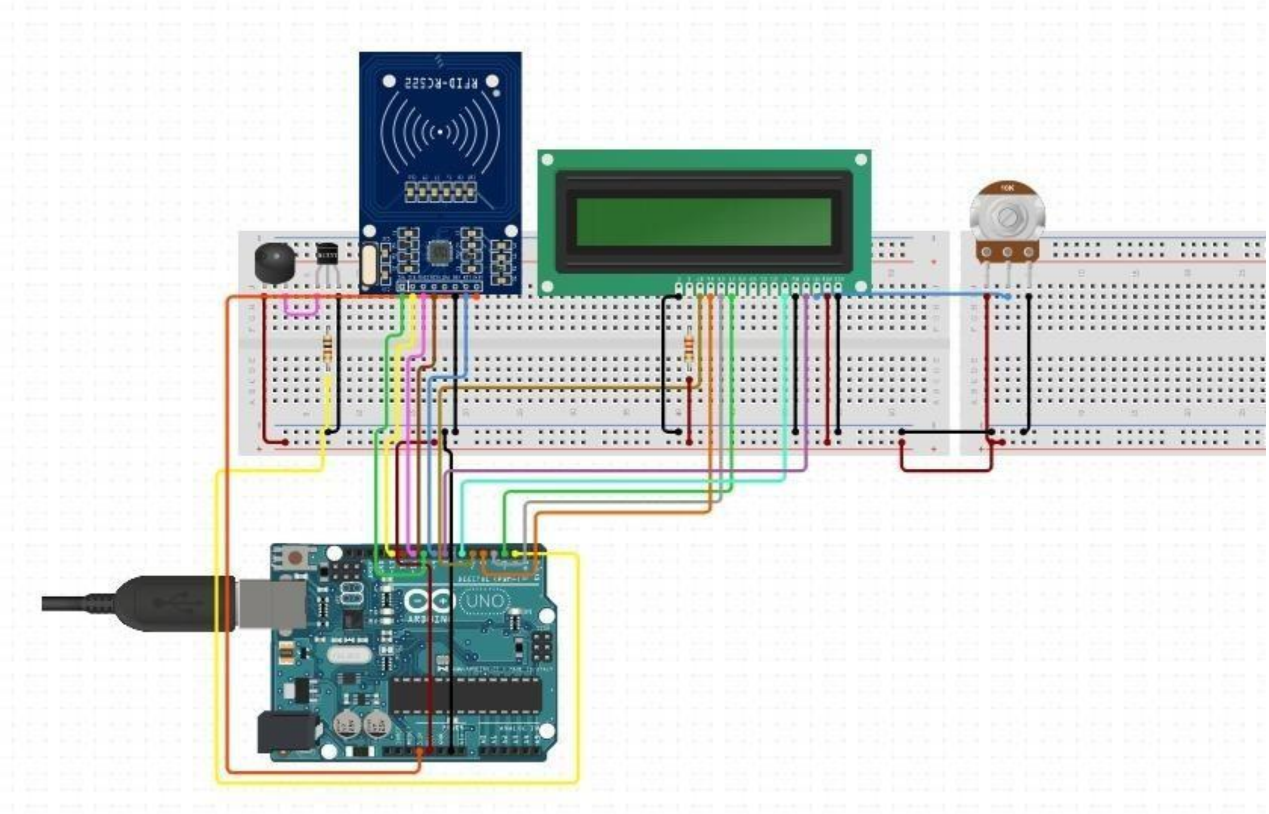
[3] Maria Grazia GNONI, Alessandra ROLLO, Piergiuseppe TUNDO, "A smart model for urban ticketing based on RFID applications," IEEM09-P0572, 2009 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM).

[4] Ana Aguiar, Francisco Nunes, Manuel Silva, Dirk Elias, "Personal Navigator for a Public Transport System using RFIDTicketing":http://inmotion09.dei.uc.pt/papers/Per's onal%20Navigator%20for%20a%20Public%20Transport%20System%20using%20RFID%20Ticketing.p df

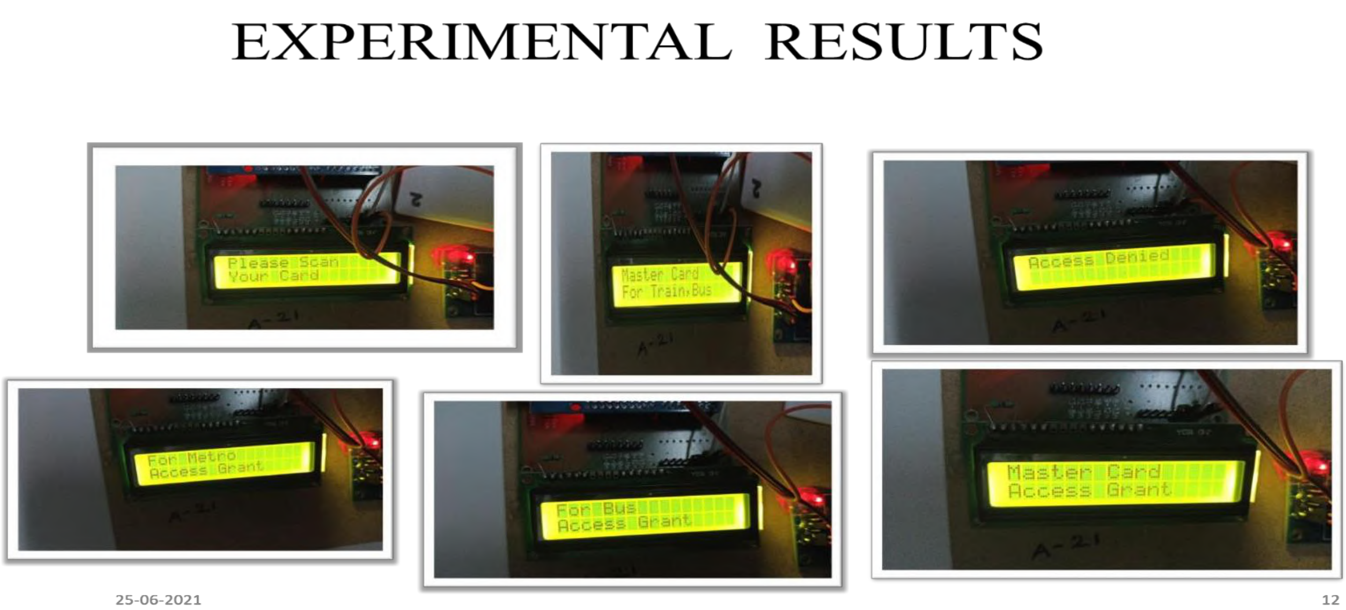
[5] http://www.hbeonlabs.com/synopsis/RFID%20Based% 20Road%20Toll%20Tax.doc http://www.gojohnedwards.com/RFIDMontreal.pdfb

[6] Bernard Menezes1, Kamlesh Laddhad , Karthik B. KReSIT, "Challenges in RFID Deployment – A Case Study in Public Transportation" http://www.it.iitb.ac.in/~kamlesh/Page/Reports/iceg06.p df

[7] "Vehicle Tracking and Ticketing System (VTTS) Using RFID"http://www.slideshare.net/computercriminals/co mpleterfidproject-document-i-presentation



SCHEMATIC DIAGRAM



EXPIREMENTAL RESULTS

A

PROJECT REPORT

on

POWER OPTIMIZATION USING SPECTRUM SHARING FOR NEXT GENERATION CELLULAR NETWORKS

Submitted by

Ms. S.Keerthi Sai	**(18K85A0404)**
Mr. D.Praveen Sagar	**(18K85A0403)**
Ms. K.Shiny	**(17K81A0423)**

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under the Guidance of

Ms. R. ALEKYA, M.E

Assistant Professor

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad-500 100

2020-2021

ST. MARTIN'S ENGINEERING COLLEGE
UGC AUTONOMOUS
(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)
Dhulapally, Secunderabad-500 100
NBA &NAAC A+ Accredited
2020-2021

BONAFIDE CERTIFICATE

This is to certify that the project entitled "Power Optimization Using Spectrum Sharing for Next Generation Cellular Networks" is being submitted by **Ms. S. Keerthi Sai 18K85A0404, Mr. D. Praveen Sagar 18K85A0403, Ms. K. Shiny 17K81A0423** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

INTERNAL GUIDE	**HEAD OF DEPARTMENT**
Ms. R. ALEKYA	**Dr.B. HARI KRISHNA**
M E, Assistant Professor (ECE)	PH. D(ECE)

EXTERNAL EXAMINER

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **DINDU PRAVEEN SAGAR** WITH ROLL NO.**18K85A0403**, **KARROLLA SHINY** WITH ROLL NO.**17K81A0423**, **S KEERTHI SAI** WITH ROLL NO.**18K85A0404**, OF B. TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY**.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**POWER OPTIMIZATION USING SPECTRUM SHARING FOR NEXT GENERATION CELLULAR NETWORKS**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of '**Bachelor of Technology in Department of Electronics and Communication Engineering**', session: 2017-2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project work entitled '**POWER OPTIMIZATION USING SPECTRUM SHARING FOR NEXT GENERATION CELLULAR NETWORKS**' is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

S KEERTHI SAI	**D PRAVEEN SAGAR**	**K SHINY**
18K85A0404	**18K85A0403**	**17K81A0423**

// ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success**.**

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr. P. Santosh Kumar Patra**, our beloved **Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B. Hari Krishna**, **Head & Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our department **Project Coordinator,**

Dr. A. Anand Kumar, Professor for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come. We sincerely thank our **Class Project Coordinator, Mrs. A. Soumya,** Department of ECE and our project **Internal guide, Ms. R. Alekya, Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level **Project Coordinators Dr. Maalyadri**, **Professor in IT** and **Dr.R. Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. S KEERTHI SAI
2. D PRAVEEN SAGAR
3. K SHINY

ABSTRACT

The rapid increase in the number of subscribers demanding high data rate applications have resulted in maturing of the 4G networks. The next generation (5G) wireless communication networks (WCN's) are required to fulfill these rising requirements, hence aiming to utilize the available spectrum as efficiently as possible. Also this is leading to a detrimental effect on the ecological balance of the environment as the transmit power levels increase correspondingly in the atmosphere.

Hence power optimization has also become a major concern. Various technologies such as massive MIMO, spectrum sharing, device to device communication (D2D), GREEN communication have gained significant attention in aiding spectrum utilization along with power optimization. This proposal intends to optimize power using spectrum sharing for the NGN's to achieve high spectrum and energy efficiency for both the primary and secondary system without introduction of a secondary transmitter. The performance of the proposed model has been compared with the opportunistic spectrum sharing model and other popular resource allocation algorithms. The results obtained confirm the efficiency of the proposed scheme for increased performance of the system

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

1.0 Overview of the Project

A cellular network or mobile network is a communication network where the link to and from end nodes is wireless. The network is distributed over land areas called "cells", each served by at least one fixed-location transceiver (typically three cell sites or base transceiver stations). These base stations provide the cell with the network coverage which can be used for transmission of voice, data, and other types of content. A cell typically uses a different set of frequencies from neighboring cells, to avoid interference and provide guaranteed service quality within each cell. When joined together, these cells provide radio coverage over a wide geographic area. This enables numerous portable transceivers (e.g., mobile phones, tablets and laptops equipped with mobile broadband modems, pagers, etc.) to communicate with each other and with fixed transceivers and telephones anywhere in the network, via base stations, even if some of the transceivers are moving through more than one cell during transmission.

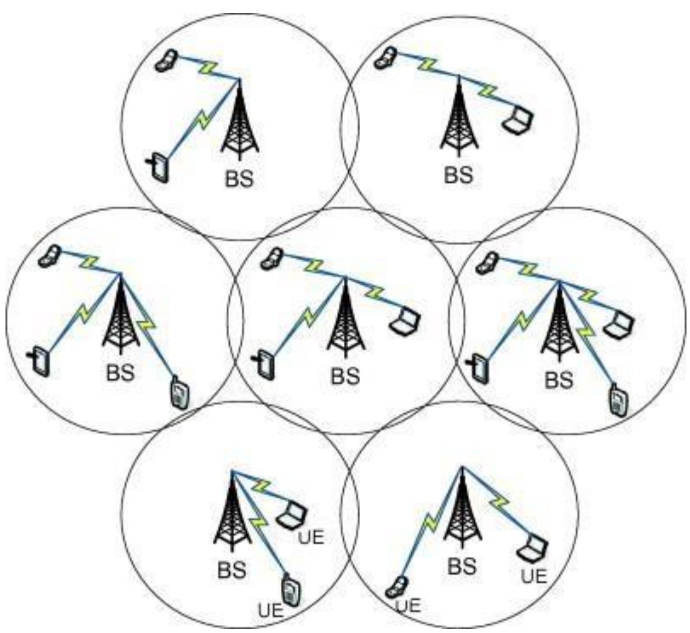


Fig:1.1 Cellular network

5G Cellular Network

5G (from "5th Generation") is the latest generation of cellular mobile communication. It succeeds the 4G (LTE-A, WiMax), 3G (UMTS, LTE) and 2G (GSM) systems. 5G performance targets high data rate, reduced latency, energy saving, cost reduction, higher system capacity, and massive device connectivity. The first phase of 5G specifications in Release-15 will be completed by April 2019 to accommodate the early commercial deployment. The second phase in Release-16 is due to be completed by April 2020 for submission to the International Telecommunication Union (ITU) as a candidate of IMT-2020 technology.

The ITU IMT-2020 specification demands speeds up to 20 Gbit/s, achievable with wide channel bandwidths and massive MIMO. 3rd Generation Partnership Project (3GPP) is going to submit 5G NR (New Radio) as its 5G communication standard proposal. 5G NR can include lower frequencies (FR1), below 6 GHz, and higher frequencies (FR2), above 24 GHz and into the millimeter waves range. However, the speed and latency in early deployments, using 5G NR software on 4G hardware (non-standalone), are only slightly better than new 4G systems, estimated at 15% to 50% better. Simulation of standalone eMBB deployments showed improved throughput between 2.5×, in the FR1 range, and nearly 20×, in the FR2 range.

Like the earlier generation 2G, 3G, and 4G mobile networks, 5G networks are digital cellular networks, in which the service area covered by providers is divided into a mosaic of small geographical areas called *cells*. Analog signals representing sounds and images are digitized in the phone, converted by an analog to digital converter and transmitted as a stream of bits. All the 5G wireless devices in a cell communicate by radio waves with a local antenna array and low power automated transceiver (transmitter and receiver) in the cell, over frequency channels assigned by the transceiver from a common pool of frequencies, which are reused in geographically separated cells. The local antennas are connected with the telephone network and the Internet by a high bandwidth optical fiber or wireless backhaul connection. Like existing cellphones, when a user crosses from one cell to another, their mobile device is automatically “handed off” seamlessly to the antenna in the new cell.

Their major advantage is that 5G networks achieve much higher data rates than previous cellular networks, up to 10 Gbit/s; which is faster than current cable internet, and 100 times faster than the previous cellular technology, 4G LTE. Another advantage is lower network latency (faster response time), below 1 ms (millisecond), compared with 30 - 70 ms for 4G. Because of the higher data rates, 5G networks will serve not just cellphones but are also envisioned as a general home and office networking provider, competing with wired internet providers like cable. Previous cellular networks provided low data rate internet access suitable for cellphones, but a cell tower could not economically provide enough bandwidth to serve as a general internet provider for home computers.

1.1 Objectives of study

5G networks achieve these higher data rates by using higher frequency radio waves, in or near the millimeter wave band from 30 to 300 GHz, whereas previous cellular networks used frequencies in the microwave band between 700 MHz and 3 GHz. A second lower frequency range in the microwave band, below 6 GHz, will be used by some 5G providers, but this will not have the high speeds of the new frequencies. Because of the more plentiful bandwidth at millimeter wave frequencies, 5G networks will use wider frequency channels to communicate with the wireless device, up to 400 MHz compared with 20 MHz in 4G LTE, which can transmit more data (bits) per second. OFDM (orthogonal frequency division multiplexing) modulation is used, in which multiple carrier waves are transmitted in the frequency channel, so multiple bits of information are being transferred simultaneously, in parallel.

Millimeter waves are absorbed by gases in the atmosphere and have shorter range than microwaves, therefore the cells are limited to smaller size; 5G cells will be the size of a city block, as opposed to the cells in previous cellular networks which could be many kilometers across. The waves also have trouble passing through building walls, requiring multiple antennas to cover a cell. Millimeter wave antennas are smaller than the large antennas used in previous cellular networks, only a few inches long, so instead of a cell tower 5G cells will be covered by many antennas mounted on telephone poles and buildings. Another technique used for increasing the data rate is massive MIMO (multiple-input multiple-output). Each cell will have multiple antennas communicating with the wireless device, each over a separate frequency channel, received by multiple antennas in the device, thus multiple bitstreams of data will be transmitted simultaneously, in parallel. In a technique called beamforming the base station computer will continuously calculate the best route for radio waves to reach each wireless device, and will organise multiple antennas to work together as phased arrays to create beams of millimeter waves to reach the device. The smaller, more numerous cells makes 5G network infrastructure more expensive to build per square kilometer of coverage than previous cellular networks. Deployment is currently limited to cities, where there will be enough users per cell to provide an adequate investment return, and there are doubts about whether this technology will ever reach rural areas.

The new 5G wireless devices also have 4G LTE capability, as the new networks use 4G for initially establishing the connection with the cell, as well as in locations where 5G access is not available.

The high data rate and low latency of 5G are envisioned as opening up new applications in the near future. One is practical virtual reality and augmented reality. Another is fast machine-to-machine interaction in the Internet of Things. For example, computers in vehicles on a round continuously communicate with each other,and with the road, by 5G.

Performance Targets

5G systems in line with IMT-2020 specifications are expected to provide enhanced device and network-level capabilities, tightly coupled with intended applications. The following eight parameters are key capabilities for IMT-2020 5G

Capability	Description	5G target	Usage scenario
Peak data rate	Maximum achievable data rate	20 Gbit/s	eMBB
User experienced data rate	Achievable data rate across the coverage area (hotspot cases)	1 Gbit/s	eMBB
User experienced data rate	Achievable data rate across the coverage area	100 Mbit/s	eMBB
Latency	Radio network contribution to packet travel time	1 ms	URLLC
Mobility	Maximum speed for handoff and QoS requirements	500 km/h	eMBB/URLLC
Connection density	Total number of devices per unit area	$10^6/km^2$	MMTC
Energy efficiency	Data sent/received per unit energy consumption (by device or network)	Equal to 4G	eMBB
Spectrum efficiency	Throughput per unit wireless bandwidth and per network cell	3–4x 4G	eMBB
Area traffic capacity	Total traffic across coverage area	1000 (Mbit/s)/m^2	eMBB

Table 1.1 Performance Targets

Note that, for 5G NR, according to 3GPP specification when using spectrum below 6 GHz, the performance would be closer to 4G.

Usage scenario

ITU-R have defined three main types of usage scenario that the capability of 5G is expected to enable. They are Enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC), and Massive Machine Type Communications (mMTC).

Enhanced Mobile Broadband (eMBB)

Enhanced Mobile Broadband (eMBB) refers to the use case of using 5G as an evolution to 4G LTE mobile broadband services with faster connection with higher throughput and more capacity. 5G would need to deliver higher capacity, enhance connectivity, and higher user mobility to match these demands, which would require capabilities in the above table with eMBB mark to deliver.

Ultra Reliable Low Latency Communications (URLLC)

Ultra-Reliable Low-Latency Communications (URLLC) refers to the use case of using 5G in mission-critical applications such as factory automation, where uninterrupted and robust exchange of data is of the utmost importance.

Massive Machine Type Communications (mMTC)

Massive Machine-Type Communications (mMTC) refers to the wide area IoT use cases consisting of large numbers of low-cost devices with high requirements on scalability and increased battery lifetime.

Advantages

Speed

5G promises superior speeds in most conditions to the 4G network. Qualcomm presented a simulation at Mobile World Congress that predicts 490 Mbit/s median speeds for 3.5 GHz 5G Massive MIMO and 1.4 Gbit/s median speed for 28 GHz mmWave. 5G NR speed in sub-6 GHz bands can be slightly higher than the 4G with a similar amount of spectrum and antennas, though some 3GPP 5G networks will be slower than some advanced 4G networks, such as T-Mobile's LTE/LAA network, which achieves 500+ Mbit/s in Manhattan.

The 5G specification allows LAA (License Assisted Access) as well but it has not yet been demonstrated. Adding LAA to an existing 4G configuration can add hundreds of megabits per second to the speed, but this is an extension of 4G, not a new part of the 5G standard.

Low communication latency

Network latency is the time it takes to pass a message from sender to receiver. 5G will have much lower latency than previous cellular networks; below 1 millisecond, compared with 30 - 70 ms for 4G.

New use cases

Features of 5G network, including extreme high bandwidth, ultra low latency, and high density connections, are expected to enable many new use cases that are impossible to be done via older network standards. (See Usage scenario) 5G can also increase the effectiveness of ecommerce vendors' activities.

1.2 Scope of the study

With the massive proliferation in the demand for the mobile wireless communications, smart devices with internet-based applications, the expectation from the from the next generation networks has increased manifolds. Also the next generation networks (NGN) are expected to meet the demands of these rising number of users with improved quality of service. Since we know that we have a limited radio frequency resource to carry out all the wireless and the mobile communications, the main aim of the NGN is to use the available radio resource most efficiently. Also with such humungous rise in the connections and the number of devices there is also a proportional rise in the amount of transmit powers emitted. This has led to a huge rise in the carbon footprint in the ecosystem. Various technologies such as massive MIMO, device-to-device (D2D) communication, spectrum sharing have been suggested to utilize the available spectrum efficiently. Wireless data traffic had been dramatically increasing over the past few ears. None the less, the existing techniques are not satisfying with the users' needs in terms of the emergence of applications for daily routines (e.g., proximity aware services). Therefore, there is a wave of popular interest to seek for new paradigms to deal with this problem. In the coming fifth generation (5G) cellular networks, emerging technologies will lead to both disruptive architectural and component design changes. For instance,in 5G wireless communication systems, diverse researchers study different aspects of millimeter wave transmission, which are plentiful because spectrums have become scarce at microwave frequencies. Massive multiple-input multiple-output, which could increase the system throughput is proposed to utilize a very high number of antennas. We know that 2G–3G–4G cellular networks were built under the design premise of having complete control from the infrastructure side.However, this assumption should be dropped in the 5G systems. The base-station-centric architecture of cellular systems may change, and intelligence at the device side, within different layers of protocol stack, should be exploited, for example, by allowing device-todevice (D2D) connectivity.

Fig1.2 Wireless Communication

Types of Wireless Communication

At present, the usage of mobiles has been increased for different requirements like the internet, talking, multimedia, gaming, photos, video capturing, etc. All these services are available on mobile. Using wireless communication services, we can transfer data, voice, images, videos, and many more.

The different services provided by the wireless communication system is a cellular telephone, Radio¸ paging, TV, video conferencing, etc use different communication services, there are different wireless communications systems are developed based on the application. Some of them are discussed below. A wireless Communication system is classified into Simplex, Half Duplex & Full Duplex. The simple wireless communication system is one-way communication. In this type, the communication can be done in one direction only. The best example is the radio broadcast system.

The half Duplex communication system is two-way communication; however, it is not simultaneous. The best example of this type of communication is walkie – talkie. The full Duplex communication system is also two-way communication & it is simultaneous. The best example of this communication system is the mobile phone. In wireless communication, the devices which are used for communication may change from one service to others because these are available in different shape, size & data throughput. The region enclosed through this type of communication system is an essential factor.

- Infrared Communication
- Broadcast Radio
- Microwave Communication
- Wi-Fi
- Mobile Communication Systems
- Bluetooth Technology
- Satellite communication
- ZigBee

Advantages

- Freedom from wires**:** Can be configured with the use of any physical connection.
- Easy to setup **:**Wireless network is easy to expand and setup
- Better or global coverage**:** It provides global reach by providing networking in places such as rural areas, battlefield, etc… where wiring is not feasible.
- Flexibility**:** Wireless network is more flexible and adaptable compared to wired network.
- Cost-effectiveness**:** Since it is easy to install and doesn't require cables, wireless network is relatively cheaper.
- Mobile and portable**:** Wireless network is easy to carry and re-install in another place.

Disadvantages

- As communication is done through open space, it is less secure.
- Unreliability
- More open to interference.
- Increased chance of jamming.
- Transmission speed is comparably less.

Applications of Wireless Communication

- Satellite system
- Television remote control
- Wi-Fi
- Paging system
- Wi-Max
- Security systems
- Cellphones
- Computer interface devices
- Bluetooth
- GPS
- GSM

1.3 Organization of Chapters

1.3.0 Introduction

Device-to-device communication defined as a direct communication between two mobile users without traversing the base station or the core network is considered to be a promising technique, which also offloads the increasing data traffic into user equipments (UEs). In a traditional cellular network, it is implicitly implied that two parties willing to establish the same call will not be in close proximity to each other. Therefore, all communications must go through the base station. However, in the age of data, mobile users in today's cellular networks are potentially in range for direct communications using high data rate services. Thus, D2D communication, which can decrease latency and increase resource utilization had been proposed as a means of taking advantage of the physical proximity of communicating devices. Figure 1 shows a simple example of D2D communication. The majority of the literatures in D2D communication proposed to use the cellular spectrum for both D2D and cellular communication. Most of these previous studies have focused on issues such as resource allocation and interference mitigation. Although, few existing studies have investigated the D2D access procedure. Here we review the literatures related to device discovery and access procedure. In TR 22.803, the D2D discovery is categorized into several types, which are summarized in. In addition, the D2D discovery procedure and long-term evolution (LTE)-based design are also discussed in. Yanget al. proposed a distributed peer discovery protocol for LTE-A networks.In, they provided an overview of the new agreements in third generation partners hipproject LTE radioaccess networks related to evaluation methodology and channel modeling for D2D discovery and communications. Hong et al. proposed a D2D discovery and link setup procedure and analyzed its performance in terms of energy consumption and delay by utilizing the measurement results of real LTE smartphones. However, all of the existing works lack the overall performance analysis based on the Markov process model. In this work, we will provide the system model based on the Markov process and present the performance analysis. Moreover,we give our proposal on theVienna Matlab platform, which is a system level Matlab simulator developed by Vienna University of Technology and obtain the simulation results.

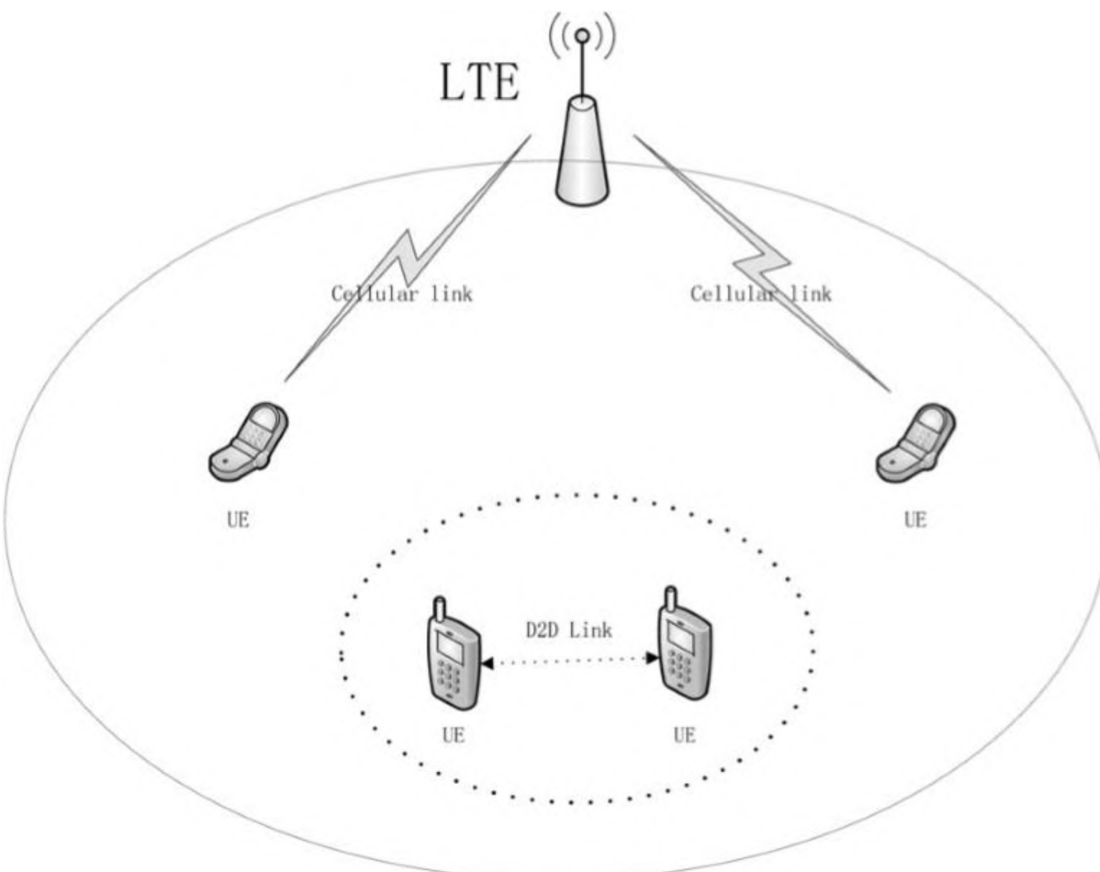


Fig 1.4 An Example of Device-to-Device Communication in a Cellular Network. LTE, Long-Term Evolution.

The rest of this paper is organized as follows: In Section 2, we will describe the system model. After that, we will present the access procedure model and performance analysis in Section 3. In Section 4, the numerical simulation will be provided.Finally,we conclude the paper in Section 5.

The next generation technologies are expected to improve the spectrum utilization along with a reduction in the power levels. Spectrum sharing is gaining popularity with regard to increasing the spectrum and the energy efficiency of the network. Spectrum sharing allows the cooperative use or the simultaneous use of the radio frequency resource by a number of independent entities in a particular geographical area. Spectrum sharing can help in effective utilization of the white spaces or the underutilized parts of the spectrum.

Also there are various power allocation strategies for optimum resource block allocation in a spectrum sharing scenario. The optimum resource block allocation strategies aims at joint optimization of the spectrum and energy efficiency of the system with increase in the quality of service for both the primary and the secondary receivers. The improved spectrum utilization can also aid in supporting the leading technology Internet of things (IoT). So we see that this area is gaining significant attention for improving the spectrum as well as energy efficiency of the network. The suggested proposal discusses both the aspects of increasing the spectrum as well as energy efficiency of the system.

1.3.1 Literature survey

Device-to-Device (D2D) communication in cellular networks is defined as direct communication between two mobile users without traversing the Base Station (BS) or core network. D2D communication is generally non-transparent to the cellular network and it can occur on the cellular frequencies (i.e., inband) or unlicensed spectrum (i.e., outband).

In a traditional cellular network, all communications must go through the BS even if communicating parties are in range for proximity-based D2D communication. Communication through BS suits conventional low data rate mobile services such as voice call and text messaging in which users are seldom close enough for direct communication. However, mobile users in today's cellular networks use high data rate services (e.g., video sharing, gaming, proximity-aware social networking) in which they could potentially be in range for direct communications (i.e., D2D). Hence, D2D communications in such scenarios can greatly increase the spectral efficiency of the network. The advantages of D2D communications go beyond spectral efficiency; they can potentially improve throughput, energy efficiency, delay, and fairness.

1.3.2 Project Design

Devices communicates with each other without intermediate notes.It uses cellular spectrum and proximity of equipments provides

- High bit rates/low delays
- Low energy consumption

Radio resources may be simultaneously used by cellular and D2D links so that the same spectral resources can be used more than once within the same cell.uses the same pre-existing cellular infrastructure supports more services and improves current services and applications.

Data Delivery in Non-Cooperative D2D communication

Existing data delivery protocols in D2D communications mainly assume that mobile nodes willingly participate in data delivery, share their resources with each other, and follow the rules of underlying networking protocols. Nevertheless, rational nodes in real-world scenarios have strategic interactions and may act selfishly for various reasons (such as resource limitations, the lack of interest in data, or social preferences).

For example, if a node has limited battery resources or the cost of the network bandwidth delivered by mobile network operators is high, it would not willingly relay data for others until appropriate incentives are provided. Meanwhile, malicious nodes may attack the network in different ways to disturb the normal operation of the data transmission process. An adversary, for example, may drop received messages but produce forged routing metrics or false information with the aim of attracting more messages or decreasing its detection probability. This issue becomes more challenging when colluding attackers boost their metrics to deceive the attack detection systems. Dealing with non-cooperative mobile nodes is very challenging because of the distributed network model and intermittent access of nodes to central authorities.

D2D applications

D2D Communications is used for

1. Local Services: In local service, user data is directly transmitted directly between the terminals and doesn't involves network side, e.g. social media apps, which are based on proximity service.

2. Emergency communications: In case of natural disasters like hurricanes, earthquakes etc., traditional communication network may not work due to the damage caused. Ad-hoc network can be established via D2D which could be used for such communication in such situations.

3. IoT Enhancement: By combining D2D with Internet of things (IoT), a truly interconnected wireless network will be created. Example of D2D-based IoT enhancement is vehicle-to-vehicle (V2V) communication in the Internet of Vehicles (IoV). When running at high speeds, a vehicle can warn nearby vehicles in D2D mode before it changes lanes or slows down.

Challenges

- Peer Discovery
- Resource Allocation
- Sharing of Spectrum
- Interference Management
- Security Problems

1.3.3 Project Implementation

Step1: start

Step2: It is used to convey an information .It transmits and receives the information

Step3: In this step the data is present after initializing the input channels

Step4: considering all the inputs HMM process starts performing the operation and as a result it produces good access rate, power allocation and throughput.

Step5: All the results are plotted and verified.

Step6: stop

1.3.4 Conclusion and Future Enhancement

The proposal presented aims to achieve joint spectrum efficiency and energy efficiency for the NGN's utilizing spectrum sharing. The model validates its performance when compared to the conventional opportunistic spectrum sharing approach and other popular resource allocation schemes. Significant improvement in the QoS and throughput has been observed in the system when Hidden markov model is used in the proposed model. Spectrum Sharing is highly susceptible to jamming attack along with interference and coverage management issues, which remains an open research field.The next generation technologies are expected to improve the spectrum utilization along with a reduction in the power levels. Spectrum sharing is gaining popularity with regard to increasing the spectrum and the energy efficiency of the network.

CHAPTER-2

LITERATURE SURVEY

2.0 Literature Review on Research Area

The future generation networks (5G) are expected to achieve high data rates, reduced latency, increased spectral efficiency and energy efficiency of the system. Since the available spectrum is a scarce resource, its efficient utilization is the prime focus of the next generation networks. Spectrum Sharing is a key aspect that is gaining significant attention as it can prove to be beneficial in meeting the above requirements. In this paper we present an exhaustive survey of spectrum sharing for future generation networks. We discuss the different techniques and methods of spectrum sharing based on which a general architecture has been presented. Next, we discuss spectrum sensing, network selection and channel allocation, power optimization in spectrum sharing as well as the security issues associated. Based on the survey a four layer architecture has been proposed depicting the complete spectrum sharing scenario from spectrum sensing till the security issues. Modern technologies such as Massive MIMO, SWIPT, spectrum harvesting, spectrum relaying have been incorporated in the architecture for optimizing the power during spectrum sharing. A detailed analysis of security attacks has also been presented in the paper. Two application scenarios have been discussed where in spectrum sharing can offer huge advantages to meet the high bandwidth requirements. The paper also includes a list of the current projects that are being conducted by various research groups and institutions on spectrum sharing, for the next generation networks.

2.1 Review Related On Literature Area

In the present scenario, an energy efficiency has become a matter of prime importance for wireless networks. To meet the demands of an increased capacity, an improved data rate, and a better quality of the service of the next-generation networks, there is a need to adopt energy-efficient architectures. Along with these requirements, it is also our social responsibility to reduce the carbon footprint by reducing the power consumption in a wireless network. Hence, a green communication is an urgent need. In this paper, we have surveyed various techniques for the power optimization of the upcoming 5G networks. The primary focus is on the use of relays and small cells to improve the energy efficiency of the network. We have discussed the various scenarios of relaying for the next-generation networks. Along with this, the importance of simultaneous wireless power and information transfer, massive multiple input multiple output, and millimeter waves has been analyzed for 5G networks.

2.2 Conclusion on Reviews

In the proposed model we have considered a spectrum sharing scenario for the next generation networks. The proposed system model has been presented in Fig. 1. Initially the network is deployed similar to a D2D pair in which the spectrum is being shared. We consider a primary transmitter communicating with ‘n’ number of primary receivers in an urban deployment scenario. The channel coefficients of the different links between the primary transmitter (PT) and primary receiver (PR) are modulated over a subset of ‘K’ subcarriers The maximum rate is achieved over the subcarriers that are OFDM modulated and is given by

$$R_{tr} = \sum_{k=1}^{K} \ln(1 + \beta_n^k p^k)$$

Where depicts the signal to noise ratio for every channel depending on the channel gain and the noise figure and is the allocated power to every channel. In the proposed model spectrum sharing is performed without the introduction of a secondary transmitter in assisting the primary system or the secondary receivers. The primary receiver that is best served assists the secondary system in achieving its target utilizing its own transmit power and the remaining subcarriers left.

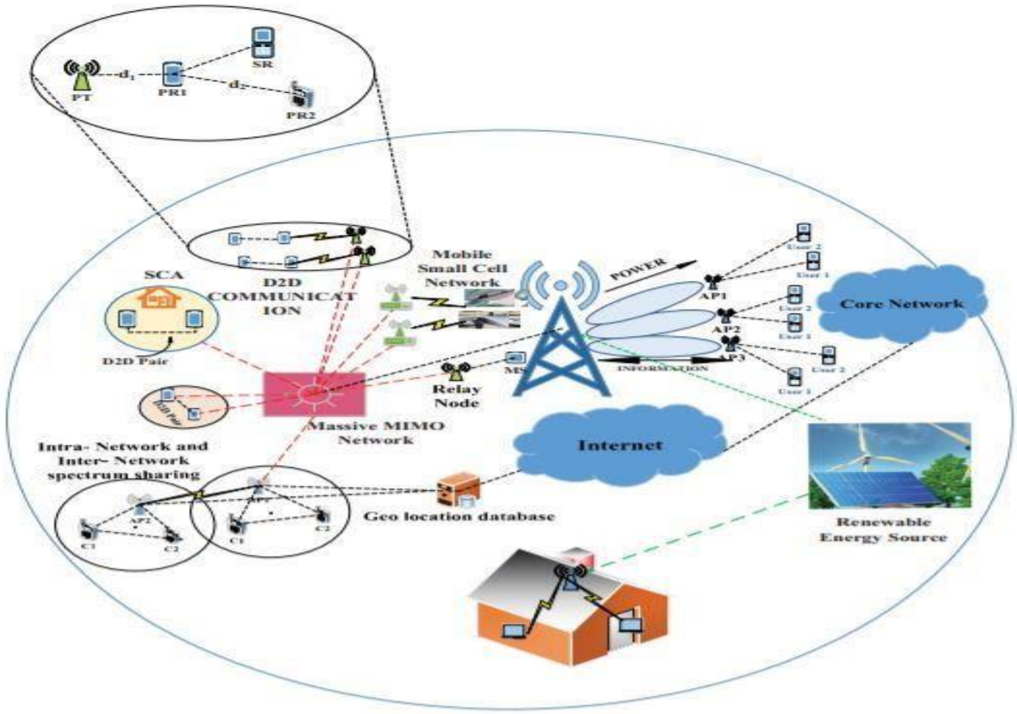


Fig. 2.1 Proposed Model for Power Optimization in a 5G Network using Spectrum Sharing

Different resource allocation techniques to allocate optimum power to the different receivers have been incorporated in the proposed model to make a comparative analysis of the most optimum resource allocation technique.

A comparison has been made with the popular opportunistic spectrum sharing technique based on full duplex cooperative OFDM relaying [13]. The various resource allocation techniques incorporated to study the performance of the model are proportional resource algorithm, water filling algorithm and Hidden Markov model (HMM). Proportional resource algorithm allocates an equal amount of transmit power to all the active users in the network. Water filling algorithm allocates optimum power in the different channels depending on the signal to noise ratio for every channel. The channels with most favorable SNR are allocated the maximum transmit power and vice versa. HMM is a probabilistic method of allocating optimum power to the active users and is based on maximum likelihood criterion. It models both the signal and the noise simultaneously. This model allocates optimum power considering various parameters which are active user distance, SNR and application. The performance of both the opportunistic spectrum sharing model and the proposed model is compared in terms of the power allocated, access rate achieved, throughput and the Mean opinion score (MOS).

The throughput computation for the model is given by the expression where depicts the bandwidth of the resource block. The Mean opinion score gives a measure of the QoE (Quality of Experience) of the users on a scale of 1-5 with 1 depicting the worst experience and 5 the best experience or the quality of the signal. It can be computed with the following expression

$$MOS = 5 - \frac{578}{1 + \left[\frac{B + 541.1}{45.98}\right]^2}$$

Where + is the throughput expressed in Kbps. Following section presents the results and analysis for the proposed model.

CHAPTER-3

PROJECT DESIGN

3.0 Overview of the Design

The players are adverting their network services and presence information through two protocols:

▪ Bonjour

▪ SSDP / UPnP (added in firmware 4.1.0 and Elementi 2018)

Elementi is using the information advertised by the players through any of these two protocols to list the players under Devices tab in Browse panel. If device discovery doesn't work for any reason, the players can be manually added into Elementi.

Both services are using multicast packets, which are restricted to the local area network where the players are located (thus the PC running Elementi must be in the same LAN), might get discarded on WiFi network (especially when crowded) and don't work over VPN connection.

Notes:

● On HMP350, HMP300 and DiVA players, both services can be disabled from Control Center > Network page, should the security policy requires that.

● Discovery across subnetworks is normally prohibited - to activate it, either enable multicast routing of SSDP on the site (applies to SpinetiX players with firmware 4.1.0 or later) or install an mDNS reflector (Avahi has this capability) on the gateway between the subnetworks.

SSDP / UPnP

● Added in firmware 4.1.0 and Elementi 2018.

● Universal Plug and Play (UPnP) is a set of networking protocols that permits networked devices, such as SpinetiX players, personal computers, printers, Internet gateways, Wi-Fi access points and mobile devices to seamlessly discover each other's presence on the network and establish functional network services for data sharing and communications. UPnP discovery protocol, known as Simple Service Discovery Protocol (SSDP), accomplishes the advertisement and discovery of network services and presence information without assistance of server-based configuration mechanisms, such as DHCP or DNS, and without special static configuration of a network host.

● SSDP / UPnP removes the dependency on Bonjour name resolution as IP addresses are used to contact players discovered via UPnP. Furthermore, HMP350, HMP300 and DiVA players show up in the Windows Explorer's Network view in Windows computers when Network Discovery is enabled.

● UPnP uses UDP port 1900 and all used TCP ports are derived from the SSDP alive and response messages.

Bonjour

● Bonjour is Apple Inc's implementation of zero-configuration networking (zeroconf), a group of technologies that includes service discovery, address assignment, and hostname resolution. Bonjour locates devices such as SpinetiX players, printers, other computers, and the services that those devices offer on a local area network using multicast Domain Name System (mDNS) service records and DNS Service Discovery (DNS-SD).

Information Advertised

● The HMP advertises the following services using Bonjour (via mDNS):

http._tcp on port 80 with name "HMP - _device name_";

● This is the standard service web browsers look for.

TXT records:

● path=/ (standard way to signify URL path)

● webdav._tcp on port 81 with name "HMP - _device name_"

This is the standard service WebDAV clients look for.

TXT records:

● path=/ (standard way to signify URL path)

● spx-hmp._tcp on port 80 with name "_device name_"

TXT records:

● txtvers=1 (standard way to support future changes)

● cport=81 (this is the content server port)

● mode= one of

normal when running normally

safe when started in safe mode

recovery when in recovery console

● serial=_serial number_ (e.g., 001d5000001a)

● firmware=_firmware version_ (e.g., 2.1.0-0.1.7844)

● model=_model_ (e.g. HMP100, HMP130 or HMP200)

spx-im._tcp on the port used by the network API (usually 1234)

This is only advertised if the network API is enabled

The device name is the name of the device as configured in Control Center.

Web Browser

Discovery from a web browser is simple if a Bonjour finder is installed (native with Safari). For instance with Internet Explorer open the Bonjour tab, all the HMP devices in the local network will be listed as "HMP - _device name_".

If the Bonjour finder is not available then one can just type spx-hmp-_serial_number_.local. (e.g., spx-hmp-001d5000001a.local.) in the URL bar of the browser, this should open the device's Control Center.

Command Line

If the Bonjour service is installed one can use the command line dns-sd tool to browse the local network.

To discover all HMP devices type (the device names will be listed under the Instance Name column)

dns-sd -B _spx-hmp._tcp

To look up a particular HMP device type (replace instance name by the name returned using the -B option). It will return the device address as spx-hmp-_serial_number_.local

dns-sd -L "instance name" _spx-hmp._tcp

To know the IP address of a device you can use ping like the following (it will show the IP address)

ping spx-hmp-_serial_number_.local.

3.1 Spectrum Sharing

Spectrum sharing is a way to optimize the use of the airwaves, or wireless communications channels, by enabling multiple categories of users to safely share the same frequency bands. Spectrum sharing is necessary because growing demand is crowding the airwaves. Smartphones, the Internet of Things, military and public safety radios, wearable devices, smart vehicles and countless other devices all depend on the same wireless bands of the electromagnetic spectrum to share data, voice and images.

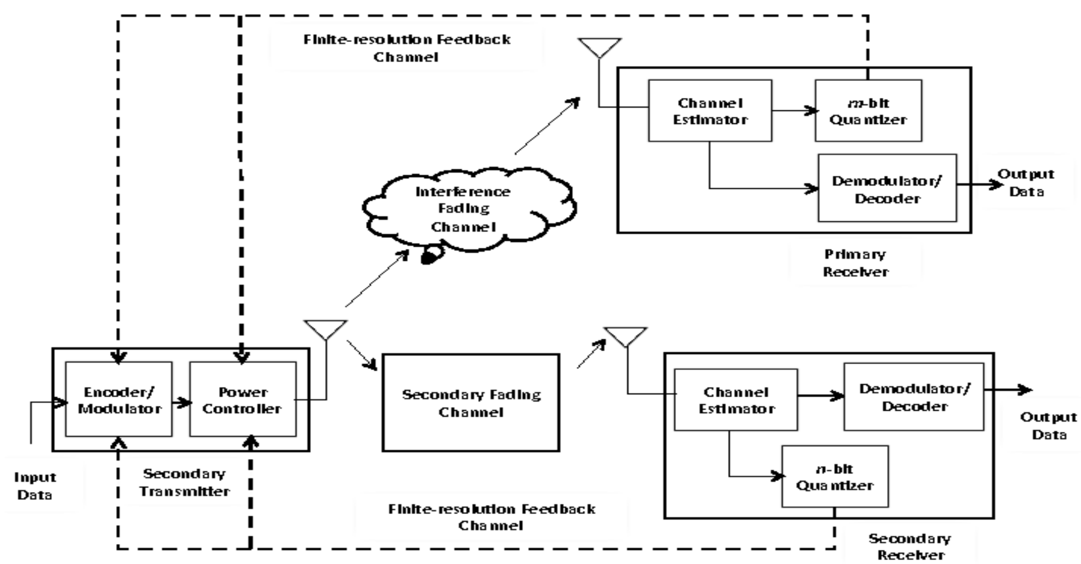


Fig 3.1 Block Diagram of Spectrum Sharing

3.2 Next Generation Network

A next-generation network (NGN) is a packet-based network which can provide services including Telecommunication Services and is able to make use of multiple broadband, quality of service-enabled transport technologies and in which service-related functions are independent from underlying transport-related technologies. It offers unrestricted access by users to different service providers. It supports generalized mobility which will allow consistent and ubiquitous provision of services to users. The next-generation network is a body of key architectural changes in telecommunication core and access networks. The general idea behind the NGN is that one network transports all information and services by encapsulating these into IP packets, similar to those used on the Internet.

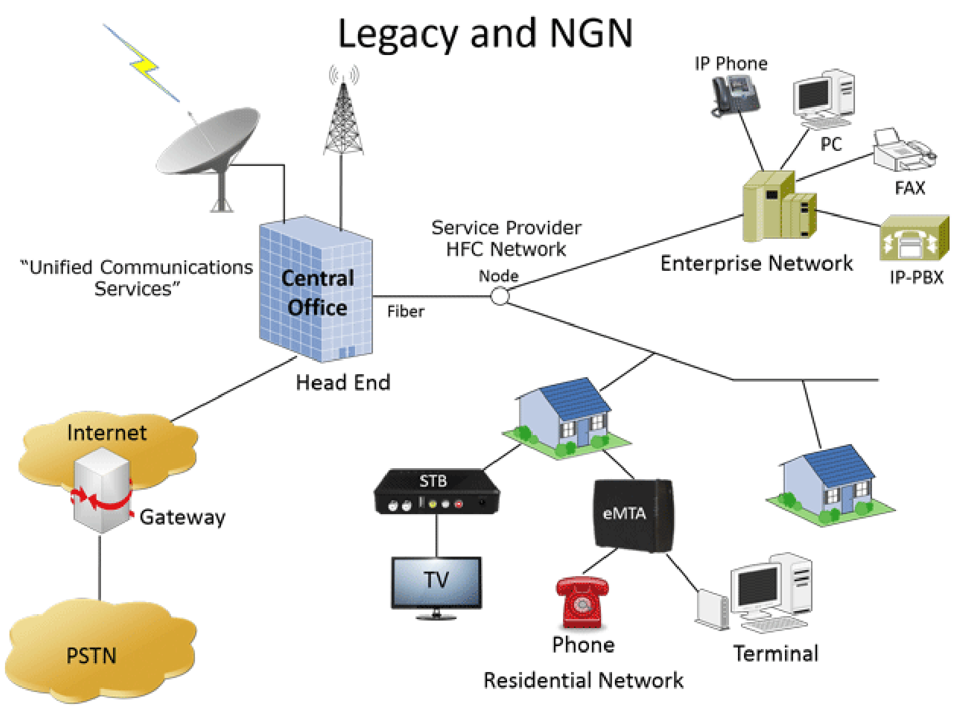


Fig 3.2 Next Generation Network

3.2 5G NR Initial Access Procedure

All the messages exchanged between UE and gNB in 5G NR Initial Access procedure i.e. Random Access Procedure are described. It is also known as initial cell search procedure. 5G NR (New Radio) is the latest cellular wireless technology developed to deliver 10 times fast data rate compare to LTE (i.e. 4G) technology. It follows 3GPP specifications release 15 and above. In this page we will understand initial access procedure i.e. random access procedure. It helps to get the initial uplink grant for UE and helps in performing synchronization with the gNB (i.e. network). It covers Random Access procedure initialization, Random Access Resource selection, Random Access Preamble transmission, Random Access Response reception, Contention Resolution and Completion of the Random Access procedure.

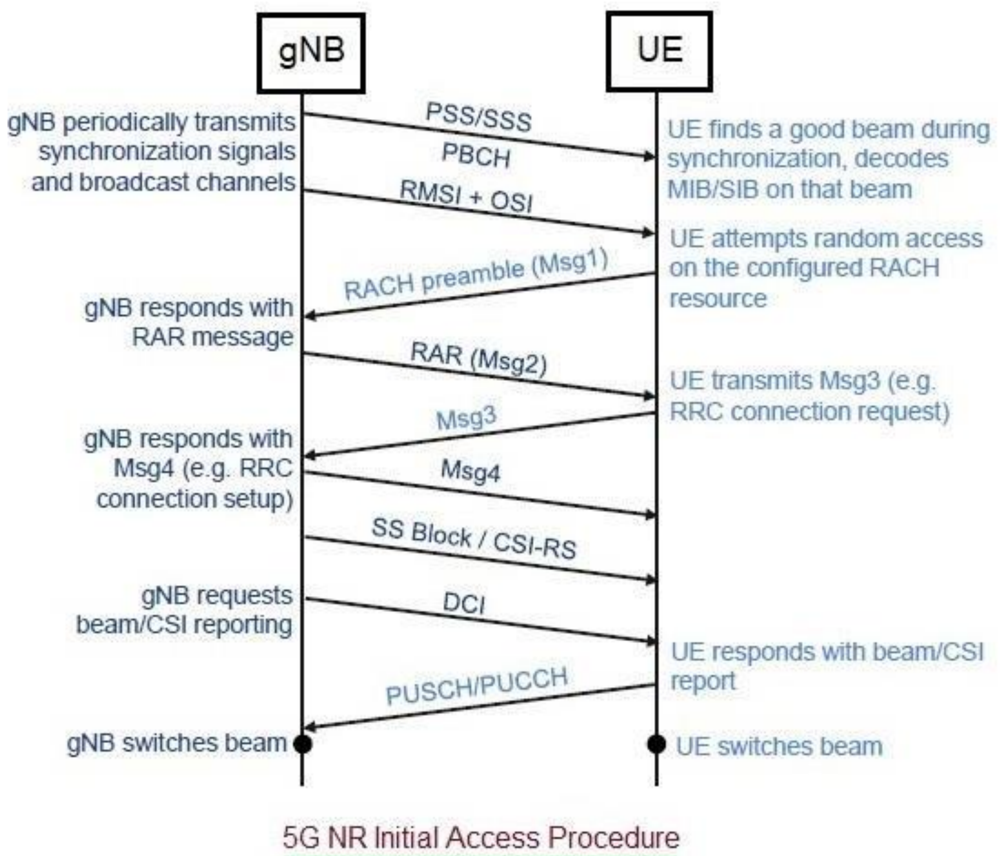
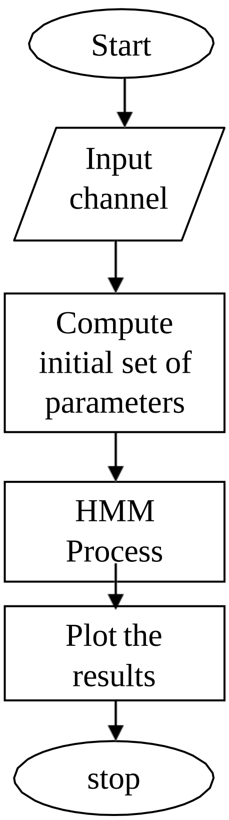


Fig 3.3 5G NR Initial Access Procedure

CHAPTER-4

PROJECT IMPLEMENTATION

Flow Chart



4.0 Implementation Stages

Step1: start

Step2: It is used to convey an information .It transmits and receives the information

Step3: In this step the data is present after initializing the input channels

Step4: considering all the inputs HMM process starts performing the operation and as a result it produces good access rate, power allocation and throughput.

Step5: All the results are plotted and verified.

Step6: stop

Introduction to MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

The MATLAB system

The MATLAB system consists of five main parts

- Development Environment

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- The MATLAB Mathematical Function Library

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- The MATLAB Language

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

- Graphics

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- The MATLAB Application Program Interface (API)

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using IMAGE PROCESSING toolbox.

Graphical User Interface (GUI)

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.

- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type

guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted,

GUIDE opens a new (i.e., blank) window.

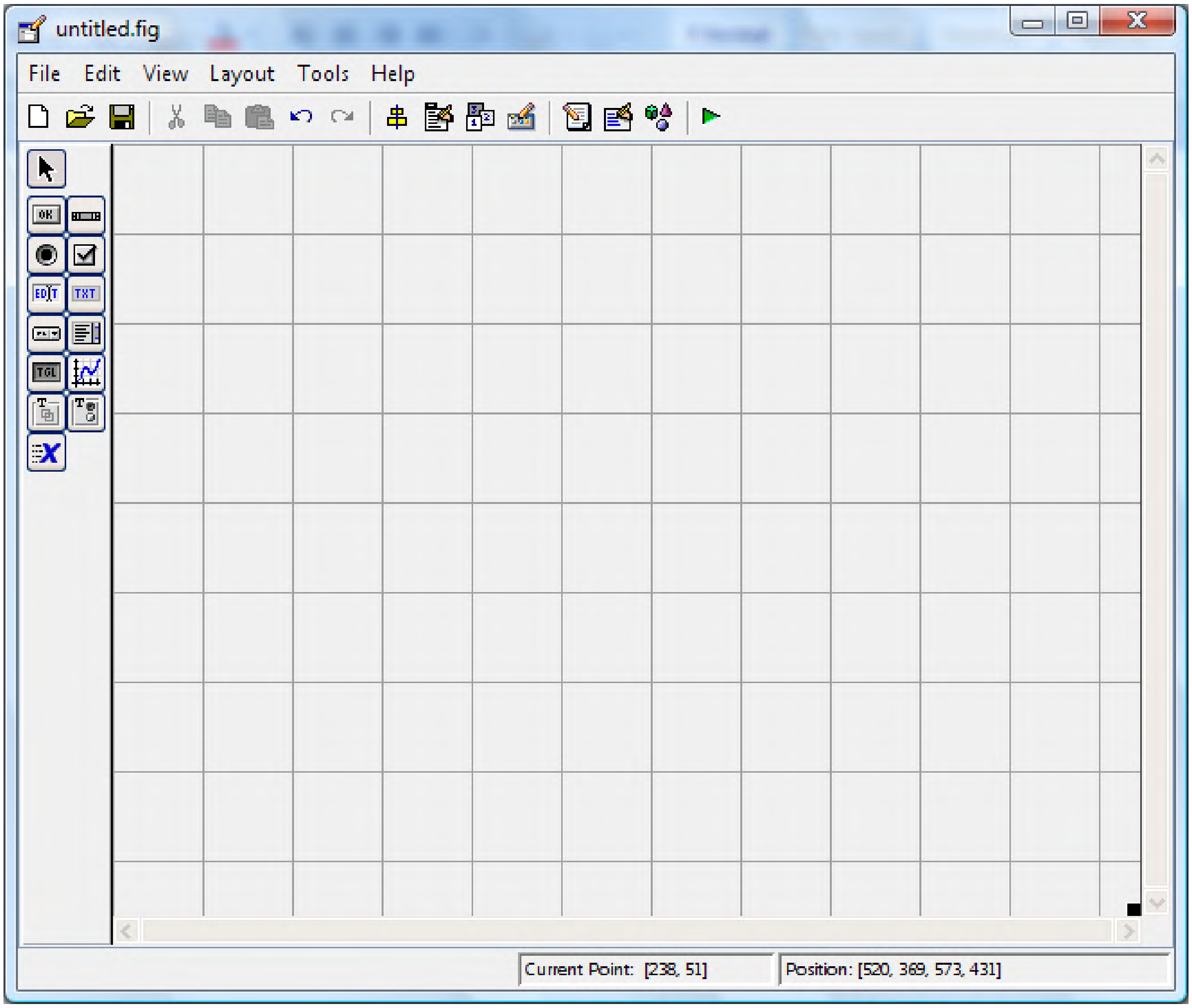


Table 4.1 Creating Matlab GUI

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

Getting Started

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

Introduction - describes the components of the MATLAB system.

Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

Development Environment

Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History:

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type help browser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to runMATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

Manipulating Matrices

Entering Matrices

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.
- Start by entering Dürer's matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]

MATLAB displays the matrix you just entered.

```
A =

    16     3     2    13

     5    10    11     8

     9     6     7    12

     4    15    14     1
```

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

Operators

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

Table 4.2 List of Operators

Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.14159265...
I	Imaginary unit, $\sqrt{-1}$
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

Table 4.3 List of Functions

GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

Creating GUIs with Guide

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

GUI Development Environment

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of its

children (uicontrols and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the

callbacks, which are defined as subfunctions. This M-file is referred to as the application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

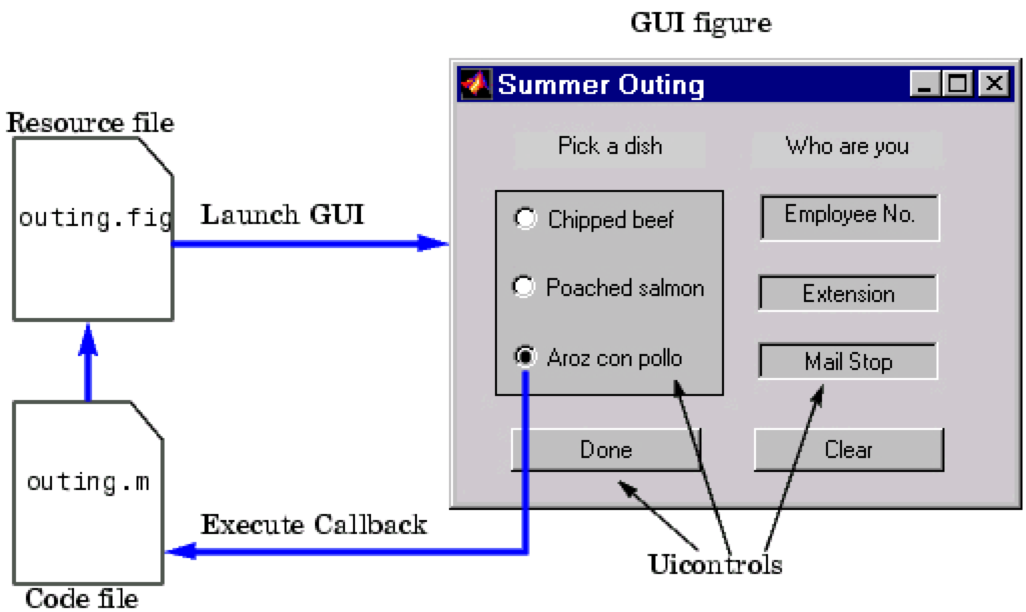


Fig 4.1 Graphical User Blocks

Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques

used in the application M-file.

Application Examples - a collection of examples that illustrate techniques

which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- Listboxes

Checkboxes

- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)

button_state = get(h,'Value');

if button_state == get(h,'Max')

   % toggle button is pressed

elseif button_state == get(h,'Min')

   % toggle button is not pressed

end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);

a(:,:,2) = rand(16,128);

a(:,:,3) = rand(16,128);

set(h,'CData',a)
```

Radio Buttons

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)
```

```
set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)

off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];

mutual_exclude(off)

% Continue with callback

   .

   .

   .
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)
if (get(h,'Value') == get(h,'Max'))
   % then checkbox is checked-take approriate action
else
   % checkbox is not checked-take approriate action
end
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)
user_string = get(h,'string');
% proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function  edittext1_Callback(h,eventdata,handles,varargin)
user_entry = str2double(get(h,'string'));
```

```
if isnan(user_entry)
    errordlg('You must enter a numeric value','Bad Input','modal')
end
% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menubar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menubar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer.

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

Listboxes differentiate between single and double clicks on an item and set the figure SelectionType property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure SelectionType property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

```
val = get(h,'Value');
switch val
case 1
% The user selected the first item
case 2
% The user selected the second item
% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)
val = get(h,'Value');
string_list = get(h,'String');
selected_string = string_list{val}; % convert from cell array to string
% etc.
```

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is

grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes ButtonDownFcn property to define the callback.

Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

```
axes(handles.axes1)
```

makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axes. See GUI with Multiple Axes for and example that uses two axes.

Figure

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control.

4.1 Results

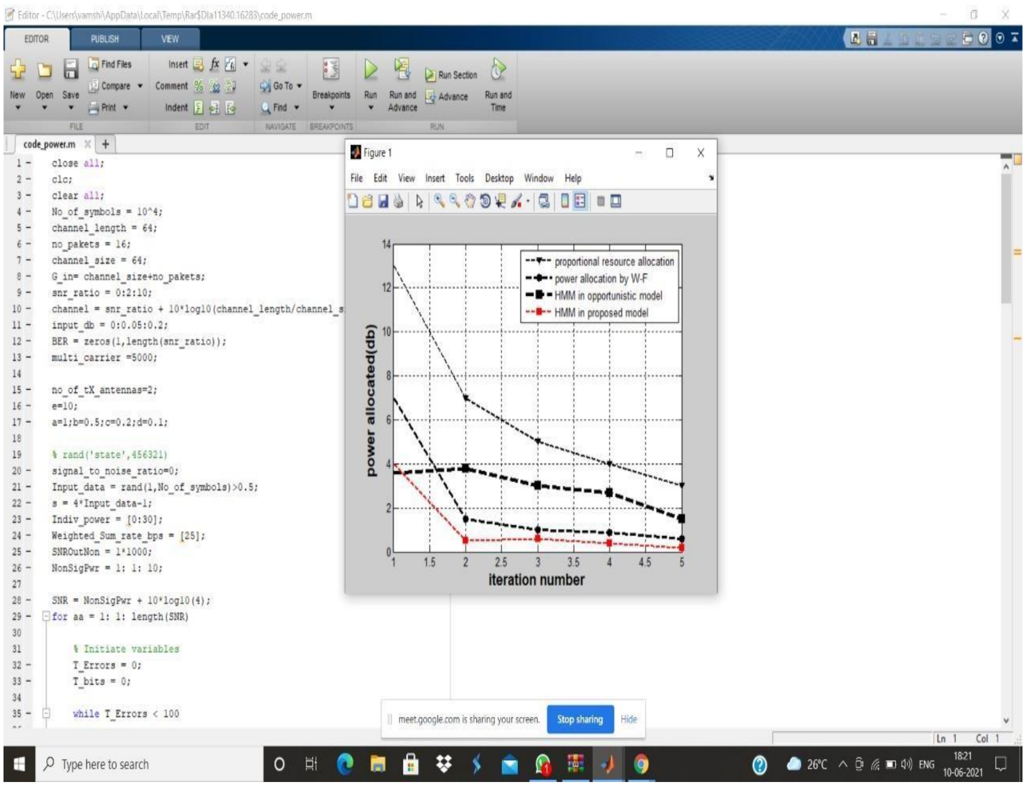


Fig 4.1. Power consumed for different algorithms v/s iteration number

It is clearly observed that it depict the power consumed by the different active pairs using the proportional resource algorithm, waterfilling algorithm, using HMM in opportunistic model and HMM in the proposed model. It is evident from the graph that maximum power is allocated to each pair in the proportional algorithm and least power is utilized in the proposed approach.

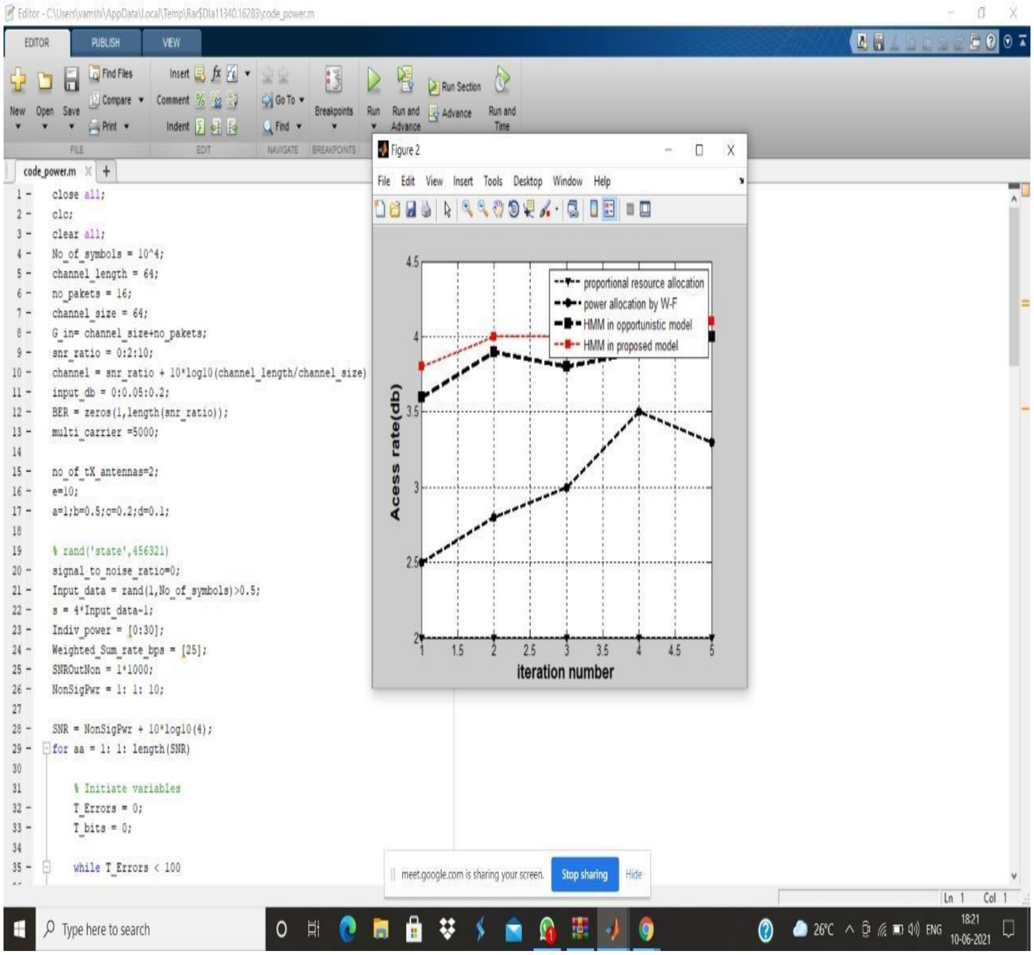


Fig 4.2. Access Rate Achieved in Different Iterations

It presents the access rate achieved for the active pairs with different resource allocation schemes. Highest access rate is achieved using HMM in the proposed model.

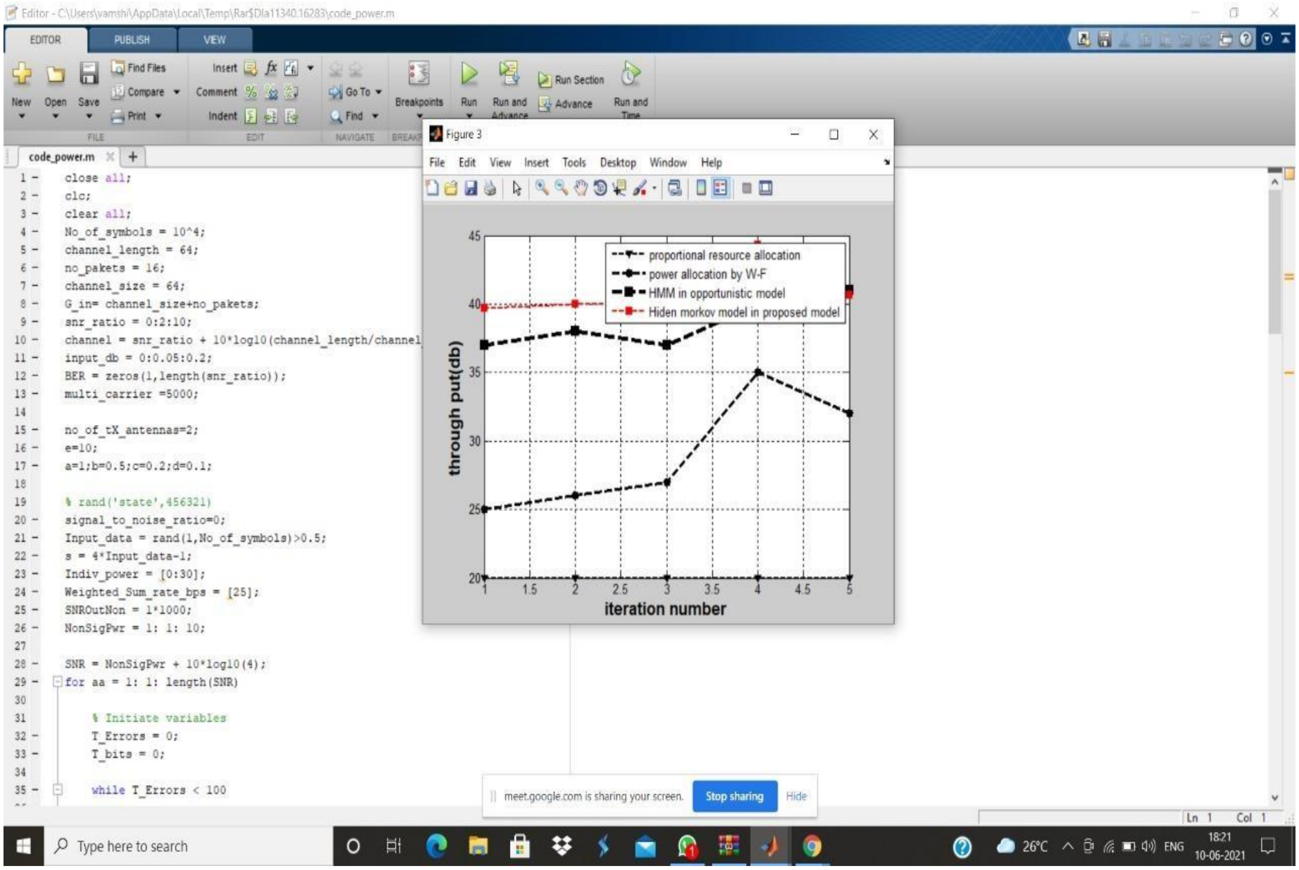


Fig 4.3 Throughput Achieved in Different Iterations

Similar trend has been found in the throughput analysis. Maximum throughput of the network if achieved when power is allocated using HMM in the proposed model.

Advantages

- High resolution and bi-directional large bandwidth shaping.
- Technology to gather all networks on one platform.
- More effective and efficient.
- Technology to facilitate subscriber supervision tools for the quick action.
- Most likely, will provide a huge broadcasting data (in Gigabit), which will support more than 60,000 connections.
- Easily manageable with the previous generations.

- Technological sound to support heterogeneous services (including private network).
- Possible to provide uniform, uninterrupted, and consistent connectivity across the world.

Disadvantages

- Technology is still under process and research on its viability is going on.
- The speed, this technology is claiming seems difficult to achieve (in future, it might be) because of the incompetent technological support in most parts of the world.
- Many of the old devices would not be competent to 5G, hence, all of them need to be replaced with new one — expensive deal.
- Developing infrastructure needs high cost.
- Security and privacy issue yet to be solved.

Applications

- It will make unified global standard for all.
- Network availability will be everywhere and will facilitate people to use their computer and such kind of mobile devices anywhere anytime.
- Because of the IPv6 technology, visiting care of mobile IP address will be assigned as per the connected network and geographical position.
- Its application will make world real Wi Fi zone.
- Its cognitive radio technology will facilitate different version of radio technologies to share the same spectrum efficiently.
- Its application will facilitate people to avail radio signal at higher altitude as well.

CHAPTER-5

CONCLUSION AND FUTURE ENHANCEMENT

5.0 Result

As we consider an urban deployment scenario with the urban macro cell radius of 0.5 km. The active users in the network form spectrum pair depending on a threshold distance criterion. The other users follow conventional cellular communication. Using MATLAB and the parameters in [14], the simulation results for the proposal have been carried out and analyzed. We assume 5 active pairs formed in different iterations. The results in Fig. 2 depict the power consumed by the different active pairs using the proportional resource algorithm, waterfilling algorithm, using HMM in opportunistic model and HMM in the proposed model. It is evident from the graph that maximum power is allocated to each pair in the proportional algorithm and least power is utilized in the proposed approach. Fig. 3 presents the access rate achieved for the active pairs with different resource allocation schemes. Highest access rate is achieved using HMM in the proposed model. Similar trend has been found in the throughput analysis. Maximum throughput of the network if achieved when power is allocated using HMM in the proposed model.

5.1 Conclusion

The proposal presented aims to achieve joint spectrum efficiency and energy efficiency for the NGN's utilizing spectrum sharing. The model validates its performance when compared to the conventional opportunistic spectrum sharing approach and other popular resource allocation schemes. Significant improvement in the QoS and throughput has been observed in the system when Hidden markov model is used in the proposed model. Spectrum Sharing is highly susceptible to jamming attack along with interference and coverage management issues, which remains an open research field.

5.2 Future Scope

The next generation technologies are expected to improve the spectrum utilization along with a reduction in the power levels. Spectrum sharing is gaining popularity with regard to increasing the spectrum and the energy efficiency of the network. Spectrum sharing allows the cooperative use or the simultaneous use of the radio frequency resource by a number of independent entities in a particular geographical area. Spectrum sharing can help in effective utilization of the white spaces or the underutilized parts of the spectrum. Also there are various power allocation strategies for optimum resource block allocation in a spectrum sharing scenario. The optimum resource block allocation strategies aims at joint optimization of the spectrum and energy efficiency of the system with increase in the quality of service for both the primary and the secondary receivers. The improved spectrum utilization can also aid in supporting the leading technology Internet of things (IoT). So we see that this area is gaining significant attention for improving the spectrum as well as energy efficiency of the network. The suggested proposal discusses both the aspects of increasing the spectrum as well as energy efficiency of the system.

5.3 Publications

Paper submitted to online mega international conference on “Smart modernistic in Electronics and Communication”(ICSME-21) with paper ID(ICSMEC21-0072) and got acceptance for the paper.

REFERENCES

[1] A. Gupta, R.K. Jha., “A Survey of 5G Network: Architecture and Emerging Technologies,” IEEE Access., vol. 3, July 2015, pp. 1206- 1231.

[2] H.Kour, et al, “A comprehensive survey on spectrum sharing: Architecture, energy efficiency and security issues”, Journal of Network and Computer Applications, Nov 2017, vol. 103, pp.29-57.

[3] X. J. Zhang, Y. Gong, and K. B. Letaief, “On the diversity gain in cooperative relaying channels with imperfect CSIT,” IEEE Trans. Commun., vol. 58, no. 4, pp. 1273–1279, Apr. 2010.

[4] C. Gong, G. Yue, and X. Wang, “A transmission protocol for a cognitive bidirectional shared relay system,” IEEE J. Sel. Topics Signal Process.,vol. 5, no. 1, pp. 160–170, Feb. 2011.

[5] Q. Li, S. H. Ting, A. Pandharipande, and Y. Han, “Cognitive spectrum sharing with two-way relaying systems,” IEEE Trans. Veh. Technol., vol. 60, no. 3, pp. 1233–1240, Mar. 2011.

[6] EARTH. Energy Aware Radio and Network Technologies Project. [Online]. Available: https://www.ict-earth.eu/default.html

[7] A. Abrol and R. K. Jha, "Power Optimization in 5G Networks: A Step Towards GrEEn Communication," in IEEE Access, vol. 4, no. , pp. 1355-1374, 2016.

[8] Xitao Gong et al., “Power Allocation and Performance Analysis in Spectrum Sharing Systems with Statistical CSI”, IEEE Transactions On Wireless Communications, Vol. 12, No. 4, April 2013, pp. 1819-1831.

[9] P Demestichas, A Georgakopoulos, D Karvounas, K Tsagkaris, V Stavroulaki, J Lu, C Xiong, J Yao, 5G on the horizon: key challenges for the radio-access network, IEEE Veh. Technol. Mag, 8, 47-53, 2013.

[10] J. Wang, D. Zhu, C. Zhao, J. C. F. Li, and M. Lei, “Resource sharing of underlaying device-to-device and uplink cellular communications,” IEEE Commun Lett., vol. 17, no. 6, June 2013.

[11] M. Pan et al., “Spectrum Harvesting and Sharing in Multi-Hop CRNs under Uncertain Spectrum Supply,”IEEE JSAC, vol. 30, no. 2, 2012, pp. 369–78.

[12] A. Ghasemi and E. S. Sousa, "Fundamental limits of spectrum-sharing in fading environments," IEEE Trans. Wireless Commun., vol. 6, no. 2, pp. 649–658, Feb. 2007.

[13] W. D. Lu, Y. Gong, S. H. Ting, X. L. Wu, and N. T. Zhang, "Cooperative OFDM relaying for opportunistic spectrum sharing: protocol design and resource allocation," IEEE Trans.

[14] K.Chandra, et al, "Unveiling capacity gains in ultradense networks", IEEE Vehicular Technology Magazine, April 2018.

APPENDIX

```
close all;

clc;

clear all;

No_of_symbols = 10^4;

channel_length = 64;

no_pakets = 16;

channel_size = 64;

G_in= channel_size+no_pakets;

snr_ratio = 0:2:10;

channel = snr_ratio + 10*log10(channel_length/channel_size) + 10*log10(channel_size/80);

input_db = 0:0.05:0.2;

BER = zeros(1,length(snr_ratio));

multi_carrier =5000;


no_of_tX_antennas=2;

e=10;

a=1;b=0.5;c=0.2;d=0.1;


% rand('state',456321)

signal_to_noise_ratio=0;

Input_data = rand(1,No_of_symbols)>0.5;
```

```
s = 4*Input_data-1;

Indiv_power = [0:30];

Weighted_Sum_rate_bps = [25];

SNROutNon = 1*1000;

NonSigPwr = 1: 1: 10;


SNR = NonSigPwr + 10*log10(4);

for aa = 1: 1: length(SNR)


    % Initiate variables

    T_Errors = 0;

    T_bits = 0;


    while T_Errors < 100


        uncoded_bits  = round(rand(1,SNROutNon));


        B = reshape(uncoded_bits,4,length(uncoded_bits)/4);

        B1 = B(1,:);

        B2 = B(2,:);

        B3 = B(3,:);

        B4 = B(4,:);
```

```
a = sqrt(1/10);

tx = a*(-2*(B3-0.5).*(3-2*B4)-j*2*(B1-0.5).*(3-2*B2));

ray = sqrt((1/2)*((randn(1,length(tx))).^2+(randn(1,length(tx))).^2));

rx = tx.*ray;

N0 = 1/10^(SNR(aa)/10);

rx = rx + sqrt(N0/2)*(randn(1,length(tx))+1i*randn(1,length(tx)));

rx = rx./ray;
    a = 1/sqrt(10);

B5 = imag(rx)<0;
B6 = (imag(rx)<2*a) & (imag(rx)>-2*a);
B7 = real(rx)<0;
B8 = (real(rx)<2*a) & (real(rx)>-2*a);
    temp = [B5;B6;B7;B8];
B_hat = reshape(temp,1,4*length(temp));
diff = uncoded_bits - B_hat ;
```

```
        T_Errors = T_Errors + sum(abs(diff));

        T_bits = T_bits + length(uncoded_bits);


    end

    BER(aa) = T_Errors / T_bits;

%     disp(sprintf('bit error probability = %f',BER(aa)));


end

db1=no_of_tX_antennas*no_of_tX_antennas;

db2=a+c;

db3=b+d;

db4=c+c;

db5=c;

x=[1 2 3 4 5];

y1=[13 7 5 4 3];

y2=[7 1.5 1 0.9 0.6];

y3=[3.6 3.8 3 2.7 1.5];

proposed=[db1 db2 db3 db4 db5];

%proposed=[4 1.2 0.6 0.4 0.2];

figure,

plot(x,y1,'k--v','linewidth',2,'markerfacecolor','k','markersize',5);

hold on;

plot(x,y2,'k--o','linewidth',2.5,'markerfacecolor','k','markersize',5);
```

```
hold on;

plot(x,y3,'k--s','linewidth',3,'markerfacecolor','k','markersize',5);

hold on;

plot(x,proposed,'r--s','linewidth',2,'markerfacecolor','r','markersize',5);

hold on;

xlabel('iteration number','fontsize',14,'fontweight','bold','color','k');

ylabel('power allocated(db)','fontsize',14,'fontweight','bold','color','k');

legend('proportional resource allocation','power allocation by W-F','HMM in opportunistic model','HMM in proposed model');

grid on;

db11=db1-c;

db12=db1;

db13=db1;

db14=db1+c+d;

db15=db1+d;

x=[1 2 3 4 5];

y1=[2 2 2 2 2];

y2=[2.5 2.8 3 3.5 3.3];

y3=[3.6 3.9 3.8 3.9 4];

proposed=[db11 db12 db13 db14 db15];

%proposed=[3.8 4 4 4.3 4.1];

figure,

plot(x,y1,'k--v','linewidth',2,'markerfacecolor','k','markersize',5);
```

```
hold on;
plot(x,y2,'k--o','linewidth',2.5,'markerfacecolor','k','markersize',5);
hold on;
plot(x,y3,'k--s','linewidth',3,'markerfacecolor','k','markersize',5);
hold on;
plot(x,proposed,'r--s','linewidth',2,'markerfacecolor','r','markersize',5);
hold on;
xlabel('iteration number','fontsize',14,'fontweight','bold','color','k');
ylabel('Acess rate(db)','fontsize',14,'fontweight','bold','color','k');
legend('proportional resource allocation','power allocation by W-F','HMM in opportunistic model','HMM in proposed model');
grid on;
db21=db1*e-a;
db22=db1*e;
db23=db22;
db24=db23+db1+a;
db25=db22+a+a;
x=[1 2 3 4 5];
y1=[20 20 20 20 20];
y2=[25 26 27 35 32];
y3=[37 38 37 40 41];
proposed=[db21 db22 db23 db24 db25];
%proposed=[39 40 40 45 42];
```

```
figure,

plot(x,y1,'k--v','linewidth',2,'markerfacecolor','k','markersize',5);

hold on;

plot(x,y2,'k--o','linewidth',2.5,'markerfacecolor','k','markersize',5);

hold on;

plot(x,y3,'k--s','linewidth',3,'markerfacecolor','k','markersize',5);

hold on;

plot(x,proposed,'r--s','linewidth',2,'markerfacecolor','r','markersize',5);

hold on;

xlabel('iteration number','fontsize',14,'fontweight','bold','color','k');

ylabel('through put(db)','fontsize',14,'fontweight','bold','color','k');

legend('proportional resource allocation','power allocation by W-F','HMM in opportunistic model','Hiden morkov model in proposed model');

grid on;
```

A

PROJECT REPORT

On

IOT TECHNOLOGY BASED TRAFFIC SIGNAL AND DENSITY CONTROL

Submitted by

1) Mr.A.Nihanth Reddy (17K81A0465)

2) Ms.K.Manasi (17K81A0484)

3) Mr.U.Jagadish (17K81A04B6)

in partial fulfillment for the award of the

degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

K.S. SAGAR

Assistant professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINNERING

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

St. MARTIN’S ENGINEERING COLLEGE

(An Autonomous Institute)

NBA, NAAC A+ Accredited

(Approved by A.I.C.T.E, New Delhi, Affiliated to JNTU Hyderabad) Dhulapally, Secunderabad, Telangana (India)-500100

BONAFIDE CERTIFICATE

This is to certify that the project entitled “IOT TECHNOLOGY BASED TRAFFIC SIGNAL AND DENSITY CONTROL”, is being submitted by **Mr. A.Nihanth Reddy (17K81A0465), Ms. K.Manasi (17K81A0484), Mr. U.Jagadish (17K81A04B6)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** IN ELECTRONICS AND COMMUNICATION ENGINEERING is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

	Head of the Department
K.S.Sagar	Dr. B. Hari Krishna
Department of ECE	Department of ECE

Internal Examiner External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **ARVA NIHANTH REDDY** WITH ROLL NO.**17K81A0465, KARANAM MANASI** WITH ROLL NO.**17K81A0484**, **U. JAGADISH** WITH ROLL NO.**17K81A04B6**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**IOT TECHNOLOGY BASED TRAFFIC SIGNAL AND DENSITY CONTROL**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. ★Kompally-500014★

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

DECLARATION

We, the student of **Bachelor of Technology** in Department of 'Electronics and Communication Engineering', session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled "IOT TECHNOLOGY BASED TRAFFIC SIGNAL AND DENSITY CONTROL" is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university foraward of any degree.

A.Nihanth reddy	(17K81A0465)
K.Manasi	(17K81A0484)
U.Jagadish	(17K81A04B6)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr. B. Hari Krishna**, **Head &Professor**, in the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept. project coordinator Dr.A.Anand Kumar**, professor in ECE for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide, **Mr. K.S.Sagar, Assistant professor,** Department of ECE and our class project coordinator **Mr.K.Karthik, Assistant professor,** Department of ECE for encouragement in carrying out this project work.

Finally, We are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar** , **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. A.Nihanth Reddy
2. K. Manasi
3. U.Jagadish

ABSTRACT

In the world of Innovative and automotive world everything is getting computerized. Each data is in effect effortlessly available. Yet, the movement signals checking is as yet done physically. The activity signals are observed physically from the control room by the executives or a predictable time is settled for signals evolving. Rather than this a computerized controller-based activity checking framework will be useful for controlling the movement. This plan of movement foundation will be useful in decreasing the activity clog issue in urban communities. This paper depicts a framework where IR sensors are incorporated with an Arduino to work the paths which measure the movement thickness. This incorporated arrangement of movement is Internet of Things (IoT) based which likewise empowers to clear the activity for emergency vehicle by giving a catch in rescue vehicle so the activity gets cleared on that side. It additionally empowers the vehicles tally that move over the sensors. Subsequently, movement controlling gets upgraded effectively, which in the end prompts huge change in rush hour gridlock framework.

TABLE OF CONTENT

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Overview of the Project

In the world of Innovative and automotive world everything isgetting computerized. Each data is in effect effortlessly available. Yet, the movement signals checking is as yet done physically. The activity signals are observed physically from the control room by the executives or a predictable time is settled for signals evolving. Rather than this a computerized controller-based activity checking framework will be useful for controlling the movement. This plan of movement foundation will be useful in decreasing the activity clog issue in urban communities. This paper depicts a framework where IR sensors are incorporated with an Arduino to work for the paths which measure the movement thickness. This incorporated arrangement of movement is Internet of Things (IoT) based which likewise empowers to clear the activity for emergency vehicle by giving a catch in rescue vehicle so the activity gets cleared on that side [1]. It additionally empowers the vehicles tally that move over the sensors. Subsequently, movement controlling gets upgraded effectively, which in the end prompts huge change in rush hour gridlock framework. Keywords Automated traffic monitoring and controlling, Arduino, IR sensors, Traffic congestion and IoT. In this quick moving world everything is mechanized. Beginning from home apparatuses to following of vehicles every last framework is implanted. Installed frameworks are those which are intended for a specific undertaking rather that for broadly useful arrangement of taking care of numerous assignments [2]. Installed frameworks are intended to do some particular errand, as opposed to be a broadly useful PC for different undertakings. Some likewise have continuous execution limitations that must be met, for reason, for example, wellbeing and ease of use; others may have low or no execution prerequisites, enabling the framework equipment to be streamlined to lessen costs. An installed framework isn't generally a different piece - all the time it is physically worked in to the gadget it is controlling. The product composed for implanted frameworks is regularly called firmware and is put away in read-just memory or glimmer convector chips as opposed to a circle drive. It frequently keeps running with constrained PC equipment assets: little or no console, screen, and little memory. Remote correspondence has turned into an essential element for business items and a well known research subject inside the most recent ten years. There are currently more cell phone memberships than wired-line memberships. With new innovations and gadgets come new business exercises, and the requirement for representatives in these mechanical territories. Architects who know about installed frameworks and remote interchanges will be popular. Shockingly, there are couple of cute situations accessible for advancement and classroom utilize, so understudies frequently don't find out about these innovations amid hands-on lab works out.

Objective of the Project

Presently a-days, every one of the works and things are getting to be mechanized and computerized. It began from looking through the notable places and came to home conveyance of anything. Be that as it may, in this quick age the activity framework is still carried on physically. Along these lines, to computerize it we have built up an undertaking. The framework is an IoT based framework which flags the movement lights by thinking about the vehicles thickness. To do this we utilize sensors. The sensors will quantify the activity thickness and isolate it into low, medium and high classifications. In view of that outcome the signs get changed. This task points in planning an IoT based framework in which movement signals are checked and controlled consequently by utilizing sensors. The framework utilizes an Arduino based circuit which controls the signs in view of the thickness of vehicles and transmit the information to the server. The Arduino is a group of microcontroller sheets to improve electronic plan, prototyping and testing for specialists, programmers, specialists, yet additionally numerous experts. Arduinos (we utilize the standard Arduino Uno) are worked around an AtMega microcontroller basically an entire PC with CPU, RAM, Flash memory, and information/yield sticks, all on a solitary chip. The Arduino associates with your PC by means of USB, where you program it in a straightforward dialect (C/C++, like Java) from inside the free Arduino IDE by transferring your arranged code to the board. Once modified, the Arduino can keep running with the USB interface back to your PC, or remain solitary without it — no console or screen required, simply control. The framework measures the thickness of movement utilizing sensors that are set in an interim of separation. This framework likewise gives an alternative to avoid the flag to green at whatever point there a rescue vehicle is going through that course by keeping different sides in Red by giving a catch. Thus the movement signals are observed consequently with the assistance of sensors. In this, we are utilizing IR sensors. IR sensors work by utilizing a particular light sensor to recognize a select light wavelength in the InfraRed (IR) range. By utilizing a LED which creates light at an indistinguishable wavelength from what the sensor is searching. At the point when a question is near the sensor, the light from the LED bobs off the protest and into the light sensor. By utilizing the IR sensors, the thickness is estimated and the signs gets observed and controlled. The information got from the sensors are refreshed at time to time to the server utilizing IoT.

Material Requirement

Hardware Requirements

- Arduino UNO
- IR Sensor
- Wi-Fi Module
- Traffic Signal Module
- LCD
- Power Supply Unit

Software Requirements

- Arduino IDE
- Proteus Software

Procurement of Equipment's

We brought all the required hardware components in an electronic manufacturing shop and before assembling and interfacing hardware components to Arduino UNO micro-controller board, we connected individual sensors to Arduino in Proteus Software and understood working of every individual sensor in Proteus Software.

Now, we assembled all sensors and interfaced them to Arduino UNO micro-controller board using connecting wires through bread board and we wrote program for the working of all the hardware sensors in Arduino IDE, we understood the working of all sensors in a practical way.

We monitored traffic signals and checked the display on LCD and also updated on Server. Here we can also monitor the traffic when a VIP vehicle is approaching with the help of TELNET app which works based on internet of things.

Introduction to Embedded System

An embedded system is a combination of computer hardware and software, either fixed in capability or programmable, designed for a specific function or functions within a larger system. Industrial machines, agricultural and process industry devices, automobiles, medical equipment, cameras, household appliances, airplanes, vending machines and toys, as well as mobile devices, are possible locations for an embedded system. An embedded system is a dedicated computer system designed for one or two specific functions. This system is embedded as a part of a complete device system that includes, such as electrical and mechanical components[1]. The embedded system is unlike the generalpurpose computer, which is engineered to manage a wide range of processing tasks. Because an embedded system is engineered to perform certain tasks only, design engineers may optimize size, cost, power consumption, reliability and performance. Embedded systems are typically produced on broad scales and share across a variety of environments and applications. An embedded system is one that has computer-hardware with software embedded in it as one of its most important components. It is a dedicated computer-based system for an application or product. An embedded system has three main components.

- It has hardware
- It has main application software.
- It has a real time operating system (RTOS)

An embedded system has software designed to keep in view three constraints:

(i) Available system memory.

(ii) Available processor speed.

(iii) The need to limit power dissipation when running the system .

An2 embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer

may be used to control the operation of a large complex processing plant, and its presence will be obvious. Some of these computers are however very simple systems as compared with a personal computer[2]. The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes[2]. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes. The applications software on such processors is sometimes referred to as firmware. The simplest devices consist of a single microprocessor (often called a chip)
which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine[3]. As the embedded system is the combination of both software and hardware.

History of Embedded Systems

Embedded systems date back to the 1960s. Charles Stark Draper developed an integrated circuit (IC) in 1961 to reduce the size and weight of the Apollo Guidance Computer, the digital system installed on the Apollo Command Module and Lunar Module. The first computer to use ICs, it helped astronauts collect realtime flight data. In 1965, Auto now a part of Boeing, developedthe D-17B, the computer used in the Minuteman I missile guidance system.

Block Diagram of Embedded System

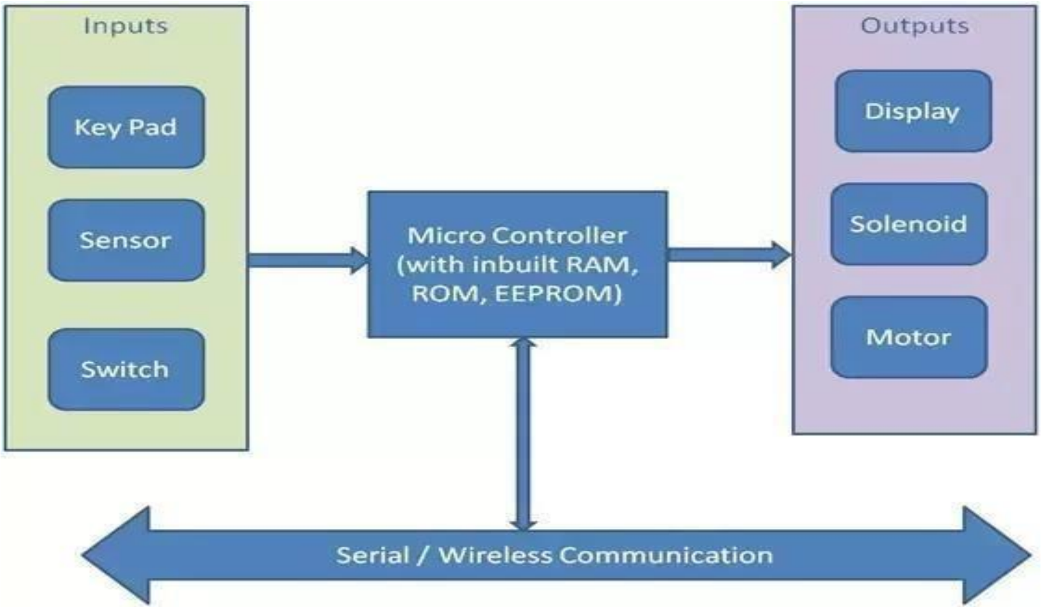


Fig.1.5.1: Block diagram of embedded system

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

Embedded System Hardware

Embedded system hardware can be microprocessor- or microcontroller-based. In either case, an integrated circuit is at the heart of the product that is generally designed to carry out computation for real-time operations[4]. Microprocessors are visually indistinguishable from microcontrollers, but while the microprocessor only implements a central processing unit (CPU) and, thus, requires the addition of other components such as memory chips, microcontrollers are designed as self contained systems.

Embedded System Software

A typical industrial microcontroller is unsophisticated compared to the typical enterprise desktop computer and generally depends on a simpler, less-memory intensive program environment. The simplest devices run on bare metal and are programmed directly using the chip CPU's machine code language. Often, embedded systems use operating systems or language platforms tailored to embedded use, particularly where real-time operating environments must be served. At higher levels of chip capability, such as those found in SoCs designers have increasingly decided the systems are generally fast enough and the tasks tolerant of slight variations in reaction time that near-real-time approaches are suitable.

Examples of Embedded System

- ATM
- Digital Cameras
- Microwave ovens
- Factory controllers
- Washing machine
- Calculator
- TV remote
- Traffic lights
- Digital watches
- Mp3 player

Applications of Embedded Systems

- Manufacturing and process control
- Construction industry
- Transport
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical Systems
- Robotics
- Instrumentation
- Military Application

Microcontroller

A microcontroller is a small computer on a single metal oxide semiconductor (MOS) integrated circuit (IC) chip. A microcontroller contains one or more CPUs along with memory and programmable input/output peripherals. Program memory in the form of ferroelectric RAM, NOR flash or OTP ROM is also often included on chip, as well as a small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications consisting of various discrete chips[7]. In modern terminology, a microcontroller is similar to, but less sophisticated than, a system on a chip (SoC). SoC may include a microcontroller as one of its components, but usually integrates it with advanced peripherals like graphics processing unit (GPU), Wi-Fi module, or one or more coprocessors.

Fig.1.6.1: Microcontroller

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems. In the context of the internet of things, microcontrollers are an economical and popular means of data collection, sensing and actuating the physical world as edge devices.

Some microcontrollers may use four-bit words and operate at frequencies as low as 4 kHz for low power consumption (single-digit milliwatts or microwatts). They generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (CPU clock and most peripherals off) may be just nanowatts, making many of them well suited for long lasting battery applications[8]. Other microcontrollers may serve performance-critical roles, where they may need to act more like a digital signal processor (DSP), with higher clock speeds and power consumption.

Microprocessor	Microcontroller
CPU is stand alone, RAM,ROM, I/O & timer are separate.	CPU, RAM,ROM, I/O & timer all are on single chip.
Designer can decide amount of RAM,ROM, & I/O ports.	Fixed amount of on-chip RAM,ROM, & I/O ports.
High processing power	Low processing power
High power consumption	Low power consumption
Typically 32/64 bit	8/16 bit
General purpose	Single purpose(control oriented)
Less reliable	Highly reliable
Eg.- 8086,8085	8051

Table 1.6.1: Difference between microprocessor and microcontroller

History

The origins of both the microprocessor and the microcontroller can be traced back to the invention of the MOSFET (metal-oxide-semiconductor field-effect transistor), also known as the MOS transistor. It was invented by Mohamed M. Atalla and Dawon Kahng at Bell Labs in 1959, and first demonstrated in 1960.

The same year, Atalla proposed the concept of the MOS integrated circuit, which was an integrated circuit chip fabricated from MOSFETs. By 1964, MOS chips had reached higher transistor density and lower manufacturing costs than bipolar chips[2]. MOS chips further increased in complexity at a rate predicted by Moore's law, leading to large-scale integration (LSI) with hundreds of transistors on a single MOS chip by the late 1960s. The application of MOS LSI chips to computing was the basis for the first microprocessors, as engineers began recognizing that a complete computer processor could be contained on a single MOS LSI chip.

Fig.1.6.2: 8051 Microcontroller Board

Most microcontrollers at this time had concurrent variants. One had EPROM program memory, with a transparent quartz window in the lid of the package to allow it to be erased by exposure to ultraviolet light. These erasable chips were often used for prototyping. The other variant was either a mask programmed ROM or a PROM variant which was only programmable once. For the latter, sometimes the designation OTP was used, standing for "one-time programmable". In an OTP microcontroller, the PROM was usually of identical type as the EPROM, but the chip packagehad no quartz window; because there was no way to expose the EPROM to ultraviolet light, it could not be erased. Because the erasable versions required ceramic packages with quartz windows, they were significantly more expensive than the OTP versions, which could be made in lower-cost opaque plastic packages. For the erasable variants, quartz was required, instead of less expensive glass, for its transparency to ultraviolet light— to which glass is largely opaque—but the main cost differentiator was the ceramic package itself.

In 1993, the introduction of EEPROM memory allowed microcontrollers (beginning with the Microchip PIC16C84) to be electrically erased quickly without an expensive package as required for EPROM, allowing both rapid prototyping, and in-system programming. (EEPROM technology had been available prior to this time, but the earlier EEPROM was more expensive and less durable, making it unsuitable for low-cost mass-produced microcontrollers.) The same year, Atmel introduced the first microcontroller using Flash memory, a special type of EEPROM. Other companies rapidly followed suit, with both memory types.

IoT Technology

The Internet of things (IoT) describes the network of physical objects—a.k.a. "things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet.[20][21]

Things have evolved due to the convergence of multiple technologies, realtime analytics, machine learning, ubiquitous computing, commodity sensors, and embedded systems[20]. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", including devices and appliances (such as lighting fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers. The IoT can also be used in healthcare systems.

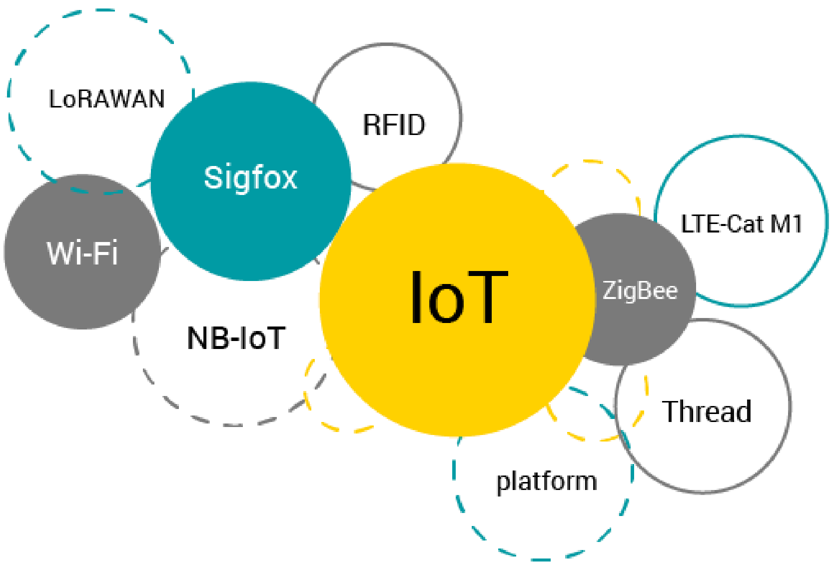


Fig 1.7.1: IoT Technology

There are a number of serious concerns about dangers in the growth of the IoT, especially in the areas of privacy and security, and consequently industry and governmental moves to address these concerns have begun including the development of international standards.

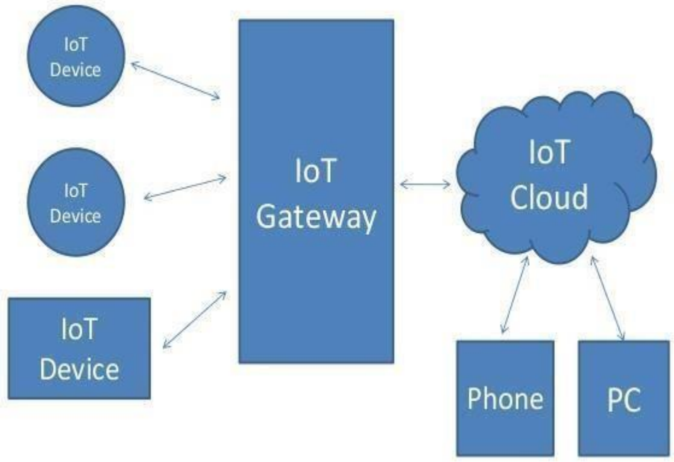


Fig 1.7.2: IoT Block diagram

The IoT technology stack and its Connectivity Solutions

It can prove a hard task if you'd like to find your way through the IoT technological maze given the diversity and sheer numerousness of the technology solutions that surround it. However, for matters of simplicity, we could break down the IoT technology stack into four basic technology layers involved in making the Internet of Things work. These are the following:

Fig 1.7.3: IoT technology stack

1. Device Hardware

Devices are objects which actually constitute the 'things' within the Internet of Things. Acting as an interface between the real and the digital worlds, they may take different sizes, shapes and levels of technological complexity depending on the task they are required to perform within the specific IoT deployment. Whether pinhead sized microphones or heavy construction machines, practically every material object (even the animate ones, like animals or humans) can be turned into a connected device by the addition of necessary instrumentation (by adding sensors or actuators along with the appropriate software) to measure and collect the necessary data. Obviously, sensors, actuators or other telemetry gear can also constitute standalone smart devices by themselves. The only limitation to be encountered here is the actual IoT use case and its hardware requirements (size, ease of deployment and management, reliability, useful lifetime, cost-effectiveness)

2. Device Software

This is what actually makes the connected devices 'smart'. Software is responsible for implementing the communication with the Cloud, collecting data, integrating devices as well as performing real-time data analysis within the IoT network. What is more, it is device software that also caters for application-level capabilities for users to visualize data and interact with the IoT system.

3. Communications

Having the device hardware and software in place, there must be another layer which will provide the smart objects with ways and means of exchanging information with the rest of the IoT world. While it is true that communications mechanisms are strongly tied to device hardware and software, it is vital to consider them as a separate layer. Communication layer includes both physical connectivity solutions (cellular, satellite, LAN) and specific protocols used in varying IoT environments (ZigBee, Thread, Z-Wave, MQTT, LwM2M). Choosing the relevant communications solution is one of the vital parts in constructing every IoT technology stack. The technology chosen will determine not only the ways in which data is sent to/received from the Cloud, but also how the devices are managed and how they communicate with third party devices. For the purpose of the present article, we will go into the details of some of the present-day communications solutions later in the text.

4. Platform

As mentioned earlier, thanks to the 'smart' hardware and the software installed the device is able to 'sense' what is going on around it and communicate that to the user via a specific communications channel. An IoT platform is the place where all of these data is gathered, managed, processed, analysed and presented in a user-friendly way. Thus, what makes such a solution especially valuable is not merely its data collection and IoT device management capabilities, but rather its ability to analyse and find useful insights from the portions of data provided by the devices via the communications layer. Again, there is quite a number of IoT platforms on the market, with choice depending on the requirements of the specific IoT project and such factors as architecture and IoT technology stack, reliability, customization properties, protocols used, hardware agnosticism, security and cost-effectiveness. It is also worth mentioning that platforms can be either installed on-premise or cloud-based. Coiote IoT Device Management platform is a good example of such a platform as it can be deployed on-site as well as in the cloud. The same applies to another IoT platform by AVSystem — Coiote IoT Data Orchestration.

Connectivity solutions

As many as there are possible real-life applications of the IoT technologies, there is no shortage of connectivity solutions behind them. Depending on the specifications of a given IoT use case, each communications option may offer different service enablement scenarios while having tradeoffs between power consumption, range and bandwidth. With this multiplicity and diversity of communication standards and protocols in mind, one may raise a question about the actual need for developing new solutions while there are some well-proven Internet protocols that have been in use already for decades. The reason for this is that existing Internet protocols, such as Transmission Control Protocol / Internet Protocol (TCP/IP), are often not effective enough and too power-consuming to be able to work efficiently within the emerging IoT technology applications. This section will present a short overview of the major alternative Internet protocols specially dedicated for use by IoT systems.

The overview concerns the most popular IoT radio technologies broken down by radio-frequency range achieved by each of the solutions: short range IoT radio solutions, medium range solutions, and long-range Wide Area Networks solutions.

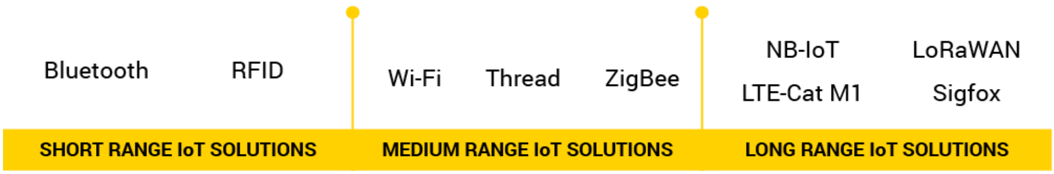


Fig 1.7.4: Connectivity Solutions

Short range IoT radio solutions

1. Bluetooth

As a well-established short-range connectivity technology, Bluetooth is considered to be the key solution particularly for the future of the wearable electronics market such as wireless headphones or geolocation sensors, especially given its widespread integration with smartphones. Designed with cost-effectiveness and reduced power consumption in mind, the Bluetooth Low-Energy (BLE) protocol requires very little power from the device. Yet, this comes with a compromise: when transferring frequently higher amounts of data, BLE may not be the most effective solution.

2. RFID

Being among the first IoT applications ever implemented, Radio-frequency identification (RFID) offers positioning solutions for IoT applications, especially in supply chain management and logistics, which require the ability of determining the object position inside buildings. The future of RFID technology clearly goes far beyond the simple localisation services, with possible applications ranging from tracking hospital patients to improving efficiency in healthcare to providing real-time merchandise location data to minimize out-of-stock situations for retail stores.

Medium Range Solutions

1. Wi-Fi

Developed based on IEEE 802.11, it remains the most widespread and generally known wireless communications protocol. Its broad usage across the IoT world is mainly limited by higher-than-average power consumption resulting from the need of retaining high signal strength and fast data transfer for better connectivity and reliability. As a key technology in the development of IoT, WiFi provides a wide-ranging ground to staggering number of IoT solutions, yet it also needs to be managed and used in terms of marketing to yield profits to service providers and users alike. A fine example of a WiFi management platform that offers a value-added service empowering public WiFi access points is Linkify. As one of AVSystem's cutting-edge solutions, Linkify allows for practically limitless guest WiFi customization and marketing options.

2. Zigbee

This popular wireless mesh networking standard finds its most frequent applications in traffic management systems, household electronics, and machine industry. Built on top of the IEEE 802.15.4 standard, Zigbee supports low data exchange rates, low power operation, security, and reliability.

Long Range Wide Area Networks (WAN) solutions:

1. NB-IOT

A product of existing 3GPP technologies, Narrowband IoT is a brand-new radio technology standard that ensures extremely low power consumption (10 years of battery power operation) and provides connectivity with signal strength approx. 23 dB lower than in the case of 2G. What is more, it uses existing network infrastructure, which ensures not only global coverage in LTE networks, but also guaranteed signal quality. In many cases, this fact allows for implementing NB-IoT instead of solutions that required the construction of local networks, such as LoRa or Sigfox.

2. LoRaWAN

LoRaWAN is a low-power Long Range Wide-Area Networking protocol optimized for low-power consumption and supporting large networks with millions of devices. Aiming at wide-area network (WAN) applications, LoRaWAN is designed to furnish low-power WANs with features required to support low-cost, mobile and secure bi-directional communication within IoT, M2M, smart city, and industrialapplications.

3. Sigfox

The concept behind Sigfox is to provide an effective connectivity solution for low-power M2M applications requiring low levels of data transfer for which the WiFi range is too short, and cellular range is too expensive and too power-hungry. Sigfox employs UNB, a technology that enables it to handle low data-transfer speeds of 10 to 1,000 bits per second. Consuming up to 100 times less energy compared to cellular communication solutions, it delivers a typical stand-by time of 20 years for a 2.5Ah battery. Offering a robust, energy-efficient and scalable network able to support communication between thousands of thousands of battery-operated devices across areas of several square kilometres, Sigfox proves suitable for various M2M applications, including smart street lighting, intelligent meters, patient monitors, security devices, and environmental sensors

4. LTE-Cat M1

LTE-Cat M1 is a low-power wide-area (LPWA) connectivity standard that connects IoT and M2M devices with medium data rate requirements. It supports longer battery lifecycles and offers enhanced in-building range as compared to cellular technologies such as 2G, 3G, or LTE-Cat 1. Being compatible with the existing LTE network, CAT M1 doesn't require the carriers to build new infrastructure to implement it. As compared to NB-IoT, LTE Cat M1 proves to be perfect for mobile use cases, as its handling of hand-over between cell sites is significantly better and is very similar to high speed LTE.

IoT vs Embedded Systems

	IoT	Embedded System
Definition	Things that contain computer, software and networking capabilities	Things that contain computers and software.
Example	A talking doll that accepts voice commands and regularly learns new words	A talking doll that never learns but that isn't a privacy concern because it transmits no data.

Table 1.7.1: IoT vs Embedded system

IoT Technology in Transportation

The IoT can assist in the integration of communications, control, and information processing across various transportation systems. Application of the IoT extends to all aspects of transportation systems (i.e. the vehicle, the infrastructure, and the driver or user). Dynamic interaction between these components of a transport system enables inter- and intra-vehicular communication, smart traffic control, smart parking, electronic toll collection systems, logistics and fleet management, vehicle control, safety, and road assistance.

Motivation of the Project

In the present day's scenario, with growing technologies and adverse development in the metropolitan cities, traffic administration has become one of the most important fields to be dealt with. The main role of a traffic administration is to constantly improve the traffic control system and effectively regulate the same. With the number of vehicle users constantly increasing, the facility provided by the current system is limited and inefficient concerning the energy and time consumed. The objective of the project is that we focus on the traffic density rather than just giving control tothe signals on a fixed time basics. It is based on vehicle detection by IR sensors, analysis, and communicates the scenarios to the Arduino UNO and it gives the control depending on the control time. On a whole, this system senses the presence of the vehicle in that given lane and suitably gives control to the signals which help in controlling traffic.

CHAPTER 2

LITERATURE SURVEY

Literature review on research area

As per the exploration paper by G.Kavya and B.Saranya , Density Based Traffic Signal System Using Microcontroller, the enhancement of activity light controller in a city utilizing sensors and microcontroller. By utilizing this framework setup the potential outcomes of congested roads are lessened to some degree which are caused by activity lights, and effectively gets the outcomes. Number of passing vehicles in the settled availability out and about chooses the thickness scope of movement .The recorded information can be downloaded to the PC through correspondenceamongst microcontroller and the PC. Pramod Sharma et al proposed a density based intelligent traffic control system using IR sensors using the ATMEL 89C51 Microcontroller were IR sensors are used for the vehicle detection and the control of the signals are taken over by the microcontroller [5]. As per Shwetha N. Pable, Density, speed, and stream are the three basic parameters for street movement examination. Superior street movement administration and control require ongoing estimation of room, mean speed and thickness as contribution for substantial spatial and fleeting scope of the roadway organize. As Saiba P A et al proposed a density based traffic control system using PIC microcontroller were three IR sensors are employed in each road for the detection of the presence of the automobile and the density is identified in different levels such as low, medium and high. Based on the density recorded, the duration of the green light is decided by the microcontroller [4]. As indicated by K.Vidhya and A.Bazila Banu , utilize the Density estimation by utilizing open instrument as programming for picture preparing by simply showing the different change of picture in the screen lastly encompassing the container on the vehicle in the given picture, the quantity of vehicles are computed. They can figure the thickness of the vehicle by utilizing mat lab device by looking at the four side of the picture which is given as an information. D. D. Pukale et al proposed a density based traffic control system using video processing. With the help of Arduino, the implementation is carried out by analysing the video recordings and extracting the frames using which the vehicle densities is found. Corresponding the density present, the time duration for the specific control of the signal is decided [6].

Review on Literature Survey

The purpose of this project is to control the traffic based on density control and also to reduce the manpower required for controlling. In today's world traffic regulation is one of the biggest problems that need to be solved. Our project helps in controlling the traffic by checking the density present at roads and accordingly changing the traffic signals for smooth flow of vehicles.

Existing Models

Under present scenario, traffic control is achieved by the use of a system of hand signs by traffic police personnel, traffic signals, and markings. Under current circumstances, traffic lights are set on in the different directions with fixed time delay, following a particular cycle while switching from one signal to other creating unwanted and wasteful congestion on one lane while the other lanes remain vacant.

Drawbacks in the existing models:

- The presence of a fixed time period of signal operation irrespective of the lack of comparison of the vehicle densities in the different parts of the road at a junction.
- Even if the road is empty and lacks the presence of vehicles, still receives a fixed time period of green signal which is of no use.
- This can lead to a large levels of traffic congestion since there is no uniform flow of vehicles from all the roads at the junction.
- Human irritability increases and this can be one of the main reasons forsome of the traffic violations such as jumping signals, over speeding, etc. [4]

Proposed model

The main aim of the proposed model is to constantly monitor the vehicle density present in all parts of the road at the junction. The basic flow of operation is as follows: collection of vehicle density data from the roads; next is to send the same data to the device which compares the same and arrives at a particular characteristic output pattern; then the execution of the output pattern which is reflected in the signal pattern. In this model, the IR sensors are used to detect the presence of any vehicle in that part of the road. When detected it sends a triggered output to Arduino UNO which is the heart of the project. Then Arduino analyses the number of such triggered outputs from the set of sensors placed in the different roads at the junction and correspondingly triggers the different LED lights in the signals in order to felicitate the vehicle movement

Advantages in the proposed model

- Traffic can be cleared without any difficulty
- Time can be shared evenly for all intersections
- Effective time management

Conclusion on review

To Overcome all the problems in the existing system in monitoring traffic signals based on density control, in the proposed system we introduced IoT based traffic signal and density control system using Arduino.

Proposed system, consists of mainly two units- in the first unit, we connect the sensors to monitor the traffic and get the information regarding density at roads and Arduino UNO microcontroller board will analyse all those information and change the traffic signals based on the information obtained.

In the second unit, after analysing all the information by the Arduino UNO, It will display the information regarding traffic signals on LCD and also updates all information in the Sever

Here we can monitor the traffic for VIP vehicles by controlling the traffic signals from telnet app that works with the help of IoT technology.

CHAPTER 3

PROJECT DESCRIPTION

Block Diagram

In the present day's scenario, with growing technologies and adverse developmentin the metropolitan cities, traffic administration has become one of the most important fields to be dealt with. The main role of a traffic administration is to constantly improve the traffic control system and effectively regulate the same. With the number of vehicle users constantly increasing, the facility provided by the current system is limited and inefficient concerning the energy and time consumed. The objective of the project is that we focus on the traffic density rather than just giving control to the signals on a fixed time basics. It is based on vehicle detection by IR sensors, where it analysis, and communicates the scenarios to the Arduino UNO and it gives the control depending on the control time. On a whole, this system senses the presence of the vehicle in that given lane and suitably gives control to the signals which help in controlling traffic.

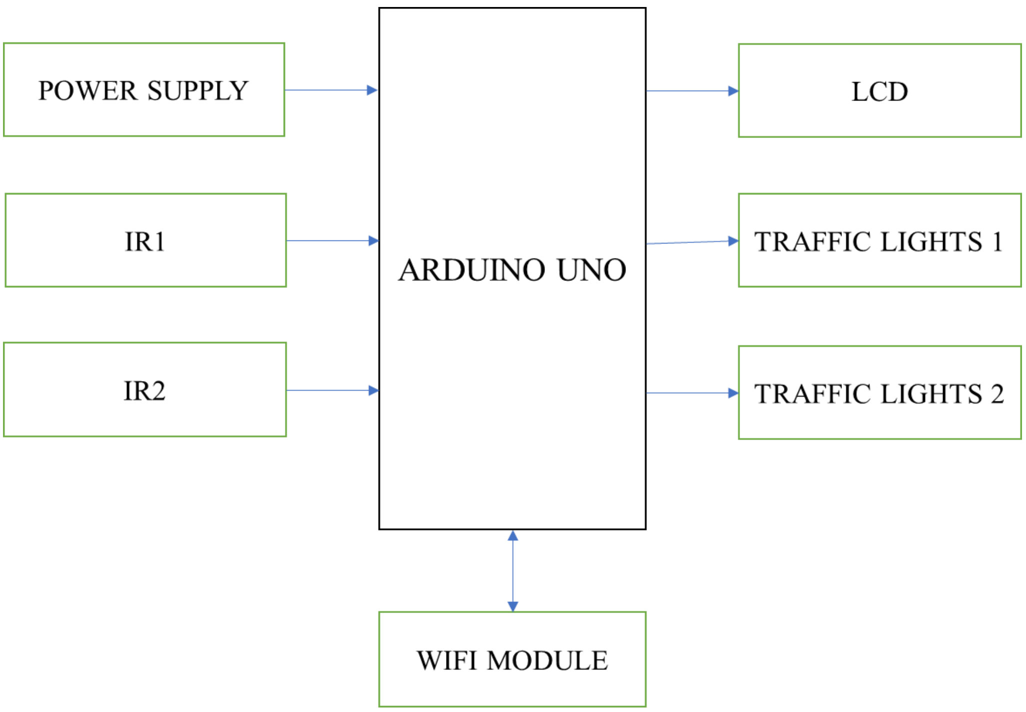


Fig 3.1.1: Block diagram

The blocks of the circuit are:

1. Arduino UNO

2. IR Sensor

3. Traffic Signal module

4. Power supply

5. LCD

6. Wi-Fi Module

1. Arduino UNO

Arduino is open source platform based on easy to use hardware and software. Arduino Uno is a microcontroller board, which has ATmega328P microcontroller in it. It has 14 digital i/o pins and 6 analog input pin, a 16 MHz quartz crystal, a USB connection, a power jack and a reset button. Simply connect to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. Embedded C program is used for coding.

2. IR Sensor

IR sensor is an electronic device, that emits the light in order to sense some object of the surroundings. IR Infrared Obstacle Avoidance Sensor Module has a pair of infrared transmitting and receiving tubes. When the transmitted light waves are reflected back, the reflected IR waves will be received by the receiver tube. Here the IR Sensor is used to detect the density present at the roads and send the information to the Arduino UNO.

3. Traffic Lights module

The LED Traffic Lights Signal Module / Digital Signal Output Traffic Light Module is a mini-traffic light display module with high brightness and very suitable for the production of a traffic light system model. Here this module is helping us to control the flow of traffic in different lanes by changing the led lights depending upon the density present at the roads.

4. Wi-Fi Module

The ESP8266 Wi-Fi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much Wi-Fi-ability.

5. LCD

The LCD here is used to display us the information regarding the traffic density present at roads. When the traffic density is detected by IR sensor at road 1, the LCD is going to display us ‘Density at Road 1’ and similarly when density is detected at road 2 then the LCD is going to display us that ‘Density at Road 2’. It is also going to display us regarding the manual control for VIP vehicles like ‘VIP vehicle on road 1’ or ‘VIP vehicle on road 2’.

6. Power Supply

Every electrical and electronic device that we use in our day-to-day life will require a power supply. In general, we use an AC supply of 230V 50Hz, but this power has to be changed into the required form with required values or voltage range for providing power supply to different types of devices.

Schematic Diagram

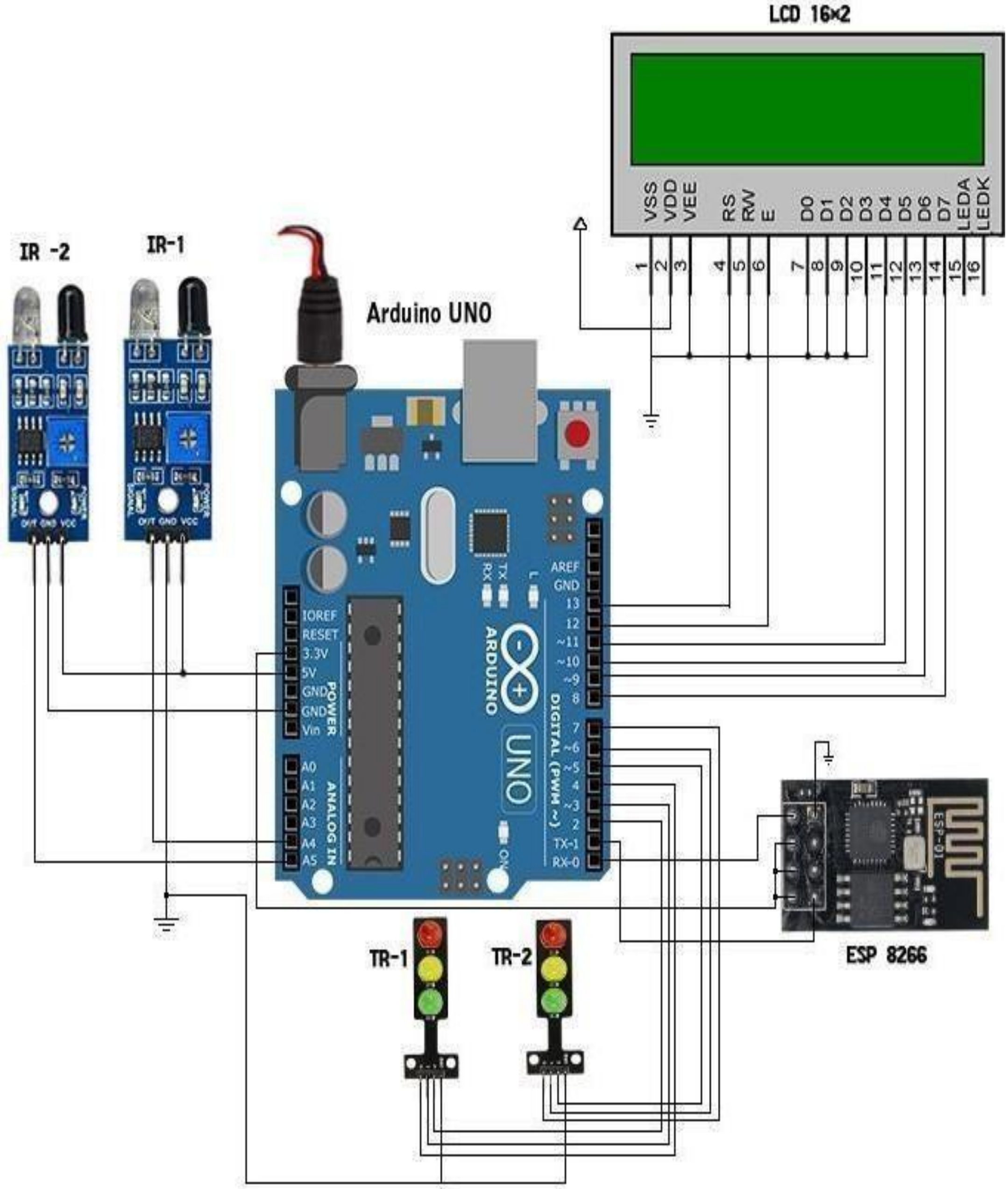


Fig 3.2.1: Schematic Diagram

3.2.1 Schematic Explanation

Arduino UNO has 28 pins among which we has 6 analog pins i.e A0-A5 to which we connect the input and 14 digital pins i.e D0 - D15 where the output devices are connected. The two IR sensors which are our input device has 3 pins each i.e vcc, gnd, out and vcc is connected to 5v supply pin of Arduino, gnd is given to the common ground and out pin is given to the input pin of Arduino A5 and A4 one from each sensor. We use two output devices like traffic lights module which has 4 pins each gnd and other 3 pins are voltage pins of each LED present in it red, yellow, green. So the gnd pin of both the modules are again connected to the common ground and 3 voltage pins of traffic signal module 1 is connected to 2,3,4 digital pins of Arduino and similarly 3 pins of traffic signal module 2 pins are connected to 5,6,7 digital pins of Arduino. We have a Wi-Fi module here where tx pin and rx pin are connected to rx and tx pin the Arduino respectively. Here reset pin, chip enable pin and vcc are connected to Arduino 3.3v pin. We also has a LCD whose data pins 14,13,12,11 are connected to 8,9,10,11 pins of Arduino respectively.

CHAPTER-4

HARDWARE AND SOFTWARE DESCRIPTION

Hardware Components

The hardware components used for the implementation of "IoT Technology based Traffic signal and Density control" are

1. Arduino UNO
2. Power Supply
3. IR sensor
4. LCD
5. Wi-Fi module
6. Traffic lights

Arduino UNO

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message and turn it into an output activating a motor, turning on an led, publishing as it reached a wider community, the Arduino board started something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board.

To do so you use the Arduino programming language (based Wiring), the Arduino Software (IDE), based on Processing. Arduino was born at the area Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming wearable, 3dprinting, and embedded environments. All Arduino boards are completely open-source, empowering from everyday objects to complex scientific instruments.

Professional gathered around open-source plat form, the software too is open-source, it is growing through the contributions of users worldwide, the Arduino board is shown in below figure.

Fig.4.1.1: Arduino uno board

Why Arduino?

Arduino Uno is inexpensive. Arduino is cross platform, its IDE can run on multiple Operating Systems such as Windows, Macintosh, Linux and many more. It has simple and clear programming environment. Its software and hardware are open-source and extensible, that is, Arduino programming can be expanded through C++ libraries and can also leap to AVR ‘C’ language programming.

It is simple and accessible to user experience; Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove the chemistry and physics principles or to get started with programming and robotics.

Designers and the architects build interactive prototypes, musicians, artists use it for installations and to experiment with new musical instruments. Makers of course use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers can start tinkering just following the step by step I constructions of a kit or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handy board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-touse package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems.

- **Inexpensive**
  Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50
- **Cross-platform**
  The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux-operating systems. Most microcontroller systems are limited to Windows.
- **Simple, clear programming environment**
  The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- **Open source and extensible software**
  The Arduino software is published as opensource tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- **Open source and extensible hardware**
  The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money and the parts of Arduino board is shown below.

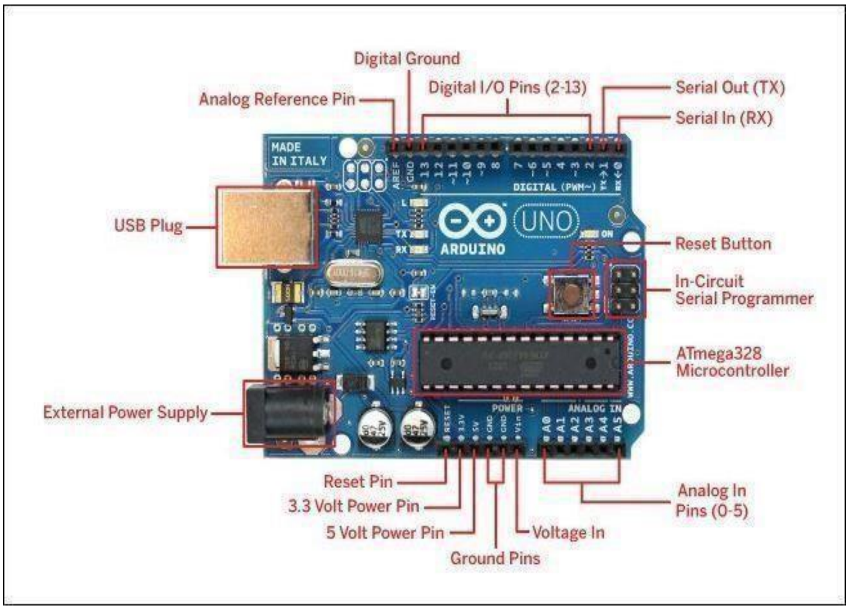


Fig.4.1.2: Arduino board description

Features of Arduino

Micro controller	Atmega328
Operating voltage	5V
Input Voltage(recommended)	7-12V
Input Voltage	6-20V
Digital I/O	14(of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 Ma
DC Current for 3.3V Pin	50 Ma
Flash Memory	32 KB(Atmega328 of which 0.5 KB used by bootloader)
SRAM	2 KB (ATmega32)
EEPROM	1 KB(ATmeha328)
Clock Speed	16 MHz

Table 4.1.1: Features of Arduino uno

- It is an easy USB interface. This allows interface with USB as this is like a serial device.
- The chip on the board plugs straight into your USB port and supports on your computer as a virtual serial port. The benefit of this setup is that serial communication is an extremely easy protocol which is time-tested and USB makes connection with modern computers and makes it comfortable.
- It is easy-to-find the microcontroller brain which is the ATmega328 chip. It has more number of hardware features like timers, external and internal interrupts, PWM pins and multiple sleep modes.
- It is an open source design and there is an advantage of being open source is that it has a large community of people using and troubleshooting it. This makes easy to help in debugging projects.
- It is 16MHz clock which is fast enough for most applications and does not speeds up the microcontroller.
- It is very convenient to manage power inside it and it had a feature of builtin voltage regulation. This also be powered directly off a USB port without any external power. You can connect an external power source of up to 12v and this regulates it to both 5v and 3.3v.
- 14 digital pins and 6 analog pins. This sort of pins allows you to connect hardware to your Arduino Uno board externally. These pins areused as a key for extending the computing capability of the Arduino Uno into the real world. This has an ICSP connector for bypassing the USB port and interfacing the Arduino directly as a serial device. This port is necessary to re-boot load your chip if it corrupts and can no longer used to your computer.
- It has a 32 KB of flash memory for storing your code.
- An on-board led is attached to digital pin 13 to make fast the debugging of code and to make the debug process easy finally it has the button to reset.

Pin Configuration of Arduino UNO

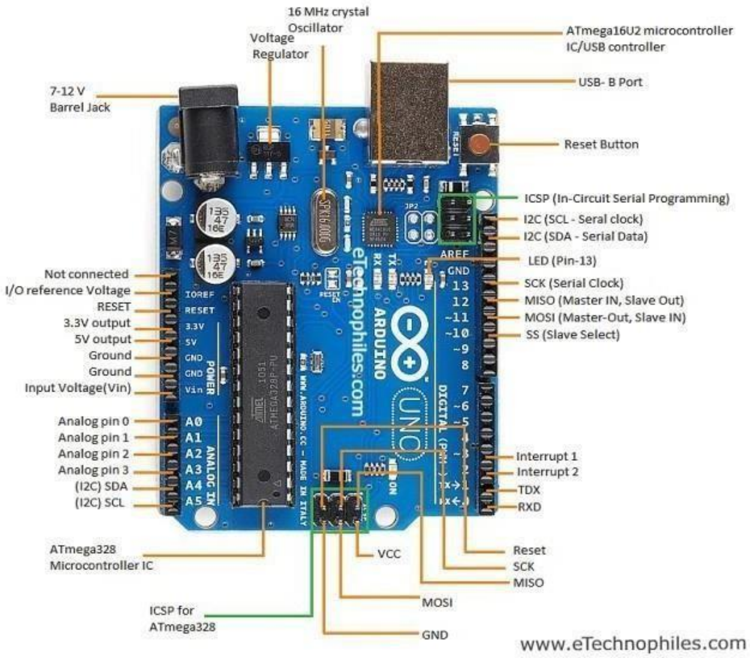


Fig 4.1.3: Pin configuration

- **Reset Pin:** resets the microcontroller.
- **GND :** used to connect the circuit to Ground
- **A0-A5:** used to provide analog input in the range of 0-5V.
- **Digital pins(0-13) :** can be used as input or output pins.
- **0(RX),1(TX) :** used to receive and transmit TTL serial data.
- **AREF :** To provide reference voltage for input voltage.
- **SPI :** used for SPI communication.

Working of Arduino Uno

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The 14 digital input/output pins can be used as input or output pins by using pinMode(), digitalRead() and digitalWrite() functions in arduino programming.

Each pin operate at 5V and can provide or receive a maximum of 40mA current, and has an internal pull-up resistor of 20-50 KOhms which are disconnected by default. Out of these 14 pins, some pins have specific functions as listedbelow:

- **Serial Pins 0 (Rx) and 1 (Tx):** Rx and Tx pins are used to receive andtransmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.
- **External Interrupt Pins 2 and 3:** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM Pins 3, 5, 6, 9 and 11:** These pins provide an 8-bit PWM output by using analogWrite() function.
- **SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK):** These pins are used for SPI communication.
- **In-built LED Pin 13:** This pin is connected with an built-in LED, when pin 13 is HIGH – LED is on and when pin 13 is LOW, its off.
- **Analog pins (A0-A5) :** used to provide analog inputs in the range of 0-5V.

- Analog pin 4 (SDA) and pin 5 (SCA) also used for TWI communication using Wire library.

Arduino Uno has a couple of other pins as explained below:

- **AREF:** Used to provide reference voltage for analog inputs with analog Reference() function.
- **Reset Pin:** resets the microcontroller.

Power

The Arduino Uno can be powered via through USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wallwart) or battery. The adapter can be connected by plugging a 2.1mm centrepositive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. the power pins are as follows.

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an onboard regulator, or be supplied by USB or another regulated 5V supply.
- **3.3V.** A 3.3volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.

Programming

The Arduino Uno can be programmed with the Arduino Software (IDE). Select Arduino Uno from the Tools Board After your code is compiled using Arduino IDE, it should be uploaded to the main microcontroller of the Arduino UNO using a USB connection. Because the main microcontroller doesn’t have a USB transceiver, you need a bridge to convert signals between the serial interface (UART interface) of the microcontroller and the host USB signals. The ATmega328 on the Arduino Uno comes pre-programmed with a boot loader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino are similar. The ATmega16U2 (or 8U2 in the rev1&rev2 boards) firmware source code is available in the Arduino repository. The ATmega16U2 or 8U2 is loaded with a DFU bootloader, which can be activated by On Rev1 boards connecting the solder jumper on the back of the board and then resetting the 8U2.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). Arduino is an open-source hardware platform that is being used by people around the globe for building electronics projects. It is an integrated platform which contains both the physical and programmable circuit otherwise known as microcontroller and a software (or IDE) that you can run on your computer to write and upload the code onto the physical board.

Arduino Board is quite popular among many people who want to get started with electronics, and unlike other embedded system boards Arduino does not require any additional hardware to upload the code (generally known as programmer). Here we will try to understand about the Arduino Architecture and its functionalities. The processor of the Arduino Board uses Harvard Architecture for which the program code and program data have separate memory. The memory of it is divided into two namely program memory and data memory. The data will be stored in the data memory whereas the program code will be stored in the flash program memory. For ex: The Atmega328 microcontroller has 32kb of flash memory, 2kb of SRAM, 1kb of EPROM and operates at 16MHz clock speed. Some of the other basic functions of Arduino are:

- **Digital write pin** is used to write the digital value of the given pin.
- **Pin mode pin** is used to set the pin to I/O mode.
- **Analog read pin** Reads the value from the specified Analog pin.
- **Analog write pin** writes the value from the specified Analog pin
- **Serial. Begins pin** sets the beginning of serial communication by setting the rate of bit
- **Digital read pin** reads the digital value of given pin.

Advantages

Inexpensive -Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50.

1. Cross-platform - The Arduino software runs on Windows, Macintosh OSX, and the Linux operating systems. Most microcontroller systems are limited to Windows.
2. Simple programming environment- Here the Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced

users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino.

3. Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you canadd AVR-C code directly into your Arduino programs if you want to.

Arduino UNO Atmega328

The Arduino UNO is a micro controller board based on the ATmega. UNP means one in Italian and is named to mark the upcoming release of the given Arduino 1.0 it has 14 digital input/output pins, 6 analog inputs, 16MHz ceramic resonator. A USB connection. A power jack, An ICSP header and a reset button.it contains information everything needed to support the microcontroller. simply connect it to a computer with a USB cable or power it with a ac-to-dc adapter or battery to get started.

The UNO differs from all preceding boards in that it does not use the FTDI USB-to serial driver chip. Instead it features the ATMega 16U2ATMega 8U2 up to version instead R2) programmed as a USB to-serial converter

Introduction to Atmega328

ATmega328 is an eight (8) bit Microcontroller. It can handle the data sized of up to eight (8) bits. It is an AVR based micro-controller. Its built-in internal memory is around 32KB. It operates ranging from 3.3V to 5V. It has an ability to store the data even when the electrical supply is removed from its biasing terminals. it is the excellent features include the cost efficiency and low power dissipation, programming lock for security purposes, real timer counter with the separate oscillator and It is generally or normally used in Embedded Systems applications. ATmega-328 is shown in the figure given below.

Fig.4.1.4: ATMega328P Microcontroller

Atmega328 Pin Description

Functions associated with the pins must be known in order to use the device appropriately. ATmega-328 pins are divided into different ports which are given in detail below.

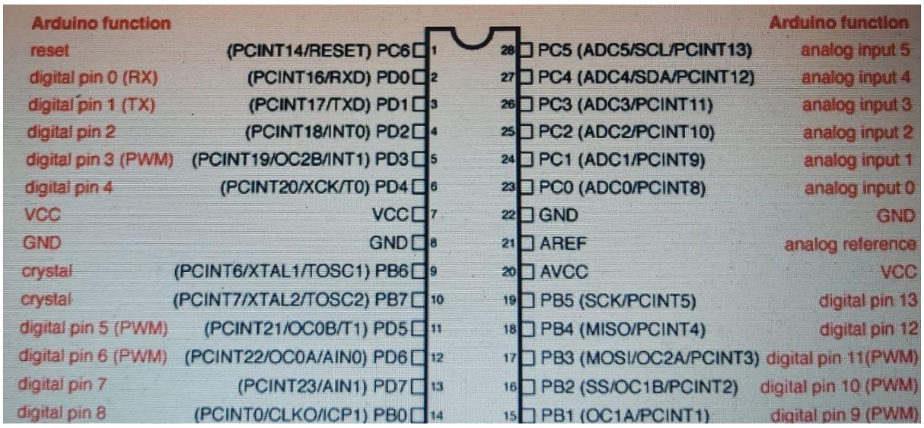


Fig.4.1.5: Arduino UNO to ATmega328P mapping

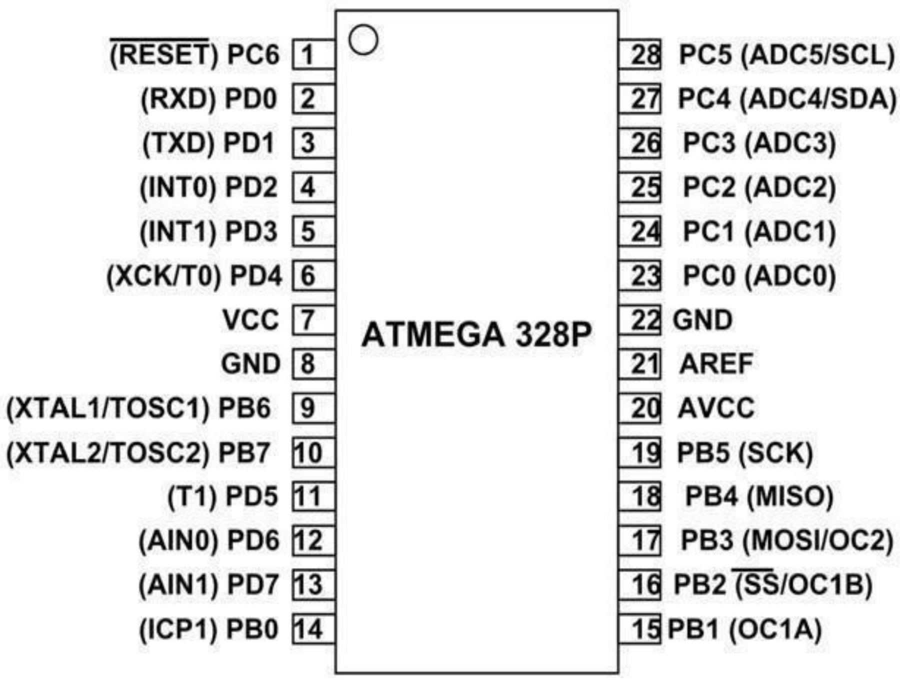


Fig.4.1.6: ATmega328P Ports

AVCC is a supply voltage pin for analog to digital converter.

VCC is a digital voltage supply

GND denotes Ground and it has a 0V.

AREF AREF is the analog reference pin for the A/D converter.

Port B consists of the pins from PB0 to PB5. This port is an 8-bit bidirectional I/O port having an internal pull-up resistor.

Port C consists of the pins from PC0 to PC6. This port is an 7-bit bidirectional I/O port, where PC6 is used as a reset input.

Port D consists of the pins from PD0 to PD7. It is also an 8bit input/output port having an internal pull-up resistor.

Applications of Arduino Uno

- Arduino Uno is used in Do-it-Yourself projects prototyping.
- In developing projects based on code-based control
- Development of Automation System
- Designing of basic circuit designs.

Power Supply

Every electrical and electronic device that we use in our day-to-day life will require a power supply. In general, we use an AC supply of 230V 50Hz, but this power has to be changed into the required form with required values or voltage range for providing power supply to different types of devices.

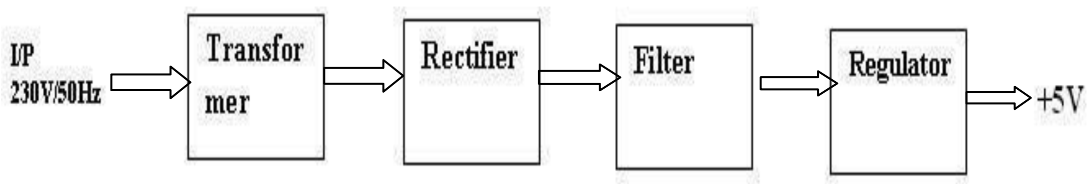


Fig 4.1.7: Block Diagram of Power supply

There are various types of power electronic converters such as step-down converter, step-up converter, voltage stabilizer, AC to DC converter, DC 25 to DC converter, DC to AC converter, and so on. For example, consider the microcontrollers that are used frequently for developing many embedded systems based projects and kits used in real-time applications. These microcontrollers require a 5V DC supply

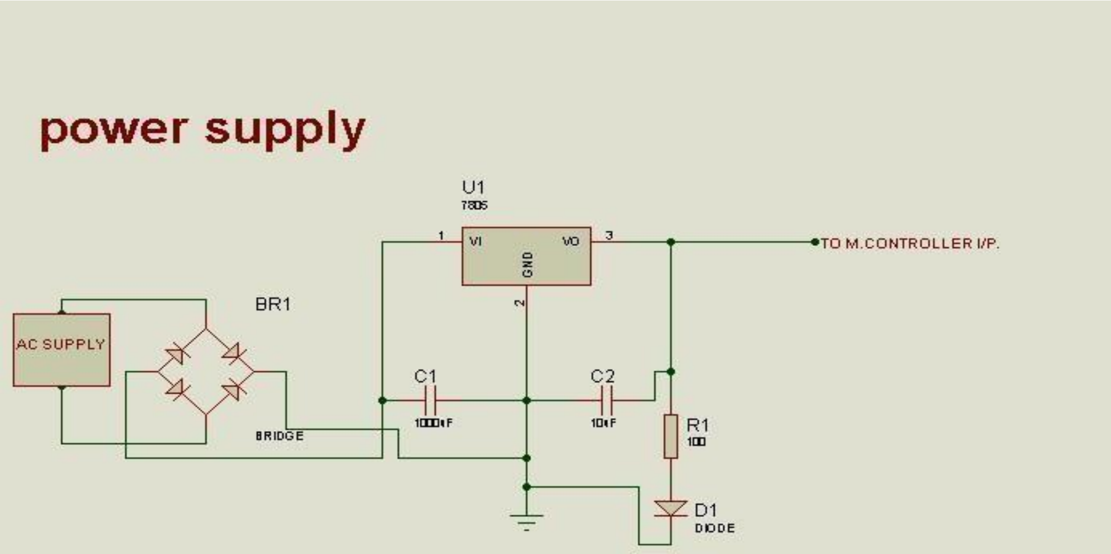


Fig 4.1.8: Circuit diagram of Power supply

IR SENSOR

Infrared technology addresses a wide variety of wireless applications. The main areas are sensing and remote controls. In the electromagnetic spectrum, the infrared portion is divided into three regions: near infrared region, mid infrared region and far infrared region.

The wavelengths of these regions and their applications are shown below.

- Near infrared region — 700 nm to 1400 nm — IR sensors, fibre optic
- Mid infrared region — 1400 nm to 3000 nm — Heat sensing
- Far infrared region — 3000 nm to 1 mm — Thermal imaging

The frequency range of infrared is higher than microwave and lesser than visible light.

For optical sensing and optical communication, photo optics technologies are used in the near infrared region as the light is less complex than RF when implemented as a source of signal. Optical wireless communication is done with IR data transmission for short range applications

An infrared sensor emits and/or detects infrared radiation to sense its surroundings. The working of any Infrared sensor is governed by three laws: Planck's Radiation law, Stephen – Boltzmann law and Wien's Displacement law. Planck's law states that "every object emits radiation at a temperature not equal to 0^0K". Stephen – Boltzmann law states that "at all wavelengths, the total energy emitted by a black body is proportional to the fourth power of the absolute temperature". According to Wien's Displacement law, "the radiation curve of a black body for different temperatures will reach its peak at a wavelength inversely proportional to the temperature".

The basic concept of an Infrared Sensor which is used as Obstacle detector is to transmit an infrared signal, this infrared signal bounces from the surface of an object and the signal is received at the infrared receiver. There are five basic elements used in a typical infrared detection system: an infrared source, a transmission medium, optical component, infrared detectors or receivers and signal processing. Infrared lasers and Infrared LED's of specific wavelength can be used as infrared sources. The three main types of media used for infrared transmission are vacuum, atmosphere and optical fibres. Optical components are used to focus the infrared radiation or to limit the spectral response.

Optical lenses made of Quartz, Germanium and Silicon are used to focus the infrared radiation. Infrared receivers can be photodiodes, phototransistors etc. some important specifications of infrared receivers are photosensitivity, detectivity and noise equivalent power. Signal processing is done by amplifiers as the output of infrared detector is very small.

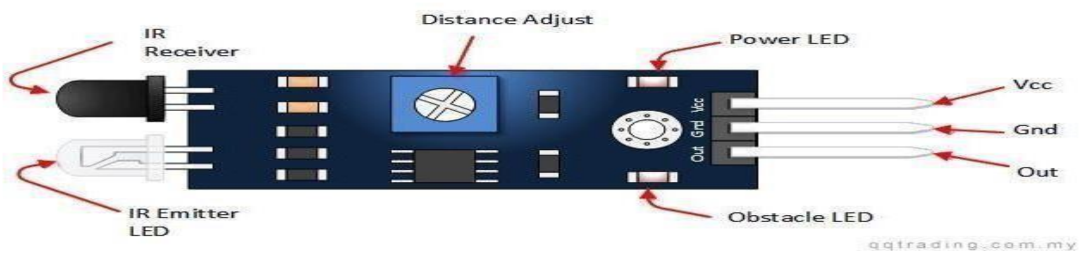


Fig 4.1.9: Basic parts of IR Sensor

Types of IR Sensors

Infrared sensors can be passive or active. Passive infrared sensors are basically Infrared detectors. Passive infrared sensors do not use any infrared source and detects energy emitted by obstacles in the field of view. They are of two types: quantum and thermal. Thermal infrared sensors use infrared energy as the source of heat and are independent of wavelength. Thermocouples, pyroelectric detectors and bolometers are the common types of thermal infrared detectors.

Quantum type infrared detectors offer higher detection performance and are faster than thermal type infrared detectors. The photosensitivity of quantum type detectors is wavelength dependent. Quantum type detectors are further classified into two types: intrinsic and extrinsic types. Intrinsic type quantum detectors are photoconductive cells and photovoltaic cells.

Active infrared sensors consist of two elements: infrared source and infrared detector. Infrared sources include an LED or infrared laser diode. Infrared detectors include photodiodes or phototransistors. The energy emitted by the infrared source is reflected by an object and falls on the infrared detector.

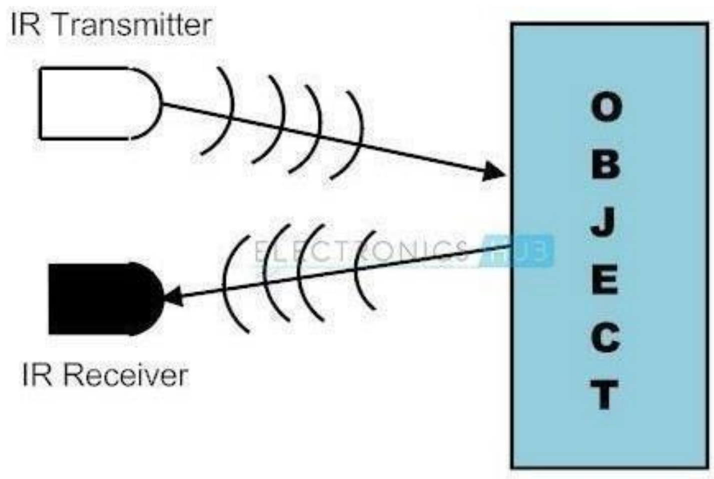


Fig 4.1.10: IR Sensor

IR Transmitter

Infrared Transmitter is a light emitting diode (LED) which emits infrared radiations. Hence, they are called IR LED's. Even though an IR LED looks like a normal LED, the radiation emitted by it is invisible to the human eye.

Fig 4.1.11: IR Transmitter

There are different types of infrared transmitters depending on their wavelengths, output power and response time. A simple infrared transmitter can be constructed using an infrared LED, a current limiting resistor and a power supply.

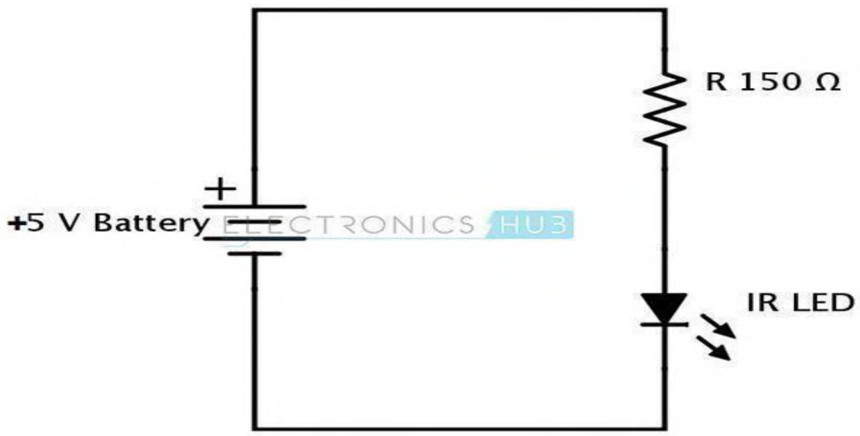


Fig.4.1.12: Schematic of IR Transmitter

When operated at a supply of 5V, the IR transmitter consumes about 3 to 5 mA of current. Infrared transmitters can be modulated to produce a particular frequency of infrared light. The most commonly used modulation is OOK (ON – OFF – KEYING) modulation.

IR transmitters can be found in several applications. Some applicationsrequire infrared heat and the best infrared source is infrared transmitter. When infrared emitters are used with Quartz, solar cells can be made.

IR Receiver

Infrared receivers are also called as infrared sensors as they detect the radiation from an IR transmitter. IR receivers come in the form of photodiodes and phototransistors. Infrared Photodiodes are different from normal photo diodes as they detect only infrared radiation.

Fig.4.1.13: IR Receiver

Different types of IR receivers exist based on the wavelength, voltage, package, etc. When used in an infrared transmitter – receiver combination, the wavelength of the receiver should match with that of the transmitter.

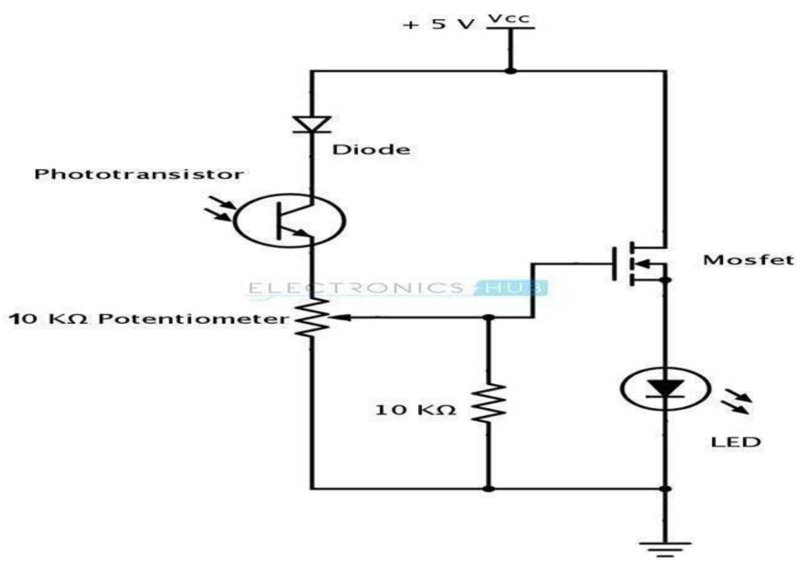


Fig.4.1.14: Infrared Receiver circuit using a phototransistor

It Consist of an IR phototransistor, a diode, a MOSFET, a potentiometer and an LED. When the phototransistor receives any infrared radiation, current flows through it and MOSFET turns on. This in turn lights up the LED which acts as a load. The potentiometer is used to control the sensitivity of the phototransistor.

Principle of Working

The principle of an IR sensor working as an Object Detection Sensor can be explained using the following figure. An IR sensor consists of an IR LED and an IR Photodiode; together they are called as Photo – Coupler or Opto – Coupler.

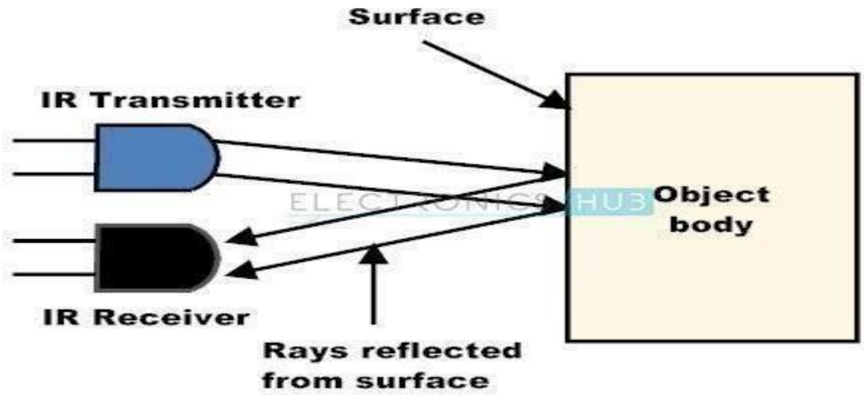


Fig.4.1.15: Working of IR Sensor

When the IR transmitter emits radiation, it reaches the object and some of the radiation reflects back to the IR receiver. Based on the intensity of the reception by the IR receiver, the output of the sensor is defined.

Pin Configuration

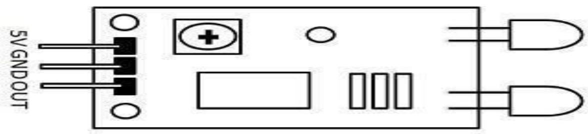


Fig 4.1.16: Pin Configuration

- Vcc – Power supply input of 5v
- GND – Power supply ground
- OUT - Active high output

Features

- 5VDC Operating voltage
- I/O pins are 5V and 3.3V compliant
- Range: Up to 20cm
- Adjustable Sensing range
- Built-in Ambient Light Sensor
- 20mA supply current

Applications

- Obstacle Detection
- Industrial safety devices
- Wheel encode

LCD (Liquid Crystal Display)

A liquid crystal display (LCD) is a thin, flat display device made up of any number of colour or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCDs connected to the controllers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0) For a 4-bit data bus it only requires the supply lines plus 6 extra lines (RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

Pin	Symbol	I/O	Description
1	GND	-	Ground
2	Vcc	-	+5V power supply
3	VEE	-	Contrast control
4	RS	I	command/data register selection
5	R/W	I	write/read selection
6	E	I/O	Enable
7	DB0	I/O	The 8-bit data bus
8	DB1	I/O	The 8-bit data bus
9	DB2	I/O	The 8-bit data bus
10	DB3	I/O	The 8-bit data bus
11	DB4	I/O	The 8-bit data bus
12	DB5	I/O	The 8-bit data bus
13	DB6	I/O	The 8-bit data bus
14	DB7	I/O	The 8-bit data bus

Table.4.1.2: LCD Pin Description

Pin Description Of 16x2:

- Pin1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.
- Pin2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
- Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
- Pin4 (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).
- Pin5 (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).

- Pin 6 (Enable/Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit & constantly held high.
- Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.
- Pin15 (+ve pin of the LED): This pin is connected to +5V
- Pin 16 (-ve pin of the LED): This pin is connected to GND.

Schematic Diagram

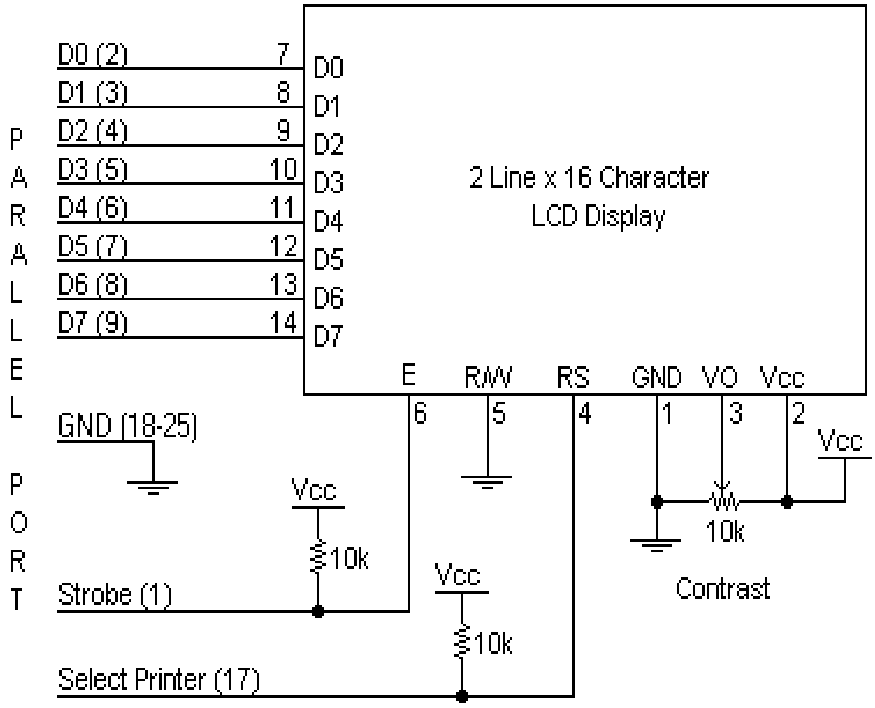


Fig.4.1.17: Schematic Diagram of LCD

- Above is the quite simple schematic. The LCD panel's Enable and register select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit ismore portable for a wider range of computers, some of which may have no internal pull up resistors.

- We make no effort to place the Data bus into reverse direction. Therefore we hard wire the R/W line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.
- The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you havetrouble with the circuit working properly.

Code (hex)	Command to LCD Instruction Register
1	Clear display screen
2	Return home
4	Shift cursor to left
5	Shift display right
6	Shift cursor to right
7	Shift display left
8	Display off, Cursor off
A	Display off, Cursor on
C	Display on, cursor off
E	Display on, cursor blinking
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80	Force cursor to beginning of 1st line
C0	Force cursor to beginning of 2nd line
38	2 lines and 5x7 matrix

Table 4.1.3: LCD commands

Features

- Operating Voltage is 4.7V to 5.3V
- Current consumption is 1mA without backlight
- Alphanumeric LCD display module, meaning can display alphabets
- Consists of two rows and each row can print 16 characters.
- Each character is build by a 5×8 pixel box
- Can work on both 8-bit and 4-bit mode
- It can also display any custom generated characters
- Available in Green and Blue Backlight

Applications

- In most of the applications that’s have only small values to show, uses the LCD.
- Most of the commercial meters use this module to represent the data output.
- In the toys and developing projects, it is still vastly in use.
- In black and white printers, it helps to show the printer settings and status.

Wi-Fi Module ESP8266

ESP8266 is a complete and self-contained Wi-Fi network solutions that can carry software applications, or through Another application processor uninstall all Wi-Fi networking capabilities. ESP8266 when the device is mounted and as the only application of the application processor, the flash memory can be started directly from an external Move. Built-in cache memory will help improve system performance and reduce memory requirements. Another situation is when wireless Internet access assume the task of Wi-Fi adapter, you can add it to any microcontroller-based design, the connection is simple, just by SPI / SDIO interface or central processor AHB bridge interface. Processing and storage capacity on ESP8266 powerful piece, it can be integrated via GPIO ports sensors and other applications specific equipment to achieve the lowest early in the development and operation of at least occupy system resources. The ESP8266 highly integrated chip, including antenna switch balun, power management converter, so with minimal external circuitry, and includes front-end module, including the entire solution designed to minimize the space occupied by PCB. The system is equipped with ESP8266 manifested leading features are: energy saving VoIP quickly switch between the sleep / wake patterns, with low-power operation adaptive radio bias, front-end signal processing functions, troubleshooting and radio systems coexist characteristics eliminate cellular / Bluetooth / DDR / LVDS / LCD interference.

Fig 4.1.18: Wi-fi Module

Pin Configuration

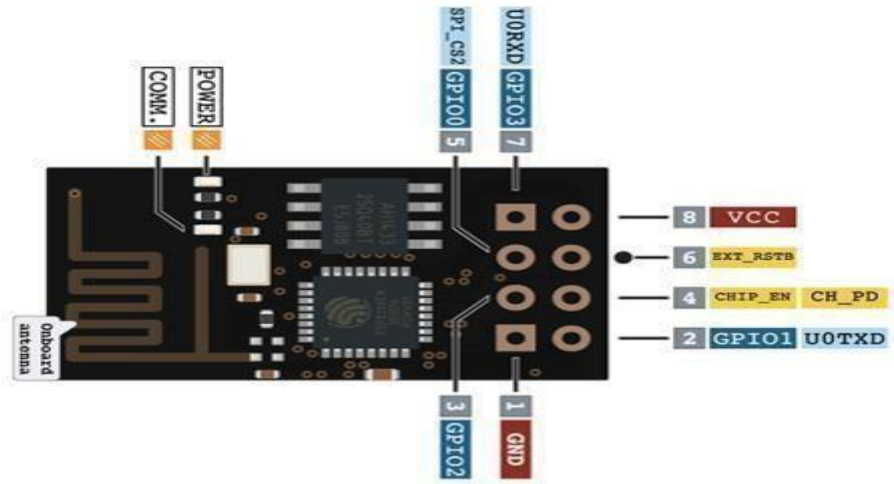


Fig 4.1.19: Pin Configuration

- Ground - Connected to the ground of the circuit
- Tx - Connected to Rx pin of programmer/microcontroller to upload program
- GPIO 2 - General purpose Input/output pin
- CH_EN - Chip Enable – Active high
- GPIO 0 - General purpose Input/output pin
- Reset – Reset pin is the Active low pin that resets the Wi-Fi module
- Rx - Connected to Tx pin of microcontroller to collect the data
- Vcc - Connect to +3.3V only

Ultra-low power technology

ESP8266 specifically for mobile devices, wearable electronics and networking applications design and make the machine to achieve the lowest energy consumption, together with several other patented technology. This energyefficient construction in three modes: active mode, sleep mode and deep sleep mode type. When ESP8266 using high-end power management technology and logic systems to reduce non-essential functions of the power conversion regulate sleep patterns and work modes, in sleep mode, it consumes less than the current 12uA, is connected, it consumes less power to 1.0mW (DTIM = 3) or 0.5mW (DTIM = 10). Sleep mode, only calibrated real-time clock and watchdog in working condition. Real-time clock can be programmed to wake ESP8266 within a specific period of time. Through programming, ESP8266 will automatically wake up when detected certain to happen. ESP8266 automatic wake-up in the shortest time, this feature can be applied to the SOC for mobile devices, so before you turn Wi- Fi SOC are in a low-power standby mode. To meet the power requirements of mobile devices and wearable electronics products, ESP8266 at close range when the PA output power can be reduced through software programming to reduce overall power consumption in order to adapt to different applications.

Specifications

- 802.11 b / g / n
- Wi-Fi Direct (P2P), soft-AP
- Built-in TCP / IP protocol stack
- Built-in TR switch, balun, LNA, power amplifier and matching network
- Built-in PLL, voltage regulator and power management components
- 802.11b mode + 19.5dBm output power
- Built-in temperature sensor
- Support antenna diversity
- off leakage current is less than 10uA
- Built-in low-power 32-bit CPU: can double as an application processor

- SDIO 2.0, SPI, UART
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU, A-MSDU aggregation and the 0.4 Within wake
- 2ms, connect and transfer data packets
- standby power consumption of less than 1.0mW (DTIM3)

Applications

- Smart Power Plug
- Home Automation
- mesh network
- industrial wireless control
- Baby Monitor
- Network Camera
- sensor networks
- wearable electronics
- wireless location-aware devices
- Security ID tag

Traffic Signal Module

The LED Traffic Lights Signal Module / Digital Signal Output Traffic Light Module is a mini-traffic light display module, high brightness, very suitable for the production of a traffic light system model.

Fig 4.1.20: Traffic Signal Module

1. GND – connected to the ground of the circuit
2. R – Red LED voltage pin

3. Y – Yellow LED voltage pin
4. G – Green LED voltage pin

Specifications

- PCB Size: 60 x 20 x 13.5 (LxWxH)mm.
- Colour: red, yellow, green
- LED Size: 10mm x 7.
- Brightness: Normal brightness.
- Voltage: 5V.
- Interface: common cathode red yellow green control

Application

- Analog traffic signal system
- Teaching demonstration
- Indication sign
- Light alarm system

Software Requirement

- Arduino IDE software
- Proteus

Arduino IDE software:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

How to Download the Arduino Software (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

Installation

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.4.2.1: USB Cable A plug to B plug

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig.4.2.2: USB Cable A plug to Mini-B plug

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

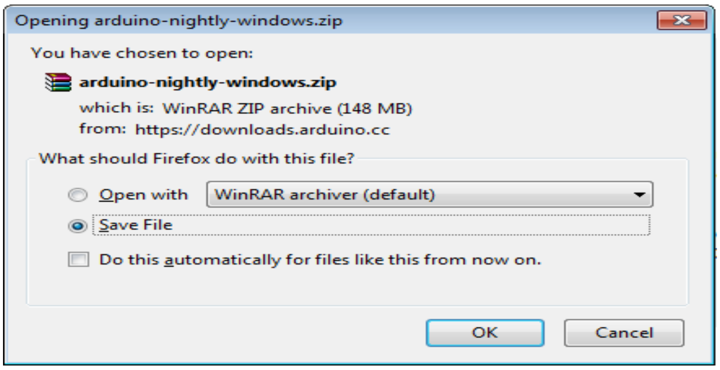


Fig.4.2.3: Unzip of Arduino IDE software

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port. Connect the Arduino board to your computer using the USB cable. The green power LED (labelled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

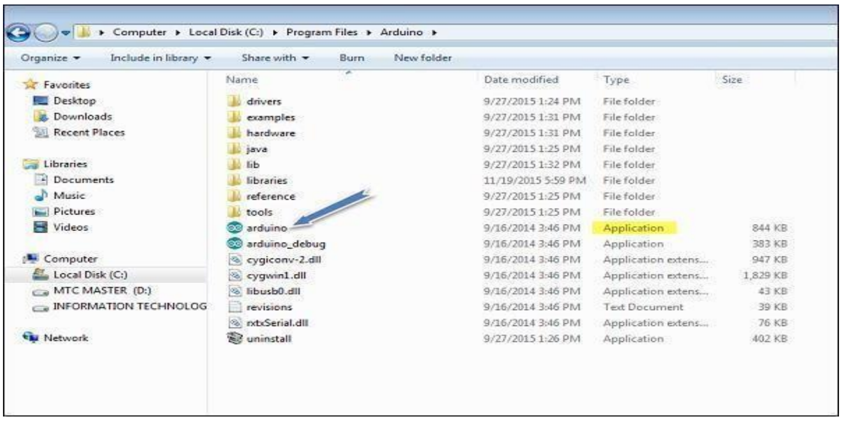


Fig.4.2.4: Open IDE software

Step 5 − Open your first project.

Once the software starts, you have two options:

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**

To open an existing project example, select

File → Example → Basics → Blink.

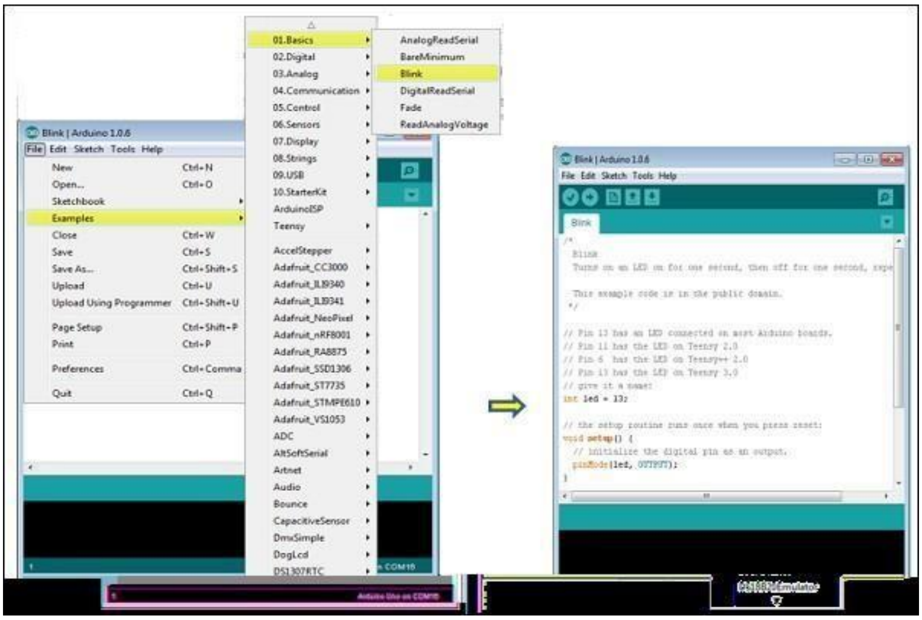


Fig.4.2.5: Open existing Project

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

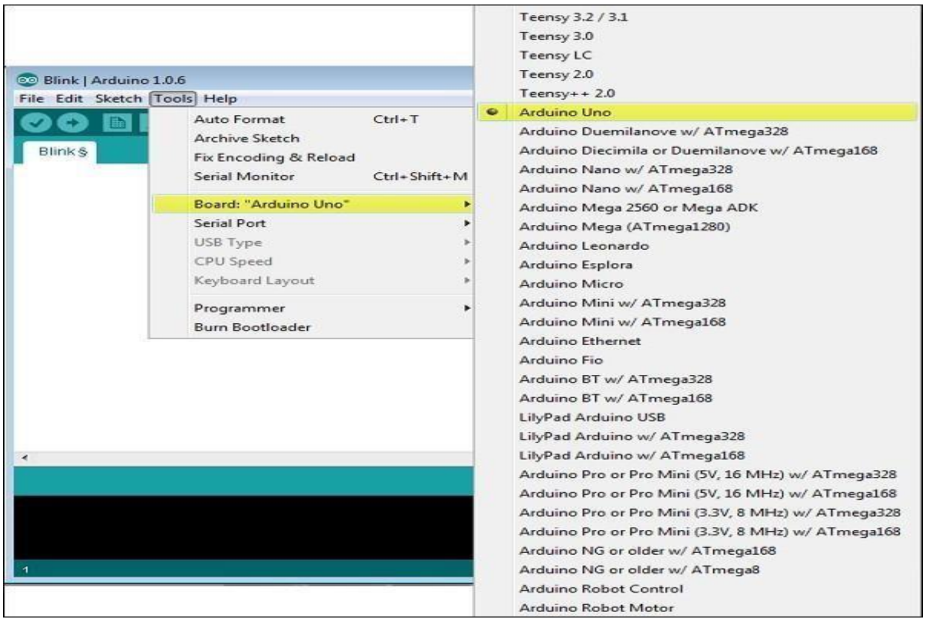


Fig.4.2.6: Selection of Arduino board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

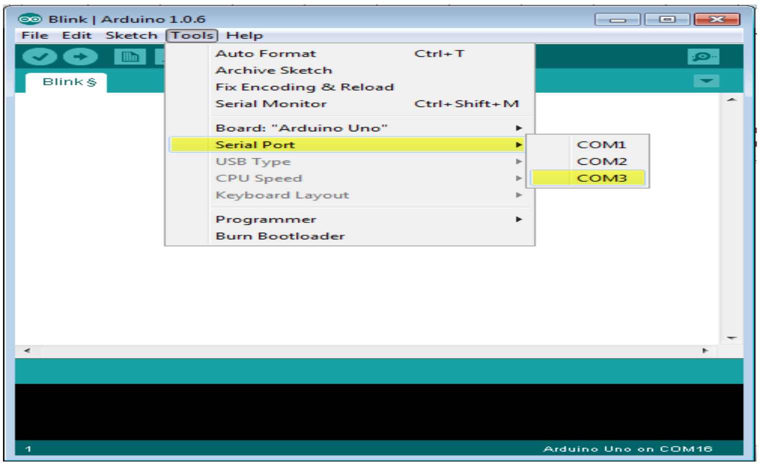


Fig.4.2.7: Selection of Serial Port

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

Fig.4.2.8: Uploading the program to board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24 V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. Here is what this looks like.

Fig.4.2.9: Battery connected to Arduino

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving on

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called “sketches”, but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

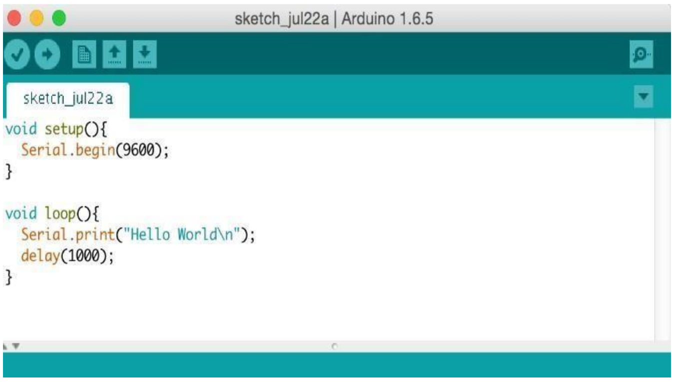


Fig.4.2.10: Program in Arduino-speak

Proteus

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline "From concept to completion" .

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing.

ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.

The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design

Step 1: Open ISIS software and select New design in File menu

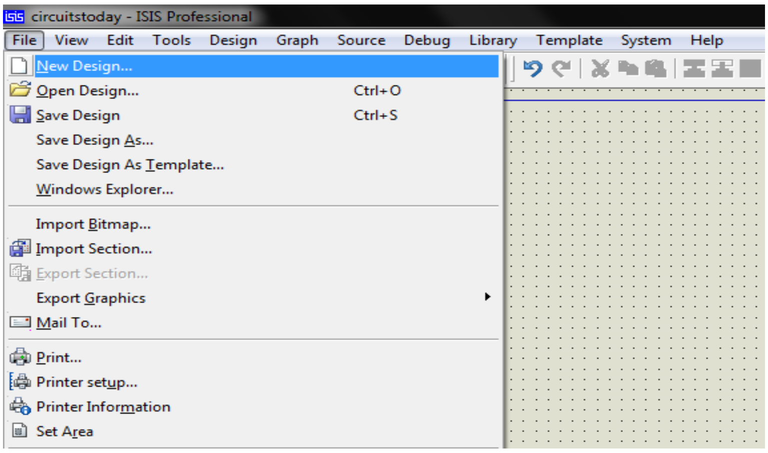


Fig.4.2.11: Proteus file menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

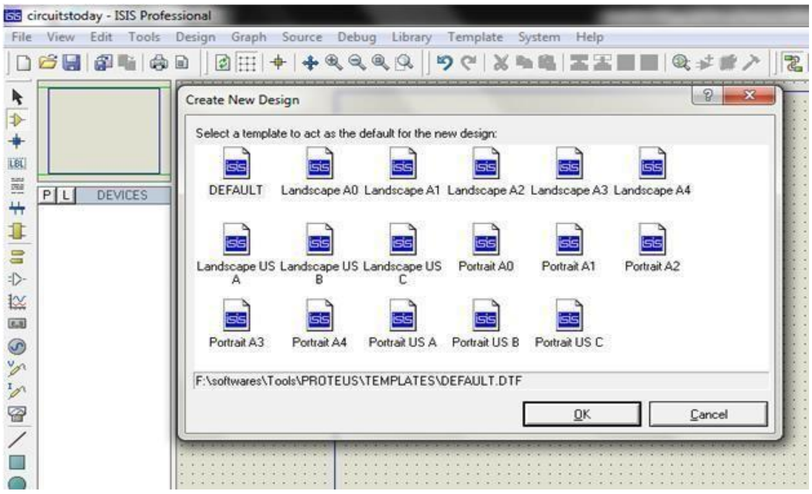


Fig. 4.2.12: Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout.

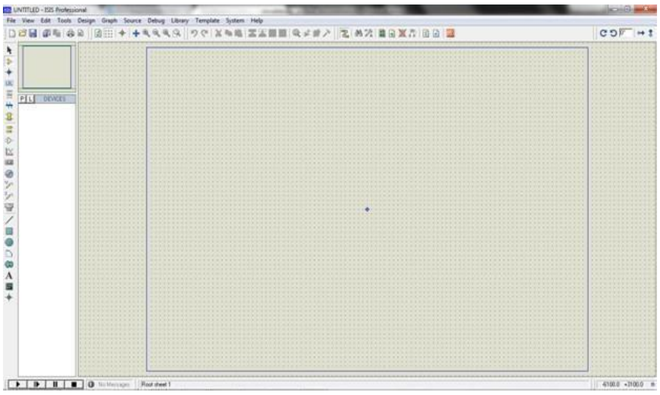


Fig. 4.2.13: Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

Step 6: Select the components from categories or type the part name in Keywords text box.

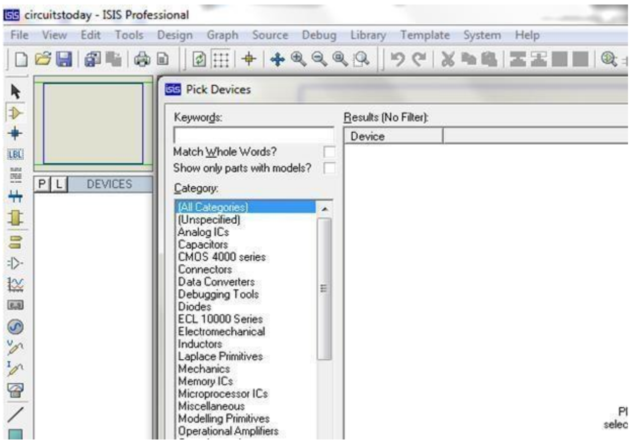


Fig.4.2.14: Keywords Textbox

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

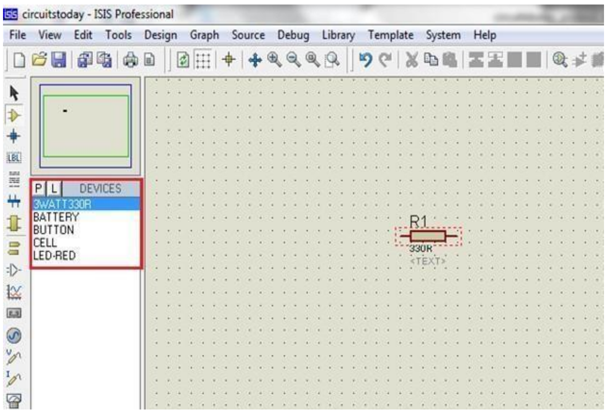


Fig. 4.2.15: Component Selection

Place all the required components and route the wires i.e., make connections. Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

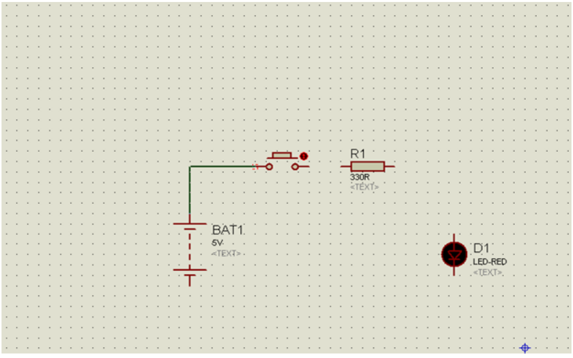


Fig.4.2.16: Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

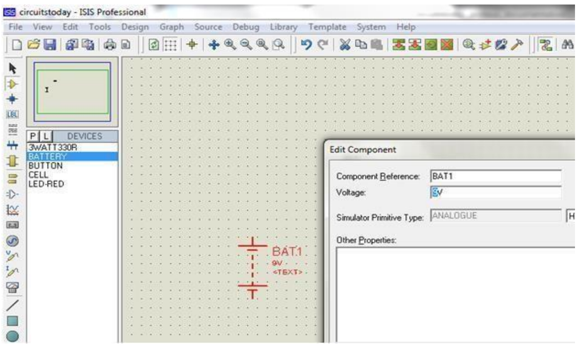


Fig.4.2.17: Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

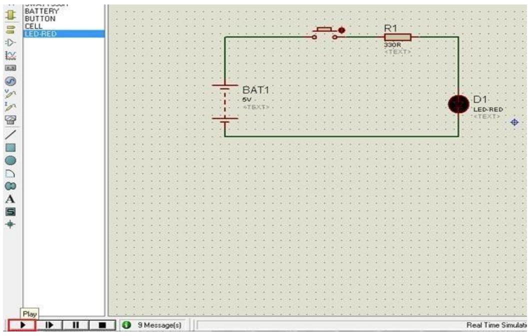


Fig.4.2.18: Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

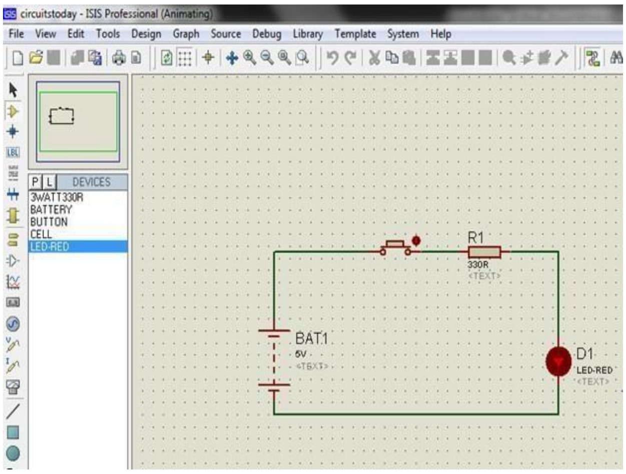


Fig.4.2.19: Simulation Animating

Simulation can be stepped, paused or stopped at any time

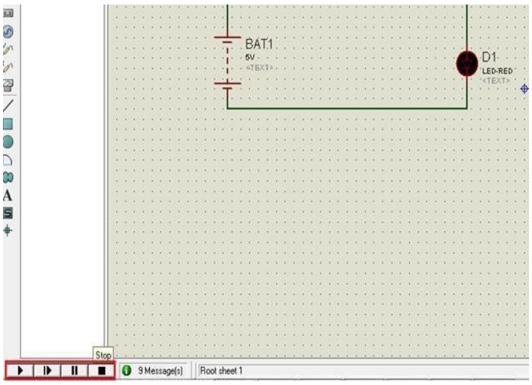


Fig.4.2.20: Simulation Step-Pause-Stop Buttons

CHAPTER 5

SOFTWARE TESTING AND CODE IMPLEMENTATION

Software Testing

Software Testing is a process of executing the application with an intent to find any software bugs. It is used to check whether the application met its expectations and all the functionalities of the application is working. The final goal of testing is to check whether the application is behaving in the way it is supposed to under specified conditions. All aspects of the code are examined to check the quality of application. The primary purpose of testing is to detect software failures so that defects may be uncovered and corrected. The test cases are designed in such way that scope of finding the bugs is maximum.

Testing Levels

There are various testing levels based on the specificity of test.

- **Unit testing:** Unit testing refers to tests conducted on a section of code in order to verify the functionality of that piece of code. This is done at the function level.
- **Integration Testing:** Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Its primary purpose is to expose the defects associated with the interfacing of modules.
- **System Testing:** System testing tests a completely integrated system to verify that the system meets its requirements.
- **Acceptance testing:** Acceptance testing tests the readiness of application, satisfying all requirements.
- **Performance testing:** Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or device such as response time or millions of instructions per second etc.

System Test Cases

A test case is a set of test data, preconditions, expected results and post conditions, developed for a test scenario to verify compliance against a specific requirement.

We have designed and executed a few test cases to check if the project meets the functional requirements.

Code Implementation

5.4.1 Project Code

```
#include"LiquidCrystal.h"

LiquidCrystal lcd(13,12,11,10,9,8);


int
red_led_road_1=2,yellow_led_road_1=3,green_led_road_1=4,red_led_road_2=5,yellow_le
d_road_2=6,green_led_road_2=7;

int sensor_road_1=A4,sensor_road_2=A5;


#include<SoftwareSerial.h>

char inputByte;

void setup()

{

  Serial.begin(9600);

  lcd.begin(16,2);

  lcd.clear();

  lcd.setCursor(0,0);

  lcd.print("IOT Based");

  delay(200);

  lcd.setCursor(0,1);

  lcd.print("Traffic Control");

  delay(3000);


  pinMode(sensor_road_1, INPUT);

  pinMode(sensor_road_2, INPUT);

  pinMode(red_led_road_1,OUTPUT);

  pinMode(red_led_road_2,OUTPUT);
```

```
 pinMode(yellow_led_road_1,OUTPUT);
 pinMode(yellow_led_road_2,OUTPUT);
 pinMode(green_led_road_1,OUTPUT);
 pinMode(green_led_road_2,OUTPUT);
 digitalWrite(red_led_road_1,LOW);
 digitalWrite(red_led_road_2,LOW);
 digitalWrite(yellow_led_road_1,LOW);
 digitalWrite(yellow_led_road_2,LOW);
 digitalWrite(green_led_road_1,LOW);
 digitalWrite(green_led_road_2,LOW);
}

void loop()
{
 lcd.clear();lcd.setCursor(0,0);lcd.print("NORMAL");

 digitalWrite(red_led_road_1,LOW);digitalWrite(yellow_led_road_2,LOW);
 digitalWrite(red_led_road_2,HIGH);digitalWrite(yellow_led_road_1,HIGH);
 digitalWrite(green_led_road_1,HIGH);digitalWrite(green_led_road_2,LOW);
 delay(7000);
 digitalWrite(red_led_road_1,LOW);digitalWrite(yellow_led_road_2,HIGH);
 digitalWrite(red_led_road_2,HIGH);digitalWrite(yellow_led_road_1,LOW);
 digitalWrite(green_led_road_1,HIGH);digitalWrite(green_led_road_2,LOW);
 delay(7000);
 digitalWrite(red_led_road_1,HIGH);digitalWrite(yellow_led_road_2,LOW);
 digitalWrite(red_led_road_2,LOW);digitalWrite(yellow_led_road_1,HIGH);
 digitalWrite(green_led_road_1,LOW);digitalWrite(green_led_road_2,HIGH);
 delay(7000);

 if(digitalRead(sensor_road_1)==HIGH)
```

```
 {
  lcd.clear();lcd.setCursor(0, 0);lcd.print("Density ON: 1");
  delay(800);
  digitalWrite(red_led_road_1,LOW);digitalWrite(yellow_led_road_2,LOW); //OFF
  digitalWrite(red_led_road_2,HIGH);digitalWrite(yellow_led_road_1,LOW);  //ON
  digitalWrite(green_led_road_1,HIGH);digitalWrite(green_led_road_2,LOW);//ON
  delay(7000); }
 if(digitalRead(sensor_road_2)==HIGH)
 {
  lcd.clear();lcd.setCursor(0, 0);lcd.print("Density ON: 2");
  delay(500);
  digitalWrite(red_led_road_1,HIGH);digitalWrite(yellow_led_road_2,LOW);
  digitalWrite(red_led_road_2,LOW);digitalWrite(yellow_led_road_1,LOW);
  digitalWrite(green_led_road_2,HIGH);digitalWrite(green_led_road_1,LOW);
  delay(7000);
 }

while(Serial.available()>0)
{
  inputByte=  Serial.read();
  Serial.println(inputByte);
  if (inputByte=='*')
  {
    lcd.clear();lcd.setCursor(0, 0);lcd.print("VIP Vehicle ON:1");
    delay(800);
    digitalWrite(red_led_road_1,LOW);digitalWrite(yellow_led_road_2,LOW);
    digitalWrite(red_led_road_2,HIGH);digitalWrite(yellow_led_road_1,LOW);
    digitalWrite(green_led_road_1,HIGH);digitalWrite(green_led_road_2,LOW);
    delay(9000);
  }
```

```
  if (inputByte=='#')
  {
    lcd.clear();lcd.setCursor(0, 0);lcd.print("VIP Vehicle ON:2");
    delay(900);
    digitalWrite(red_led_road_1,HIGH);digitalWrite(yellow_led_road_2,LOW);
    digitalWrite(red_led_road_2,LOW);digitalWrite(yellow_led_road_1,LOW);
    digitalWrite(green_led_road_2,HIGH);digitalWrite(green_led_road_1,LOW);
    delay(9000);
  }
  }
}
```

CHAPTER 6

RESULT ANALYSIS

Working of traffic lights at normal condition

This hardware setup gives a brief idea as to how the project “IoT Technology based Traffic Signal and Density Control” works. As Initially when there is normal flow of traffic then this system works normally with a fixed time delay that we have set. The LCD which we have used displays us that is it working in normal condition.

The code below is the condition for normal functioning of traffic signals.

Fig 6.1.1: LCD display for Normal condition

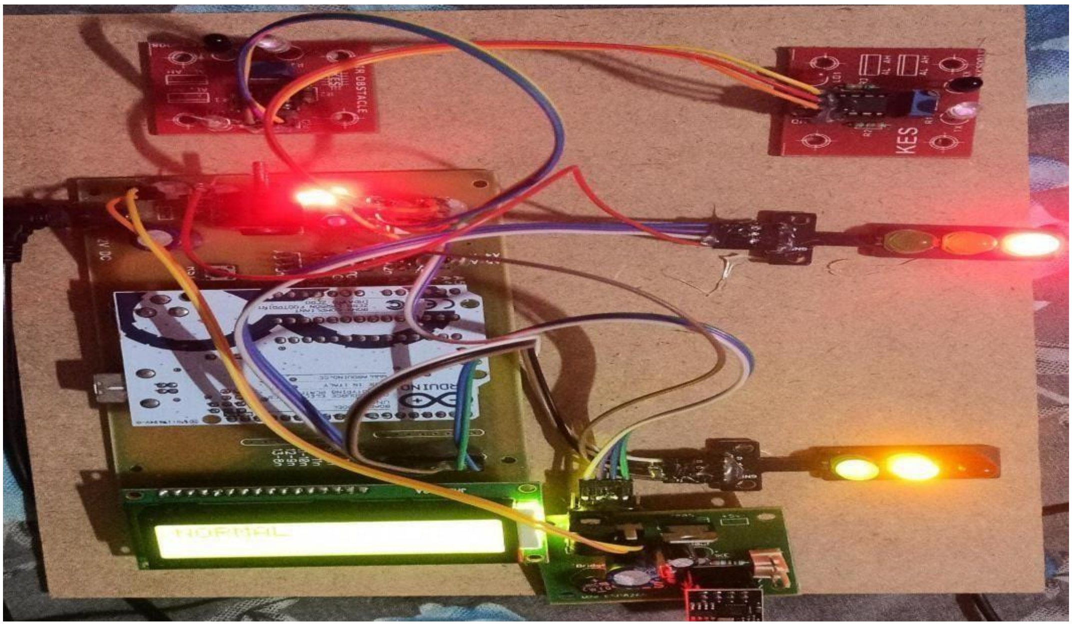


Fig 6.1.2: Normal functioning of the traffic signal

Working of traffic lights based on Density

When the IR Sensor senses that there is density present at 1 road and not in the other, then we use this kind of condition depending on the density present on which road.

Density at road 1

- When the IR Sensor senses that there is density present at road 1 and not in the road 2, then we use this kind of condition for working of traffic signals.
- Condition for the working of traffic signal when density is present in road 1

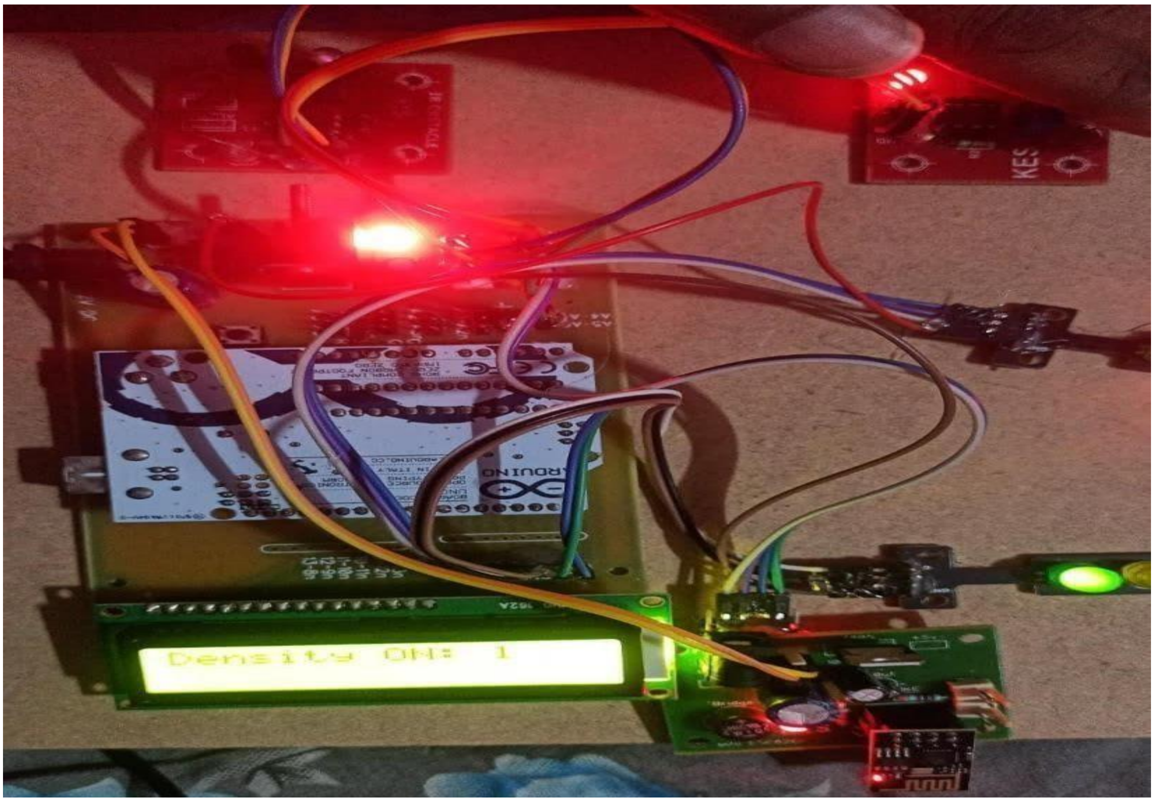


Fig 6.2.1: Working of traffic signals when Density is at road 1

Fig 6.2.2: LCD display when Density is at road 1

Density at road 2

- When the IR Sensor senses that there is density present at road 2 and not in the road 1, then we use this kind of condition for working of traffic signals.
- Condition for the working of traffic signal when density is present in road 2

Fig 6.2.3: LCD display when Density is at road 2

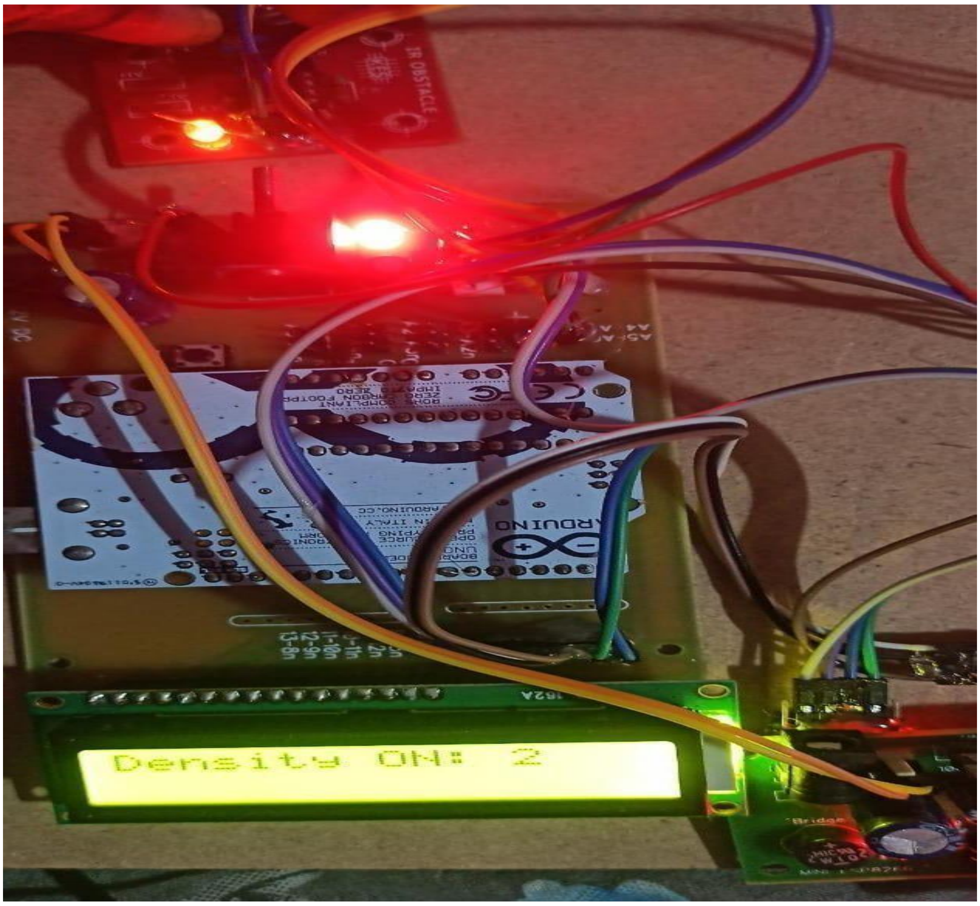


Fig 6.2.4: Working of traffic signals when Density is at road 2

Manually monitoring for VIP Vehicle

- Here we are manually monitoring the control for VIP vehicles using TELNET app through the mobile phone.

VIP vehicle at road 1

- This condition is used where ever we want to manually allow any VIP vehicle that is struck at road 1. This condition allows us to clear the lane when we spot a VIP vehicle on the road 1.
- Here we need to click * in the Telnet app so that green signal is at road 1 and red signal at road 2.
- Condition for the working of traffic signal when VIP vehicle is present at road 1

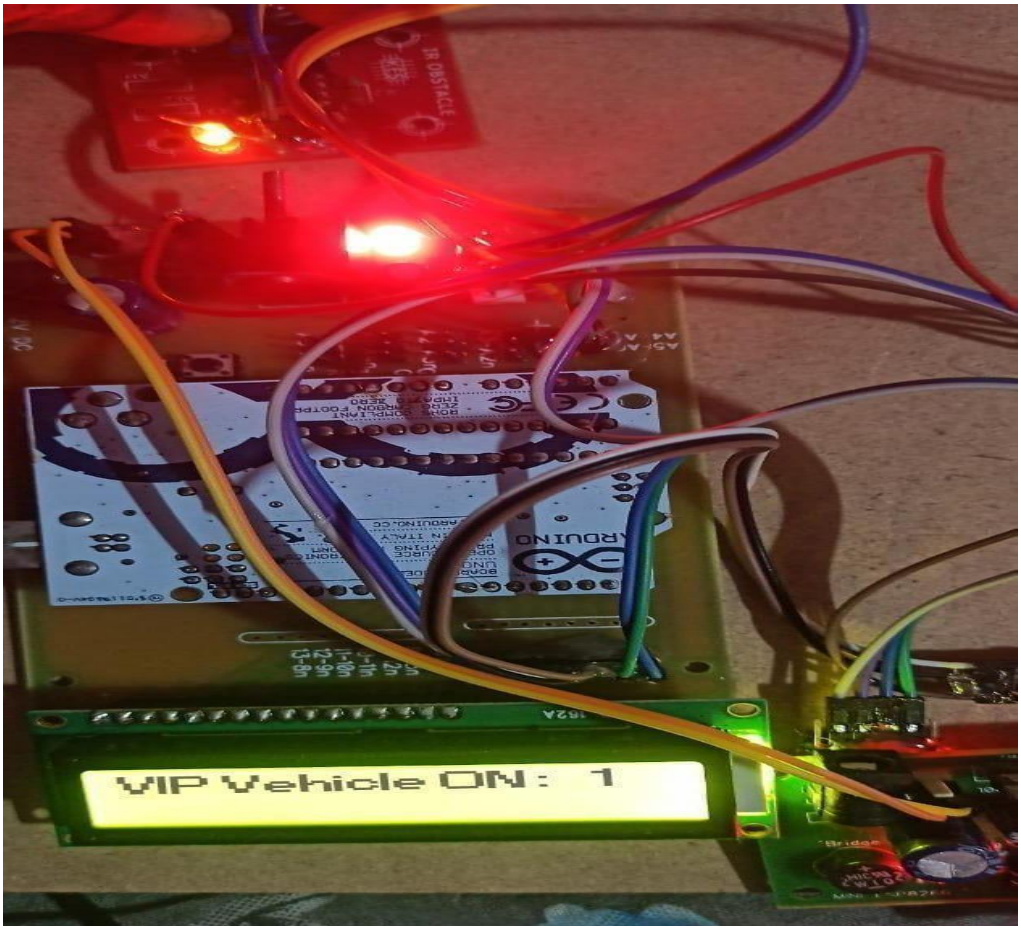


Fig 6.3.1: Working of traffic signals when VIP Vehicle is at road 1

VIP vehicle at road 2

- This condition is used where ever we want to manually allow any VIP vehicle that is struck at road 2. This condition allows us to clear the lane when we spot a VIP vehicle on the road 2.

- Here we need to click # in the Telnet app so that green signal is at road 2 and red signal at road 1.
- Condition for the working of traffic signal when VIP vehicle is present at road 2.

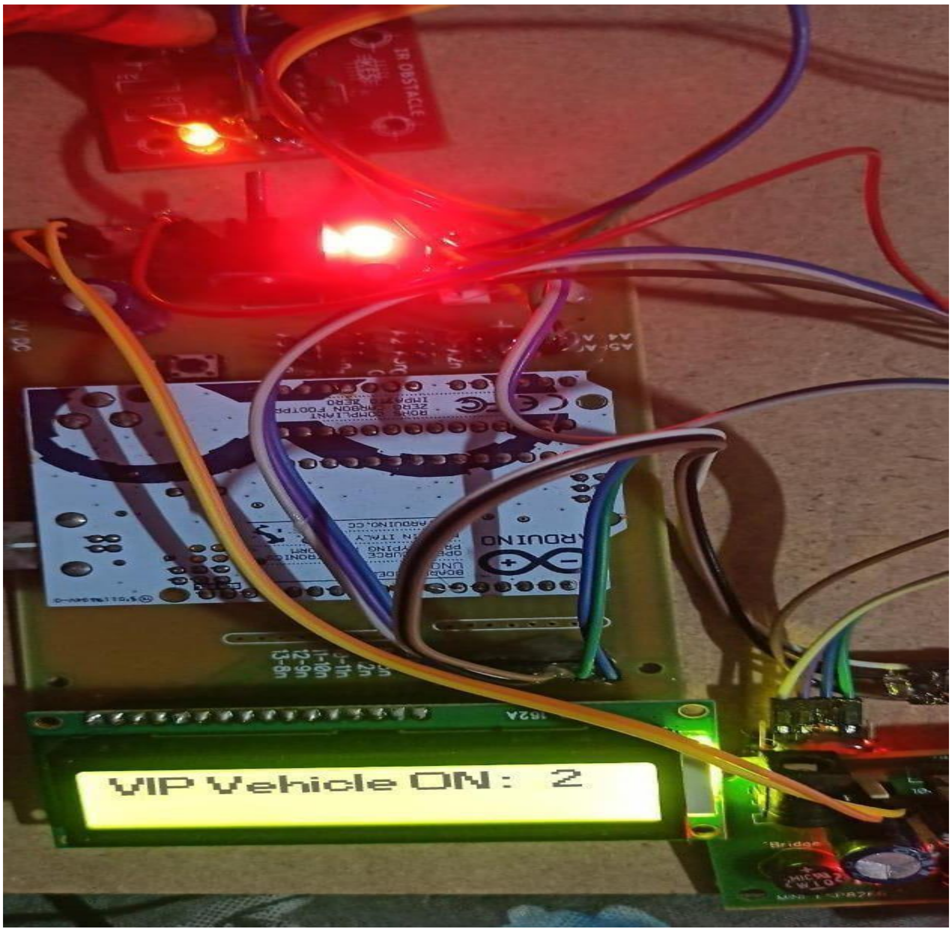


Fig 6.3.2: Working of traffic signals when VIP Vehicle is at road 2

CHAPTER 7

LIMITATIONS AND ADVANTAGES

Limitations

- Power Supply must be 5V.
- Manual errors are possible

Advantages

- Reduces traffic jams at the signals and on the streets
- A large chunk of vehicles can transit the signals efficiently
- Instant traffic clearance for emergency vehicle
- Effective time management
- Time can be shared evenly for all intersections

CHAPTER 8
CONCLUSION AND FUTURE SCOPE

Conclusion

There is exigent need of efficient traffic management system in our country, as India meets with many road accidents every day. To reduce this congestion and unwanted time delay in traffic an advanced system is designed here in this project. With field application of this technology, the maddening chaos of traffic can be effectively channelized by distributing the time slots based on the merit of the vehicle load in certain lanes of multi junction crossing.

The proposed system aims to save the number of manhours wasted at the signals and hence making effective utilization of time.

So in this way, apart from operating the signal manually or by keeping them constant, the signal can be monitored and traffic can be controlled using the sensors and by controlling the density of traffic. Even, instead of clearing the traffic by the traffic police, the green will be signaled automatically, to give way for the ambulance by clicking the button provided.

Future Scope

Though the prototype model worked very efficiently with remarkable outputs, the real-life situation is going to be way more challenging and demanding.

Few of the challenges that should be taken into account are listed as follows

- Low range IR sensors may not be an answer for long range signaling system. We may resort to ultrasound or radar techniques for big scale set-ups.
- Next is the influence of stray signals that may alter the reading of sensor receptors and lead to conveying false information to the microcontroller

Safety has to be absolutely made sure that no compromise is being made on safety issues, i.e. a secondary stand-by set-up that can switch over from automated to manual mode, should be provided in case of sensor or circuit malfunctions so that vehicular crowd does not go beyond control.

As part of future advancements, the traffic check post may be connected by wireless transmitters by which the crossings ahead may be an anticipation of the traffic that is approaching. This may be achieved the connecting the sensor network with GPS connectivity and shortwave radio transmission signals. This will act as a feed forward system making the signaling system even more smooth and congestion free.

REFERENCES

[1]. R. Dhakad and M. Jain, "GPS based road traffic congestion reportingsystem," 2014 IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, 2014, pp. 1-6. doi: 10.1109/ICCIC.2014.7238547

[2] Mangesh Jadhawar1, Gauri Kandepalli2, Ashlesha Kohade3, Rajkumar Komati4 1,2,3,4MIT College of Engineering, ENTC Department, Pune.

[3] Haran P C and Suriyanarayani R (2012), Embedded System Based Automobile Accident Prevention, Proc. of the Intl. Conf. on Advances in Computer Science and Electronics Engineering.

[4] Saiba P A, Afeefa M U, Aruna T S, Clincy Jose, & Radhika V M (2017). Density Based Traffic Signal System using PIC Microcontroller. International Journal of Computer Trends and Technology (IJCTT). ISSN: 2231-2803. Volume 47. Number 1.

[5] Er. Pramod Sharma, Akanksha Mishra, & Kaumudi Singh (2015). Density Based Intelligent Traffic Control System Using IR Sensors. International Journal of Scientific Research (IJSR). ISSN No 2277 – 817. Volume: 4. Issue: 5.

[6] D. D. Pukale1, Palak Chauhan, Adhikari Siddhi Satish, Preeti Nawal, & Neha Kumari (2016). Density Based Traffic Control System Using Video Processing. Imperial Journal of Interdisciplinary Research (IJIR). Vol-2. Issue-6, 2016 ISSN: 2454-1362.

[7]. Er. Pramod Sharma, Akanksha Mishra, & Kaumudi Singh (2015). Density Based Intelligent Traffic Control System Using IR Sensors. International Journal of Scientific Research (IJSR). ISSN No 2277 – 817. Volume: 4. Issue: 5.

[8]. Sakshi Pandey, Mittal Jethva, Sailee Jangam, & Yashika Panwar (2017). Density based Traffic Control System. International Journal for Scientific Research & Development (IJSRD). ISSN: 2321-0613, Vol. 4, Issue 11.

[9]. Ghazal, B., ElKhatib, K., Chahine, K., & Kherfan, M. (2016). Smart traffic light control system. 2016 Third International Conference on Electrical, Electronics, Computer Engineering and Their Applications (EECEA). doi:10.1109/eecea.2016.7470780.

[10]. K.Vidhya,& A.Bazila Banu (2014). Density Based Traffic Signal System. International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET). ISSN: 2319 – 8753. Volume 3. Special Issue 3.

[11]. D. D. Pukale1, Palak Chauhan, Adhikari Siddhi Satish, Preeti Nawal, & Neha Kumari (2016). Density Based Traffic Control System Using Video Processing. Imperial Journal of Interdisciplinary Research (IJIR). Vol-2. Issue-6, 2016 ISSN: 2454-1362.

[12]. K. Lalitha1, & M. Pounambal (2018). Density Based Traffic Management Using IoT. International Journal of Pure and Applied Mathematics. ISSN: 1314-3395. Volume 120 No. 6.15

[13]. Ashok. P.V, SivaSankari.S , Vignesh Mani, & Suresh Sankaranarayanan (2017). IoT Based Traffic Signalling System. International Journal of Applied Engineering Research. ISSN 0973-4562. Volume 12. pp. 8264-8269.

[14]. Basil, E., & Sawant, S. D. (2017). IoT based traffic light control system using Raspberry Pi. 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS). doi:10.1109/icecds.2017.8389604.

[15]. Badamasi, Y. A. (2014). The working principle of an Arduino. 2014 11th International Conference on Electronics, Computer and Computation (ICECCO). doi:10.1109/icecco.2014.6997578.

[16]. Barrett, S. F. (2012). Arduino Microcontroller: Processing for Everyone! Second Edition. Synthesis Lectures on Digital Circuits and Systems, 7(2),1371. doi:10.2200/s00421ed1v01y201205dcs038

[17]. Mohammed Fayaz, Pooja K, Pranitha P Reddy, & Swathi T (2019). Density based Traffic Control System with Ambulance Detection. International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181. Volume 7, Issue 08.

A Major Project report on

A Preliminary Study On Projection Denoising For Low-Dose CT Imaging

Submitted in partial fulfilment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

IN

Electronics &Communication Engineering

Submitted

By

V.Vamsee Mohan (17K81A04C0)

K.Varun (17K81A0483)

K.Rakesh(18K85A0423)

Under the Guidance of

Mrs.Sameen Azhar

M.Tech. (Asst.Professor)

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

ST MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

(An UGC Autonomous Institute)

Dhulapally, Secunderabad-500 100.

NBA &NAAC A+ Accredited 2020-2021

St.MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the major-project work entitled **"A Preliminary Study On Projection Denoising For Low-Dose CT Imaging"** is a bonafide work carried out by**V.Vamsee Mohan(17K81A04C0), K.Varun (17K81A0483), K.Rakesh(18K85A0423)**in partial fulfilment of the requirements for the degree of **Bachelor of Technology in Electronics & Communication Engineering** by the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2020-21.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree.

INTERNALGUIDE

Mrs.Sameen Azhar
AssociateProfessor,
DepartmentofECE.

HEAD ofTHE
DEPARTMENT, ECE

Dr.B.Hari Krishna,
Professor.

EXTERNAL EXAMINER

Date: 18-02-2021

This is to certify that V.Vamsee Mohan(17K81A04C0), K.Varun(17K81A0483), K.Rakesh(18K85A0423) of B.Tech – IV Year, ECE department of St. Martin's Engineering College, Kompally, Secunderabad, have completed one month Internship Program at Lasya IT Solution Pvt. Ltd, Kompally(V). During the period, they have successfully completed Major Project, "APRELIMINARY STUDY ON PROJECTION DENOISING FOR LOW-DOSE CT IMAGING" at our Development Center.

We wish them every success in life.

Oruganti Venkat
Director
Trainings& Placements
Lasya IT Solution Pvt. Ltd.

Date: 18-02-2021

Lasya Infotech, Behind Cine Planet, Kompally, Email :contact@lasyainfotech.com, ov@lasyainfotech.comWebsiet : www.lasyainfotech.com | contact: 73300666884/83/82/81/86

ACKNOWLEDGEMENT

I sincerely thank the management of our college **St.MARTIN'S ENGINEERING COLLEGE** for providing required facilities during our project work.

Ideriveourgreatpleasureinexpressingoursinceregratitudetoourprincipal**Dr.P.Santosh Kumar Patra** for his timely suggestions, which helped us to complete the project work successfully.

Itistheveryauspiciousmomenttoexpressourgratitudeto**Dr.B.HariKrishna**,Headofthe Department,ECEforhisconsistentencouragementduringtheprogressofthisproject.Wetake it as a privilege to thank our project coordinator **Mr.K.Karthik**Assistant Professor, department ofECE fortheideasthatledtocompletetheprojectworkandwealsothankthemfortheircontinuous guidance, support and unfailing patience, throughout the course of thiswork.

We sincerely thank our project Internal guide **Mrs.Sameen Azhar,** Associate Professor**,** Department of ECE for guidance and encouragement in carrying out this project work.

DECLARATION

We, the students of '**Bachelor of Technology in Department of Electronics and Communication Engineering**', session: 2017 – 2021, **St. Martin's Engineering College**, hereby declare that the project work entitled '**A Preliminary Study On Projection Denoising For Low-Dose CT Imaging**' is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

V.VamseeMohan
17K81A04C0

K.Varun
17K81A0483

K.Rakesh
18K85A0423

ABSTRACT

Recently, low-dose computed tomography (CT) was considered by many researchers to be a good solution to reduce radiation risks of patients. However, lowering X-ray tube current will make reconstructed images quality be significantly degraded. To improve image quality, in this paper, we proposed a modified dual-domain U-net (MDD-U-net) that combines the projection domain and image domain losses. The proposed MDD-U-net can effectively suppress the projection domain noise and reduce the error in reconstructed images. The simulation experiment results showed that the proposed method effectively reduce the noise in the low-dose CT images while preserved images details information.

TABLES OF CONTENT

LIST OF FIGURES

CHAPTER-1 INTRODUCTION

CHAPTER-3 PROJECT DESCRIPTION

CHAPTER-5 RESULT ANALYSIS

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE OF THEPROJECT

Recently, low-dose computed tomography (CT) was considered by many researchers to be a good solution to reduce radiation risks of patients. However, lowering X-ray tube current will make reconstructed images quality be significantly degraded. To improve image quality, in this paper, we proposed a modified dual-domain U-net (MDD-U-net) that combines the projection domain and image domain losses. The proposed MDD-U-net can effectively suppress the projection domain noise and reduce the error in reconstructed images. The simulation experiment results showed that the proposed method effectively reduce the noise in the low-dose CT images while preserved images details information.

1.2 Introduction:

MATLAB is a programming language developed by MathWorks. It started out as a matrix programming language where linear algebra programming was simple. It can be run both under interactive sessions and as a batch job. This tutorial gives you aggressively a gentle introduction of MATLAB programming language. It is designed to give students fluency in MATLAB programming language. Problem-based MATLAB examples have been given in simple and easy way to make your learning fast and effective.

This tutorial has been prepared for the beginners to help them understand basic to advanced functionality of MATLAB. After completing this tutorial you will find yourself at a moderate level of expertise in using MATLAB from where you can take yourself to next levels.

We assume you have a little knowledge of any computer programming and understand concepts like variables, constants, expression, statements, etc. If you have done programming in any other high-level programming language like C, C++ or Java, then it will be very much beneficial and learning MATLAB will be like a fun for you.

MATLAB (matrix laboratory) is a fourth-generation high-level programming language and interactive environment for numerical computation, visualization and programming.

It allows matrix manipulations; plotting of functions and data; implementation of algorithms; creation of user interfaces; interfacing with programs written in other languages, including C, C++, Java, and FORTRAN; analyze data; develop algorithms; and create models and applications.

It has numerous built-in commands and math functions that help you in mathematical calculations, generating plots, and performing numerical methods.

MATLAB is used in every facet of computational mathematics. Following are some commonly used mathematical calculations where it is used most commonly −

- Dealing with Matrices and Arrays
- 2-D and 3-D Plotting and graphics
- Linear Algebra
- Algebraic Equations

- Non-linear Functions
- Statistics
- Data Analysis
- Calculus and Differential Equations
- Numerical Calculations
- Integration
- Transforms
- Curve Fitting
- Various other special functions

Features of MATLAB

Following are the basic features of MATLAB −

It is a high-level language for numerical computation, visualization and application development.

It also provides an interactive environment for iterative exploration, design and problem solving.

It provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations.

It provides built-in graphics for visualizing data and tools for creating custom plots.

MATLAB's programming interface gives development tools for improving code quality maintainability and maximizing performance.

It provides tools for building applications with custom graphical interfaces.

It provides functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET and Microsoft Excel.

Uses of MATLAB

MATLAB is widely used as a computational tool in science and engineering encompassing the fields of physics, chemistry, math and all engineering streams. It is used in a range of applications including −

- Signal Processing and Communications
- Image and Video Processing
- Control Systems
- Test and Measurement
- Computational Finance
- Computational Biology

In recent decades, computed tomography (CT) has been widely used in the diagnosis and intervention tasks of modern hospitals and clinics. Given the potential radiation risk of CT scans, there is growing concern about the harm caused by radiation dose to humans. In order to reduce the radiation dose received by patients and the risk of related diseases, low-dose CT has gradually become the focus of researchers in the field of CT imaging. At present, there are two main ways to reduce the radiation dose. The first method is to reduce time of CT scans, and the second method is to lower the operating current of the X-ray source tube. Because the second way is simple and effective, it has become one of the most popular methods to reduce theradiation dose. However, directly lowering the tube current will severely degrade the images quality due to the excessive quantum noise caused by an insufficient number of photons in the projections, thereby affecting the doctor’s diagnosis of thedisease. Many different methods have been proposed to improve the quality of low-dose CT imaging. In 2008, Sidkyet al.[1] integrated the total variation (TV) regularization into traditional projection onto convex sets (POCS) and proposed the adaptive-steepest-descent POCS (ASD-POCS) algorithm to suppress images noise. In 2012, Chen et al.[2] proposed a processing scheme called 'artifact suppressed large-scale nonlocal means' for suppressing both noise and artifact in low-dose CT images. In 2016, Chen et al.[3] proposed an improved context-based block matching and three-dimension (BM3D) filtering method to reduce the noise of low-dose CT and improve the images quality. However, the above methods belong to the general denoising method, and using them may not be enough to suppress strong and widelydistributed noise in low-dose CTimages. In recent years, deep learning methods have been widely used in the field of low-dose CT images noise reduction. In 2017, Chen et al.[4] proposed residual encoder-decoder convolutional neural network (RED-CNN) that combines the auto-encoder, the deconvolution network, and shortcut connections for improving low-dose CT imaging quality. In 2017, Kang et al.[5] proposed a deep convolution neural network using a directional wavelet transform to extract the directional component of noise to suppress low-dose CT images noise. In 2018, Lee et al.[6] proposed a deep-neuralnetwork for interpolating sparsely-sampled sinogram to reduce low-dose CT images noise. Although this method have shown impressive performance capabilities, there are still errors between the processed projection and the real projection. During the image reconstruction process, these errors will be magnified, resulting in blurred details in the reconstructed images. In this paper, we proposed a modified dual-domain U-net (MDD-U-net) to preprocess the projection data to effectively suppress the projection domain noise. In addition, in order to reduce the error of the low-dose CT reconstructed images, a filtered back-projection (FBP) module with backpropagation capability is added into the network structure to implement image domain constraints. The trained network model can be directly applied to low-dose CT projection denoising, which greatly reduces the calculation time and improving the quality of low-dose CT reconstructed images.

II. METHODS

A. Noise Model X-ray photon statistical noise and electronic background noise are the two main types of noise in the process of lowdose CT imaging. They approximately obey the Poisson distribution and Gaussian distribution, respectively.[7] Zheng et al.conducted a detailed study for the noise in the low-dose CT images and showed that the Poisson-Gaussian mixed noise is closerto the real noise in the low-dose CT images[8]. Therefore, in order to simulate the low-dose CT images more exactly, the Poisson-Gaussian mixed noise was used to 223 2020 3rd International Conference on Artificial Intelligence and Big Data

 simulate the real low-dose CT projection data acquisition process in this paper. The specific noise model is shown as follow: (1) where is the path of the X-ray, parameter is used to simulate the conversion gain of x-ray photons to electrons, is the initial incident intensity of X-rays, and is the variance of Gaussian noise. In our study, was set to 10000 and was set to 25. *B. Network Architecture* Recently, the U-net proposed by Ronnebergeret al.[9] not only have achieved good performance on many different kinds of biomedical segmentation applications but also have been applied in CT image denoising tasks[10].

Inspired by the successful application of the U-net, we proposed a novel low-dose CT images denoising network that can be directly applied to the projection denoising of low-dose CT. In order to further improve the performance of the network in suppressing noise of the projection domain, we proposed a MDD-U-net to more effectively extract projection domain features.

The proposed network structure consists of an encoding structure (left side) and a decoding structure (right side) is illustrated in Fig. 1. In contrast to the U-net architecture composed of four convolutional layers and four deconvolution layers in the original paper, the MDD-U-net encoding structure contains five convolution layers (CONV) and the decoding structure contains five deconvolution layers (DECONV). All layers have a 3×3 kernel, and each layer is followed by a rectified linear unit (ReLU) to prevent vanish of gradient and accelerate the convergence rate. Downsampling operate in the encoding structure and up-sampling operate in the decoding structure are implemented by 2×2 max-pooling operation and 2×2 up-convolution operation, respectively. The number of feature channels doubles with each down-sampling operation, however, the number of feature channels is halved after each up-sampling operation. We added five shortcut connections to the network to reduce the loss of pixels caused by each convolution. Due to sinogram edge vanish after each convolution, we used the zero-padding scheme to maintain sinogram size. The proposed network reduces noise by learning an endto-end mapping from containing noise low-dose CT sinogram (input) to its corresponding noise-free normal-dose CT sinogram (label). The deconvolution layers use the projection features extracted from the input by the convolution layers to construct the predicted output.

C. Objective Function

For the purpose of improving imaging quality, different loss functions were used in the proposed method. We reduced the projection domain noise by minimizing the projection domain loss. Let be a low-dose CT sinogram as an input and be a normal-dose CT sinogram as a label. The MDD-U-net obtains the predicted output by learning the end-to-end mapping from to . And we minimized image domain loss to further reduce the error of low-dose CT reconstruction image. To achieve the constraint on the image domain, the ASTRA-toolbox[11] was used for implementing the FBP module with the ability for error backward-propagation from image to sinogram. Let be a FBP image of the predicted output and be a FBP image of the label . Given the good noise suppression capability of the TV regularization term, in this work, TV loss is added to the objective function. The reduction of TV loss can decrease the difference between adjacent pixel values in the low-dose CT images[12]. TV regularization model can describe the sparsity of gradient magnitude images. TV of an image is defined asfollow[13]:

(2) where represents discrete directional gradient operators. Specifically, the objective function is expressed as follow: (3)

where represents the loss of the projection domain, represents the loss of the image domain, both are based on norm loss function, and represent the parameters of the projection domain loss and the image domain loss, respectively. represents the TV loss.

.

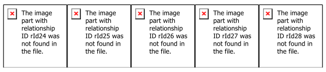
D. Network Training

In this study, network training is implemented by minimizing the objective function (2). Parameter $\lambda 1$ and $\lambda 2$ values were set to 1 and 100, respectively. The $\lambda 3$ was set to 0.5. We used the adaptive momentum estimator (Adam) optimizer[14] to optimize our proposed network. It is a gradient-based optimizer that considers first-order moment estimation and second-order moment estimation of a gradient. We set the first momentum to 0.9 and the second momentum to 0.999 and ε to 10-8 , respectively, the same setting as in the original paper. We set the initial learning rate to 0.0001. We used the Tensorflow deep learning framework to train the proposed network on a GPU (NVIDIA TITAN V, 12GB). The proposed network model was trained for approximately 17 hours.

Matlab can be used as a tool for simulating various electrical networks but the recent developments in matlab make it a very competitive tool for Artificial Intelligence, Robotics, Image processing, Wireless communication, Machine learning, Data analytics and whatnot. Though its mostly used by circuit branches and mechanical in the engineering domain to solve a basic set of problems its application is vast. It is a tool that enables computation, programming and graphically visualizing the results.

The basic data element of MATLAB as the name suggests is the Matrix or an array. MATLAB toolboxes are professionally built and enable you to turn your imaginations into reality. MATLAB programming is quite similar to C programming and just requires a little brush up of your basic programming skills to start working with.

1.3 Block diagram of embeddedsystem:



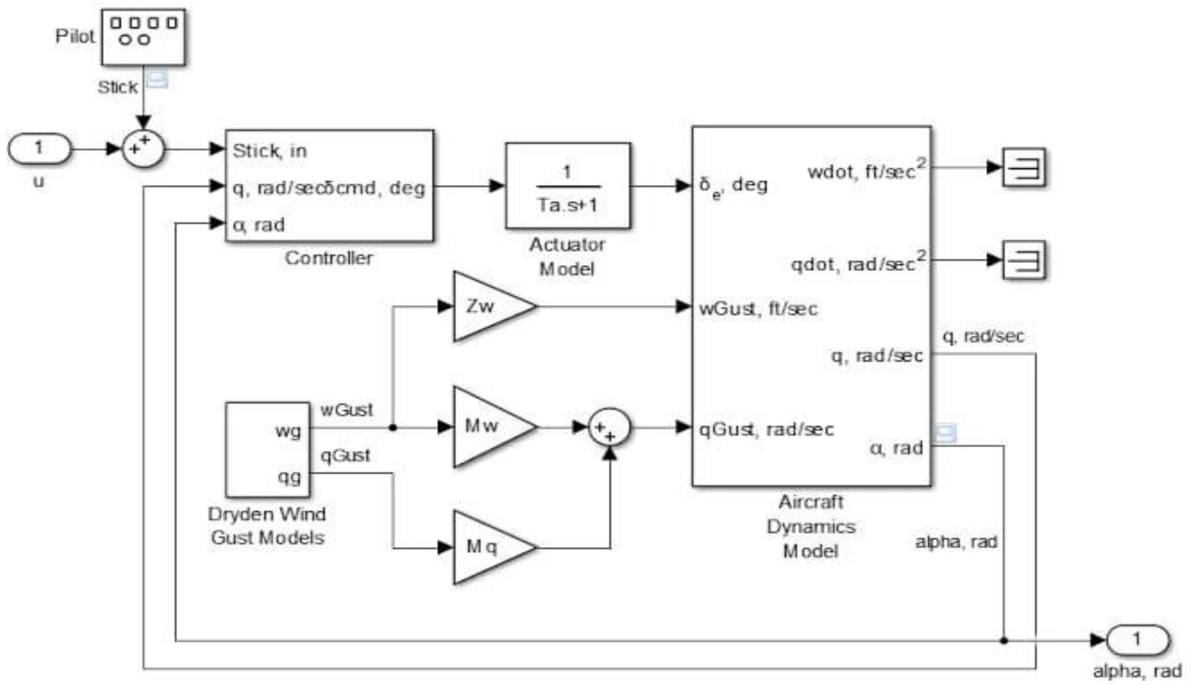


Fig.1.1: Block diagram of Matlab

1.4 Applications of Matlab:

- Math and computation.
- Algorithm development.
- Modeling, simulation, and prototyping.
- Data analysis, exploration, and visualization.
- Scientific and engineering graphics.
- **Application** development, including Graphical User Interface building.

1.5 Matlab Parts

The **Matlab** screen mainly include 4 **parts**:Command Window, Workspace window,Command History window and Current Directory window.

1.5.1 Command window

The MATLAB Command Window is the main window where you type commands directly to the MATLAB interpreter. The MATLAB Editor Window is a simple text editor where you can load, edit and save complete MATLAB programs. The Editor window also has a menu command (Debug/Run) which allows you to submit the program to the command window. The MATLAB Help Window gives you access to a great deal of useful information about the MATLAB language and MATLAB computing environment. It also has a number of example programs and tutorials.

The Command Window is always open. To restore the Command Window to the default location, go to the Home tab, and in the Environment section, click Layout. Then, select from one of the default layout options. To bring focus to the Command Window from another tool such as the Editor, type commandwindow .

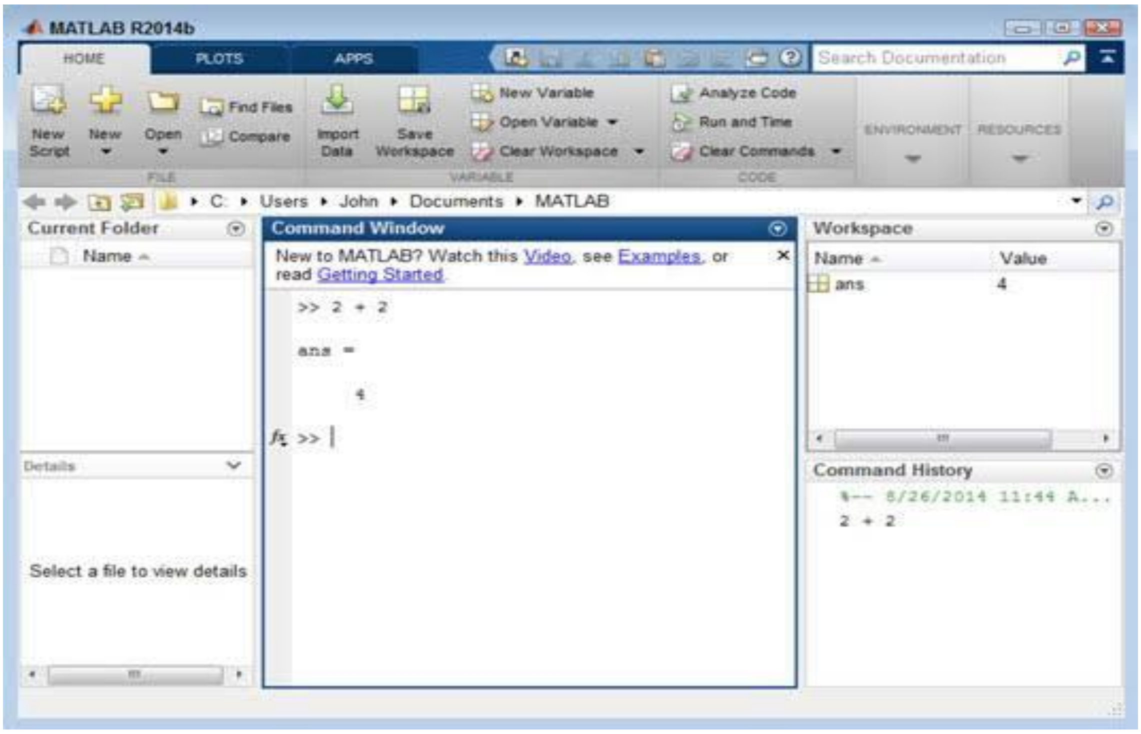


Fig:1.2: Command window

1.5.2 Workspace window

The workspace contains variables that you create or import into MATLAB from data files or other programs. You can view and edit the contents of the workspace in the Workspace browser or in the Command Window. For more information, see Create and Edit Variables. Workspace variables do not persist after you exit MATLAB.

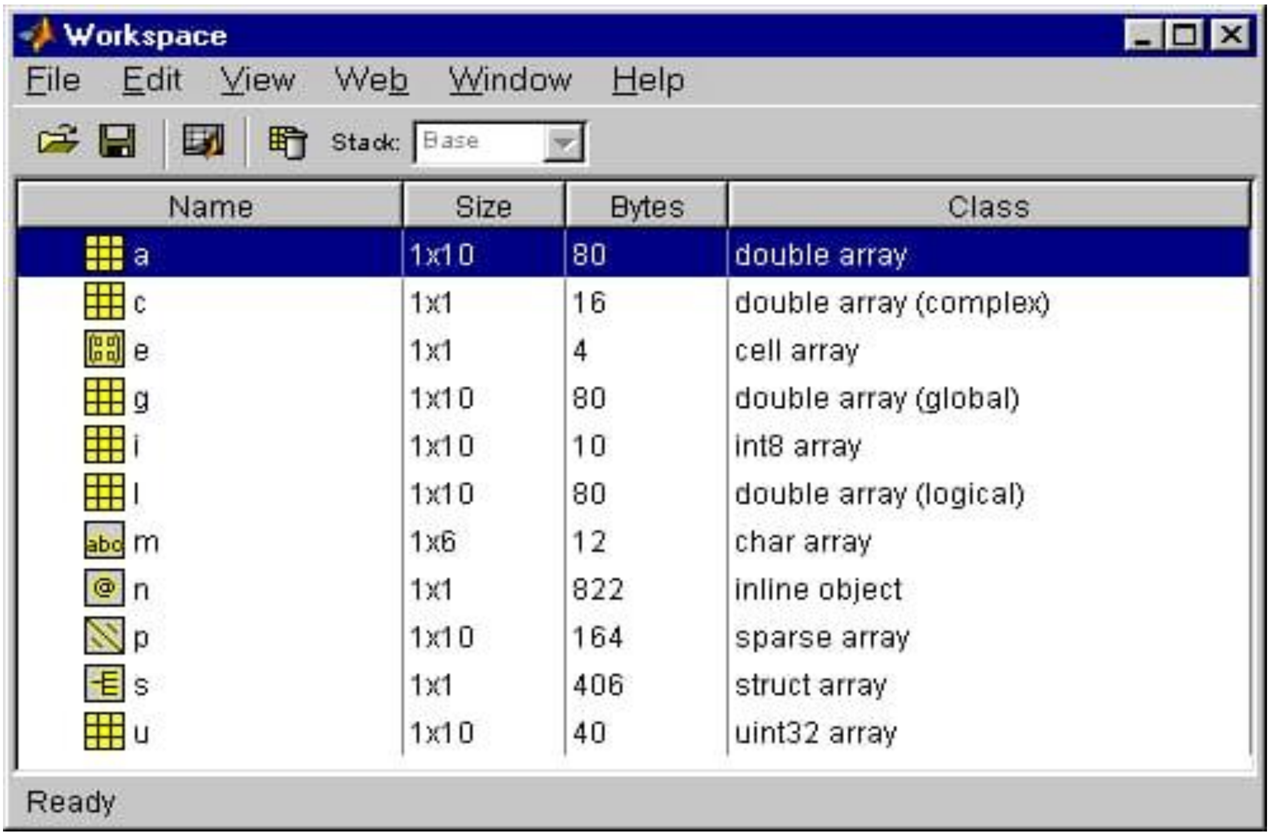


Fig:1.3:Work space in matlab

1.5.3 Command History Window

To open the **Command History** window with all **history** showing, in the **Command** Window, press the Up Arrow key (↑) or enter commandhistory . To open the **Command History** window and display a specific statement, type any part of the statement at the prompt and then press the Up Arrow key.

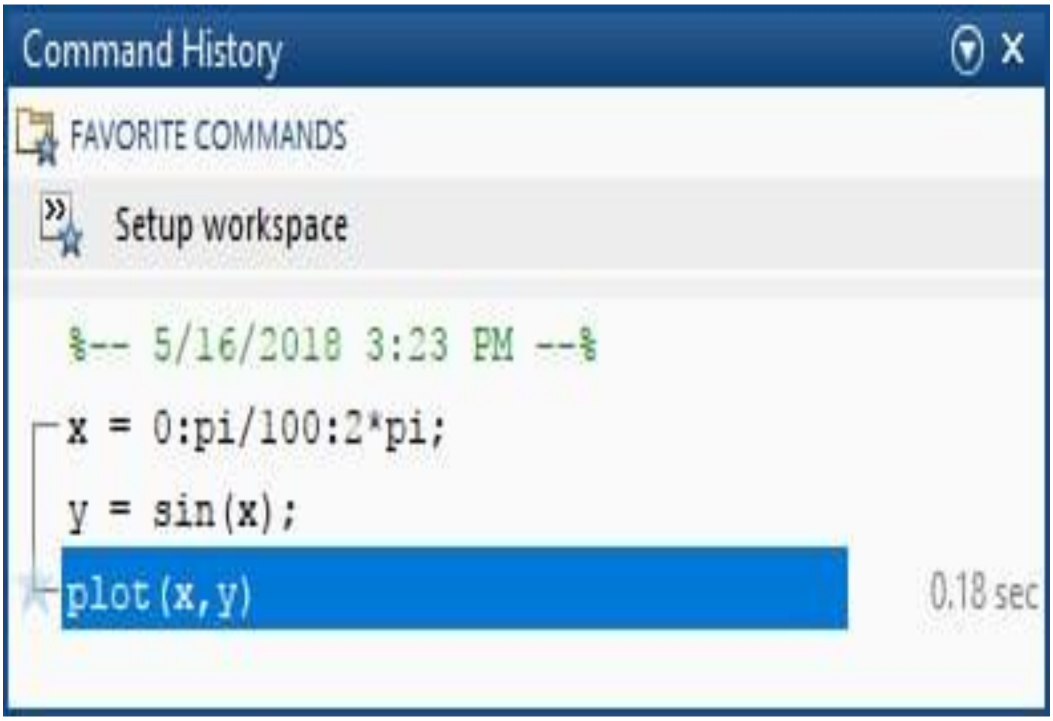


Fig:1.3:Command history window

.

1.5.4 Current Directory Window

To open the Current Folder browser if it is not currently visible, do one of the following: MATLAB Toolstrip: On the Home tab, in the Environment section, click Layout. Then, in the Show section, select Current Folder. MATLAB command prompt: Enter filebrowser .

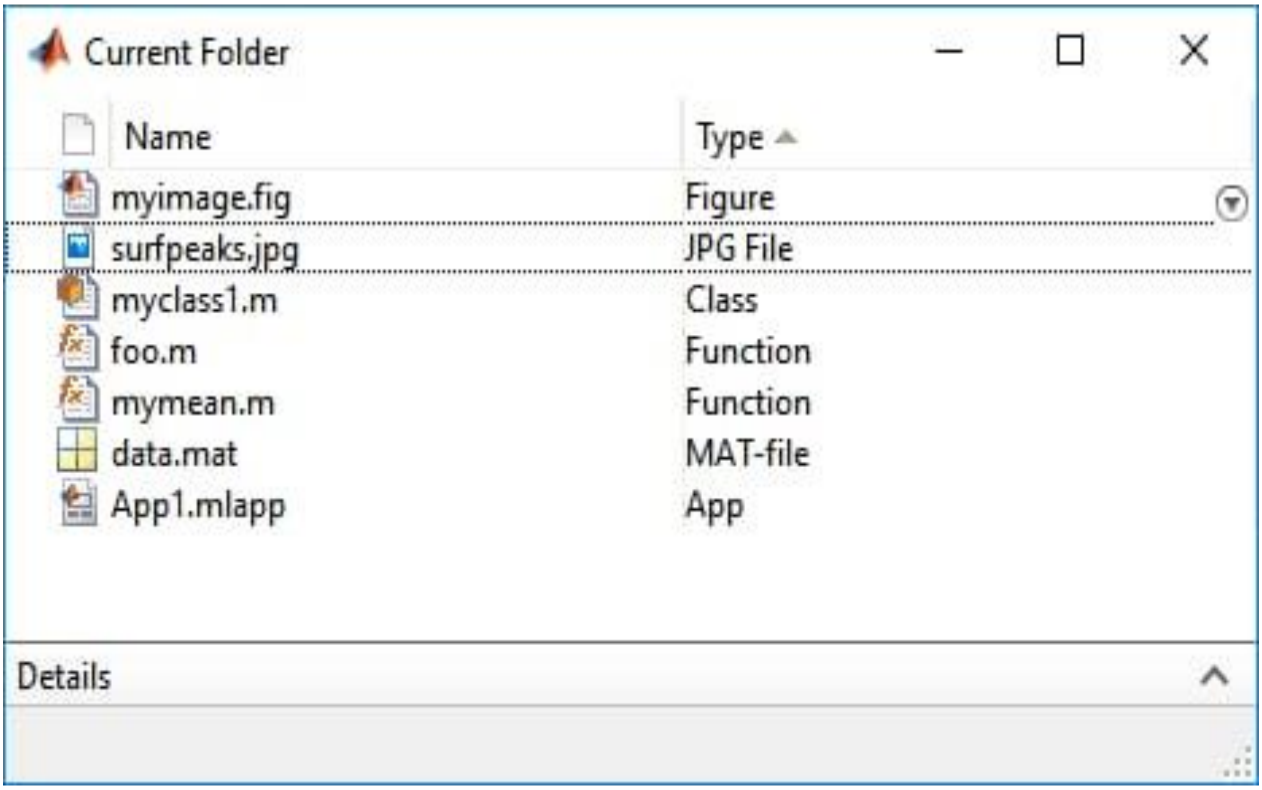


Fig:1.5: Command Directory Window

CHAPTER-2 LITERATURE SURVEY

2.1 Literature survey:

This literature review focused on the use of MATLAB, a computer algebra system, as a didactical tool for mathematics. It has also looked at some theories of learning related to the teaching and learning of mathematics with the computer technology. The review aimed at summarizing the different ways in which the software has been used as a tool for visualization, computation, problem-solving, fostering positive attitudes and math confidence, increasing motivation & math performance, and projects. It has also included the proposals in which the software has been recommended to be used. It is our hope that this review could be useful for those interested in integrating MATLAB in mathematics curriculum. Keywords: CAS, MATLAB, Teaching & Learning, Mathematics, Literature Review. Introduction The Computer Algebra System, popularly known as CAS, is a modern computing technology that provides great learning and teaching opportunities to both students as well as instructors. MATLAB is one of the most popular CAS programs that is widely used in mathematics classrooms especially at the university level. As more and more universities are adopting its use in education, it is necessary to study and analyze how the software could be used in teaching and learning of mathematics. In this literature, focus has been given on how MATLAB has been used in problem solving, computation, and visualization of mathematics through mathematical explorations etc. It has been shown to be a technology that could be used for developing mathematical thinking, concepts and skills, and a source of motivating students for learning mathematics. It is also a very useful tool in demonstrating connections in various underlying mathematical concepts in the teaching and learning processes. 23 24 M. Abdul Majid et al. The modern computer algebra systems are cognitive tools, Heid (2001). According to Pea (1987, p. 91), a cognitive technology is “any medium that helps transcend the limitations of the mind in thinking, learning, and problem-solving activities”. Cognitive tools are both mental and computational devices that support, guide, and extend the thinking processes of their users, Derry (1990). They are used as a didactical tool to enhance students' mathematical learning and to provide them with essential career-related skills. Both advantages and disadvantages of using computer technology are reported in the literature. Those who oppose the didactical use of computer tools voice the declining mental and manual mathematical skills of students. However, those who support its use believe that these tools would not only enhance the presentation of mathematics but also could provide the necessary skills needed in the 21st century. Undoubtedly, technology has become ubiquitous in mathematics education. According to National Council of Teachers’ of Mathematics (2006), “Technology is essential in teaching and learning mathematics; it influences the mathematics that is taught and enhances students' learning”. Does the use of computer tools or technologies per se enhance students learning? What factors influence classroom-based math learning? The positive change in the teaching and learning lies in several factors. “Education technology is neither inherently effective nor inherently ineffective; instead, its degree of effectiveness depends upon the congruence among the goals of instruction, characteristics of the learners, design of the software, and educator training and decision-making, among other factors", Schneider (2000). However, for some, its use is still questionable. The research question which will guide this review is the following: "In what ways MATLAB has been used in the literature for teaching and learning of mathematics in the university education?" MATLAB MATLAB, the product of Mathworks Company, is the general purpose computing software. It contains a vast range of specialized toolboxes and also works as the computer algebra system through its symbolic math toolbox. This toolbox performs symbolic algebraic/mathematical manipulative operations with a lot of built-in interactivity. MATLAB undoubtedly is popular among computer and multi-disciplinary scientists, engineers and particularly with experts in the area of computational mathematics. It integrates numerical, symbolic and the state-of-the-art graphic visualization capabilities with quite intuitive computer programming environment. Methodology The systematically classified and integrated methods were used to summarize the relevant articles and presented them in a narrative manner. A literature review for the publications between 1988 and 2012 was carried out from the sources such as mathwork.com, ERIC, IEEE xplore, ProQuest, Academic Search, Springer Link, Taylor and Francis, Science Direct, Proceedings for the conferences, the Google and the Google scholar search. The keywords used for the search were "CAS and mathematics education", and "CAS/MATLAB as a teaching & learning tool for math", "CAS/MATLAB as a visualization tool for math", "CAS/MATLAB as problem-solving tool in math" etc. Our primary focus was the empirical studies based on classroom and computer-lab where MATLAB was used as a pedagogical tool for mathematics. MATLAB as a Teaching and Learning Tool for Mathematics... 25 Theories of Learning and CAS It is important to know how theories may help in technology integration for mathematics instruction. The theoretical and conceptual basis of using computer technology in the teaching and learning of mathematics can be linked with the Constructivist Theory. According to this theory, a learner cognitively constructs a mathematical concept through the active involvement in the processes of experimenting, conjecturing, proving, and applying. Using CAS, students actively construct and acquire knowledge in mathematics rather than being passive receivers of knowledge. There are a number of other theories derived from the cognitive psychology relating to cognitive and technology-aided cognitive processes such as Cognitive Load Theory, Cognitive Theory for Multimedia Learning and Distributed Cognition Theory. According to Sweller (2004), Cognitive Load Theory suggests the learning activities to be aligned with the human cognitive structure. The structure of human cognitive architecture, while not known precisely, is discernible through the results of experimental research. According to Hutchins (1990), “Distributed cognition is a branch of cognitive science that proposes cognition and knowledge are not confined to an individual; rather, it is distributed across objects, individuals, artefacts, and tools in the environment.” According to Cognitive Multimedia Theory, the information presented through the integration of limited visual and auditory channels will be filtered, selected, organized, and integrated by the learner actively, figure 1, Mayer & Moreno (2003). However, an understanding of the human cognitive processes in a mathematics teaching and learning environment which involves human computer interaction still needs much exploration. Figure 1: Cognitive Multimedia Theory by Mayer & Moreno, (2003). Duval (1993) suggested an absolute necessity of the application of different semiotic representations for a solid understanding of mathematical concepts. This can be achieved in a technology-enriched teaching and learning environment. The theoretical framework in figure 2 is based on the notion of 'procept' given by Gray and Tall (1994) and the semiotic representation of mathematical concepts by Duval (1993). This theoretical framework was used by Abdul Majid et al (2012). 26 M. Abdul Majid et al. Figure 2: A theoretical framework for teaching and learning of mathematics with CAS. Literature Review Summary The literature review search has not revealed any study which focused specifically on the review of literature on MATLAB as a teaching and learning tool for math. The selected articles and papers that were reviewed, compiled and summarized are as follows: Proposals to use MATLAB as a Teaching and Learning Tool for Mathematics A number of articles have been written proposing the use of MATLAB in teaching and learning of mathematics, Kazimovich et al (2012), Tahir et al (2010), Chaamwe (2010), Mckenzie (2008), Friedrich et al (2008), Stanoyevitch (2005), Shi (2004), Wu (2004), Colgan (2000), Kawski (2003), Dunn et al (2002), Pester et al (2001), Markanda (1995) and Katende in his unpublished paper. Kazimovich et al (2012) illustrated the capabilities of the applications of symbolic computation techniques of MATLAB for engineering students. The capabilities of MuPAD and its combination with MATLAB were described for writing formulas, factorizing an expression, solving equations, solving systems of equations and differential equations, finding derivatives and antiderivatives as well as plotting functions. In addition, some of the useful functions like computing Fourier transforms and their inverses have also been given. The paper also provided sketches of the transfer between MuPAD and MATLAB. Tahir et al (2010) proposed the integration of MATLAB into science, technology and engineering disciplines as a teaching and learning tool for undergraduate studies to the Ministry of Science and Technology, Iraq. The graphical user interface programs created for Math 1 and Math 2 courses were proposed to be used in teaching linear algebra, graphing, Laplace Transform, differential and integral calculus courses. The proposed programs were aimed at motivating students and demonstrating the relationships between pure mathematical concepts to the real world. The authors planned to develop a supplemental booklet for using computer programs in tutorial sessions. The authors also recommended the use of the software throughout the science and engineering institutions in the country and also emphasized the need for providing necessary training on the use of MATLAB to the teaching faculty. Chaamwe (2010) provided a few illustrations for sketching mathematical graphs of functions using MATLAB. The study highlighted some general benefits of integrating the software. Students motivation to learning can be enhanced, they can work collaboratively in teams. It could provide a problem-centered approach in teaching and learning. Their skills could be honed. The software can also be used for demonstrating a connection between the theory and MATLAB as a Teaching and Learning Tool for Mathematics... 27 the practice and as well as students actively participate in the learning process. The study also highlighted some relevant studies on the role of Information and Computer Technology. In this report by Mckenzie (2008), MATLAB has been recommended to be used as a new pedagogical model for illustrating and reinforcing mathematical concepts to students at the Oregon Institute of Technology. This recommendation was for a replacing the existing system of math instruction. The purpose was to provide interactive demonstrations and visualizations into mathematics classes. Primarily, two objectives were aimed at, firstly, to demonstrate the link between the use of MATLAB to the enhancement of the understanding of the math and secondly, to equip students with the necessary professional software skills in their careers. To make the efficient use of the new model, the report also recommended creating a website for MATLAB, integrating teaching of the software as a one-credit course into the math curriculum and also providing the information on the website about the connection of mathematical contents to students' specific engineering fields. The report concluded that students' motivation, interest, their appreciation to the software, communications and critical thinking would be positive. Computer tools could also be used to overcome math-phobia as termed by Pappert (1993). Friedrich et al (2008) described the cognitive role of computer technologies such as MATLAB in the development of students' mathematical reasoning, logic and problem solving skills. The graphing capabilities of various CAS programs such as Maxima, Scilab, Micro Worlds, Dynamic and Interactive Geometry as well as MATLAB have been given. The study concluded that the issue of declining interest in math and in its allied disciplines such as engineering is due to the lack of well-educated technical experts in cognitive technologies. In this paper, the author Stanoyevitch (2005) presented the benefits and versatility of introducing MATLAB into university mathematics programs and how the software could be incorporated. A detailed description was given on how the implementation of the software could enhance mathematics courses both from the academic aspect as well as opening up the employment prospects for graduates. Students could achieve a higher level of learning in courses like abstract algebra with the incorporation of MATLAB into it. The issues such as licensing the software, using it as an isolated technology, from a minimal lower level to an advance levels, have also been raised in the study. Some illustrations of its use such as logistics of air shipping and computation of worst-case-scenario number of a directed network were provided in detail. The study by Shi (2004) was aimed at implementing MATLAB as a tool into teaching and learning of calculus for students majoring in Physics. It was also suggested dividing students into normal and advanced levels based upon their aptitudes as well as including the group work for them. The need for the change in the pedagogy was felt due to a host of problems related to students learning of mathematics. The focus for the normal group of students was given for learning the basic underlying concepts and principles of mathematics with less emphasis on the subject’s rigor. The advanced group of students would have the normal course without providing them the details of the course putting greater self-responsibility for learning. A number of illustrations of the visualization of the graphs of the mathematical functions were given with explanation of how they could be used in teaching. The software tool was also recommeded to be used for enhancing the key concepts such as tangent line of a curve at some point, geometric interpretation of the definite integral as well as some examples of mathematical modelling. The study also highlighted the importance of group work and suggested to form a group of 4 to 5 students to cultivate necessary communication and collaborative skills among them. The assessment scheme to the new approach was suggested to be in place. The author believed that 28 M. Abdul Majid et al. the proposed new approach would be more student-centered and it would provide necessary important skills to students. In this study, Wu (2004) proposed MATLAB to be used as a teaching and learning tool at the School of Science, Beijing Institute of Technology. The study aimed at improving the teaching and learning of linear algebra and to equip students with modern computing technology. The proposal also suggested establishing the website for the course. It was also highlighted that the software could be used as a data visualization tool to promote deeper understanding of the concepts. Dunn et al (2002) used MATLAB for teaching and learning of introductory calculus at the University of Southern Queensland, Australia. Five MATLAB based graphical interface programs were used to demonstrate the mathematical concepts such as Newton's Method, differentiation and integration. The programs present some unusual and important cases of the calculus concepts. The authors argued that MATLAB could be incorporated into learning of mathematics in three ways, namely, firstly as a numerical and graphical tool, secondly as an aid to learning of ideas and concepts and thirdly to show the procedure or concept in a more sophisticated way. In this work, the authors have used MATLAB using the third approach. The main focus of the use of these programs was to demonstrate mathematical concepts rather than doing a particular task. The study also suggested that the computer technology should be used a supplement to the traditional teaching and learning. In this proposal, Kawski (2003) compared the arguments of the integration of MATLAB with that of the other CAS programs such as Maple and Mathematica in the first year college mathematics. The author comparatively put forward the argument that integrating a CAS in mathematics curriculum is much easier than MATLAB due to the pedagogical and planning related challenges with the later. The study further stated that the traditional calculus is still an "algebra course" whereas MATLAB provides numeric computing in which "almost everything is a matrix". Katende in his unpublished paper reviewed the role of MATLAB and other symbolic algebra systems for engineering mathematics with some illustrations. The study outlined some of the potentials and challenges in undergraduate engineering mathematics. Illustrations such as expanding cos(7x), differentiating sin(E^x(x^2+1)) and integration problems were given and how quickly the algebra systems provided output to such complicated computations. Some examples of 2-D and 3-D graphical visualization were also provided in the paper. The study concluded highlighting the need to change the curriculum or the approach to teaching engineering mathematics. MATLAB Web Server was proposed to be used in this study by Pester et al (2001) for electrical engineering and mathematical modeling. The interactive software could be used to save students' time and to increase their motivation. It could be used for simulation and visualization of 2 and 3-dimension graphs. A few illustrations of how the software could be used were given as well as with some advantages and disadvantage of using web-based MATLAB. Solutions to some network related problems were also proposed. Markanda (1995) described how the students' version of MATLAB could be used to compare several direct, elimination, and iterative methods for solving efficiently and with less calculations systems of n equations in n variables. The illustrations of Jacobi and Gauss-Seidel methods using the software were given in the paper. In a study by Dios et al (2012), MATLAB and Learning Management System "Moodle" were introduced in the traditional teaching and learning of graduate engineering mathematics course. The aim was to give students a better understanding of the course contents. The paper MATLAB as a Teaching and Learning Tool for Mathematics... 29 discussed the mechanism of the new structure of the course including the web-based assessment. The Learning Management System "Moodle" was created as a virtual learning environment to work as a pedagogical model. The online quizzes were used using WIRIS, an editor to write symbols and formulas. Kovacheva (2008) outlined the main features of many popular mathematical application packages including MATLAB. The capabilities of these programs for research as well as for teaching were described briefly. MATLAB as a Tool for Graphing and Visualization of Mathematical Data and Concepts One of the greatest benefits of computer technology is the accessibility of the mathematical concepts through visualization. MATLAB is one of the best known software which contains mathematical data visualization capabilities, Palais (1999). MATLAB as a visualization tool has been used in the courses like linear algebra, calculus and differential equations etc. Szurley (2007) used MATLAB as a visualization tool for applications of linear algebra course. Two examples of population modeling and one example of Genetics were used in this study. The aim of the study was to provide students with the graphing capabilities for illustrating the insights into the material. In an extension of the study, the author created some graphical user interfaces in MATLAB to address students' unsatisfactory responses to the use of the software in an earlier study. The aim was to provide interactivity and visuals for presenting mathematical concepts. Palmer (2008) also aimed at visualization of linear algebra concepts through the writing and the development of simple animation programs in MATLAB. The author claimed that creating animation programs is possible with little knowledge of linear algebra, no computer programming experience and without MATLAB's familiarity. The study concluded that students' learning of matrices enhanced, however, no students' feedback was given. In a collaborative project between Swedish and British instructors (two mathematicians and one didactic expert), MATLAB was used to teach undergraduate engineering mathematics students, Burton et al (2004). The aims of the project were to support students' visual and graphical understanding of mathematics and their access to the numerical computation of the course. Individual interviews were conducted to get the feedback from students. Their overall feelings were mixed on their experience of using the software. To achieve effective software integration, the study has also provided suggestions such as ensuring clarity in learning goals, adequate number of MATLAB classes, and the preparation of a manual to support MATLAB programming by providing necessary reference material on the web. Students expressed their dissatisfaction with their experiences of using MATLAB as there was too much elapsed time between two MATLAB sessions consequently students had to learn the material over and over again. The study concluded that the integration of the software should be planned thoroughly with enough software documentation. In several studies, MATLAB was integrated into the mathematics curriculum as an add-on teaching and learning tool. In a slightly different approach, it was integrated to an interactive "Adaptive Book" system with the aim to demonstrate the linear algebra concepts and as well as the course's algorithms, Gunawardena and Jain (2002). The concepts were presented both in the mathematics classroom/laboratory followed by the software demonstrations. In this study, MATLAB was also used as a drill and practice tool in which a combination of text and visuals were presented in multiple forms. The project such as Singular Value Decomposition for performing data compression was based on practical applications. The study described linear algebra as an ideal course for integrating technology to enhance the teaching and learning processes. It was concluded that the software 30 M. Abdul Majid et al. like MATLAB could provide a good platform for not only visualization but also for numerical computation. Stewart et al (2005) used MATLAB to assess the process of students' instrumentation of technology in the teaching of mathematics courses. The authors' called the instrumentation as a student's transformation and adaptation of computer technology to the assigned mathematical task and "it is through the actions and decisions that the student makes that the CAS tool or artefact is transformed into an instrument". The case study involved first and second year students enrolled in scientific and engineering disciplines at the University of Canterbury. The second year students used MATLAB in tutorials and computer laboratory sessions for an advance linear algebra course. The aim of the software use was to solve problems but the main focus was to help students with understanding the steps involved in the solution. The data were collected through a questionnaire in which questions were given relating to students' attitudes to the use of computers in learning math; use of computers in linear algebra course and some openended questions on the value of computers for the course. Students' feedback to the benefits of using MATLAB for visualization, conceptualization, and problem-solving was positive. In the analysis, the study reported three students' types regarding the instrumentation of computer technology for university mathematics courses. These types were firstly, those who opposed to the computer work, secondly those who were happy to use computer with a limited instrumentation, and thirdly those who exhibited too much reliance on the use of computer for mathematics. The study concluded with the recommendation of the Leigh-Lancaster's view of achieving the integration of computer technology through the process of having congruency between curriculum, pedagogy and assessment making it clear that CAS must be specified in the curriculum, teaching as well as assessment. The assessment in mathematics must be in tandem with its teaching and learning. In a study, Colgan (2000) described the integration of MATLAB to the teaching of traditional engineering mathematics course at the University of South Australia. The course included elementary linear algebra, the differential and the integral calculus. The need for the software integration was felt due to the decline in the mathematical skills of students entering into the engineering disciplines. MATLAB was chosen because of two reasons, firstly it was extensively used in some of the engineering programs at the university and secondly it was believed to increase students' employability after graduation. A MATLAB guide specifically covering the course contents was prepared to incorporate the software into the teaching. This was considered a major component of integrating software into the syllabus. The software use was demonstrated in the classroom lectures. The graphing capability of the MATLAB was shown to the students illustrating graphs of functions including features such as zoom on and multiple commands for plotting functions etc. MATLAB was used to improve students understanding of the mathematical concepts such as limits, continuity, differentiability, multiple representations of the Mean Value Theorem, Riemann Sums, and finding the volumes of revolution etc. The methodologies of teaching the course were redesigned keeping in view of the qualities of the engineering programs mentioned by the professional engineering body of Australia. The syllabus was rewritten outlining clearly the delivery and the assessment criteria. Weekly five contact hours were allocated for the course in which three hours of regular lecture, one hour of tutorial and one hour was dedicated for working with the software. Apart from the software guide, the syllabus also included the engineering applied examples relevant to the course. The emphasis throughout the process of technology integration was to enhance students' understanding of mathematics. The paper also provided some of the illustration of the engineering examples such as the classical problem of Snell's Law and the minimization of the MATLAB as a Teaching and Learning Tool for Mathematics... 31 time in the refraction of light. To inculcate collaborative spirit among students, they worked in groups of four on MATLAB sessions and the engineering projects/assignments. A part of the group's work was performing the analysis of the engineering project and presenting that in the form of a report. The projects involved the knowledge of the engineering as well as the use of the mathematical formulas. The study concluded that the course was considered to be the most successful without providing any data. The author highlighted the need for striking a good balance between the amounts of time to be spent on MATLAB to the course requirement. Schlatter (1999) described the visualization benefits of using MATLAB in a multivariable calculus for various concepts such as three-dimensional surfaces, contours plots, gradient fields and parametric curves etc. The software, to some extent, was also used for symbolic integration and for setting up integrals. The author developed two- and three- dimensional interactive packages in MATLAB and they were used; firstly to help students in visualization of mathematical objects and surfaces and; secondly to familiarize them with the use of MATLAB. The study concluded that the students could cover a more variety of integrals when the software would be used as a pedagogical tool. Apart from linear algebra and multivariable calculus courses, the software was also used for teaching concepts such as series convergence and numerical computation and visualization of power series approximations. Moore (1988) used MATLAB to teach "Series and Differential Equations" at the Grinnell College. The goal was to provide graphic visualization through the experimentation of various functions, and numeric computations for Taylor polynomials approximations, partial sums of power series, definite integrals and differential equations. Students' used MATLAB-based programs. The study concluded that the software tools not only widen students' problem-solving ability, insight, flexibility but also their ability to use them. However, the study has not provided any students' feedback. In addition to the above studies, the software was also reported to be effective tool in the visualization of mathematical concepts, (Abdul Majid et al 2012, Puhak 2011, Liang 2009, Stewart 2005, Henderson 2002, Smits 1992). MATLAB as a Computational Tool Another extremely powerful feature of MATLAB is its interactive numerical computation environment which is based on the matrix structure. Computational methods and skills of students in topics such as matrix computation, numerical differentiation and integration, and differential equations etc. can be enhanced with the use of the software. Particularly, engineering students need this skill to solve many engineering problems, Canfield (2012). Addabbo (2010) presented the approach that was used for teaching Introduction to MATLAB course for engineering students of Vaughn College who had completed calculus as one of the requirements. The course was divided into two parts; part 1 was related with programming techniques based on calculus problems, and the part 2 was the study of the numerical solutions of differential equations studied through linear and non-linear physical models. This approach was developed to demonstrate the relationship between the pre-requisite courses such as physics, statics, dynamics, writing and the vibration with that of calculus as well as to encourage higher order learning. Several projects such as matrix operations, summing series, Newton's Root Finding Method, Weather Data, and the numerical integration were given to students. Students were expected to do projects in groups of maximum three students. Upon completion, they were required to give a presentation and their performance was assessed on a set criteria. However, the study has not reported the effects of the approach and no conclusions were drawn. In a similar study, MATLAB was used for teaching computational modules at Southern Queensland University, Tonkes (2005). The learning model was developed and integrated for the first year mathematics courses. The aim of the study was to primarily reinforce traditional mathematical 32 M. Abdul Majid et al. learning, teaching concepts in numerical math and as well as for engaging students in computer aided problem solving. The model development included an interactive workbook, web-based solutions, Graphical User Interfaces and programs developed in MATLAB. The study highlighted some of the challenges encountered in the integration of the software in the mathematics curriculum. They were time allocation for the software use, students' efforts in learning it, numeric nature of the software, its syntax, and differences in the meaning of some of the signs in computers and math as well as the attitudes of teachers using it. Students progressed through their use of the software in stages through the single line commands environment to launching MATLAB's GUIs and then to developing their own MATLAB code. All assignments/exercises and experiments were performed in pairs. The outcome of the study was assessed through students' feedback survey and the programs downloads log. The students demonstrated an enhanced understanding of math, interaction with the tutors increased, and students' retention improved. The model had been reportedly successful from all stake-holders; students, lecturers, and tutors. Linear Algebra course was among the most popular courses in which MATLAB was used as reinforcement to classrooms lecture. At the University of Sydney, MATLAB was used as a supplement for teaching an undergraduate linear algebra course, Henderson (2002). The aims were to familiarize students with basic MATLAB commands so as to improve their conceptual understanding of the course. In this study, students used MATLAB to solve 4 to 5 problems in a week over one full semester. The use was mainly towards mathematical exploration, visualization of graphs and multiple representations of mathematical concepts. The study recommended maintaining an appropriate balance between the use of the software for learning mathematics and the traditional learning of mathematics. The data were collected through a questionnaire which contained both open-ended and closed-ended questions. The feedback to students' attitudes toward mathematics have also been discussed. The study reported positive and negative views toward the use of the software for learning mathematics. The software had also been used to study some mathematical functions such as Weirstrass function and to study numerical approximation of ʌ, Feng (2011). In this study, the author has also used the software for testing the convergence of an infinite series using ratio test, root test and the integral test. Liang 2009, Burton 2004, Gunawardana 2002, Smits 1992, & Moore 1988 have also reported the benefits of using MATLAB as a computational tool. MATLAB as a Tool for Improving Math Performance/Math Achievement The positive impact of computer technology on students' learning and performance has also been examined and reported, Abdul Majid et al (2012), Strayhorn (2006), Kulik & Kulik (1991). Abdul Majid et al (2012) have used MATLAB as a supplement for teaching Integral Calculus course to 77 engineering students at the University of Ha'il. The study aimed at firstly, to enhance students' conceptual understanding, their performance in mathematics and secondly, to foster positive attitudes towards mathematics and attitudes toward the computer technology. The case study involved an experimental group and a control group. Among the important objectives of incorporating the software were to address the declining mathematical skills of students needed for their core engineering courses as well as establishing a smooth transition to the upper level courses. The software had been used for a number of instructional objectives such as for visualization of graphs of mathematical functions, for identifying and exploring mathematical patterns in evaluating definite integrals, for multiple representations of a mathematical concept, and as well as for computation. The study reported the positive effects on students' math achievement compared to the performance of control group of students in the final examination. MATLAB as a Teaching and Learning Tool for Mathematics... 33 The conclusions drawn from the study were that the integration of MATLAB could be effective through the appropriate instructional design combined with students using the software in groups of 2-3. The integration of MATLAB has also been reportedly positive on students' performance in mathematics with "Moodle" an e-learning management system. The MATLAB was integrated with Moodle as a supplement to the classroom lecture for teaching calculus to undergraduate students in Geodesy at the Vienna University of Technology, Judex et al (2008). The blended teaching and learning approach was used to address the issue of students' lack of interest in mathematics and their weak background in the subject. To integrate MATLAB into the course, the software was used as an instructional tool and its basic course was included into the exercises of the mathematics courses. Specifically, the software was used for the large geometric mathematical drawings used in the geo science. Over 90 examples were developed and were made available via "Moodle" for students. The blended approach proved to be successful for the presentation of the heavy calculus course contents in a short time without sacrificing the true comprehension of the course. It enhanced the quality of lectures, improved students average grades as well as their programming skills. The integration of MATLAB would not only provide students a familiarity with modern learning tool but also an added advantage in their careers. The linear algebra course is among the most popular courses in which MATLAB has been integrated in its teaching. Smits et al (1992) described the impact of the use of MATLAB on mathematics achievement of the first year linear algebra students at the Eindhoven University of Technology. A detailed overview was given on how the course contents were presented in lectures and tutorials. In their study, the first nine weeks of the term, linear algebra was taught in which a 3-hour lecture with a tutorial were given. During the second half of the term, the course primarily focused on eigenvalues and eigenvectors followed by the last 3-weeks of the course on ordinary differential equations. MATLAB was used for time-consuming calculations, for motivating students, the interpretation of the row-echelon form, for doing more practice with the problems, for illustrating some aspects of the theory, geometric interpretation of the matrix of a linear transformation, and for introducing unknown techniques or concepts. Students played an active role in learning, conjecturing through guided experiments, and acquainted themselves with the real use of the mathematical techniques. An analysis of the results revealed that students' performance was better in the first term compared to the second term due to the complex nature of the concepts dealt with in the second. The study described the positive effects of the use of the software on students' performance as well as on equipping them with modern computing skills. MATLAB as a Tool for Fostering Positive Attitudes Towards Math, Math Confidence, and Enhancing Motivation Towards Learning Math Attitudes are defined as positive or negative emotional dispositions, McLeod, D. B. (1992) & Aiken (2000). Students' attitudes toward mathematics are complex and a number of measurement tools and constructs are reported in the literature. According to Willis (1995), "Human-computer interaction is a complex phenomenon and the attitudes and feelings involved with this relationship are difficult to identify" and the author has also reported the positive attitudes towards computer technology. This positive change in attitudes also reported to have motivated students in mastering the necessary computer skills quickly. Confidence in learning mathematics, mathematics anxiety, and perceived usefulness of mathematics as the most frequently studied and identified affective variables associated with student achievement in mathematics, Reyes (1984). One of the strategies indicated by Bretscher (1989) is the use of a computer lab with 34 M. Abdul Majid et al. tutors providing individual instruction and hands-on practice in concepts and skills. This strategy may help enhancing students' motivation and overcoming their math anxiety. The computer technology fosters positive attitudes towards mathematics, Lim et al (2009), Yashau (2006), Ocak (2006) &Gorev et al (2004), Cretchley et al (2000), increases students confidence in math, Brake (2007) and as well as it enhances their motivation towards learning math, Puhak (2011) & Carol (2012) . Figure 3: Ernest (2003) on failure and success cycles in math. Yushau (2006) used a blended learning approach in which MATLAB was used as a learning tool by 70 pre-calculus students at King Fahd University of Petroleum & Minerals. Students used MATLAB in weekly computer lab sessions. Some related online learning resources were also provided to students. The study was conducted over a semester and the pre- and post-study students' attitude towards mathematics and computers were examined. The concluding results reported a statistical significance of students' positive attitude towards mathematics and computers. The use of technology had reportedly positive effects on students' attitudes for a variety of courses such as single and multivariable calculus and courses which require solid background in mathematics such as engineering and technology. However, some studies such as the one on mathematical modeling by Lim et al (2009) reported no change in students' attitudes. In this study, the effects on students' attitudes were examined when applied mathematical modeling project was introduced with MATLAB. The project was concerning the volcanic ash fall after an eruption. The study involved third year students of Earth Science at the National Taiwan Normal University. It aimed at assessing students' attitudes towards mathematics in the pre- and post-modeling project. Some of the anticipated benefits of the projects were making the learning of the course interesting, providing a link between mathematical theory and the practical applications thereby reinforcing the bond with the real world, in addition to the fact that the computer modeling is cheaper as it does not require expensive equipment. The partial differential equation of the advection-dispersion model was used for the study. Due to the time constraints, students were involved only in the graphical interpretation of the solution by writing programs in MATLAB. The study highlighted that for practical purposes, the experts require the graphical interpretation of the solution rather than the methods which are employed. Students were also required to orally present and report their findings. Data were collected through preand post-study questionnaires and the post-study semi-structured individual interview with 7 students to assess their attitudes towards the project. Questionnaires consisted of four categories of questions relating to students beliefs, usefulness, enjoyment and anxiety. The study, however, concluded that there was no significant change in the attitudes of students due to a relatively short period of time. Nevertheless, students' enjoyment and interest towards the project and the use of MATLAB had reportedly positive. MATLAB as a Teaching and Learning Tool for Mathematics... 35 Computers have also been used for motivating students toward mathematics and their use reportedly had positive impact. Puhak (2011) used MATLAB as a teaching tool to enhance students' motivation and understanding of applied mathematics. The method used for the implementation of the software included choosing one traditional calculus concept, using its application in related areas, performing related calculations for enhanced understanding, and dealing with associated questions to further students comprehension. This method was illustrated with the example 'Center of Mass' with details of the role of MATLAB in the activity with the needed programming code and its visualization. The study has not used the formal measurement techniques, but recommended, to assess the effects of the use of the approach on students' learning. However, routine course evaluations and volunteered feedback were received. In another study by Carol (2012), MATLAB was used for students of Sports Technology who were enrolled into a first year mathematics course at Loughborough University. The aim of the study was to enhance students' motivation and engagement towards learning of mathematics through the introduction of appropriate sports-related group projects and applications. The need for the study was to address the problem of students' lack of engagement and motivation in the learning of mathematics, their poor attendance and high failure rate. An electronic voting system was also used to increase students' engagement and interaction with the lectures. The study highlighted the importance of the group projects and the use of the computer software for realistic problems for engineering students. The author compared the outcome of the initiative which included students' attendance record, motivation and their post-study results with the previous historical data. In the post-study analysis, the students' attendance had reportedly increased compared to the previous data. This increase was attributed to the change in the tutorial classes from the traditional math tutorial to the tutorial held in the computer laboratory where the active learning took place. The change was also evident in their quality of work and the feedback given through the questionnaire. Moreover, a noticeable jump of 33% in the students' success rate was reported. However, the study lacked the control group. Brake (2007) used MATLAB with the aim to increase the math self-confidence and the math ability of freshmen engineering technology students at the School of Engineering Technology, Eastern Michigan University. The software was used to see if this could improve students’ learning. The author believed that self-confident students could be transformed from novice learners to expert learners and MATLAB would increase that self-confidence in them in analyzing problems in mathematics. In this study, the researcher taught two sections of ET 100, Introduction to Engineering Technology, a freshmen engineering pre-requisite course for electronics engineering technology, mechanical engineering technology, and computer engineering technology. MATLAB was introduced into the curriculum, not just as a tool, but to be able to solve engineering problems. The number of students enrolled in this course was 38. A survey was carried out to obtain demographic information as well as data on students past mathematical training including how confident they felt in solving two rather difficult problems relating to finding roots of numbers and applying the math to a real world. A second survey was conducted at the end of the semester and a record of quiz, midterm and final grades on specific questions using MATLAB was kept. Since one of the goals was to turn the inexperienced students into expert learners, it was found that graphing helped many students learn to think more deeply about problems, however, further interpretation skills could not be enhanced due to the insufficient time of using the software. Nevertheless, this study touched upon a very important issue of incorporating computational package into the learning of engineering technology course which requires a thorough understanding of underlying mathematical principles. There are some issues unaddressed relating to the low class size of students, improper 36 M. Abdul Majid et al. brushing up of students skills in learning MATLAB, inadequate background of students in mathematics, and hence statistical insignificance in terms of its applicability to a wider audience and areas. There is a need for more research on the use of MATLAB as a tool to increase self confidence and math ability in freshmen engineering students which the author also felt strongly about. The skills acquired through the use of MATLAB in the teaching and learning will prepare future engineers to meet the demands of the 21st century. Ocak (2006) investigated the relationship between gender and students' attitudes and their prior experience of using MATLAB. Three calculus I classes comprising of 23 female and 37 male students from three different colleges were chosen for this study. All students of these classes had used MATLAB. Students' attitudes were assessed with an attitudes survey instrument and the prior experience with MATLAB was measured with a closed-ended response format questions. The author found a slightly positive correlation between students' attitudes and a prior experience on using the mathematical software program. The study concluded that there was no correlation between gender differences, students' attitude and the software program experience. The study also suggested that an assessment of the students' attitudes should be taken into consideration while using programs like MATLAB. Students' familiarity with features of the software program and their ease of use also affect their enthusiasm and curiosity. This study was conducted by Gorevet al (2004) to examine the effectiveness of the use of computer technologies in solving and doing mathematical problems by pre-service teachers. In addition, the assessments of teachers' skills in using these computer tools; their attitudes toward them; and their willingness to use these tools in their careers as teachers were also considered. MATLAB was also used among the many other computer software tools. Students were given assignment sheets which included both routine and non-routine questions without the use of computers. They were further encouraged to use the computerized tools to solve the problems which they could not solve by hand as well as to enhance their previous solutions. Students' solutions were analyzed using Van Hiele theory to understand the development of their insights. This theory defines insight as a combination of the ability to perform in a possible unfamiliar situation, to perform competently as per the requirement, and resolving the solution consciously and deliberately. Their responses were also collected through their opinions on the use of computer tools in their careers. A vast majority of students expressed strong support to the use of computerized tools for improving their understanding and as well as to find solutions to problems. Some qualitative responses were also provided in the study which, by and large, favored their strong willingness to the use of these tools in their careers despite some unanticipated technical problems that may come in the way. Cretchley et al (2000) investigated the effects of the use of MATLAB on the attitudes and learning of undergraduate mathematics students at the University of Southern Queensland. The study also highlighted the relationships between attitudinal effects and performance. The student body had comprised of a diverse group, on-campus and distance learners, registered in the first semester Algebra and Calculus course. MATLAB was used in lectures, tutorial and assignments. In order to develop mathematical concepts and necessary skills, tasks were designed and a MATLAB manual was written. The use of the software was demonstrated in lectures and oncampus students attended weekly two-hour small group tutorials conducted by tutors and the distance learners used scheduled tutorial on telephones, emails and newsgroups. To encourage and motivate students, small prize money was announced. The data collection instruments used were a pre-study diagnostic test, two post-study examinations, questionnaires related to MATLAB-based assignments, interviews and focus groups, and a pre- and post-study survey questionnaire. The study reported that students' appreciation of the use of the software gradually MATLAB as a Teaching and Learning Tool for Mathematics... 37 increased over time and their ease of use developed as they overcome frustration and difficulty. The assessment of students' knowledge was based on an open book examination without the use of the computers to maintain equity. Students were not examined on their knowledge of the software. Pre- and post-study tests and assignments were conducted to evaluate students' mathematics learning experience, their opinions, skills and interviews were also conducted through focus discussion groups and case studies. Students' opinions were sought on categories such as their learning of mathematics, their experience of using computer software in the learning of mathematics, and self-efficacy in math. MATLAB based tasks particularly explorations were reportedly successful and students reported their positive experiences such as visualizing and computing areas under curves, and the concept of definite integral by taking Riemann sums. The study also reported a wide acceptance, by both on-campus and external students, of the use of technology as an aid for learning and, in particular, of MATLAB as an appropriate software package. The authors reported a strong positive reaction to the experience of using MATLAB to support students learning of mathematics where inculcating motivation towards the study of mathematics many a times is a hardest job. Another motivating factor towards the use of the MATLAB was students' need and importance of its familiarity would help in their relevant engineering subjects. The software's graphical capacity, power and novelty were much appreciated by students. Other features students liked were MATLAB's speed in performing computation with ease, clarity in understanding through explorations, and use the software as a technology check for confirmation of their answers. MATLAB was favoured over graphing calculators due to the computer power with large screen and 3-D graphing capacity. Several others studies have also reported the benefits of using the software as a tool for increasing motivation (Pennel 2009, Ohrstrom 2005, Smits 1992) and fostering positive attitudes, Abdul Majid et al 2012. MATLAB as a Tool for Projects Han (2008) used MATLAB to teach elementary linear algebra to a class of 23 students at the Eastern Michigan University. Three software-based projects and some homework assignments problems were given to students. This innovation was used as a supplement to the traditional method of instruction and its aim was to examine students' perceptions of its usefulness in learning the course material. The approach provided an opportunity to students to solve realworld application problems. MATLAB was chosen to be the right choice over the other software due to its wide popularity in science and engineering, interactive calculations, visualizations capabilities together with its broader applications and audiences. A MATLAB manual was written and the tutorial sessions were held to familiarize students of its use. The three projects designed for the study were; firstly the application of Gaussian elimination, secondly inverse computing methods to solve large linear systems of equations, solving linear systems using Cramer's rule and thirdly the application of the singular value decomposition method. Students performed these projects individually and as well as in groups of 2 to 3. Students' feedback was received through a survey using yes/no types of questions and their opinions were also taken. More than 3/4th of the students expressed their overall satisfaction with the software and more than 57% thought that the software helped them in their learning. Chapin et al (2004) supplemented MATLAB with the standard calculus course for giving simple homework assignments to students at Ohio University. The authors outlined three categories, integrated use, project format and textbook add-on, of the uses of technology in the literature. They, however, implemented the use of the software for simple assignments for 38 M. Abdul Majid et al. making its application easy and practical for both students and instructors. The study highlighted the incorporation of the technology through firstly, use of simple and basic calculations; secondly, giving very clear instructions on assignments; and thirdly, giving very clear and concise technical information. In conclusion, the authors proposed the use of software in undergraduate mathematics courses for simple homework and included a sample MATLAB assignment. The study has not discussed its effects on students learning. Horwitz (2002) described collaborative engineering and science projects between mathematics and engineering departments for the purpose of enhancing first year differential and integral calculus courses. Students used MATLAB in projects individually and in teams and some real data were used. The first semester projects were Hydraulic Engineering, designing a pipeline with minimum cost, optimization of irrigation channel and the second semester projects were Exponential Decay, Skydiver free fall, Wrecking ball, and the Automobile velocity data. A students survey was carried out and they were asked to rate the projects on the scale of 1 to 5. Students reported some complains on the difficulty level of the projects. Some variation was observed in the comments of students belonging to engineering disciplines compared with the science majors. In addition, some students did take part in the projects but did not fully understand the mathematics involved. The study concluded firstly that the students' motivation for doing projects did not seem to have increased at the end of the study, secondly, they were simply involved in the project without themselves understanding the process, and thirdly, in depth project understanding could not be achieved due to the pressure of covering the regular syllabus of the courses. The authors suggested offering one extra credit hour course which could enhance the calculus program. The positive feedback was received from students as they enjoyed doing the projects with MATLAB and they saw a connection of mathematics with the real world. A thorough understanding of mathematical concepts with their applications in engineering and science proved useful to students. Szurley (2011) discussed the application of MATLAB for providing simulation to the projects involving solution of Differential Equations. The modeling projects involved the simulation of the industrial process of film casting and the conduction of heat. The projects were aimed at exposing students to real world applications, familiarizing them with MATLAB, demonstrating the need for numerical computation, and providing the necessary presentation skills. The study reported positive responses from students about real-world applications and in general considered a success. However, the use of MATLAB and the numerical schemes were not much favoured by them. In another study, the author incorporated MATLAB within mathematics courses at Francis Marion University. MATLAB was used for Calculus III projects involving numerical solutions and also for understanding the software's programming concepts. The projects involved L'Hopital Rule, improper integrals, sequences, power series, Taylor and Maclaurin series and as well as Linear Programming and Linear Algebra. The study concluded that students' responses were not very positive about the use of MATLAB partly due to nongraphical interfaces provided to them but in the long run this exposure was believed to help them. The future plan was set up to develop interactive graphical interfaces. Hailan et al (2012) highlighted various benefits of using MATLAB in applications of engineering as an innovative and scientific tool. The author illustrated with examples of how the software could be used as a tool for exploration and experimentation in courses like linear algebra and higher mathematics. The software's commands for finding reduced row echelon form and solving linear equations are used with examples including examples from statistics. Some applications in textile engineering and textile materials have also been given in this paper. MATLAB as a Teaching and Learning Tool for Mathematics... 39 Ohrstrom et al (2005) conducted a study at Chalmers University of Technology. An integrated approach to using MATLAB in chemistry and mathematics courses was introduced and its impact on students learning of solving chemical problems was evaluated. This curriculum reform approach was developed by experts in math, chemistry and education. Students used MATLAB-based assignments individually and in groups in the laboratory. The process of integration of the software into the course was achieved through the exchange of teachers and course material, schedules and planning together with the number of MATLABrelated projects and exercises/assignments. The projects were meant to teach mathematical concepts and skills followed by the development of MATLAB programs. The projects included were the introduction of the 3d geometry of molecules and mathematics, differential equations and kinetics, systems of nonlinear equations and equilibrium analysis as well as computer laboratory on acids and bases. Various pedagogical benefits of using the software were highlighted in the paper. The process of evaluation of its impact involved students, teachers and the educational consultants. The new reform resulted in a positive impact on students learning, and their motivation towards the subject increased. One of the main challenges highlighted by the study was maintaining a right balance in providing the necessary MATLAB skills to students for the future courses. MATLAB as a Tool for Business and Engineering-Related Applications The study conducted by Liang et al (2009) at the University of New Haven is an illustration and a summary of the use of MATLAB for teaching mathematics. The software use aimed at helping students' learning of calculus-based business mathematics and apply the calculus principles to find solutions to real quantitative business problems. The software was implemented to address students' varying levels of algebraic knowledge and skills and the challenge of completing the course in one semester. MATLAB was used to teach topics such as limits, derivatives, optimization, integration and their applications in business as well as the introduction to multivariate calculus and partial derivatives. MATLAB-aided solutions to the mathematical problems were presented step-by-step. The researchers concluded that the software use had helped students in focusing on key objectives of the course through the visualization of the examples and quick numerical computation. The study concluded with the statement that the application of computer technology would prepare students for modern day challenging careers in the workplace. An innovative approach, in the study by Pennell et al (2009), was introduced to the traditionally taught Differential Equations course for the students of engineering majors. The need for innovation was felt due to the fact that most engineering students were unaware of the relationship between mathematics courses and the their core engineering courses as well as the importance of mathematics in their careers as engineers. The new interdisciplinary project was designed in collaboration between mathematicians and engineers in an attempt to overcome students' lack of motivation towards mathematics and to address their poor retention of the material. The lack of engineering-orientation in the traditional teaching of a course like differential equations was highlighted. For instance the main focus was given on techniques for generating solution formulas. From the practical engineering application view point, the system under consideration was to understand how it responds to classes of inputs. The methods of incorporating engineering points of views of first and second order differential equations were given through the illustrations of circuit series, a mass-spring system and the Laplace transform. For students to investigate the behavior of the systems, authors prepared a brief theoretical 40 M. Abdul Majid et al. description together with the software tool developed using MATLAB and its allied toolboxes. MATLAB's GUI was prepared for students to interactively investigate various plots and behaviors of the systems. No feedback on the approach has been given. Naimark (2002) pointed the problems of a lack of connection in mathematics curriculum with the engineering disciplines which are applications oriented. The author alternatively presented the argument of what he called modernizing the mathematics courses by introducing modern computing-based applications by incorporating MATLAB. Mathematics syllabus should be viewed and taught as a central concept and a unified picture, "concepts as unifying vehicles". The illustrations of the use of MATLAB's Polyfit, MATLAB's view and Fourier Transforms were given. The linear Algebra course was taught using MATLAB in this study. Some of the software-based applications were the visualization of parametric equations, matrix of Linear Operator, matrix multiplication, Gaussian elimination method, rotations, projections, eigenvalues and eigenvectors etc. The study highlighted some pedagogical benefits of the new approach such as making the theory more transparent; students could concentrate on the structure of operations and as well as an additional method of memorization of concepts and procedures for students. The effects of the modern approach enhanced students understanding. The course objectives were achieved through the internalization of the central concept of the general linear operator. The use of the equality of inverse matrices, systems of linear equations, and the associative property of matrix multiplication improved students learning. Some of the challenges noted in the study were: the amount of time required for MATLAB-based exercises, instantaneous rectification of students' errors and the delay caused in teaching concepts like determinants and eigenvectors. In a study by Brown et al (2012), MATLAB has been used as a complementary tool to the traditional lecture for exploring mathematical concepts for the 1st year Electrical Engineering students at the University of Manchester. The introduction of the software was to provide mathematics contextually to see the connection with engineering concepts and to do calculations with varied parameters. Students used the software notebooks developed in Mupad-MATLAB. The study has not described the effects on students learning. MATLAB as a Tool for Improving Problem-Solving Techniques Talbert (2011) used MATLAB to teach problem-solving techniques to first-year Liberal Arts students who were enrolled in Calculus III at the Franklin College. Students spent most of the classtime doing hands-on lab exercises, watching video lectures and completing online tutorial work. The inverted classroom model was used in this study. This model provided students the ability to learn on their own and to apply new concepts. Students took some time adjusting to the innovative learning model, however, their feedback was reportedly positive. Legua et al (2001) in their study provided a comparison of the use of DERIVE, Mathematica and MATLAB to solve the line integral, an engineering mathematical problem. The aim of the study was to harmonize the use of the software with the teaching of mathematics at the Escuela Universitaria de IngenieraTcnica Industrial. A comparison of how the line integral concept could be explained using DERIVE, Mathematica and MATLAB was given with screenshots. The study concluded the DERIVE software as the most efficient of the other programs for evaluating line integrals. The study has not reported the effects of the use of this approach in the classroom teaching. Nyondo (1992) used MATLAB as a problem-solving tool for teaching some topics in linear algebra course at The Papua New Guinea University of Technology. The software was used to MATLAB as a Teaching and Learning Tool for Mathematics... 41 reinforce theoretical concepts taught in conventional lectures. The instructors had designed exercises and problems appropriate to the level of the course and suitable to the students' engineering specialization. The emphasis was also given on using MATLAB and mathematics as problem solving tools for engineering. Using the software, students practiced a variety of problems through experimentation. The study, however, has not reported the effects of the use of MATLAB on students' learning. Conclusion In this review, we have shown the different ways in which MATLAB was used as a teaching and learning tool for mathematics. It has been used as a tool for the visualization of the graphs of mathematical functions, computation, problem-solving, increasing motivation and selfconfidence and as well as fostering positive attitudes toward mathematics. However, a number of factors need to be taken into consideration concerning the technology integration in the teaching and learning of mathematics as highlighted by Schneider (2000). A vast majority of studies have used MATLAB as a supplement to the traditional teaching and learning of mathematics. References Abdul Majid, M., Huneiti, Z. A., Balachandran, W., & Al-Naafa, M. A. (2012). A study of the effects of using MATLAB as a pedagogical tool for engineering mathematics students, 15th International Conference on Interactive Collaborative Learning (ICL), 10.1109/ICL.2012.6402183, Villach, Austria. Addabbo, R. (2010). The integration of technology, writing and mathematics into an introductory MATLAB course for engineering students, www.asee.org/...2010/Integration-of-Technology-Writing-and-Math.pdf . Aiken, L. R. (2000). Psychological Testing and Assessment (10th ed.). Boston, MA: Allyn and Bacon. Brake, M. L. (2007). MATLAB as a Tool to Increase the Math Self-Confidence and the Math Ability of First-Year Engineering Technology Students, The Scholarship of Teaching and Learning at EMU: Vol. 1, Article 5. Bretscher, A. (1989). Success or failure: Variables affecting mathematics performance. Paper presented at the Annual Meeting of the National Association of Developmental Education, Cincinnati, OH. Brown, M. Steele, C., D., C., &Alsusa, E. (2012). The Mathexplore System: Student Exploration with a MATLAB-based system, http://sefi.htw-aalen.de/Seminars/Salamanca2012/16thSEFIMWGSeminar/ficheros/ lecturas/Documents_pdf/SoftwareDemonstrations/SEFIMWG12_Brown.pdf. Burton, L. Falk, L., &Jarner, S. (2004). Too much, too seldom, International Journal Mathematical Education in Science and Technology, 35:2, 219-226. Canfield, R. A. (2012). Teaching Computational Methods to 150+ Second-Year Engineering Students at Virginia Tech, http://www.mathworks.com/tagteam/71057_91984v00_teaching-computational-methods-to-studentsat-virginia-tech.pdf Chaamwe, N. (2010). Integrating ICTS in the teaching and learning of mathematics: an Overview, 2nd International Workshop on Education Technology and Computer Science, 10.1109/ETCS.2010.163. Chapin, S. & Young, T. (2004). A simple approach to technology in calculus, Proceedings of the Fifteenth Annual International Conference on Technology in Collegiate Mathematics, Edited by Corinna Mansfield, ISBN 0- 321-20570-7, Addison-Wesley Publishing Company, Inc. Colgan, L. (2000). MATLAB in first year engineering math, International Journal of Mathematical Education in Science and Technology, Vol. 31, No.1, 15-25, ISSN 0020-739x. Cretchely, P., Harman, C., Ellerton, N. & Fogarty, G. (2000). MATLAB in early undergraduate mathematics: An investigation into the effects of scientific software on learning, Mathematics Education Research Journal, Vol. 12, No. 3 219-233. 42 M. Abdul Majid et al. Derry.

CHAPTER-3 PROJECT

DESCRIPTION 3.1

SoftwareRequirement

- **Matlab**
- **CT Images**
- **Personal Computer**

3.2 BLOCK DIAGRAM OF PROJECT:

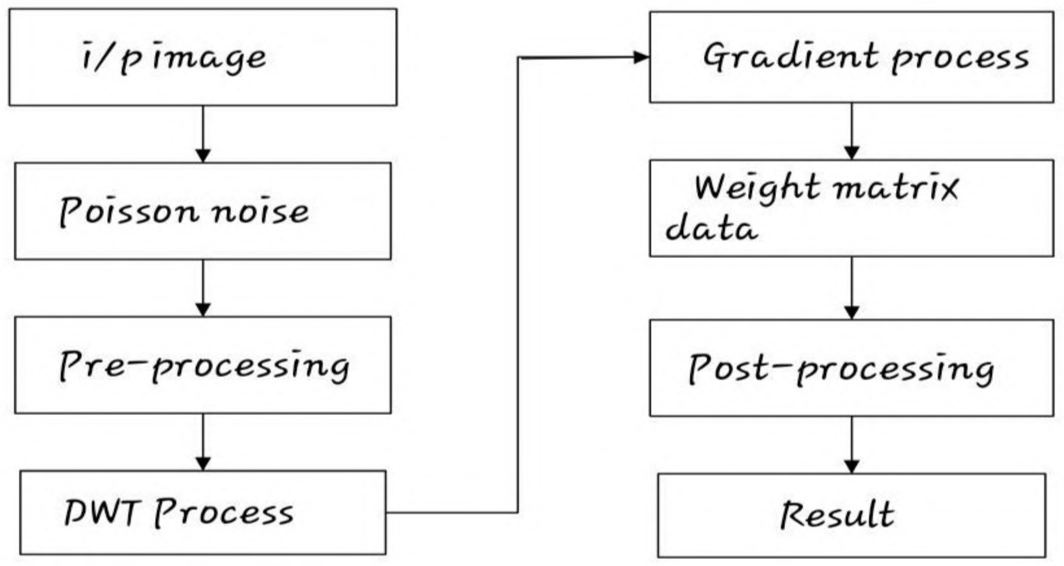


Fig.3.1: Block diagram of the project

Matlab

MATLAB is a programming language developed by MathWorks. It started out as a matrix programming language where linear algebra programming was simple. It can be run both under interactive sessions and as a batch job. This tutorial gives you aggressively a gentle introduction of MATLAB programming language. It is designed to give students fluency in MATLAB programming language. Problem-based MATLAB examples have been given in simple and easy way to make your learning fast and effective.

This tutorial has been prepared for the beginners to help them understand basic to advanced functionality of MATLAB. After completing this tutorial you will find yourself at a moderate level of expertise in using MATLAB from where you can take yourself to next levels.

We assume you have a little knowledge of any computer programming and understand concepts like variables, constants, expression, statements, etc. If you have done programming in any other high-level programming language like C, C++ or Java, then it will be very much beneficial and learning MATLAB will be like a fun for you.

MATLAB (matrix laboratory) is a fourth-generation high-level programming language and interactive environment for numerical computation, visualization and programming.

It allows matrix manipulations; plotting of functions and data; implementation of algorithms; creation of user interfaces; interfacing with programs written in other languages, including C, C++, Java, and FORTRAN; analyze data; develop algorithms; and create models and applications.

It has numerous built-in commands and math functions that help you in mathematical calculations, generating plots, and performing numerical methods.

MATLAB is used in every facet of computational mathematics. Following are some commonly used mathematical calculations where it is used most commonly.

Following are the basic features of MATLAB −

It is a high-level language for numerical computation, visualization and application development.

It also provides an interactive environment for iterative exploration, design and problem solving.

It provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations.

It provides built-in graphics for visualizing data and tools for creating custom plots.

MATLAB's programming interface gives development tools for improving code quality maintainability and maximizing performance.

It provides tools for building applications with custom graphical interfaces.

It provides functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET and Microsoft Excel.

The positive change in the teaching and learning lies in several factors. “Education technology is neither inherently effective nor inherently ineffective; instead, its degree of effectiveness depends upon the congruence among the goals of instruction, characteristics of the learners, design of the software, and educator training and decision-making, among other factors", Schneider (2000). However, for some, its use is still questionable. The research question which will guide this review is the following: "In what ways MATLAB has been used in the literature for teaching and learning of mathematics in the university education?" MATLAB MATLAB, the product of Mathworks Company, is the general purpose computing software. It contains a vast range of specialized toolboxes and also works as the computer algebra system through its symbolic math toolbox. This toolbox performs symbolic algebraic/mathematical manipulative operations with a lot of built-in interactivity. MATLAB undoubtedly is popular among computer and multi-disciplinary scientists, engineers and particularly with experts in the area of computational mathematics. It integrates numerical, symbolic and the state-of-the-art graphic visualization capabilities with quite intuitive computer programming environment. Methodology The systematically classified and integrated methods were used to summarize the relevant articles and presented them in a narrative manner. A literature review for the publications between 1988 and 2012 was carried out from the sources such as mathwork.com, ERIC, IEEE xplore, ProQuest, Academic Search, Springer Link, Taylor and Francis, Science Direct, Proceedings for the conferences, the Google and the Google scholar search. The keywords used for the search were "CAS and mathematics education", and "CAS/MATLAB as a teaching & learning tool for math", "CAS/MATLAB as a visualization tool for math", "CAS/MATLAB as problem-solving tool in math" etc. Our primary focus was the empirical studies based on classroom and computer-lab where MATLAB was used as a pedagogical tool for mathematics. MATLAB as a Teaching and Learning Tool for Mathematics... 25 Theories of Learning and CAS It is important to know how theories may help in technology integration for mathematics instruction. The theoretical and conceptual basis of using computer technology in the teaching and learning of mathematics can be linked with the Constructivist Theory. According to this theory, a learner cognitively constructs a mathematical concept through the active involvement in the processes of experimenting, conjecturing, proving, and applying. Using CAS, students actively construct and acquire knowledge in mathematics rather than being passive receivers of knowledge. There are a number of other theories derived from the cognitive psychology relating to cognitive and technology-aided cognitive processes such as Cognitive Load Theory, Cognitive Theory for Multimedia Learning and Distributed Cognition Theory. According to Sweller (2004), Cognitive Load Theory suggests the learning activities to be aligned with the human cognitive structure. The structure of human cognitive architecture, while not known precisely, is discernible through the results of experimental research. According to Hutchins (1990), "Distributed cognition is a branch of cognitive science that proposes cognition and knowledge are not confined to an individual; rather, it is distributed across objects, individuals, artefacts, and tools in the environment." According to Cognitive Multimedia Theory, the information presented through the integration of limited visual and auditory channels will be filtered, selected, organized, and integrated by the learner actively, figure 1, Mayer & Moreno (2003). However, an understanding of the human cognitive processes in a mathematics teaching and learning environment which involves human computer interaction still needs much exploration. Figure 1: Cognitive Multimedia Theory by Mayer & Moreno, (2003). Duval (1993) suggested an absolute necessity of the application of different semiotic representations for a solid understanding of mathematical concepts. This can be achieved in a technology-enriched teaching and learning environment. The theoretical framework in figure 2 is based on the notion of 'procept' given by Gray and Tall (1994) and the semiotic representation of mathematical concepts by Duval (1993). This theoretical framework was used by Abdul Majid et al (2012). 26 M. Abdul Majid et al. Figure 2: A theoretical framework for teaching and learning of mathematics with CAS. Literature Review Summary The literature review search has not revealed any study which focused specifically on the review of literature on MATLAB as a teaching and learning tool for math. The selected articles and papers that were reviewed, compiled and summarized are as follows: Proposals to use MATLAB as a Teaching and Learning Tool for Mathematics A number of articles have been written proposing the use of MATLAB in teaching and learning of mathematics, Kazimovich et al (2012), Tahir et al (2010), Chaamwe (2010), Mckenzie (2008), Friedrich et al (2008), Stanoyevitch (2005), Shi (2004), Wu (2004), Colgan (2000), Kawski (2003), Dunn et al (2002), Pester et al (2001), Markanda (1995) and Katende in his unpublished paper. Kazimovich et al (2012) illustrated the capabilities of the applications of symbolic computation techniques of MATLAB for engineering students. The capabilities of MuPAD and its combination with MATLAB were described for writing formulas, factorizing an expression, solving equations, solving systems of equations and differential equations, finding derivatives and antiderivatives as well as plotting functions. In addition, some of the useful functions like computing Fourier transforms and their inverses have also been given. The paper also provided sketches of the transfer between MuPAD and MATLAB. Tahir et al (2010) proposed the integration of MATLAB into science, technology and engineering disciplines as a teaching and learning tool for undergraduate studies to the Ministry of Science and Technology, Iraq. The graphical user interface programs created for Math 1 and Math 2 courses were proposed to be used in teaching linear algebra, graphing, Laplace Transform, differential and integral calculus courses. The proposed programs were aimed at motivating students and demonstrating the relationships between pure mathematical concepts to the real world. The authors planned to develop a supplemental booklet for using computer programs in tutorial sessions. The authors also recommended the use of the software throughout the science and engineering institutions in the country and also emphasized the need for providing necessary training on the use of MATLAB to the teaching faculty. Chaamwe (2010) provided a few illustrations for sketching mathematical graphs of functions using MATLAB. The study highlighted some general benefits of integrating the software. Students motivation to learning can be enhanced, they can work collaboratively in teams. It could provide a problem-centered approach in teaching and learning. Their skills could be honed. The software can also be used for demonstrating a connection between the theory and MATLAB as a Teaching and Learning Tool for Mathematics... 27 the practice and as well as students actively participate in the learning process. The study also highlighted some relevant studies on the role of Information and Computer Technology. In this report by Mckenzie (2008), MATLAB has been recommended to be used as a new pedagogical model for illustrating and reinforcing mathematical concepts to students at the Oregon Institute of Technology. This recommendation was for a replacing the existing system of math instruction. The purpose was to provide interactive demonstrations and visualizations into mathematics classes. Primarily, two objectives were aimed at, firstly, to demonstrate the link between the use of MATLAB to the enhancement of the understanding of the math and secondly, to equip students with the necessary professional software skills in their careers. To make the efficient use of the new model, the report also recommended creating a website for MATLAB, integrating teaching of the software as a one-credit course into the math curriculum and also providing the information on the website about the connection of mathematical contents to students' specific engineering fields. The report concluded that students' motivation, interest, their appreciation to the software, communications and critical thinking would be positive. Computer tools could also be used to overcome math-phobia as termed by Pappert (1993). Friedrich et al (2008) described the cognitive role of computer technologies such as MATLAB in the development of students' mathematical reasoning, logic and problem solving skills. The graphing capabilities of various CAS programs such as Maxima, Scilab, Micro Worlds, Dynamic and Interactive Geometry as well as MATLAB have been given. The study concluded that the issue of declining interest in math and in its allied disciplines such as engineering is due to the lack of well-educated technical experts in cognitive technologies. In this paper, the author Stanoyevitch (2005) presented the benefits and versatility of introducing MATLAB into university mathematics programs and how the software could be incorporated. A detailed description was given on how the implementation of the software could enhance mathematics courses both from the academic aspect as well as opening up the employment prospects for graduates. Students could achieve a higher level of learning in courses like abstract algebra with the incorporation of MATLAB into it. The issues such as licensing the software, using it as an isolated technology, from a minimal lower level to an advance levels, have also been raised in the study. Some illustrations of its use such as logistics of air shipping and computation of worst-case-scenario number of a directed network were provided in detail. The study by Shi (2004) was aimed at implementing MATLAB as a tool into teaching and learning of calculus for students majoring in Physics. It was also suggested dividing students into normal and advanced levels based upon their aptitudes as well as including the group work for them. The need for the change in the pedagogy was felt due to a host of problems related to students learning of mathematics. The focus for the normal group of students was given for learning the basic underlying concepts and principles of mathematics with less emphasis on the subject’s rigor. The advanced group of students would have the normal course without providing them the details of the course putting greater self-responsibility for learning. A number of illustrations of the visualization of the graphs of the mathematical functions were given with explanation of how they could be used in teaching. The software tool was also recommeded to be used for enhancing the key concepts such as tangent line of a curve at some point, geometric interpretation of the definite integral as well as some examples of mathematical modelling. The study also highlighted the importance of group work and suggested to form a group of 4 to 5 students to cultivate necessary communication and collaborative skills among them. The assessment scheme to the new approach was suggested to be in place. The author believed that 28 M. Abdul Majid et al. the proposed new approach would be more student-centered and it would provide necessary important skills to students. In this study, Wu (2004) proposed MATLAB to be used as a teaching and learning tool at the School of Science, Beijing Institute of Technology. The study aimed at improving the teaching and learning of linear algebra and to equip students with modern computing technology. The proposal also suggested establishing the website for the course. It was also highlighted that the software could be used as a data visualization tool to promote deeper understanding of the concepts. Dunn et al (2002) used MATLAB for teaching and learning of introductory calculus at the University of Southern Queensland, Australia. Five MATLAB based graphical interface programs were used to demonstrate the mathematical concepts such as Newton’s Method, differentiation and integration. The programs present some unusual and important cases of the calculus concepts. The authors argued that MATLAB could be incorporated into learning of mathematics in three ways, namely, firstly as a numerical and graphical tool, secondly as an aid to learning of ideas and concepts and thirdly to show the procedure or concept in a more sophisticated way. In this work, the authors have used MATLAB using the third approach. The main focus of the use of these programs was to demonstrate mathematical concepts rather than doing a particular task. The study also suggested that the computer technology should be used a supplement to the traditional teaching and learning. In this proposal, Kawski (2003) compared the arguments of the integration of MATLAB with that of the other CAS programs such as Maple and Mathematica in the first year college mathematics. The author comparatively put forward the argument that integrating a CAS in mathematics curriculum is much easier than MATLAB due to the pedagogical and planning related challenges with the later. The study further stated that the traditional calculus is still an "algebra course" whereas MATLAB provides numeric computing in which "almost everything is a matrix". Katende in his unpublished paper reviewed the role of MATLAB and other symbolic algebra systems for engineering mathematics with some illustrations. The study outlined some of the potentials and challenges in undergraduate engineering mathematics. Illustrations such as expanding cos(7x), differentiating sin(E^x(x^2+1)) and integration problems were given and how quickly the algebra systems provided output to such complicated computations. Some examples of 2-D and 3-D graphical visualization were also provided in the paper. The study concluded highlighting the need to change the curriculum or the approach to teaching engineering mathematics. MATLAB Web Server was proposed to be used in this study by Pester et al (2001) for electrical engineering and mathematical modeling. The interactive software could be used to save students' time and to increase their motivation. It could be used for simulation and visualization of 2 and 3-dimension graphs. A few illustrations of how the software could be used were given as well as with some advantages and disadvantage of using web-based MATLAB. Solutions to some network related problems were also proposed. Markanda (1995) described how the students' version of MATLAB could be used to compare several direct, elimination, and iterative methods for solving efficiently and with less calculations systems of n equations in n variables. The illustrations of Jacobi and Gauss-Seidel methods using the software were given in the paper. In a study by Dios et al (2012), MATLAB and Learning Management System "Moodle" were introduced in the traditional teaching and learning of graduate engineering mathematics course. The aim was to give students a better understanding of the course contents. The paper MATLAB as a Teaching and Learning Tool for Mathematics... 29 discussed the mechanism of the new structure of the course including the web-based assessment. The Learning Management System "Moodle" was created as a virtual learning environment to work as a pedagogical model. The online quizzes were used using WIRIS, an editor to write symbols and formulas. Kovacheva (2008) outlined the main features of many popular mathematical application packages including MATLAB. The capabilities of these programs for research as well as for teaching were described briefly. MATLAB as a Tool for Graphing and Visualization of Mathematical Data and Concepts One of the greatest benefits of computer technology is the accessibility of the mathematical concepts through visualization. MATLAB is one of the best known software which contains mathematical data visualization capabilities, Palais (1999). MATLAB as a visualization tool has been used in the courses like linear algebra, calculus and differential equations etc. Szurley (2007) used MATLAB as a visualization tool for applications of linear algebra course. Two examples of population modeling and one example of Genetics were used in this study. The aim of the study was to provide students with the graphing capabilities for illustrating the insights into the material. In an extension of the study, the author created some graphical user interfaces in MATLAB to address students' unsatisfactory responses to the use of the software in an earlier study. The aim was to provide interactivity and visuals for presenting mathematical concepts. Palmer (2008) also aimed at visualization of linear algebra concepts through the writing and the development of simple animation programs in MATLAB. The author claimed that creating animation programs is possible with little knowledge of linear algebra, no computer programming experience and without MATLAB’s familiarity. The study concluded that students' learning of matrices enhanced, however, no students' feedback was given. In a collaborative project between Swedish and British instructors (two mathematicians and one didactic expert), MATLAB was used to teach undergraduate engineering mathematics students, Burton et al (2004). The aims of the project were to support students' visual and graphical understanding of mathematics and their access to the numerical computation of the course. Individual interviews were conducted to get the feedback from students. Their overall feelings were mixed on their experience of using the software. To achieve effective software integration, the study has also provided suggestions such as ensuring clarity in learning goals, adequate number of MATLAB classes, and the preparation of a manual to support MATLAB programming by providing necessary reference material on the web. Students expressed their dissatisfaction with their experiences of using MATLAB as there was too much elapsed time between two MATLAB sessions consequently students had to learn the material over and over again. The study concluded that the integration of the software should be planned thoroughly with enough software documentation. In several studies, MATLAB was integrated into the mathematics curriculum as an add-on teaching and learning tool. In a slightly different approach, it was integrated to an interactive "Adaptive Book" system with the aim to demonstrate the linear algebra concepts and as well as the course's algorithms, Gunawardena and Jain (2002). The concepts were presented both in the mathematics classroom/laboratory followed by the software demonstrations. In this study, MATLAB was also used as a drill and practice tool in which a combination of text and visuals were presented in multiple forms. The project such as Singular Value Decomposition for performing data compression was based on practical applications. The study described linear algebra as an ideal course for integrating technology to enhance the teaching and learning processes. It was concluded that the software 30 M. Abdul Majid et al. like MATLAB could provide a good platform for not only visualization but also for numerical computation. Stewart et al (2005) used MATLAB to assess the process of students' instrumentation of technology in the teaching of mathematics courses. The authors' called the instrumentation as a student's transformation and adaptation of computer technology to the assigned mathematical task and "it is through the actions and decisions that the student makes that the CAS tool or artefact is transformed into an instrument". The case study involved first and second year students enrolled in scientific and engineering disciplines at the University of Canterbury. The second year students used MATLAB in tutorials and computer laboratory sessions for an advance linear algebra course. The aim of the software use was to solve problems but the main focus was to help students with understanding the steps involved in the solution. The data were collected through a questionnaire in which questions were given relating to students' attitudes to the use of computers in learning math; use of computers in linear algebra course and some openended questions on the value of computers for the course. Students' feedback to the benefits of using MATLAB for visualization, conceptualization, and problem-solving was positive. In the analysis, the study reported three students' types regarding the instrumentation of computer technology for university mathematics courses. These types were firstly, those who opposed to the computer work, secondly those who were happy to use computer with a limited instrumentation, and thirdly those who exhibited too much reliance on the use of computer for mathematics. The study concluded with the recommendation of the Leigh-Lancaster's view of achieving the integration of computer technology through the process of having congruency between curriculum, pedagogy and assessment making it clear that CAS must be specified in the curriculum, teaching as well as assessment. The assessment in mathematics must be in tandem with its teaching and learning. In a study, Colgan (2000) described the integration of MATLAB to the teaching of traditional engineering mathematics course at the University of South Australia. The course included elementary linear algebra, the differential and the integral calculus. The need for the software integration was felt due to the decline in the mathematical skills of students entering into the engineering disciplines. MATLAB was chosen because of two reasons, firstly it was extensively used in some of the engineering programs at the university and secondly it was believed to increase students' employability after graduation. A MATLAB guide specifically covering the course contents was prepared to incorporate the software into the teaching. This was considered a major component of integrating software into the syllabus. The software use was demonstrated in the classroom lectures. The graphing capability of the MATLAB was shown to the students illustrating graphs of functions including features such as zoom on and multiple commands for plotting functions etc. MATLAB was used to improve students understanding of the mathematical concepts such as limits, continuity, differentiability, multiple representations of the Mean Value Theorem, Riemann Sums, and finding the volumes of revolution etc. The methodologies of teaching the course were redesigned keeping in view of the qualities of the engineering programs mentioned by the professional engineering body of Australia. The syllabus was rewritten outlining clearly the delivery and the assessment criteria. Weekly five contact hours were allocated for the course in which three hours of regular lecture, one hour of tutorial and one hour was dedicated for working with the software. Apart from the software guide, the syllabus also included the engineering applied examples relevant to the course. The emphasis throughout the process of technology integration was to enhance students' understanding of mathematics. The paper also provided some of the illustration of the engineering examples such as the classical problem of Snell's Law and the minimization of the MATLAB as a Teaching and Learning Tool for Mathematics... 31 time in the refraction of light. To inculcate collaborative spirit among students, they worked in groups of four on MATLAB sessions and the engineering projects/assignments. A part of the group's work was performing the analysis of the engineering project and presenting that in the form of a report. The projects involved the knowledge of the engineering as well as the use of the mathematical formulas. The study concluded that the course was considered to be the most successful without providing any data. The author highlighted the need for striking a good balance between the amounts of time to be spent on MATLAB to the course requirement. Schlatter (1999) described the visualization benefits of using MATLAB in a multivariable calculus for various concepts such as three-dimensional surfaces, contours plots, gradient fields and parametric curves etc. The software, to some extent, was also used for symbolic integration and for setting up integrals. The author developed two- and three-dimensional interactive packages in MATLAB and they were used; firstly to help students in visualization of mathematical objects and surfaces and; secondly to familiarize them with the use of MATLAB. The study concluded that the students could cover a more variety of integrals when the software would be used as a pedagogical tool. Apart from linear algebra and multivariable calculus courses, the software was also used for teaching concepts such as series convergence and numerical computation and visualization of power series approximations. Moore (1988) used MATLAB to teach "Series and Differential Equations" at the Grinnell College. The goal was to provide graphic visualization through the experimentation of various functions, and numeric computations for Taylor polynomials approximations, partial sums of power series, definite integrals and differential equations. Students' used MATLAB-based programs. The study concluded that the software tools not only widen students' problem-solving ability, insight, flexibility but also their ability to use them. However, the study has not provided any students' feedback. In addition to the above studies, the software was also reported to be effective tool in the visualization of mathematical concepts, (Abdul Majid et al 2012, Puhak 2011, Liang 2009, Stewart 2005, Henderson 2002, Smits 1992). MATLAB as a Computational Tool Another extremely powerful feature of MATLAB is its interactive numerical computation environment which is based on the matrix structure. Computational methods and skills of students in topics such as matrix computation, numerical differentiation and integration, and differential equations etc. can be enhanced with the use of the software. Particularly, engineering students need this skill to solve many engineering problems, Canfield (2012). Addabbo (2010) presented the approach that was used for teaching Introduction to MATLAB course for engineering students of Vaughn College who had completed calculus as one of the requirements. The course was divided into two parts; part 1 was related with programming techniques based on calculus problems, and the part 2 was the study of the numerical solutions of differential equations studied through linear and non-linear physical models. This approach was developed to demonstrate the relationship between the pre-requisite courses such as physics, statics, dynamics, writing and the vibration with that of calculus as well as to encourage higher order learning. Several projects such as matrix operations, summing series, Newton's Root Finding Method, Weather Data, and the numerical integration were given to students. Students were expected to do projects in groups of maximum three students. Upon completion, they were required to give a presentation and their performance was assessed on a set criteria. However, the study has not reported the effects of the approach and no conclusions were drawn. In a similar study, MATLAB was used for teaching computational modules at Southern Queensland University, Tonkes (2005). The learning model was developed and integrated for the first year mathematics courses. The aim of the study was to primarily reinforce traditional mathematical 32 M. Abdul Majid et al. learning, teaching concepts in numerical math and as well as for engaging students in computer aided problem solving. The model development included an interactive workbook, web-based solutions, Graphical User Interfaces and programs developed in MATLAB. The study highlighted some of the challenges encountered in the integration of the software in the mathematics curriculum. They were time allocation for the software use, students' efforts in learning it, numeric nature of the software, its syntax, and differences in the meaning of some of the signs in computers and math as well as the attitudes of teachers using it. Students progressed through their use of the software in stages through the single line commands environment to launching MATLAB's GUIs and then to developing their own MATLAB code. All assignments/exercises and experiments were performed in pairs. The outcome of the study was assessed through students' feedback survey and the programs downloads log. The students demonstrated an enhanced understanding of math, interaction with the tutors increased, and students' retention improved. The model had been reportedly successful from all stake-holders; students, lecturers, and tutors. Linear Algebra course was among the most popular courses in which MATLAB was used as reinforcement to classrooms lecture. At the University of Sydney, MATLAB was used as a supplement for teaching an undergraduate linear algebra course, Henderson (2002). The aims were to familiarize students with basic MATLAB commands so as to improve their conceptual understanding of the course. In this study, students used MATLAB to solve 4 to 5 problems in a week over one full semester. The use was mainly towards mathematical exploration, visualization of graphs and multiple representations of mathematical concepts. The study recommended maintaining an appropriate balance between the use of the software for learning mathematics and the traditional learning of mathematics. The data were collected through a questionnaire which contained both open-ended and closed-ended questions. The feedback to students' attitudes toward mathematics have also been discussed. The study reported positive and negative views toward the use of the software for learning mathematics. The software had also been used to study some mathematical functions such as Weirstrass function and to study numerical approximation of ʌ, Feng (2011). In this study, the author has also used the software for testing the convergence of an infinite series using ratio test, root test and the integral test. Liang 2009, Burton 2004, Gunawardana 2002, Smits 1992, & Moore 1988 have also reported the benefits of using MATLAB as a computational tool. MATLAB as a Tool for Improving Math Performance/Math Achievement The positive impact of computer technology on students' learning and performance has also been examined and reported, Abdul Majid et al (2012), Strayhorn (2006), Kulik & Kulik (1991). Abdul Majid et al (2012) have used MATLAB as a supplement for teaching Integral Calculus course to 77 engineering students at the University of Ha'il. The study aimed at firstly, to enhance students' conceptual understanding, their performance in mathematics and secondly, to foster positive attitudes towards mathematics and attitudes toward the computer technology. The case study involved an experimental group and a control group. Among the important objectives of incorporating the software were to address the declining mathematical skills of students needed for their core engineering courses as well as establishing a smooth transition to the upper level courses. The software had been used for a number of instructional objectives such as for visualization of graphs of mathematical functions, for identifying and exploring mathematical patterns in evaluating definite integrals, for multiple representations of a mathematical concept, and as well as for computation. The study reported the positive effects on students' math achievement compared to the performance of control group of students in the final examination. MATLAB as a Teaching and Learning Tool for Mathematics... 33 The conclusions drawn from the study were that the integration of MATLAB could be effective through the appropriate instructional design combined with students using the software in groups of 2-3. The integration of MATLAB has also been reportedly positive on students' performance in mathematics with "Moodle" an e-learning management system. The MATLAB was integrated with Moodle as a supplement to the classroom lecture for teaching calculus to undergraduate students in Geodesy at the Vienna University of Technology, Judex et al (2008). The blended teaching and learning approach was used to address the issue of students' lack of interest in mathematics and their weak background in the subject. To integrate MATLAB into the course, the software was used as an instructional tool and its basic course was included into the exercises of the mathematics courses. Specifically, the software was used for the large geometric mathematical drawings used in the geo science. Over 90 examples were developed and were made available via "Moodle" for students. The blended approach proved to be successful for the presentation of the heavy calculus course contents in a short time without sacrificing the true comprehension of the course. It enhanced the quality of lectures, improved students average grades as well as their programming skills. The integration of MATLAB would not only provide students a familiarity with modern learning tool but also an added advantage in their careers. The linear algebra course is among the most popular courses in which MATLAB has been integrated in its teaching. Smits et al (1992) described the impact of the use of MATLAB on mathematics achievement of the first year linear algebra students at the Eindhoven University of Technology. A detailed overview was given on how the course contents were presented in lectures and tutorials. In their study, the first nine weeks of the term, linear algebra was taught in which a 3-hour lecture with a tutorial were given. During the second half of the term, the course primarily focused on eigenvalues and eigenvectors followed by the last 3-weeks of the course on ordinary differential equations. MATLAB was used for time-consuming calculations, for motivating students, the interpretation of the row-echelon form, for doing more practice with the problems, for illustrating some aspects of the theory, geometric interpretation of the matrix of a linear transformation, and for introducing unknown techniques or concepts. Students played an active role in learning, conjecturing through guided experiments, and acquainted themselves with the real use of the mathematical techniques. An analysis of the results revealed that students' performance was better in the first term compared to the second term due to the complex nature of the concepts dealt with in the second. The study described the positive effects of the use of the software on students' performance as well as on equipping them with modern computing skills. MATLAB as a Tool for Fostering Positive Attitudes Towards Math, Math Confidence, and Enhancing Motivation Towards Learning Math Attitudes are defined as positive or negative emotional dispositions, McLeod, D. B. (1992) & Aiken (2000). Students' attitudes toward mathematics are complex and a number of measurement tools and constructs are reported in the literature. According to Willis (1995), "Human-computer interaction is a complex phenomenon and the attitudes and feelings involved with this relationship are difficult to identify" and the author has also reported the positive attitudes towards computer technology. This positive change in attitudes also reported to have motivated students in mastering the necessary computer skills quickly. Confidence in learning mathematics, mathematics anxiety, and perceived usefulness of mathematics as the most frequently studied and identified affective variables associated with student achievement in mathematics, Reyes (1984). One of the strategies indicated by Bretscher (1989) is the use of a computer lab with 34 M. Abdul Majid et al. tutors providing individual instruction and hands-on practice in concepts and skills. This strategy may help enhancing students' motivation and overcoming their math anxiety. The computer technology fosters positive attitudes towards mathematics, Lim et al (2009), Yashau (2006), Ocak (2006) &Gorev et al (2004), Cretchley et al (2000), increases students confidence in math, Brake (2007) and as well as it enhances their motivation towards learning math, Puhak (2011) & Carol (2012) . Figure 3: Ernest (2003) on failure and success cycles in math. Yushau (2006) used a blended learning approach in which MATLAB was used as a learning tool by 70 pre-calculus students at King Fahd University of Petroleum & Minerals. Students used MATLAB in weekly computer lab sessions. Some related online learning resources were also provided to students. The study was conducted over a semester and the pre- and post-study students' attitude towards mathematics and computers were examined. The concluding results reported a statistical significance of students' positive attitude towards mathematics and computers. The use of technology had reportedly positive effects on students' attitudes for a variety of courses such as single and multivariable calculus and courses which require solid background in mathematics such as engineering and technology. However, some studies such as the one on mathematical modeling by Lim et al (2009) reported no change in students' attitudes. In this study, the effects on students' attitudes were examined when applied mathematical modeling project was introduced with MATLAB. The project was concerning the volcanic ash fall after an eruption. The study involved third year students of Earth Science at the National Taiwan Normal University. It aimed at assessing students' attitudes towards mathematics in the pre- and post-modeling project. Some of the anticipated benefits of the projects were making the learning of the course interesting, providing a link between mathematical theory and the practical applications thereby reinforcing the bond with the real world, in addition to the fact that the computer modeling is cheaper as it does not require expensive equipment. The partial differential equation of the advection-dispersion model was used for the study. Due to the time constraints, students were involved only in the graphical interpretation of the solution by writing programs in MATLAB. The study highlighted that for practical purposes, the experts require the graphical interpretation of the solution rather than the methods which are employed. Students were also required to orally present and report their findings. Data were collected through preand post-study questionnaires and the post-study semi-structured individual interview with 7 students to assess their attitudes towards the project. Questionnaires consisted of four categories of questions relating to students beliefs, usefulness, enjoyment and anxiety. The study, however, concluded that there was no significant change in the attitudes of students due to a relatively short period of time. Nevertheless, students' enjoyment and interest towards the project and the use of MATLAB had reportedly positive. MATLAB as a Teaching and Learning Tool for Mathematics... 35 Computers have also been used for motivating students toward mathematics and their use reportedly had positive impact. Puhak (2011) used MATLAB as a teaching tool to enhance students' motivation and understanding of applied mathematics. The method used for the implementation of the software included choosing one traditional calculus concept, using its application in related areas, performing related calculations for enhanced understanding, and dealing with associated questions to further students comprehension. This method was illustrated with the example 'Center of Mass' with details of the role of MATLAB in the activity with the needed programming code and its visualization. The study has not used the formal measurement techniques, but recommended, to assess the effects of the use of the approach on students' learning. However, routine course evaluations and volunteered feedback were received. In another study by Carol (2012), MATLAB was used for students of Sports Technology who were enrolled into a first year mathematics course at Loughborough University. The aim of the study was to enhance students' motivation and engagement towards learning of mathematics through the introduction of appropriate sports-related group projects and applications. The need for the study was to address the problem of students' lack of engagement and motivation in the learning of mathematics, their poor attendance and high failure rate. An electronic voting system was also used to increase students' engagement and interaction with the lectures. The study highlighted the importance of the group projects and the use of the computer software for realistic problems for engineering students. The author compared the outcome of the initiative which included students' attendance record, motivation and their post-study results with the previous historical data. In the post-study analysis, the students' attendance had reportedly increased compared to the previous data. This increase was attributed to the change in the tutorial classes from the traditional math tutorial to the tutorial held in the computer laboratory where the active learning took place. The change was also evident in their quality of work and the feedback given through the questionnaire. Moreover, a noticeable jump of 33% in the students' success rate was reported. However, the study lacked the control group. Brake (2007) used MATLAB with the aim to increase the math self-confidence and the math ability of freshmen engineering technology students at the School of Engineering Technology, Eastern Michigan University. The software was used to see if this could improve students' learning. The author believed that self-confident students could be transformed from novice learners to expert learners and MATLAB would increase that self-confidence in them in analyzing problems in mathematics. In this study, the researcher taught two sections of ET 100, Introduction to Engineering Technology, a freshmen engineering pre-requisite course for electronics engineering technology, mechanical engineering technology, and computer engineering technology. MATLAB was introduced into the curriculum, not just as a tool, but to be able to solve engineering problems. The number of students enrolled in this course was 38. A survey was carried out to obtain demographic information as well as data on students past mathematical training including how confident they felt in solving two rather difficult problems relating to finding roots of numbers and applying the math to a real world. A second survey was conducted at the end of the semester and a record of quiz, midterm and final grades on specific questions using MATLAB was kept. Since one of the goals was to turn the inexperienced students into expert learners, it was found that graphing helped many students learn to think more deeply about problems, however, further interpretation skills could not be enhanced due to the insufficient time of using the software. Nevertheless, this study touched upon a very important issue of incorporating computational package into the learning of engineering technology course which requires a thorough understanding of underlying mathematical principles. There are some issues unaddressed relating to the low class size of students, improper 36 M. Abdul Majid et al. brushing up of students skills in learning MATLAB, inadequate background of students in mathematics, and hence statistical insignificance in terms of its applicability to a wider audience and areas. There is a need for more research on the use of MATLAB as a tool to increase self confidence and math ability in freshmen engineering students which the author also felt strongly about. The skills acquired through the use of MATLAB in the teaching and learning will prepare future engineers to meet the demands of the 21st century. Ocak (2006) investigated the relationship between gender and students' attitudes and their prior experience of using MATLAB. Three calculus I classes comprising of 23 female and 37 male students from three different colleges were chosen for this study. All students of these classes had used MATLAB. Students' attitudes were assessed with an attitudes survey instrument and the prior experience with MATLAB was measured with a closed-ended response format questions. The author found a slightly positive correlation between students' attitudes and a prior experience on using the mathematical software program. The study concluded that there was no correlation between gender differences, students' attitude and the software program experience. The study also suggested that an assessment of the students' attitudes should be taken into consideration while using programs like MATLAB. Students' familiarity with features of the software program and their ease of use also affect their enthusiasm and curiosity. This study was conducted by Gorev et al (2004) to examine the effectiveness of the use of computer technologies in solving and doing mathematical problems by pre-service teachers. In addition, the assessments of teachers' skills in using these computer tools; their attitudes toward them; and their willingness to use these tools in their careers as teachers were also considered. MATLAB was also used among the many other computer software tools. Students were given assignment sheets which included both routine and non-routine questions without the use of computers. They were further encouraged to use the computerized tools to solve the problems which they could not solve by hand as well as to enhance their previous solutions. Students' solutions were analyzed using Van Hiele theory to understand the development of their insights. This theory defines insight as a combination of the ability to perform in a possible unfamiliar situation, to perform competently as per the requirement, and resolving the solution consciously and deliberately. Their responses were also collected through their opinions on the use of computer tools in their careers. A vast majority of students expressed strong support to the use of computerized tools for improving their understanding and as well as to find solutions to problems. Some qualitative responses were also provided in the study which, by and large, favored their strong willingness to the use of these tools in their careers despite some unanticipated technical problems that may come in the way. Cretchley et al (2000) investigated the effects of the use of MATLAB on the attitudes and learning of undergraduate mathematics students at the University of Southern Queensland. The study also highlighted the relationships between attitudinal effects and performance. The student body had comprised of a diverse group, on-campus and distance learners, registered in the first semester Algebra and Calculus course. MATLAB was used in lectures, tutorial and assignments. In order to develop mathematical concepts and necessary skills, tasks were designed and a MATLAB manual was written. The use of the software was demonstrated in lectures and oncampus students attended weekly two-hour small group tutorials conducted by tutors and the distance learners used scheduled tutorial on telephones, emails and newsgroups. To encourage and motivate students, small prize money was announced. The data collection instruments used were a pre-study diagnostic test, two post-study examinations, questionnaires related to MATLAB-based assignments, interviews and focus groups, and a pre- and post-study survey questionnaire. The study reported that students' appreciation of the use of the software gradually MATLAB as a Teaching and Learning Tool for Mathematics... 37 increased over time and their ease of use developed as they overcome frustration and difficulty. The assessment of students' knowledge was based on an open book examination without the use of the computers to maintain equity. Students were not examined on their knowledge of the software. Pre- and post-study tests and assignments were conducted to evaluate students' mathematics learning experience, their opinions, skills and interviews were also conducted through focus discussion groups and case studies. Students' opinions were sought on categories such as their learning of mathematics, their experience of using computer software in the learning of mathematics, and self-efficacy in math. MATLAB based tasks particularly explorations were reportedly successful and students reported their positive experiences such as visualizing and computing areas under curves, and the concept of definite integral by taking Riemann sums. The study also reported a wide acceptance, by both on-campus and external students, of the use of technology as an aid for learning and, in particular, of MATLAB as an appropriate software package. The authors reported a strong positive reaction to the experience of using MATLAB to support students learning of mathematics where inculcating motivation towards the study of mathematics many a times is a hardest job. Another motivating factor towards the use of the MATLAB was students' need and importance of its familiarity would help in their relevant engineering subjects. The software's graphical capacity, power and novelty were much appreciated by students. Other features students liked were MATLAB's speed in performing computation with ease, clarity in understanding through explorations, and use the software as a technology check for confirmation of their answers. MATLAB was favoured over graphing calculators due to the computer power with large screen and 3-D graphing capacity. Several others studies have also reported the benefits of using the software as a tool for increasing motivation (Pennel 2009, Ohrstrom 2005, Smits 1992) and fostering positive attitudes, Abdul Majid et al 2012. MATLAB as a Tool for Projects Han (2008) used MATLAB to teach elementary linear algebra to a class of 23 students at the Eastern Michigan University. Three software-based projects and some homework assignments problems were given to students. This innovation was used as a supplement to the traditional method of instruction and its aim was to examine students' perceptions of its usefulness in learning the course material. The approach provided an opportunity to students to solve realworld application problems. MATLAB was chosen to be the right choice over the other software due to its wide popularity in science and engineering, interactive calculations, visualizations capabilities together with its broader applications and audiences. A MATLAB manual was written and the tutorial sessions were held to familiarize students of its use. The three projects designed for the study were; firstly the application of Gaussian elimination, secondly inverse computing methods to solve large linear systems of equations, solving linear systems using Cramer's rule and thirdly the application of the singular value decomposition method. Students performed these projects individually and as well as in groups of 2 to 3. Students' feedback was received through a survey using yes/no types of questions and their opinions were also taken. More than 3/4th of the students expressed their overall satisfaction with the software and more than 57% thought that the software helped them in their learning. Chapin et al (2004) supplemented MATLAB with the standard calculus course for giving simple homework assignments to students at Ohio University. The authors outlined three categories, integrated use, project format and textbook add-on, of the uses of technology in the literature. They, however, implemented the use of the software for simple assignments for 38 M. Abdul Majid et al. making its application easy and practical for both students and instructors. The study highlighted the incorporation of the technology through firstly, use of simple and basic calculations; secondly, giving very clear instructions on assignments; and thirdly, giving very clear and concise technical information. In conclusion, the authors proposed the use of software in undergraduate mathematics courses for simple homework and included a sample MATLAB assignment. The study has not discussed its effects on students learning. Horwitz

(2002) described collaborative engineering and science projects between mathematics and engineering departments for the purpose of enhancing first year differential and integral calculus courses. Students used MATLAB in projects individually and in teams and some real data were used. The first semester projects were Hydraulic Engineering, designing a pipeline with minimum cost, optimization of irrigation channel and the second semester projects were Exponential Decay, Skydiver free fall, Wrecking ball, and the Automobile velocity data. A students survey was carried out and they were asked to rate the projects on the scale of 1 to 5. Students reported some complains on the difficulty level of the projects. Some variation was observed in the comments of students belonging to engineering disciplines compared with the science majors. In addition, some students did take part in the projects but did not fully understand the mathematics involved. The study concluded firstly that the students' motivation for doing projects did not seem to have increased at the end of the study, secondly, they were simply involved in the project without themselves understanding the process, and thirdly, in depth project understanding could not be achieved due to the pressure of covering the regular syllabus of the courses. The authors suggested offering one extra credit hour course which could enhance the calculus program. The positive feedback was received from students as they enjoyed doing the projects with MATLAB and they saw a connection of mathematics with the real world. A thorough understanding of mathematical concepts with their applications in engineering and science proved useful to students. Szurley (2011) discussed the application of MATLAB for providing simulation to the projects involving solution of Differential Equations. The modeling projects involved the simulation of the industrial process of film casting and the conduction of heat. The projects were aimed at exposing students to real world applications, familiarizing them with MATLAB, demonstrating the need for numerical computation, and providing the necessary presentation skills. The study reported positive responses from students about real-world applications and in general considered a success. However, the use of MATLAB and the numerical schemes were not much favoured by them. In another study, the author incorporated MATLAB within mathematics courses at Francis Marion University. MATLAB was used for Calculus III projects involving numerical solutions and also for understanding the software's programming concepts. The projects involved L'Hopital Rule, improper integrals, sequences, power series, Taylor and Maclaurin series and as well as Linear Programming and Linear Algebra. The study concluded that students' responses were not very positive about the use of MATLAB partly due to nongraphical interfaces provided to them but in the long run this exposure was believed to help them. The future plan was set up to develop interactive graphical interfaces. Hailan et al (2012) highlighted various benefits of using MATLAB in applications of engineering as an innovative and scientific tool. The author illustrated with examples of how the software could be used as a tool for exploration and experimentation in courses like linear algebra and higher mathematics. The software's commands for finding reduced row echelon form and solving linear equations are used with examples including examples from statistics. Some applications in textile engineering and textile materials have also been given in this paper. MATLAB as a Teaching and Learning Tool for Mathematics... 39 Ohrstrom et al (2005) conducted a study at Chalmers University of Technology. An integrated approach to using MATLAB in chemistry and mathematics courses was introduced and its impact on students learning of solving chemical problems was evaluated. This curriculum reform approach was developed by experts in math, chemistry and education. Students used MATLAB-based assignments individually and in groups in the laboratory. The process of integration of the software into the course was achieved through the exchange of teachers and course material, schedules and planning together with the number of MATLABrelated projects and exercises/assignments. The projects were meant to teach mathematical concepts and skills followed by the development of MATLAB programs. The projects included were the introduction of the 3d geometry of molecules and mathematics, differential equations and kinetics, systems of nonlinear equations and equilibrium analysis as well as computer laboratory on acids and bases. Various pedagogical benefits of using the software were highlighted in the paper. The process of evaluation of its impact involved students, teachers and the educational consultants. The new reform resulted in a positive impact on students learning, and their motivation towards the subject increased. One of the main challenges highlighted by the study was maintaining a right balance in providing the necessary MATLAB skills to students for the future courses. MATLAB as a Tool for Business and Engineering-Related Applications The study conducted by Liang et al (2009) at the University of New Haven is an illustration and a summary of the use of MATLAB for teaching mathematics. The software use aimed at helping students' learning of calculus-based business mathematics and apply the calculus principles to find solutions to real quantitative business problems. The software was implemented to address students' varying levels of algebraic knowledge and skills and the challenge of completing the course in one semester. MATLAB was used to teach topics such as limits, derivatives, optimization, integration and their applications in business as well as the introduction to multivariate calculus and partial derivatives. MATLAB-aided solutions to the mathematical problems were presented step-by-step. The researchers concluded that the software use had helped students in focusing on key objectives of the course through the visualization of the examples and quick numerical computation. The study concluded with the statement that the application of computer technology would prepare students for modern day challenging careers in the workplace. An innovative approach, in the study by Pennell et al (2009), was introduced to the traditionally taught Differential Equations course for the students of engineering majors. The need for innovation was felt due to the fact that most engineering students were unaware of the relationship between mathematics courses and the their core engineering courses as well as the importance of mathematics in their careers as engineers. The new interdisciplinary project was designed in collaboration between mathematicians and engineers in an attempt to overcome students' lack of motivation towards mathematics and to address their poor retention of the material. The lack of engineering-orientation in the traditional teaching of a course like differential equations was highlighted. For instance the main focus was given on techniques for generating solution formulas. From the practical engineering application view point, the system under consideration was to understand how it responds to classes of inputs. The methods of incorporating engineering points of views of first and second order differential equations were given through the illustrations of circuit series, a mass-spring system and the Laplace transform. For students to investigate the behavior of the systems, authors prepared a brief theoretical 40 M. Abdul Majid et al. description together with the software tool developed using MATLAB and its allied toolboxes. MATLAB's GUI was prepared for students to interactively investigate various plots and behaviors of the systems. No feedback on the approach has been given. Naimark (2002) pointed the problems of a lack of connection in mathematics curriculum with the engineering disciplines which are applications oriented. The author alternatively presented the argument of what he called modernizing the mathematics courses by introducing modern computing-based applications by incorporating MATLAB. Mathematics syllabus should be viewed and taught as a central concept and a unified picture, "concepts as unifying vehicles". The illustrations of the use of MATLAB's Polyfit, MATLAB's view and Fourier Transforms were given. The linear Algebra course was taught using MATLAB in this study. Some of the software-based applications were the visualization of parametric equations, matrix of Linear Operator, matrix multiplication, Gaussian elimination method, rotations, projections, eigenvalues and eigenvectors etc. The study highlighted some pedagogical benefits of the new approach such as making the theory more transparent; students could concentrate on the structure of operations and as well as an additional method of memorization of concepts and procedures for students. The effects of the modern approach enhanced students understanding. The course objectives were achieved through the internalization of the central concept of the general linear operator. The use of the equality of inverse matrices, systems of linear equations, and the associative property of matrix multiplication improved students learning. Some of the challenges noted in the study were: the amount of time required for MATLAB-based exercises, instantaneous rectification of students' errors and the delay caused in teaching concepts like determinants and eigenvectors. In a study by Brown et al (2012), MATLAB has been used as a complementary tool to the traditional lecture for exploring mathematical concepts for the 1st year Electrical Engineering students at the University of Manchester. The introduction of the software was to provide mathematics contextually to see the connection with engineering concepts and to do calculations with varied parameters. Students used the software notebooks developed in Mupad-MATLAB. The study has not described the effects on students learning. MATLAB as a Tool for Improving Problem-Solving Techniques Talbert (2011) used MATLAB to teach problem-solving techniques to first-year Liberal Arts students who were enrolled in Calculus III at the Franklin College. Students spent most of the classtime doing hands-on lab exercises, watching video lectures and completing online tutorial work. The inverted classroom model was used in this study. This model provided students the ability to learn on their own and to apply new concepts. Students took some time adjusting to the innovative learning model, however, their feedback was reportedly positive.Legua et al (2001) in their study provided a comparison of the use of DERIVE, Mathematica and MATLAB to solve the line integral, an engineering mathematical problem. The aim of the study was to harmonize the use of the software with the teaching of mathematics at the Escuela Universitaria de IngenieraTcnica Industrial. A comparison of how the line integral concept could be explained using DERIVE, Mathematica and MATLAB was given with screenshots. The study concluded the DERIVE software as the most efficient of the other programs for evaluating line integrals. The study has not reported the effects of the use of this approach in the classroom teaching. Nyondo (1992) used MATLAB as a problem-solving tool for teaching some topics in linear algebra course at The Papua New Guinea University of Technology. The software was used to MATLAB as a Teaching and Learning Tool for Mathematics... 41 reinforce theoretical concepts taught in conventional lectures. The instructors had designed exercises and problems appropriate to the level of the course and suitable to the students' engineering specialization. The emphasis was also given on using MATLAB and mathematics as problem solving tools for engineering. Using the software, students practiced a variety of problems through experimentation. The study, however, has not reported the effects of the use of MATLAB on students' learning. Conclusion In this review, we have shown the different ways in which MATLAB was used as a teaching and learning tool for mathematics. It has been used as a tool for the visualization of the graphs of mathematical functions, computation, problem-solving, increasing motivation and selfconfidence and as well as fostering positive attitudes toward mathematics. However, a number of factors need to be taken into consideration concerning the technology integration in the teaching and learning of mathematics as highlighted by Schneider (2000). A vast majority of studies have used MATLAB as a supplement to the traditional teaching and learning of mathematics

CT Images

A computerized tomography scan (CT or CAT scan) uses computers and rotating X-ray machines to create cross-sectional images of the body. These images provide more detailed information than normal X-ray images. They can show the soft tissues, blood vessels, and bones in various parts of the body. A CT scan may be used to visualize the:

- head
- shoulders
- spine
- heart
- abdomen
- knee
- chest

During a CT scan, you lie in a tunnel-like machine while the inside of the machine rotates and takes a series of X-rays from different angles. These pictures are then sent to a computer, where they're combined to create images of slices, or cross-sections, of the body. They may also be combined to produce a 3-D image of a particular area of the body.

Why Is a CT Scan Performed?

A CT scan has many uses, but it's particularly well-suited for diagnosing diseases and evaluating injuries. The imaging technique can help your doctor:

- diagnose infections, muscle disorders, and bone fractures
- pinpoint the location of masses and tumors (including cancer)
- study the blood vessels and other internal structures
- assess the extent of internal injuries and internal bleeding
- guide procedures, such as surgeries and biopsies
- monitor the effectiveness of treatments for certain medical conditions, including cancer and heart disease

The test is minimally invasive and can be conducted quickly.

How Is a CT Scan Performed?

Your doctor may give you a special dye called a contrast materialto help internal structures show up more clearly on the X-ray images. The contrast material blocks X-rays and appears white on the images, allowing it to highlight the intestines, blood vessels, or other structures in the area being examined. Depending on the part of your body that's being inspected, you may need to drink a liquid containing the contrast. Alternatively, the contrast may need to be injected into your arm or administered through your rectum via an enema. If your doctor plans on using a contrast material, they may ask you to fast for four to six hours before your CT scan.

When it comes time to have the CT scan, you'll be asked to change into a hospital gown and to remove any metal objects. Metal can interfere with the CT scan results. These items include jewelry, glasses, and dentures. Your doctor will then ask you to lie face up on a table that slides into the CT scanner. They'll leave the exam room and go into the control room where they can see you and hear you. You'll be able to communicate with them via an intercom.

While the table slowly moves you into the scanner, the X-ray machine will rotate around you. Each rotation produces numerous images of thin slices of your body. You may hear clicking, buzzing, and whirring noises during the scan. The table will move a few millimeters at a time until the exam is finished. The entire procedure may take anywhere from 20 minutes to one hour.

It's very important to lie still while CT images are being taken because movement can result in blurry pictures. Your doctor may ask you to hold your breath for a short period during the test to prevent your chest from moving up and down. If a young child needs a CT scan, the doctor may recommend a sedative to keep the child from moving.

Once the CT scan is over, the images are sent to a radiologist for examination. A radiologist is a doctor who specializes in diagnosing and treating conditions using imaging techniques, such as CT scans and X-rays. Your doctor will follow-up with you to explain the results.

What Are the Risks Associated with a CT Scan?

There are very few risks associated with a CT scan. Though CT scans expose you to more radiation than typical X-rays, the risk of cancer caused by radiation is very small if you only have one scan. Your risk for cancer may increase over time if you have multiple X-rays or CT scans. The risk of cancer is increased in children receiving CT scans, especially to the chest and abdomen.

Some people have an allergic reaction to the contrast material. Most contrast material contains iodine, so if you've had an adverse reaction to iodine in the past, make sure to notify your doctor. Your doctor may give you allergy medication or steroids to counteract any potential side effects if you're allergic to iodine but must be given contrast.

It's also important to tell your doctor if you're pregnant. Though the radiation from a CT scan is unlikely to harm your baby, your doctor may recommend another exam, such as an ultrasound or MRI scan, to minimize risk.

What Do CT Scan Results Mean?

CT scan results are considered normal if the radiologist didn't see any tumors, blood clots, fractures, or other abnormalities in the images. If any abnormalities are detected during the CT scan, you may need further tests or treatments, depending on the type of abnormality found.

A computerized tomography scan (CT or CAT scan) uses computers and rotating X-ray machines to create cross-sectional images of the body. These images provide more detailed information than normal X-ray images. They can show the soft tissues, blood vessels, and bones in various parts of the body. A CT scan may be used to visualize the:

- head
- shoulders
- spine
- heart
- abdomen
- knee
- chest

During a CT scan, you lie in a tunnel-like machine while the inside of the machine rotates and takes a series of X-rays from different angles. These pictures are then sent to a computer, where they're combined to create images of slices, or cross-sections, of the body. They may also be combined to produce a 3-D image of a particular area of the body.

Why Is a CT Scan Performed?

A CT scan has many uses, but it's particularly well-suited for diagnosing diseases and evaluating injuries. The imaging technique can help your doctor:

- diagnose infections, muscle disorders, and bone fractures
- pinpoint the location of masses and tumors (including cancer)
- study the blood vessels and other internal structures
- assess the extent of internal injuries and internal bleeding
- guide procedures, such as surgeries and biopsies
- monitor the effectiveness of treatments for certain medical conditions, including cancer and heart disease

The test is minimally invasive and can be conducted quickly.

How Is a CT Scan Performed?

Your doctor may give you a special dye called a contrast materialto help internal structures show up more clearly on the X-ray images. The contrast material blocks X-rays and appears white on the images, allowing it to highlight the intestines, blood vessels, or other structures in the area being examined. Depending on the part of your body that's being inspected, you may need to drink a liquid containing the contrast. Alternatively, the contrast may need to be injected into your arm or administered through your rectum via an enema. If your doctor plans on using a contrast material, they may ask you to fast for four to six hours before your CT scan.

When it comes time to have the CT scan, you'll be asked to change into a hospital gown and to remove any metal objects. Metal can interfere with the CT scan results. These items include jewelry, glasses, and dentures. Your doctor will then ask you to lie face up on a table that slides into the CT scanner. They'll leave the exam room and go into the control room where they can see you and hear you. You'll be able to communicate with them via an intercom.

While the table slowly moves you into the scanner, the X-ray machine will rotate around you. Each rotation produces numerous images of thin slices of your body. You may hear clicking, buzzing, and whirring noises during the scan. The table will move a few millimeters at a time until the exam is finished. The entire procedure may take anywhere from 20 minutes to one hour.

It's very important to lie still while CT images are being taken because movement can result in blurry pictures. Your doctor may ask you to hold your breath for a short period during the test to prevent your chest from moving up and down. If a young child needs a CT scan, the doctor may recommend a sedative to keep the child from moving.

Once the CT scan is over, the images are sent to a radiologist for examination. A radiologist is a doctor who specializes in diagnosing and treating conditions using imaging techniques, such as CT scans and X-rays. Your doctor will follow-up with you to explain the results.

What Are the Risks Associated with a CT Scan?

There are very few risks associated with a CT scan. Though CT scans expose you to more radiation than typical X-rays, the risk of cancer caused by radiation is very small if you only have one scan. Your risk for cancer may increase over time if you have multiple X-rays or CT scans. The risk of cancer is increased in children receiving CT scans, especially to the chest and abdomen.

Some people have an allergic reaction to the contrast material. Most contrast material contains iodine, so if you've had an adverse reaction to iodine in the past, make sure to notify your doctor. Your doctor may give you allergy medication or steroids to counteract any potential side effects if you're allergic to iodine but must be given contrast.

It's also important to tell your doctor if you're pregnant. Though the radiation from a CT scan is unlikely to harm your baby, your doctor may recommend another exam, such as an ultrasound or MRI scan, to minimize risk.

What Do CT Scan Results Mean?

CT scan results are considered normal if the radiologist didn't see any tumors, blood clots, fractures, or other abnormalities in the images. If any abnormalities are detected during the CT scan, you may need further tests or treatments, depending on the type of abnormality found.

What is a cranial CT scan?

A cranial CT scan is a diagnostic tool used to create detailed pictures of features inside your head, such as your skull, brain, paranasal sinuses, ventricles, and eye sockets. CT stands for computed tomography, and this type of scan is also referred to as a CAT scan. A cranial CT scan is known by a variety of names as well, including brain scan, head scan, skull scan, and sinus scan.

This procedure is noninvasive, meaning it doesn’t require surgery. It’s usually suggested to investigate various symptoms involving the nervous system before turning to invasive procedures.

Reasons for a cranial CT scan

The pictures created by a cranial CT scan are far more detailed than regular X-rays. They can help diagnose a range of conditions, including:

- abnormalities of the bones of your skull
- arteriovenous malformation, or abnormal blood vessels
- atrophy of brain tissue
- birth defects
- Brain eneurysm
- hemorrhage, or bleeding, in your brain
- hydrocephalus, or fluid buildup in your skull
- infections or swelling
- injuries to your head, face, or skull
- stroke
- tumors
- Your doctor may order a cranial CT scan if you’ve had an injury or display any of these symptoms with no apparent cause:
- fainting
- headache
- seizures, especially if any occurred recently
- sudden behavioral changes or changes in thinking
- Hearing loss
- vision loss
- muscle weakness or numbness and tingling
- speech difficulty
- difficulty swallowing
- A cranial CT scan can also be used to guide other procedures such as surgery or biopsy.

What happens during a cranial CT scan

A cranial CT scanner takes a series of X-rays. A computer then puts these X-ray images together to create detailed pictures of your head. These images help your doctor make a diagnosis.

The procedure is usually done in a hospital or outpatient imaging center. It should take only about 15 minutes to complete your scan.

On the day of the procedure, you must remove jewelry and other metal objects. They can damage the scanner and interfere with the X-rays.

You’ll probably be asked to change into a hospital gown. You’ll lie on a narrow table either face up or face down, depending on the reasons for your CT scan.

It’s very important that you remain completely still during the exam. Even a little movement can blur the images.

Some people find the CT scanner stressful or claustrophobic. Your doctor may suggest a sedative to keep you calm during the procedure. A sedative will also help keep you still. If your child is having the CT scan, their doctor may recommend a sedative for these same reasons.

The table will slowly slide so that your head is inside the scanner. You may be asked to hold your breath for a short period. The scanner’s X-ray beam will rotate around your head, creating a series of images of your head from different angles. The individual images are called slices. Stacking the slices creates three-dimensional images.

Images can be seen immediately on a monitor. They will be stored for later viewing and printed. For your security, the CT scanner has a microphone and speakers for two-way communication with the scanner operator.

Contrast dye and cranial CT scans

Contrast dye helps highlight some areas better on CT images. For example, it can highlight and emphasize blood vessels, intestines, and other areas. The dye is given through an intravenous line inserted into a vein of your arm or hand.

Often, images are first taken without contrast, and then again with contrast. However, use of contrast dye isn’t always necessary. It depends on what your doctor is looking for.

Your doctor may direct you not to eat or drink for several hours before the test if you’re going to receive contrast dye. This depends on your particular medical condition. Ask your doctor for specific instructions for your CT scan.

Preparation and precautions to consider

The scanner table is very narrow. Ask if there is a weight limit for the CT scanner table if you weigh more than 300 pounds.

Be sure to tell your doctor if you’re pregnant. X-rays of any kind aren’t recommended for pregnant women.

You’ll want to be aware of some extra precautions if contrast dye will be used. For example, special measures must be taken for people on the diabetes medicine metformin (Glucophage). Be sure to let your doctor know if you take this drug. Also tell your doctor if you’ve ever suffered an adverse reaction to contrast dye.

Possible side effects or risks

Side effects and risks for a cranial CT scan involve discomfort, exposure to radiation, and allergic reaction to the contrast dye.

Discuss any concerns with your doctor before the test so you can assess the potential risks and benefits for your medical condition.

Discomfort

The CT scan itself is a painless procedure. Some people feel uncomfortable on the hard table or have difficulty remaining still.

You may feel a slight burning when the contrast dye enters your vein. Some people experience a metal taste in their mouths and a warm sensation throughout their bodies. These reactions are normal and generally last less than a minute.

Radiation exposure

CT scans expose you to some radiation. Doctors generally agree that the risks are low compared to the potential risk of not being diagnosed with a dangerous health problem. The risk from a single scan is small, but it increases if you have many X-rays or CT scans over time. Newer scanners may expose you to less radiation than older models.

Tell your doctor if you're pregnant. Your doctor may be able to avoid exposing your baby to radiation by using other tests. These may include a head MRI scan or ultrasound, which don't use radiation.

Allergic reaction to contrast

Tell your doctor before the scan if you've ever had an allergic reaction to contrast dye.

Contrast dye commonly contains iodine and may cause nausea, vomiting, rash, hives, itching, or sneezing in people who are allergic to iodine. You may be given steroids or antihistamines to help with these symptoms before you receive the dye injection. after the test, you may need to drink extra fluids to help flush the iodine from the body if you have diabetes or kidney disease.

In very rare cases, contrast dye can cause anaphylaxis, a whole-body allergic reaction that can be life-threatening. Notify the scanner operator immediately if you have trouble breathing.

Results of your cranial CT scan and follow-up

You should be able to return to your normal routine after the test. Your doctor may give you special instructions if contrast was used in your test.

A radiologist will interpret the results of the test and send a report to your doctor. The scans are stored electronically for future reference.

Your doctor will discuss the radiologist's report with you. Depending on the results, your doctor might order more tests. Or if they're able to reach a diagnosis, they will go over next steps with you, if any.

Chapter 4 SOFTWARE DESCRIPTION

4.1. Introduction to MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include • Math and computation • Algorithm development • Data acquisition • Modeling, simulation, and prototyping • Data analysis, exploration, and visualization • Scientific and engineering graphics • Application development, including graphical user interface building MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN. The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis. 49 MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others. 4.2 The MATLAB system: The MATLAB system consists of five main parts • Development Environment: This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path. • The MATLAB Mathematical Function Library: This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms. • The MATLAB Language: This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs. • Graphics: MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. 50 It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes lowlevel functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications. • The MATLAB Application Program Interface (API): This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling

MATLAB as a computational engine, and for reading and writing MAT-files. Various toolboxes are there in MATLAB for computing recognition techniques, but we are using IMAGE PROCESSING toolbox. 4.3 GRAPHICAL USER INTERFACE (GUI): MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.) and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files: • A file with extension. fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed. • A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed. 51 To launch GUIDE from the MATLAB command window, type guide filename Where filename is the name of an existing FIG-file on the current path. If filename is omitted, GUIDE opens a new (i.e., blank) window. Fig:4.3 Blank Window A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed. GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few 4.4 Getting Started If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos. 52 At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work. 4.4.1 Introduction - describes the components of the MATLAB system. 4.4.2 Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop. 4.4.3 Manipulating Matrices- introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices. 4.4.4 Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images. 4.4.5 Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays. 4.5 DEVELOPMENT ENVIRONMENT 4.5.1Introduction This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print. Starting and Quitting MATLAB 4.5.2 Starting MATLAB On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop. 53 On a UNIX platform, to start MATLAB, type MATLAB at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop. You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations. 4.5.3 Quitting MATLAB To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script. 4.5.4 MATLAB Desktop When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB. The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries. You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts. You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box. 54 4.5.5 Desktop Tools This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are: • Current Directory Browser • Workspace Browser • Array Editor • Editor/Debugger • Command Window • Command History • Launch Pad • Help Browser Command Window Use the Command Window to enter variables and run functions and M-files. Command History Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function. Running External Programs You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example, emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB. 55 Launch Pad MATLAB's Launch Pad provides easy access to tools, demos, and documentation. Help Browser Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents. To open the Help browser, click the help button in the toolbar, or type help browser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information. Help Navigator Use to Help Navigator to find information. It includes: Product filter - Set the filter to show documentation only for the products you specify. Content's tab - View the titles and tables of contents of documentation for your products. Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products. Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name. Favorite's tab - View a list of documents you previously designated as favorites. Display Pane-After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can: Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar. Bookmark pages - Click the Add to Favorites button in the toolbar. Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go. Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages Current Directory Browser MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path. Search Path To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path. Workspace Browser The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces. To view the workspace and information about each variable, use the Workspace browser, or use the functions who and who's. To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function. The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace as from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a. mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function 57 Array Editor Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace. Editor/Debugger Use the Editor/Debugger to create and debug M-files, which are programs you write to run MATLAB functions. The Editor/Debugger provides a graphical userinterface for basic text editing, as well as for M-file debugging. You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint. If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function. 4.6 MANIPULATING MATRICES 4.6.1 Entering Matrices The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example. You can enter matrices into MATLAB in several different ways: Enter an explicit list of elements.Load matrices from external data files. • Generate matrices using built-in functions. Create matrices with your own functionsin M-files. 58 Start by entering Durer's matrix as a list of its elements. You have only to follow a few basic conventions: Separate the elements of a row with blanks or commas. Use a semicolon, ; , to indicate the end of each row. Surround the entire list of elements with square brackets, [ ]. To enter Dürer's matrix, simply type in the Command Window A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1] MATLAB displays the matrix you just entered. ` A = 16 3 2 13 5 10 11 8 9 6 7 12 4 15 14 1 This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A. 4.6.2 Expressions Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are: • Variables • Numbers • Operators • Functions 59 Variables MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example, num_students = 25 Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element. Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned toany variable, simply enter the variable name. Numbers MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are 3 -99 0.0001 9.6397238 1.60210e-20 6.02252e23 1i -3.14159j 3e5i All numbers are stored internally using the long format specified by the IEEE floatingpoint standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308. 60 4.6.3 Operators Expressions use familiar arithmetic operators and precedence rule. + Addition - Subtraction * Multiplication / Division \ Left division (described in "Matrices and Linear Algebra" in Using MATLAB) ^ Power ' Complex conjugate transpose ( ) Specify evaluation order 4.6.3 Table: Operators 4.6.4 Functions MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, for a list of more advanced mathematical and matrix functions, type help specfun help elmat. 61 Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants. Pi 3.14159265... I Imaginary unit, √-1 I Same as i Eps Floating-point relative precision, 2-52 Realmin Smallest floating-point number, 2-1022 Realmax Largest floating-point number, (2- ε)21023 Inf Infinity NaN Not-a-number 4.6.4 Table: Functions 4.7 GUI A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components. Applications thatprovide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface. 62 The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include: 4.7.1 Creating GUIs with GUIDE MATLAB implements GUIs as figure windows containing various styles of uncontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must beable to save and launch your GUI. All ofthese tasks are simplified by GUIDE, MATLAB's graphical user interface developmentenvironment. 4.7.2 GUI Development Environment The process of implementing a GUI involves two basic tasks. • Laying out the GUI components • Programming the GUI components GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI. The Implementation of a GUI While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI: A FIG-file - contains a complete description of the GUI figure and all of its children (uncontrol and axes), as well as the values of all object properties. 63 An M-file - contains the functions that launch and control the GUI and the callbacks, which are defined as subfunctions. This M-file is referred to as the application M-file in this documentation. Note that the application M-file does not contain the code that lays out the uncontrol's; this information is saved in the FIG-file. The following diagram illustrates the parts of a GUI implementation. FIG 4.7.2 graphical user blocks 4.7.3 Features of the GUIDE-Generated Application M-Fil GUIDE simplifies the creation of GUI applications by automatically generating an Mfile framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages: The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). 64 The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information). The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments. You can elect to have GUIDE generate only the FIG-file and write the application Mfile yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor. 4.7.4 Beginning the Implementation Process To begin implementing your GUI, proceed to the following sections: Getting Started with GUIDE - the basics of using GUIDE. Selecting GUIDE Application Options - set both FIG-file and M-file options. Using the Layout Editor - begin laying out the GUI. Understanding the Application M-File - discussion of programming techniques used in the application M-file. Application Examples - a collection of examples that illustrate techniques which are useful for implementing GUIs. Command-Line Accessibility When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as find obj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes. 65 GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI. In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access. 4.7.5 User Interface Control The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components. ➢Push Buttons ➢ Sliders ➢Toggle Buttons ➢Frames ➢Radio Buttons ➢List boxes ➢Checkboxes ➢Popup Menus ➢Edit Text ➢Axes ➢Static Text ➢Figures Push Buttons Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event. 66 Properties to Set String - set this property to the character string you want displayed on the push button. Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close button) before activating the GUI. Programming the Callback When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state. Toggle Buttons Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback. Programming the Callback The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default). From the GUIDE Application M-File The following code illustrates how to program the callback in the GUIDE application Mfile. function varargout = togglebutton1_Callback(h,eventdata,handles,varargin) button_state = get(h,'Value'); if button_state == get(h,'Max') % Toggle button is pressed elseif button_state == get(h,'Min') % toggle button is not pressed end 67 Adding an Image to a Push Button or Toggle Button Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand). a(:,:,1) = rand (16,128); a(:,:,2) = rand (16,128); a(:,:,3) = rand(16,128); set (h,'CData’, a) Radio Buttons Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button. Implementing Mutually Exclusive Behavior Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property: Value = Max, button is selected. Value = Min, button is not selected. To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user. 68 The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected. function mutual exclude(off) set(off,'Value',0) Obtaining the Radio Button Handles. The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks. The following code shows the call to mutual exclude being made from the first radio button's callback in a group of four radio buttons. function varargout = radiobutton1_Callback (h, event data, handles, varargin) off = [handles. radiobutton2, handles. radiobutton3, handles. radiobutton4]; mutual exclude(off) % Continue with callback.

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks. Checkboxes Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes). 69 The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default): Value = Max, box is checked. Value = Min, box is not checked. You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example: function checkbox1_Callback (h, eventdata, handles, varargin) if (get(h,'Value') == get(h,'Max')) % Then checkbox is checked-take appropriate action else % Checkbox is not checked-take appropriate action end Edit Text Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user. To obtain the string typed by the user, get the String property in the callback. function edittext1_Callback (h, event data, handles,varargin) user string = get(h,'string'); % Proceed with callback... 70 4.7.6 Obtaining Numeric Data from an Edit Test Component MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns Nan. You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is Nan, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg). function edittext1_Callback(h,eventdata, handles,varargin) user entry = str2double(get(h,'string')); if isnan(user_entry) errordlg ('You must enter a numeric value','BadInput','modal') end % Proceed with callback... Triggering Callback Execution On UNIX systems, clicking on the menu bar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menu bar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback. Static Text Static text controls display lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it 71 Frames Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot). Placing Components on Top of Frames Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose. List Boxes List boxes display a list of items and enable users to select one or more items. The String property contains the list of strings displayed in the list box. The first item in the list has an index of The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty. The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer Single or Multiple Selection The values of the Min and Max properties determine whether users can make single or multiple selections: If Max - Min > 1, then list boxes allow multiple item selection. If Max - Min <= 1, then list boxes do not allow multiple item selection. 72 Selection Type List boxes differentiate between single and double clicks on an item and set the figure Selection Type property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection. Triggering Callback Execution MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure Selection Type property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either: Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items. The first choice is best if you are sure, you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file. Popup Menus Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1. 73 Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires. Programming the Popup Menu You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item. This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu are fixed, then you can use this approach. function varargout = popupmenu1_Callback (h, event data, handles, varargin) val = get(h,'Value'); switch val case 1 % The user selected the first item case 2 % The user selected the second item % etc. This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string. 74 function varargout = popupmenu1_Callback (h, event data, handles,varargin) val = get(h,'Value'); string_list = get(h,'String'); selected_string = string_list{val}; % convert from cell array to string % etc. Enabling or Disabling Controls You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states: on - The control is operational off - The control is disabled and its label (set by the string property) is grayed out. inactive - The control is disabled, but its label is not grayed out. When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's Button-down callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details. Axes Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects. 75 Axes Callbacks Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes Button-down property to define the callback. 4.7.7 Plotting to Axes in GUIs GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes. GUIs with Multiple Axes If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example, axes (handles. axes1) makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axis. See GUI with Multiple Axes for an example that uses two axes. Figure Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control. 76 Chapter 5 RESULT ANALYSIS 5.1 Results: Experiments conducted in this research using grayscale image as the cover image, and binary image as a message, see Figure as cover images and as a message. The cover image and message image have the same size, 256 * 256 in size. Here is a cover image used in this study: Fig 5.1.1 Original Image The hidden data should be robust against unauthorized extraction / viewing. In the simplest scenario, the attacker has the knowledge about the proposed method as well as the fact that only one fixed pixel pair with 2 partitions is used throughout the image to hide data. The number of possible cases (i.e., correctly guessing the threshold value used during the filter embedding step) to consider can be formulated as: $\Lambda = [\ w2/r \times (w2/r + 1)] \times 2$ where in these two partitions, each consists of w2/r pixels and (w2/r + 1) pixels, respectively. The value 2 signifies that there is r = 2 cases to consider.

CHAPTER 5 RESULT ANALYSIS

Here we take a low dose CT image which is not a clear image and contains noise.Input image is taken and noise is removed in several steps.

Fig:5.1:Input Image

FIG:5.2 LL Image

Here low low filter is applied to the image which is used to smoothen the edges.

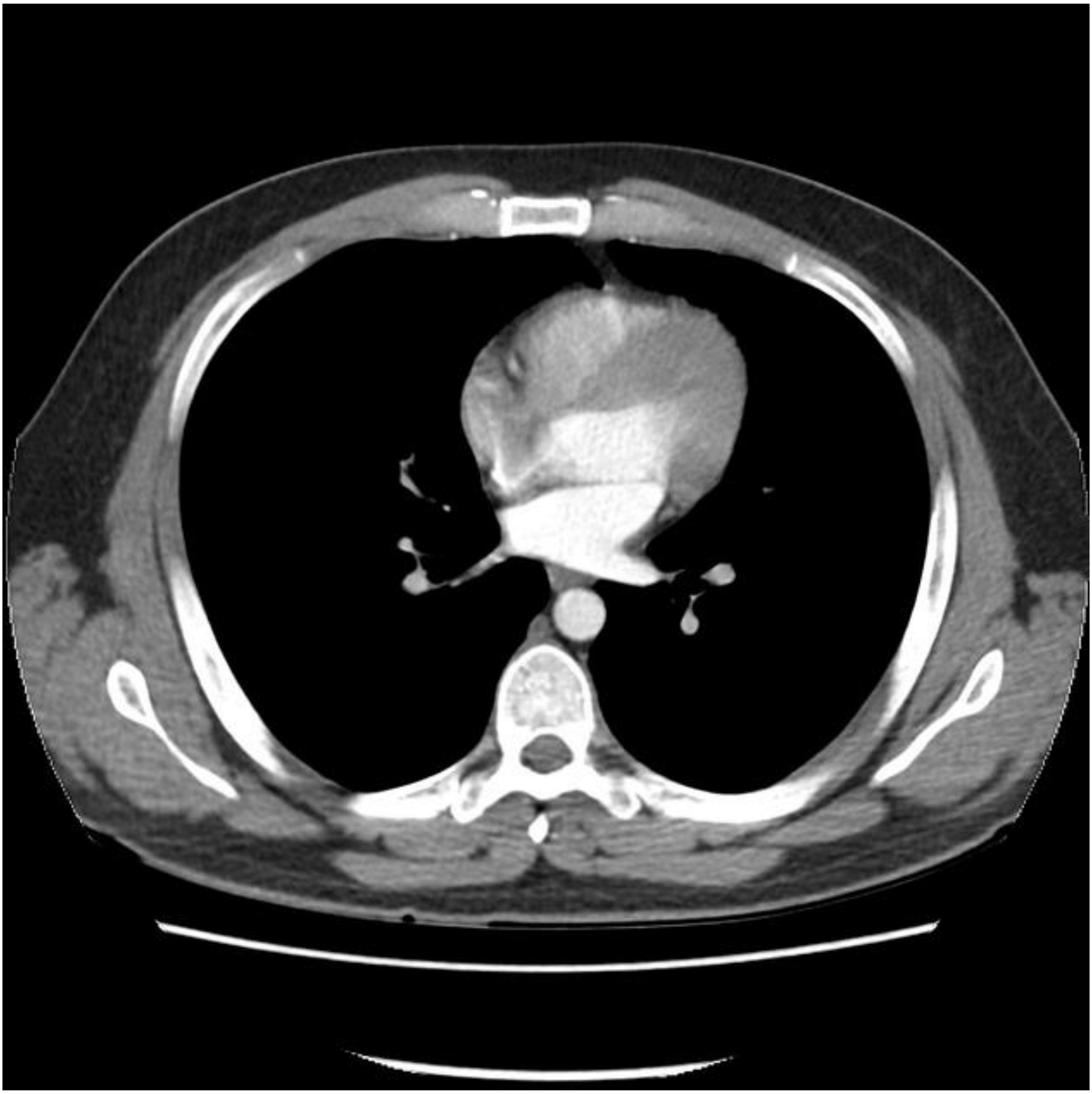


Fig :5.3:LH Image

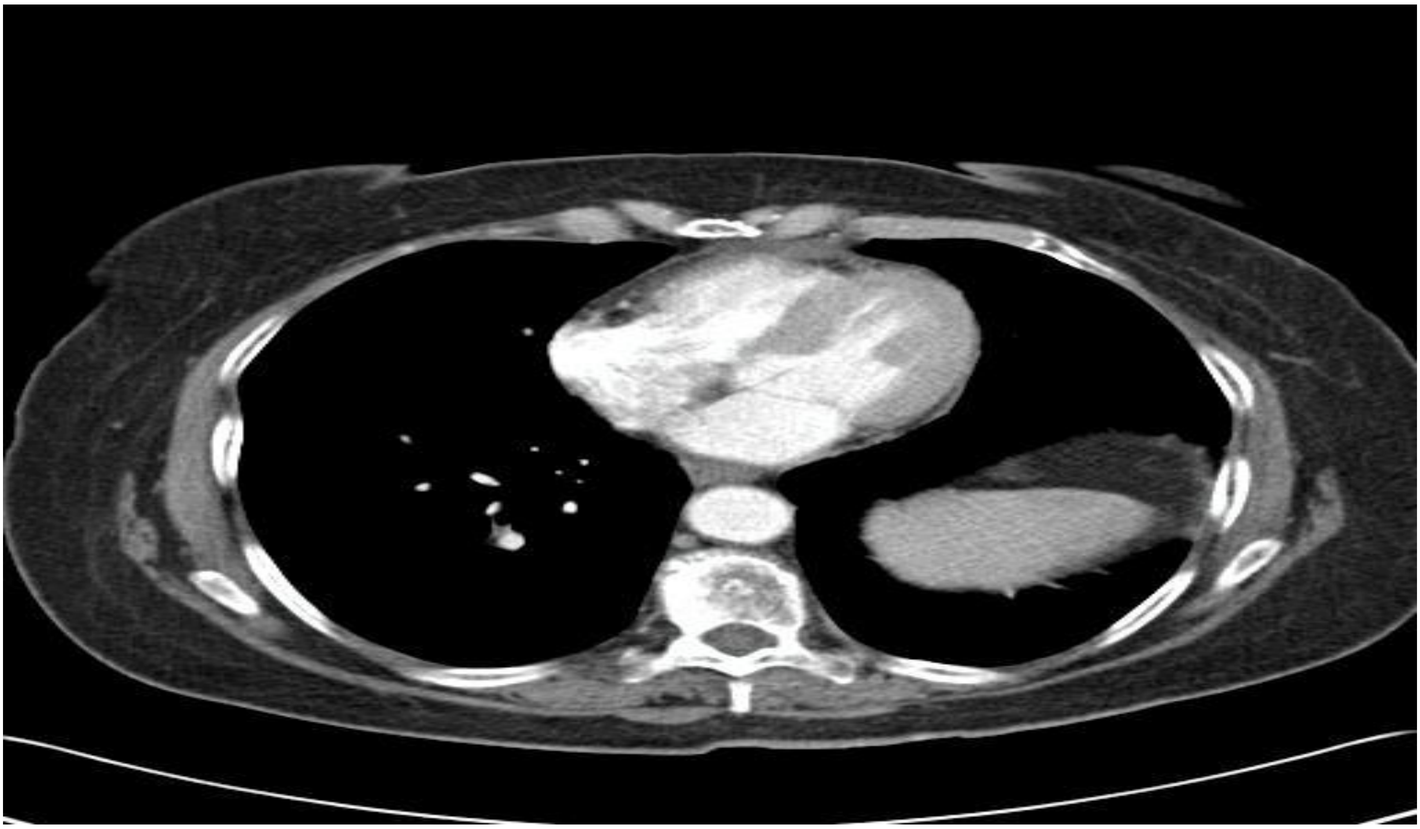


Fig:5.3 HL image

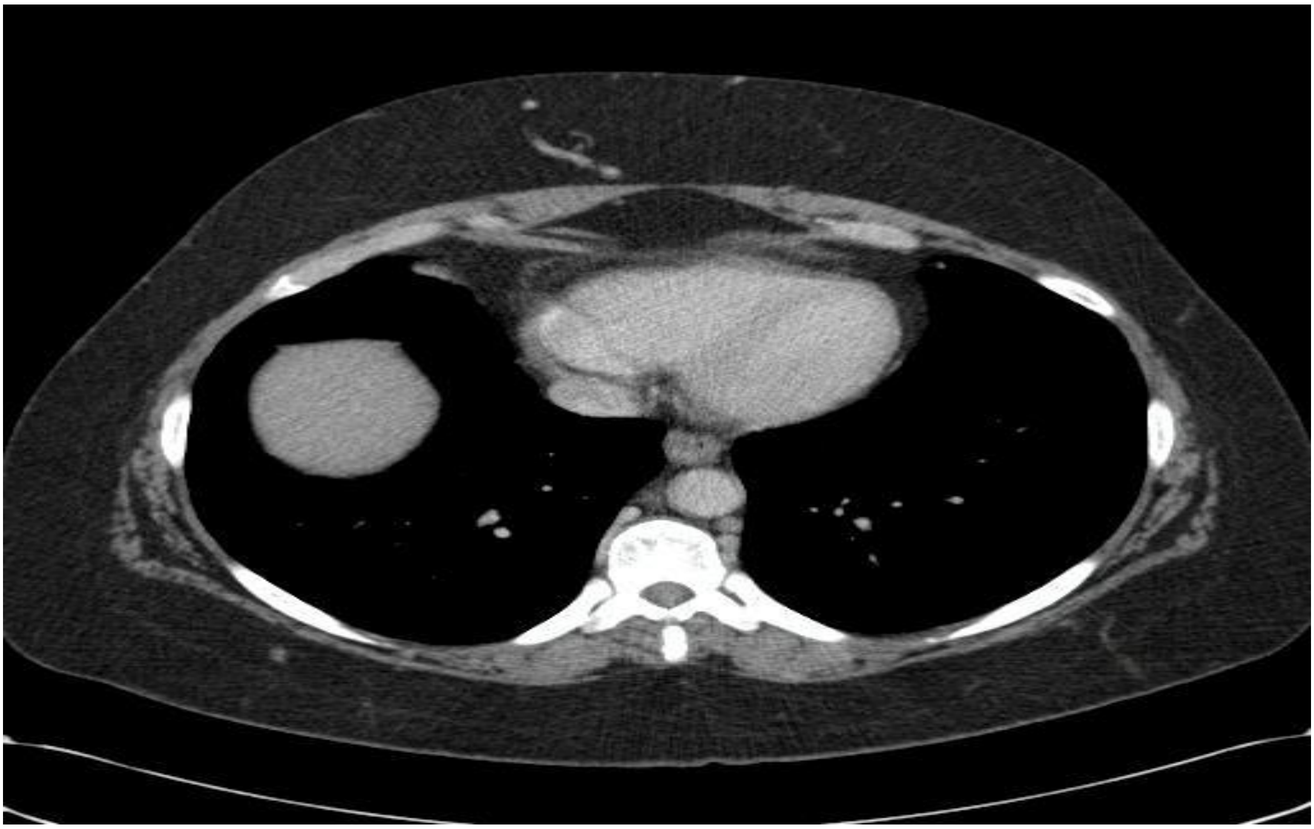


Fig:5.4:HL Image

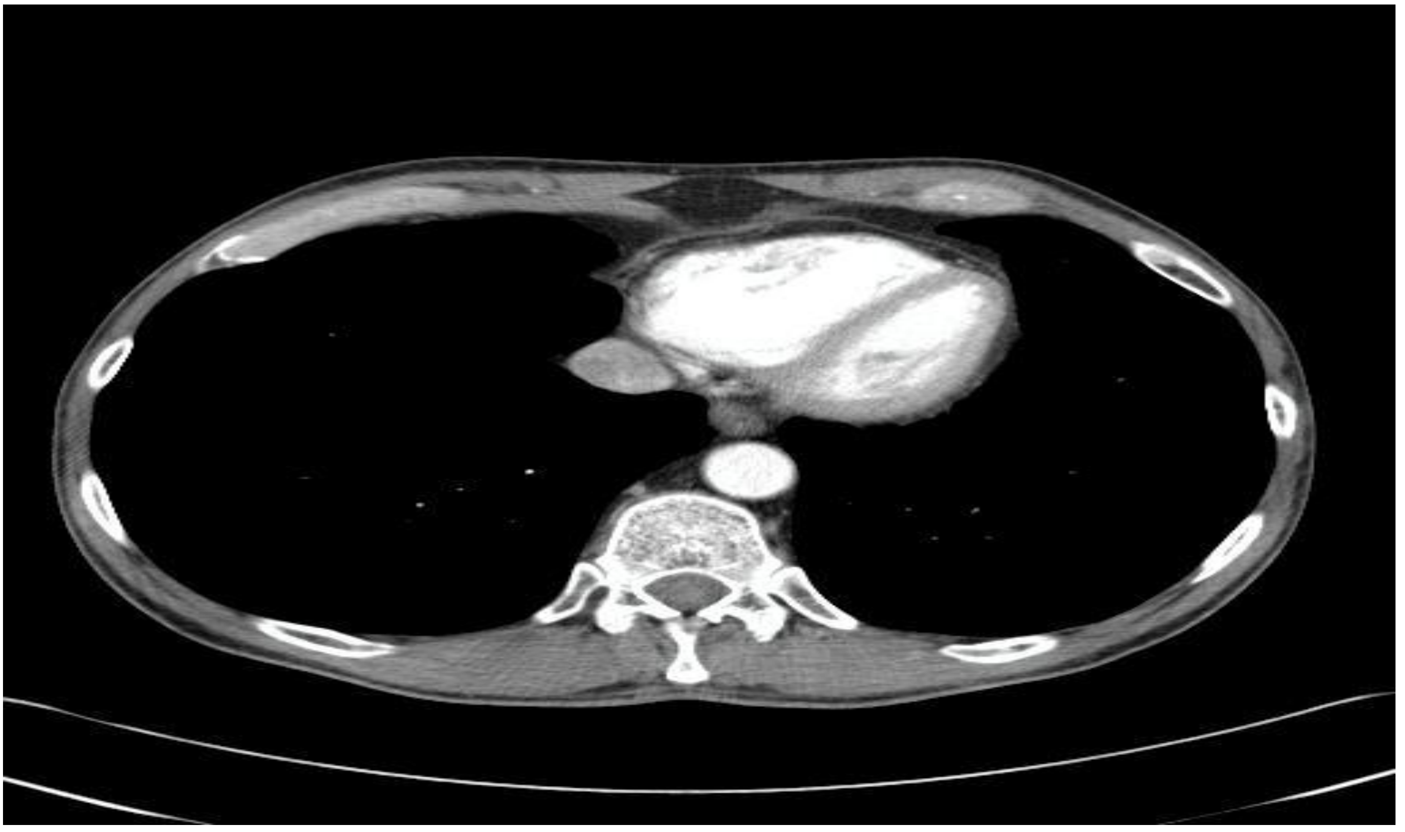


Fig:5.5:Gradient Image

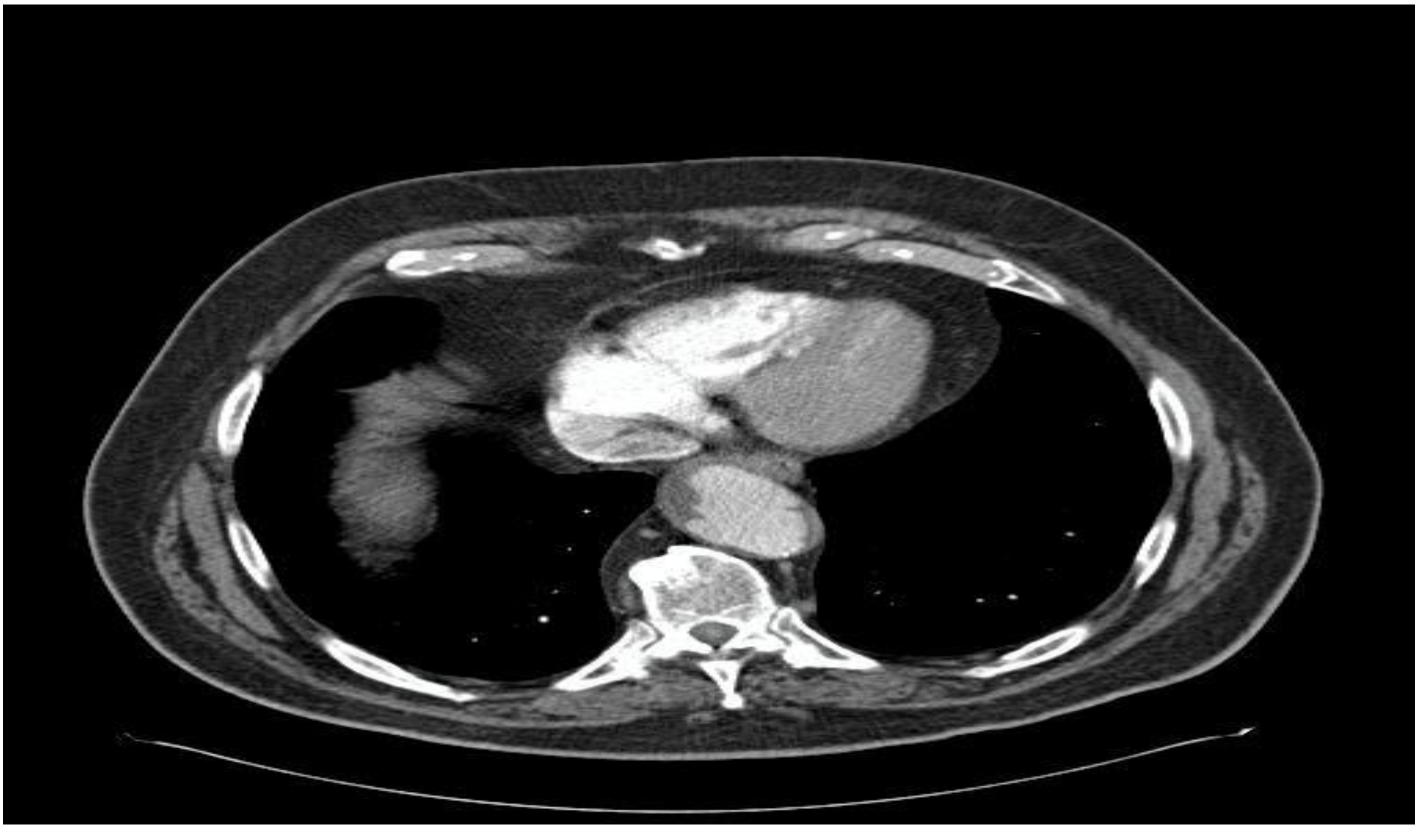


Fig:5.6:Gradient Image

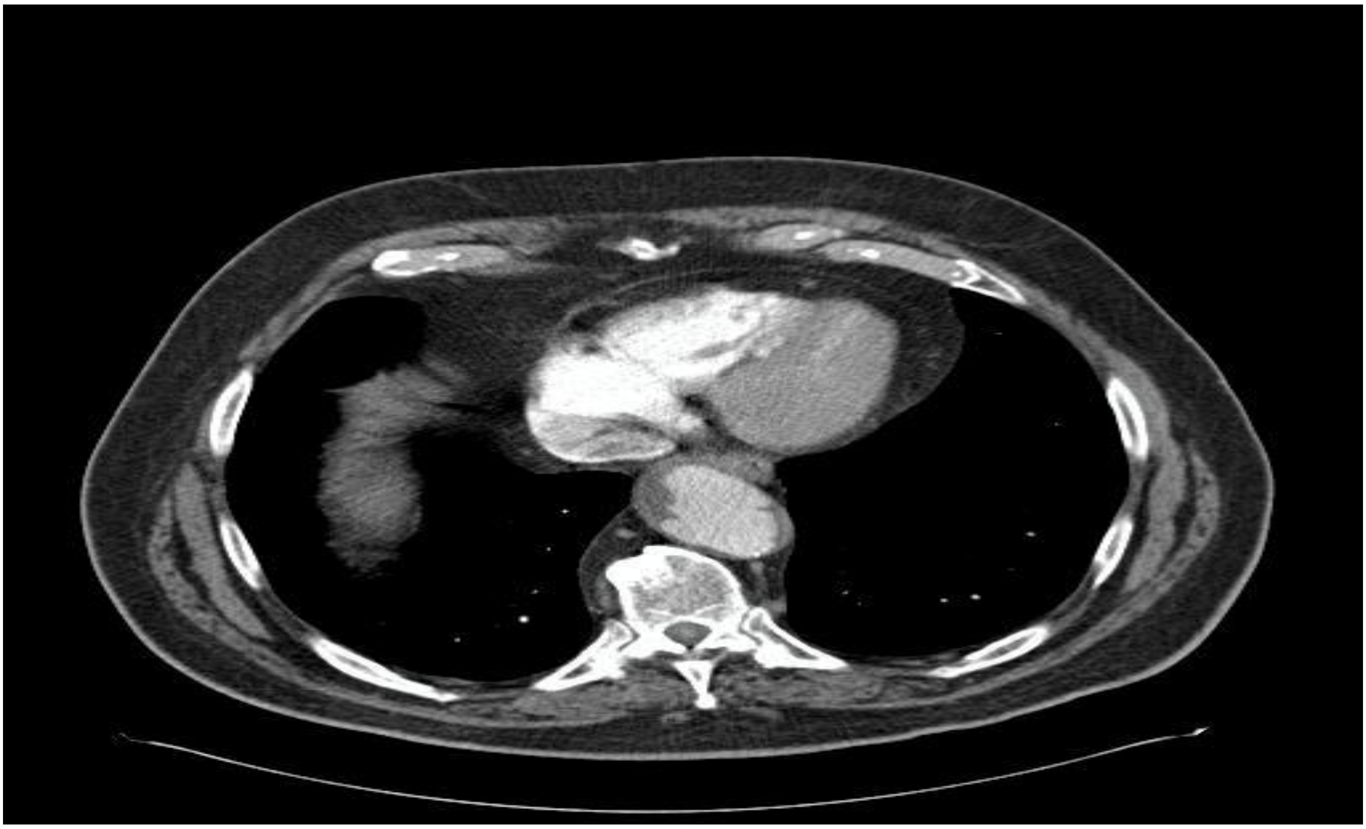


Fig:5.6:Output image

CHAPTER 6 APPLICATIONS AND ADVANTAGES

APPLICATIONS

→The application requirements for CT have almost covered all clinical departments, and have been commonly used in medical institutions.

→ Denoising can be a very effective approach to improving the quality of low-dose CT images.

→ Electronic surveillance The application concerns passive detection and finger printing of incoming radar signatures from an electronic surveillance platform.

ADVANTAGES

- It can be done faster with large amount of software.
- This method featured security, capacity and robustness, the three needed aspects of steganography that makes it useful in hidden exchange of information through text documents and establishing secrets communication.

CODE

```
% A Preliminary Study on Projection Denoising for Low-dose CT Imaging Using
% Modified Dual-Domain U-net
```

```
clc
clear all
close all
warning off
%% input image
[filename, pathname] = uigetfile({'*.jpg'},'pick an image');
I=imread(filename);
I=imresize(I,[256 256]);
I=rgb2gray(I);
nSig=imnoise(I,'poisson');
imwrite(nSig,'YO.jpg');
img=imcrop(nSig);
figure,imshow(img);title('Original image')
%% divide Low pixels to high pixels
[Low_Low,Low_High,High_Low,High_High]=dwt2(I ,'haar');
figure,subplot(2,2,1),imshow(mat2gray(Low_Low));title('LL Image');
subplot(2,2,2),imshow(Low_High);title('LH Image');
subplot(2,2,3),imshow(High_Low);title('HL Image');
subplot(2,2,4),imshow(High_High);title('HH Image');
if size(Low_Low,3)==3
Low_Low=rgb2gray(Low_Low);
Low_High=rgb2gray(Low_High);
High_Low=rgb2gray(High_Low);
High_High=rgb2gray(High_High);
end
a=imresize(I,[30 30]);
a = double(I);
[nr,nc]=size(I);
[x y] = meshgrid(1:nc,1:nr);
u = x;
v = y;
quiver(x,y,u,v);
[Gmag, Gdir] = imgradient(I,'prewitt');
figure
imshowpair(Gmag, Gdir, 'montage');
title('Gradient Magnitude, Gmag , and Gradient Direction');
%%%%%%% convolutional sparsity based morphological component analysis (CS-MCA) algoritham
T = 128;
T0 = .5;
G1 = I > T;
G2 = I <= T;
meanGL1 = mean(I(G1))
meanGL2 = mean(I(G2))
% 4)Compute a new threshold value
Tnew=(1/2) * (meanGL1 +meanGL2)
if (Tnew - T) < T0
end
k=0
I=nSig;
nSig=max(I);
k=6;
[Faf, Fsf] = FSdoubledualfilt;
[af, sf] = doubledualfilt;
im1 = double(imread('YO.jpg'));
im2 = double(imread('YO.jpg'));
% image decomposition
```

```
y = gradient_HP(im1,k);
w1=gray2rgb(y);
w1=num2cell(w1(:,1:7),2);
T = 6;
y = gradient_HP(im2,T);
w2=gray2rgb(y);
w2=num2cell(w2(:,1:7),2);
e_j=std(y);
% disp('standard deviation of enargy---');
disp(e_j);
% Image fusion process start here
for k1=1:k
    for p=1:2
        for d1=1:2
      for d2=1:3
                k=6;
                x = w1{k};
                y = w2{k};
                x=sort(x);     %%%%%%%%%%centerpixil with neighboor data caliculation
D  = (abs(x)-abs(y)) >= 0;
wf{k1}{p}{d1}{d2} = D.*x + (~D).*y; % image fusion
            end
        end
    end
end
load w
J=4;
y = Iterative_HS(w,J,Fsf,sf);
y=imadd(y,50);
out_data=double(y);
figure; imshow(mat2gray(out_data));title('final result image');
%imwrite(out_mca,'out_img.jpg');

%%%%%%%%Robust Image Watermarking Based on Multiscale Gradient Direction
Quantization%%%%%%%%%
warning off
clc;
clear all;
close all;
%%%%%%%Input Image%%%%%%%%%%%%%%%%%%%%

[filename, pathname] = uigetfile({'*.jpg'},'pick an image');

if isequal(filename,0) || isequal(pathname,0)
warndlg('File is not selected');
else
a=imread(filename);
imshow(a);title('Original image');
end

%%%%%%%%%%%%%%%%%% Apply Discrete wavelet transform %%%%%%%%%%%%%%%%%

[cA,cH,cV,cD]=dwt2(a ,'haar');
figure,subplot(2,2,1),imshow(mat2gray(cA));title('LL Image');
subplot(2,2,2),imshow(cH);title('LH Image');
subplot(2,2,3),imshow(cV);title('HL Image');
```

```
subplot(2,2,4),imshow(cD);title('HH Image');

%%%%%%%%CALICULTE gradient vector for each DWT scale%%%%%%%%%%
if size(cA,3)==3
cA=rgb2gray(cA);
cH=rgb2gray(cH);
cV=rgb2gray(cV);
cD=rgb2gray(cD);
end
[FX1, Gdir1] = imgradient(cA,'prewitt');
figure,subplot(2,2,1),imshow(mat2gray(FX1));title('gradient LL comp');
[FX2, Gdir2] = imgradient(cH,'prewitt');
subplot(2,2,2),imshow(mat2gray(FX2));title('gradient LH comp');
[FX3, Gdir3] = imgradient(cV,'prewitt');
subplot(2,2,3),imshow(mat2gray(FX3));title('gradient HL comp');
[FX4, Gdir4] = imgradient(cD,'prewitt');
subplot(2,2,4),imshow(mat2gray(FX4));title('gradient HH comp');
%FX=cat(4, FX1, FX2, FX3, FX4);
%FX=a;

a=imresize(a,[30 30]);
a = double(a);
[nr,nc]=size(a);
[dx,dy] = gradient(a);
[x y] = meshgrid(1:nc,1:nr);
u = x;
v = y;
quiver(x,y,u,v);



[Gmag, Gdir] = imgradient(a,'prewitt');
figure
imshowpair(Gmag, Gdir, 'montage');
title('Gradient Magnitude, Gmag (left), and Gradient Direction, Gdir (right), using Prewitt method')
nSig=imnoise(a,'poisson');
figure,imshow(nSig);

nSig=std(nSig);

%GHE_out=HMIS_Main(a,nSig);
%figure,imshow(GHE_out);
```

CONCLUSION AND FUTURE SCOPE

CONCLUSION

→ Denoising of the projections with the proposed idea leads to a substantial improvement of the reconstructed image in terms of noise level, spatial resolution, and visual quality.

→ Low dose images on the other hand have poor quality due to the presence of AWGN noise.

→ CT scan imaging is one of the most widely used imaging modality. The quality of the CT images improves as the radiation dose.

FUTURE SCOPE

→ Ultra low dose CT Scan:researchers pay attention to the reconstruction algorithms for high-quality ultra low dose CT scan.

REFERENCES

[1] “Computed Tomography (CT).” National Institute of Biomedical Imaging and Bioengineering, U.S. Department of Health and Human Services, www.nibib.nih.gov/science-education/sciencetopics/computed-tomography-ct.

[2] Modica, Michael J., et al. “The Obese Emergency Patient: Imaging Challenges and Solutions.” RadioGraphics, 4 May 2011, pubs.rsna.org/doi/full/10.1148/rg.313105138.

[3] Wang, Jing, et al. “An Experimental Study on the Noise Properties of X-Ray CT Sinogram Data in Radon Space.” Physics in Medicine and Biology, U.S. National Library of Medicine, 21 June 2008.

[4] Kent, Chloe. “CT Scans of Coronavirus Patients Reveal Nature and Extent of Lung Damage.” Verdict Medical Devices, 4 Mar. 2020, www.medicaldevicenetwork.com/news/coronavirus-ct-scans/.

[5] Ellenberg, Jordan (February 22, 2010). "Fill in the Blanks: Using Math to Turn Lo-Res Datasets Into Hi-Res Samples". Wired. 12 March 2020.

[6] Methods for Clinical Evaluation of Noise Reduction Techniques in Abdominopelvic CT. Ehman et. al. https://pubs.rsna.org/doi/pdf/10.1148/rg.344135128

[7] “Abdomen CT Scan.” Insight Medical Imaging, 7 Jan. 2020, x-ray.ca/services/computed-tomography/abdomenct-scan/.

[8] Orenstein, David. “CT Lung Screening Appears CostEffective.” CT Lung Screening Appears Cost-Effective | News from Brown, 5 Nov. 2014, news.brown.edu/articles/2014/11/cancercost.

[9] Naqi, Rohana&Azeemuddin, Muhammad. (2013). Naeglaeria infection of the central nervous system, CT scan findings: A case series. Journal of the Pakistan Medical Association. 63. 399 - 402.

[10] Romano, et al. “The Little Engine That Could: Regularization by Denoising (RED).” ArXiv.org, 3 Sept. 2017, arxiv.org/abs/1611.02862.

A

PROJECT REPORT

On

AIR AND NOISE POLLUTION MONITORING SYSTEM OVER AN IOT

Submitted by

1) Ms.E.Sowmyasri (17K81A0476)

2) Ms.T.Kavya (17K81A04B4)

3) Mr.M.Harish (17K81A0492)

in partial fulfillment for the award of the

degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Mrs.K.Anitha

Assistant professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINNERING

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

NBA, NAAC A+ Accredited

(Approved by A.I.C.T.E, New Delhi, Affiliated to JNTU Hyderabad) Dhulapally, Secunderabad, Telangana (India)-500100

BONAFIDE CERTIFICATE

This is to certify that the project entitled "AIR AND NOISE POLLUTION MONITORING SYSTEM OVER AN IOT", is being submitted by **Ms.E.Sowmyasri (17K81A0476), Ms. T.Kavya (17K81A04B4), Mr.M.Harish (17K81A0492)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** IN ELECTRONICS AND COMMUNICATION ENGINEERING is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

	Head of the Department
K.Anitha	Dr. B. Hari Krishna
Department of ECE	Department of ECE

Internal Examiner	External Examiner

Place:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **E.SOWMYASRI** WITH ROLL NO.**17K81A0476, T.KAVYA** WITH ROLL NO.**17K81A04B4**, **M.HARISH** WITH ROLL NO.**17K81A0492**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**AIR AND NOISE POLLUTION MONITORING SYSTEM OVER AN IOT**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of **Bachelor of Technology** in Department of 'Electronics and Communication Engineering', session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled "AIR AND NOISE POLLUTION MONITORING SYSTEM OVER AN IOT" is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

E.Sowmyasri (17K81A0476)
T.Kavya (17K81A04B4)
M.Harish (17K81A0492)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to Dr. B. Hari Krishna, **Head &Professor**, in the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept. project coordinator Dr.A.Anand**, professor in ECE for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide **Mrs.K.Anitha**, **Assistant professor**,Department of ECE and our class project coordinator **Mr.Karthik,Assistant professor**,Department of ECE for guidance and encouragement in carrying out this project work.

Finally, We are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar** , **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. E.Sowmyasri
2. T.Kavya
3. M.Harish

ABSTRACT

In the present era, air and noise pollution is the growing hazardous issue. It is necessary to monitor air quality and keep it under control for a better future and healthy living for all. Here we propose an air quality as well as sound pollution monitoring system that allows us to monitor and check live air quality as well as sound pollution in a particular area through IOT. System uses air sensors to sense presence of harmful gases/compounds in the air and constantly transmit this data to microcontroller. Also, system keeps measuring sound level and reports it to the online server over IOT. The sensors interact with microcontroller which processes this data and transmits it over internet. This allows authorities to monitor air pollution in different areas and take action against it. Also, authorities can keep a watch on the noise pollution near schools, hospitals and no honking areas, and if system detects air quality and noise issues it alerts authorities so they can take measures to control the issue.

INDEX

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

OVERVIEW TO THE PROJECT

Air and sound pollution is a growing issue these days. It is necessary to monitor air quality and keep it under control for a better future and healthy living for all. Here we propose an air quality as well as sound pollution monitoring system that allows us to monitor and check live air quality as well as sound pollution in a particular areas through IOT. System uses air sensors to sense presence of harmful gases/compounds in the air and constantly transmit this data to microcontroller. Also system keeps measuring sound level and reports it to the online server over IOT. The sensors interact with microcontroller which processes this data and transmits it over internet. This allows authorities to monitor air pollution in different areas and take action against it. Also authorities can keep a watch on the noise pollution near schools, hospitals and no honking areas, and if system detects air quality and noise issues it alerts authorities so they can take measures to control the issue.

In infrastructure and industrial plants the rapid growth creating environmental issues like pollution (Air, Water, Noise), climate change, malfunctioning and has greatly consequence for the requirement of an, operationally adaptable, efficient, cheap and smart monitoring systems. In this context where combination of many challenges of computer science, wireless communication and electronics; the Smart Sensor Networks are an emerging field of research. In this paper a solution to monitor the air and noise pollution levels in industrial environment or by using wireless embedded computing system a particular area of interest is proposed. The technology like Internet of Things (IoT) is included in the form of solution which is outcome of merged field of computer science and electronics. For monitoring the fluctuation of parameters like noise and air pollution levels from their normal levels in this case the sensing devices are connected to the embedded computing system. For the requirement of continuous monitoring, controlling and behavior analysis this model is adaptable and distributive for any infrastructural environment. For two or three parameters like noise, CO and radiation levels the implementation is tested with respect to the normal behavior levels or given specifications which provide a monitoring over the pollution control to make the environment smart and Eco-friendly.

MATERIAL REQUIREMENT:

HARDWARE REQUIREMENT:

1.Arduino uno

2.MQ2 gas sensor

3.Sound sensor

4.LCD Display

5.ESP8266 Wi-Fi Module

6.Power supply unit

SOFTWARE REQUIREMENT :

- Arduino IDE
- Proteus software

PROCUMENT OF EQUIPMENT:

We brought all the required hardware components in a electronic manufacturing shop and before assembling and interfacing hardware components to Arduino UNO micro-controller, we connected individual sensors to Arduino in Proteus Software and understood working of every individual sensor in Proteus Software. Now, we assembled all sensors and interfaced to Arduino UNO micro-controller using connecting wires through bread board and we wrote program to work all the hardware sensors in Arduino IDE, we understood the working of all sensors in a practical way. We calculated all the threshold limit for the harmful gases and sound intensity and all the information is sent to the online server.

INTRODUCTION TO EMBEDDED SYSTEM

An embedded system is a combination of computer hardware and software, either fixed in capability or programmable, designed for a specific function or functions within a larger system. Industrial machines, agricultural and process industry devices, automobiles, medical equipment, cameras, household appliances, airplanes, vending machines and toys, as well as mobile devices, are 2 possible locations for an embedded system. An embedded system is a dedicated computer system designed for one or two specific functions. This system is embedded as a part of a complete device system that includes, such as electrical and mechanical components[1]. The embedded system is unlike the generalpurpose computer, which is engineered to manage a wide range of processing tasks. Because an embedded system is engineered to perform certain tasks only, design engineers may optimize size, cost, power consumption, reliability and performance. Embedded systems are typically produced on broad scales and share across a variety of environments and applications. An embedded system is one that has computer-hardware with software embedded in it as one of its most important components. It is a dedicated computer-based system for an application or product. An embedded system has three main components. 1. It has hardware 2. It has main application software. 3. It has a real time operating system (RTOS) An embedded system has software designed to keep in view three constraints: (i) Available system memory. (ii) Available processor speed. (iii) The need to limit power dissipation when running the system . An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious. Some of these computers are however very simple systems as compared with a personal computer[2]. The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a 3 variety of different purposes[2]. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes. The applications software on such processors is sometimes referred to as firmware. The simplest devices consist of a single microprocessor (often called a "chip”), which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine[3]. As the embedded system is the combination of both software and hardware.

HISTORY OF EMBEDDED SYSTEM

Embedded systems date back to the 1960s. Charles Stark Draper developed an integrated circuit (IC) in 1961 to reduce the size and weight of the Apollo Guidance Computer, the digital system installed on the Apollo Command Module and Lunar Module. The first computer to use ICs, it helped astronauts collect realtime flight data. In 1965, Auto now a part of Boeing, developed the D-17B, the computer used in the Minuteman I missile guidance system.

BLOCK DIAGRAM OF EMBEDDED SYSTEM

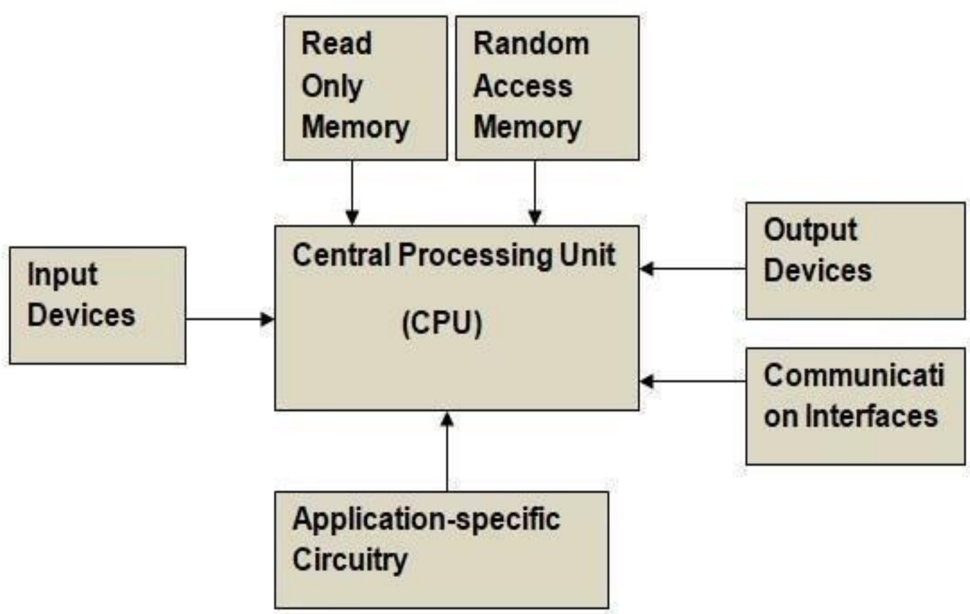


Fig.1.2.1: Block diagram of embedded system

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

EMBEDDED SYSTEM HARDWARE:

Embedded system hardware can be microprocessor- or microcontroller-based. In either case, an integrated circuit is at the heart of the product that is generally designed to carry out computation for real-time operations[4]. Microprocessors are visually indistinguishable from microcontrollers, but while the microprocessor only implements a central processing unit (CPU) and, thus, requires the addition of other components such as memory chips, microcontrollers are designed as self contained systems.

EMBEDDED SYSTEM SOFTWARE:

A typical industrial microcontroller is unsophisticated compared to the typical enterprise desktop computer and generally depends on a simpler, less-memory intensive program environment. The simplest devices run on bare metal and are programmed directly using the chip CPU's machine code language. Often, embedded systems use operating systems or language platforms tailored to embedded use, particularly where real-time operating environments must be served. At higher levels of chip capability, such as those found in SoCs designers have increasingly decided the systems are generally fast enough and the tasks tolerant of slight variations in reaction time that near-real-time approaches are suitable.

EXAMPLES OF EMBEDDED SYSTEM

Some examples of embedded systems are below

- ATM
- Digital Cameras
- Microwave ovens
- Factory controllers
- Washing machine 5
- Calculator

- TV remote
- Traffic lights
- Digital watches
- Mp3 player
- Video games consoles
- Printers
- GPS receivers
- Dishwashers
- Thermostats
- Anti-lock banking system
- Medical imaging

TYPES OF EMBEDDED SYSTEM

1. Real Time Systems

Real-time systems are those which give a quick response to critical situations. They are used in military, medical and industrial applications. Engineers working in these systems have high demand is current days. To develop the real-time embedded system we require timing analysis, multitasking design, debugging, cross-platform testing and architecture design. In these systems, quick response is very important.

2. Standalone Embedded System

This type of embedded system works for itself as a device without needing any interconnected computer. It can take data in the form of Analog or digital signals. This system first process data and then outputs data by displaying on the screen.

3. Networked Embedded System

Networked embedded systems are those systems which are connected to the network to give output to the attached resources[5]. The devices in the networked embedded system are connected to the network with network interfaces. The network can be either a local area network (LAN) or a wide area network (WAN).

4. Mobile Embedded System

Mobile embedded systems are limited in resources including memory. Examples of mobile embedded systems include

- Personal digital assistants (PDA)
- Cellular phones
- Mp3 players
- Digital cameras

5. Small-Scale Embedded System

Small-scale embedded systems consist of 8-16 bit microcontroller. This system can perform tasks at a small level. They have on-chip ROM and RAM. Small scale systems can be even activated by the battery. The tools used to develop small-scale embedded systems are an editor, cross assembler, assembler and integrated development environment (IDE). The purpose of this system is not computation but to control as a computer embedded inside it. It behaves as a component of a computer and its function is not to compute. The small-scale system is dedicated to some specific task.

6. Medium Scale Embedded System

This embedded system has 16-32 bit microprocessor or microcontroller with external RAM and ROM They can perform medium to complex level works. The integration between hardware and software is complex in these embedded systems. Programming languages used to develop medium scale embedded systems include Java, C, Visual C++, debugger, C++, RTOS, simulator, source code engineering tool and IDE.

7. Sophisticated Embedded System

The embedded system which can do large-scale works with multiple 32-64 bit chips is known as sophisticated embedded systems[6]. They can perform distributed work on a large scale. The complexity of hardware and software is very high in these systems. In sophisticated embedded systems, hardware and software are assembled together on large scale and designing of hardware products is also included in these systems.

APPLICATIONS OF EMBEDDED SYSTEM

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

MICROCONTROLLER

A microcontroller is a small computer on a single metal oxide semiconductor (MOS) integrated circuit (IC)chip. A microcontroller contains one or more CPUs along with memory and programmable input/output peripherals. Program memory in the form of ferroelectric RAM, NOR flash or OTP ROM is also often included on chip, as well as a small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications consisting of various discrete chips[7]. In modern terminology, a microcontroller is similar to, but less sophisticated than, a system on a chip (SoC). SoC may include a microcontroller as one of its components, but usually integrates it with advanced peripherals like graphics processing unit (GPU), Wi-Fi module, or one or more coprocessors.

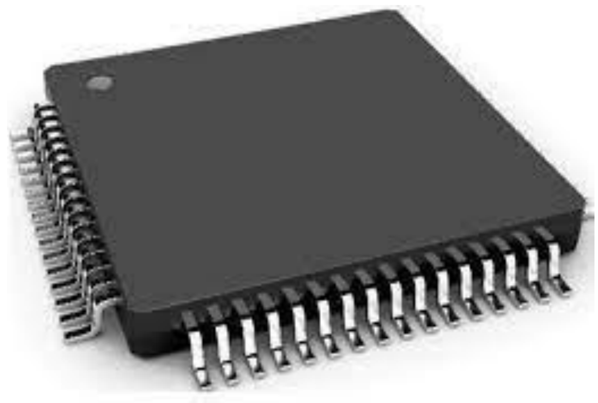


Fig.1.3.1: Microcontroller

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems. In the context of the internet of things, microcontrollers are an economical and popular means of data collection, sensing and actuating the physical world as edge devices. Some microcontrollers may use four-bit words and operate at frequencies as low as 4 kHz for low power consumption (single-digit milliwatts or microwatts). They generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (CPU clock and most peripherals off) may be just nanowatts, making many of them well suited for long lasting battery applications[8]. Other microcontrollers may serve performance-critical roles, where they may need to act more like a digital signal processor (DSP), with higher clock speeds and power consumption.

HISTORY

The origins of both the microprocessor and the microcontroller can be traced back to the invention of the MOSFET (metal-oxide-semiconductor field-effect transistor), also known as the MOS transistor. It was invented by Mohamed M. Atalla and Dawon Kahng at Bell Labs in 1959, and first demonstrated in 1960.9 The same year, Atalla proposed the concept of the MOS integrated circuit, which was an integrated circuit chip fabricated from MOSFETs. By 1964, MOS chips had reached higher transistor density and lower manufacturing costs than bipolar chips[2]. MOS chips further increased in complexity at a rate predicted by Moore's law, leading to large-scale integration (LSI) with hundreds of transistors on a single MOS chip by the late 1960s. The application of MOS LSI chips to computing was the basis for the first microprocessors, as engineers began recognizing that a complete computer processor could be contained on a single MOS LSI chip.

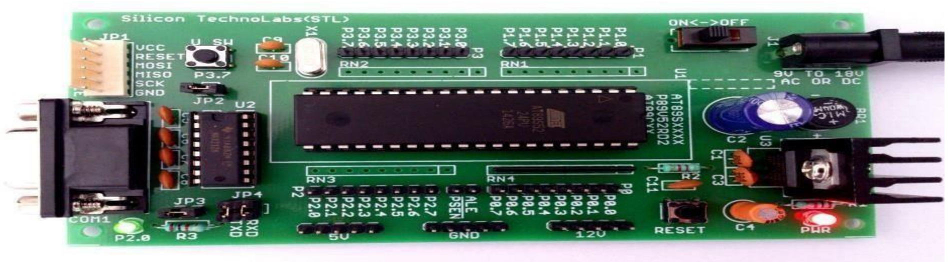


Fig.1.3.2: 8051 Microcontroller Board

Most microcontrollers at this time had concurrent variants. One had EPROM program memory, with a transparent quartz window in the lid of the package to allow it to be erased by exposure to ultraviolet light. These erasable chips were often used for prototyping. The other variant was either a mask programmed ROM or a PROM variant which was only programmable once. For the latter, sometimes the designation OTP was used, standing for "one-time programmable". In an OTP microcontroller, the PROM was usually of identical type as the EPROM, but the chip package had no quartz window; because there was no way to expose the EPROM to ultraviolet light, it could not be erased. Because the erasable versions required ceramic packages with quartz windows, they were significantly more expensive than the OTP versions, which could be made in lower-cost opaque plastic packages. For the erasable variants, quartz was required, instead of less expensive glass, for its transparency to ultraviolet light— to which glass is largely opaque—but the main cost differentiator was the ceramic package itself .

In 1993, the introduction of EEPROM memory allowed microcontrollers (beginning with the Microchip PIC16C84) to be electrically erased quickly without an expensive package as required for EPROM, allowing both rapid prototyping, and in-system programming. (EEPROM technology had been available prior to this time, but the earlier EEPROM was more expensive and less durable, making it unsuitable for low-cost mass-produced microcontrollers.) The same year, Atmel introduced the first microcontroller using Flash memory, a special type of EEPROM. Other companies rapidly followed suit, with both memory types. Nowadays microcontrollers are cheap and readily available for hobbyists, with large online communities around certain processors.

INTRODUCTION TO INTERNET OF THINGS

The Internet of things (IoT) describes the network of physical objects—a.k.a. "things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet.Things have evolved due to the convergence of multiple technologies, real-time analytics, machine learning, ubiquitous computing, commodity sensors, and embedded systems.[1] Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", including devices and appliances (such as lighting fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers. The IoT can also be used in healthcare systems.

Internet of Things (IOT) is a network of devices that connect directly with each other to capture and share data through a secure service layer that connects to a central command and control server in the cloud. The closure\look suggest that the way people collect, record and analyze data—not just in health care but in every industry today. The idea of devices connecting directly with each other is basically called Internet of Things.

The Internet of Things also a called the Internet of Objects, refers to a wireless network between objects, usually the network will be wireless and self-configuring. Internet of Things (IOT) is one of the major component advances in present time that links the internet with everyday sensor and working devices. Smart objects play an important role in the Internet of Things vision, since embedded communication and information technology would have the potential to change the utility of these objects. Using sensors they are able to recognize their condition, and via built-in connecting power they would be able to interact with each other.

Benefits of IOT

Ubiquitous networks -: Personal Wi-Fi on your smart phones and on many of the other devices. Everyone (and everything) wants as well as needs to be connected.

Connected computing -:We want all of the devices, smart phones, televisions(colored or black and white), dvd players, vehicles etc. to keep record of what we are doing, seeing, reading, and/or listening to as we sway through the day, from one place to another – the handoffs from device to device is happening already.

Analytics-as-a-Service-: The API and App economies are already wide and growing which enables to "do something interesting" as long as it can be connected to an API or can invoke an App that carries out a network-based service. The thing is a data generator as well as collector that learns from, makes forecast, and maybe even takes data-driven actions in response to the data that are collected too.

Marketing automation-: Smart phone customer engagement, geological-location, Apple's iBeacon etc. are all developing a network of knowledge and information regarding customers' locations, intentions, preferences, as well as buying patterns. Obviously, the degree of geological location-based knowledge needs to maintain the right balance between user privacy as well as the timely delivery of important and significant products and services to the particular user.

Supply Chain Analytics-: Delivering the just-in-time products at the time of need (inclusive of the use of RFID-based tracking). Significantly, everything is a customer (inclusive of machines, automobiles, manufacturing plants, ATM machines, etc.), as well as the IOT is monitoring, watching, as well as waiting for the product needs to arise.

MOTIVATION OF THE PROJECT

In this era of modernization, technologies are advancing rapidly. Every day we realize some new technology coming in market to simplify our lives more than ever. Here this system is created to make one more system in the fulfillment of the goal of smart city. Here this system used to monitor the air and noise pollution without human intervention and helps in taking measures to control the pollution if it exceeds the normal range to better quality of life.

CHAPTER 2

LITERATURE SURVEY AND EXISTING MODELS

LITERATURE REVIEW ON RESEARCH AREA:

1. IOT based Air and Sound Pollution Monitoring System : The air pollution monitoring system contains sensors to monitor the interested pollution parameter in environment. They simulated the three air pollutants gases including carbon monoxide, carbon dioxide &sulphur dioxide in air because these gases decide the degree of pollution level. They also applied the approach in various applications like leaking cooking gas in our homes, to alert the workers in oil &gas industry to detect the leakage etc.

2. Implementation of an Efficient Noise and Air Pollution Monitoring System Using Internet of Things (IoT) A Smart City is one with at least one initiative addressing one or more of the following six characteristics: Smart Governance, Smart People, Smart Living, Smart Mobility, Smart Economy and Smart Environment. In this simulation, these three gases are successfully tested in four areas. Then extended the simulated results to update in web.

3. An IoT Based Automated Noise and Air Pollution Monitoring System: The proposed embedded device for monitoring noise and air levels in atmosphere to make the environment intelligent or interactive with object. The proposed model is adaptable and distributive in nature to monitor the environmental parameters. The architecture is developed for noise and air pollution monitoring.

4. Implementation of an Efficient Noise and Air Pollution Monitoring System Using Internet of Things (IoT) : In the present era, air and noise pollution is the growing hazardous issue. It is necessary to monitor air quality and keep it under control for a better future and healthy living for all. Here we propose an air quality as well as sound pollution monitoring system that allows us to monitor and check live air quality as well as sound pollution in a particular areas through IOT.

5. A System For Monitoring Air And Sound Pollution Using Ardunio Controller With IOT Technology : Some of the research works carried out for monitoring the pollution parameters in a

particular area of interest for making the environment smart in that area, different techniques and methods which were used in the past discussed in this section

6. Industrial Air Pollution Monitoring and Analysis System: An embedded system for hazardous gas detection has been implemented; here only two gases (LPG and Propane) have been detected for demo purpose. The gas sensors and the critical level of the respective gas should be known, and then this system can be implemented for detecting various gases either in domestic area such as places of educational institutions, residential and industrial areas which avoids endangering of human lives. This system provides quick response rate and the diffusion of the critical situation can be made faster than the manual methods

REVIEW OF RELATED LITERATURE:

The purpose of this project is to identify the harm caused by the air and sound pollution to the environment. Pollution in simple words can be explained by, the presence of an foreign object in the environment which has harmful effect, we as a society have to ensure that all the pollution levels are maintained to the minimum, In today's world, pollution is the biggest concern, pollution has a lot of harmful effects on the wellbeing of the people. Our project is a small step to this big initiative to ensure that pollution stays under control. This project helps in detecting the major gases in air and the decibel levels in the surrounding environment. Our project will be a boon to the society as our project will be making sure that every individual will be able to keep a track of the pollution from the website . It is the need of the hour to monitor air quality and keep it under control for a better future and healthy living for all.

The motive of making a smart city can be fulfilled by using technology, thus making the life better and also enhancing the quality of services, therefore meeting every individual's needs. With modern technology in fields of information and communication, it has become easy to interact with the authorized people of city to tell the where abouts of the area or city, how well the city is developing and how to make it possible to achieve a better life quality. In this system, an application was created to make one more step in the fulfillment of the goal. An area is analyzed for evaluating how much pollution is affecting the area. The components of gases and sound intensity their amounts are calculated and checked. If the amount is higher than normal then the officials are reported about it. After that the people are made to clear the area and taken to a safe place.

CONCLUSION ON REVIEWS:

Hence as the technology is advancing day by day and we have to follow the new technologies so in the proposed system we used internet of things for makings things smarter. Here in the proposed system it consists of two units – in the first unit , we connect all the sensors to the Arduino uno that the sensors sense the concentration of harmful gases and sound intensity and sends to Arduino uno. In the second unit ,this transmitted information is displayed on the lcd and simultaneously sends to the online server.

CHAPTER 3

PROJECT DESIGN

BLOCK DIAGRAM:

The system is used for measuring air and noise pollution in the environment .The system consists of gas sensor to measure the concentration of harmful gases in the environment and sound sensor for measuring intensity of sound .These sensed information is transmitted to the Arduino and it process the information and continuously displays these values on the lcd display which is connected to it, simultaneously this information is transmitted to the online server by the wi-fi module .By using the website thingspeak we can monitor the level of air and noise pollution in the environment. This allows authorities to monitor air pollution in different areas and take action against it.

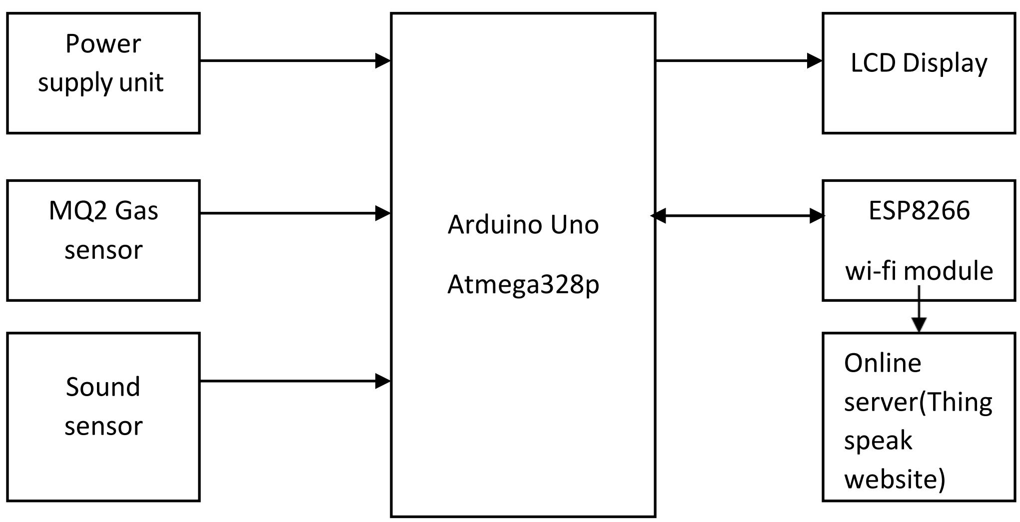


Fig.3.1.1: Block Diagram

The blocks of the circuit are :

1.Arduino uno

2. MQ2 gas sensor

3. Sound sensor

4.LCD Display

5.ESP8266 Wi-Fi Module

6.Power supply unit

7.Online server

1. Arduino UNO

Arduino is open source software. Arduino Uno is a microcontroller, ATmega328Pcontain in it. It has 14 digital i/o pins and 6 analog input pin, a 16 MHz quartz crystal, a USB connection, a power jack and a reset button. Simply connect to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. Embedded C program is used for coding.

2. MQ2 gas sensor

Gas sensors main aim is to sense hazardous gases that evolve its surroundings. It detects the concentrations of combustible gas in the air and outputs its reading as an analog voltage. The sensor can measure concentrations of flammable gas of 300 to 10,000 ppm. The sensor can operate at temperatures from -20 to 50°C and consumes less than 150 mA at 5 V. This sensor is connected to LCD and when there is a leakage of gas this sensor senses and a message is displayed on the LCD indicating the leakage of gas. This sensor is also connected to a buzzer and the buzzer sounds whenever the gas leakage is detected.

3. Sound sensor

The sound sensor is a small board that combines a microphone (50Hz-10kHz) and some processing circuitry to convert sound waves into electrical signals.This electrical signal is fed to on-board LM393 High Precision Comparator to digitize it and is made available at OUT pin. The module has a built-in potentiometer for sensitivity adjustment of the OUT signal. We can set a threshold by using a potentiometer; So that when the amplitude of the sound exceeds the threshold value, the module will output LOW otherwise HIGH.

4. LCD

The LCD is used to display when ever there is a gas leakage or fire detected in the area. When ever there is a gas leakage the Arduino that is connected to the sensor and LCD sends a message to LCD to display ‘Gas leakage’. When there is fire outbreak the Arduino again sends a message to LCD to display ‘Fire Detected’ notifying us if there is a gas leakage or fire detected in the vicinity.

5.ESP8266 Wi-Fi Module

The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all WiFi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

6. Power Supply Unit

Every electrical and electronic device that we use in our day-to-day life will require a power supply. In general, we use an AC supply of 230V 50Hz, but this power has to be changed into the required form with required values or voltage range for providing power supply to different types of devices. There are various types of power electronic converters such as step-down converter, step-up converter, voltage 31 stabilizer, AC to DC converter, DC 25 to DC converter, DC to AC converter, and so on. For example, consider the microcontrollers that are used frequently for developing many embedded systems based projects and kits used in real-time applications. These microcontrollers require a 5V DC supply.

Schematic Diagram

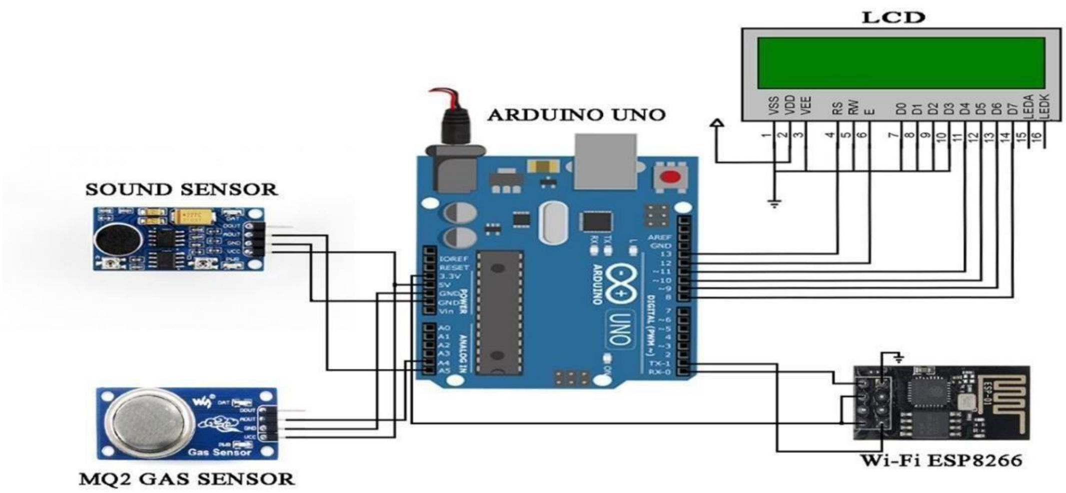


Fig.3.2.1: Schematic Diagram

Schematic Explanation

Arduino UNO has 6 analog pins from A0 to A6 which are generally input pins. It also has 14 digital pins from 0-13 which are the output pins. The analog pin A4 is connected to the output pin of the mq2 gas sensor and analog pin A5 is connected to the output pin of the sound sensor. The digital pins 8-13 is connected to the lcd to display the output values. Here pin 0 is the receiver pin and pin 1 is the transmitter pin. These pins are used for the esp8266 wi-fi module to transmit the information to the online server.

CHAPTER 4

HARDWARE AND SOFTWARE DESCRIPTION

HARDWARE COMPONENTS

The hardware components used for the implementation of Air And Noise Pollution Monitoring System Over An IOT are

1. Arduino Uno
2. MQ2 gas sensor
3. Sound sensor
4. LCD
5. ESP8266 Wi-Fi module
6. Power supply

ARDUINO UNO

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message and turn it into an output activating a motor, turning on an led, publishing as it reached a wider community, the Arduino board started something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based Wiring), the Arduino Software (IDE), based on Processing. Arduino was born at the area Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming wearable, 3dprinting, and embedded environments. All Arduino boards are completely open-source, empowering from everyday objects to complex scientific instruments. Professional gathered around open-source plat form, the software too is opensource, it is growing through the contributions of users worldwide, the Arduino board is shown in below figure.

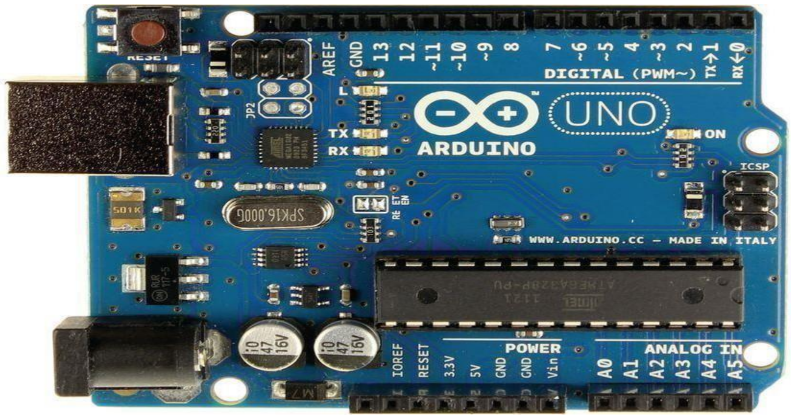


Fig.4.1.1: Arduino uno board

Why Arduino?

Arduino Uno is inexpensive. Arduino is cross platform, its IDE can run on multiple Operating Systems such as Windows, Macintosh, Linux and many more. It has simple and clear programming environment. Its software and hardware are open-source and extensible, that is, Arduino programming can be expanded through C++ libraries and can also leap to AVR ‘C’ language programming. It is simple and accessible to user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove the chemistry and physics principles or to get started with programming and robotics. Designers and the architects build interactive prototypes, musicians, artists use it for installations and to experiment with new musical instruments. Makers of course use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers can start tinkering just following the step by step I constructions of a kit or sharing ideas online with other members of the Arduino community. There are many other microcontrollers platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handy board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-touse package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems. • Inexpensive

Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50 • Cross-platform The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows. • Simple, clear programming environment The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works. • Open source and extensible software The Arduino software is published as opensource tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to. • Open source and extensible hardware The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money and the parts of Arduino board is shown below

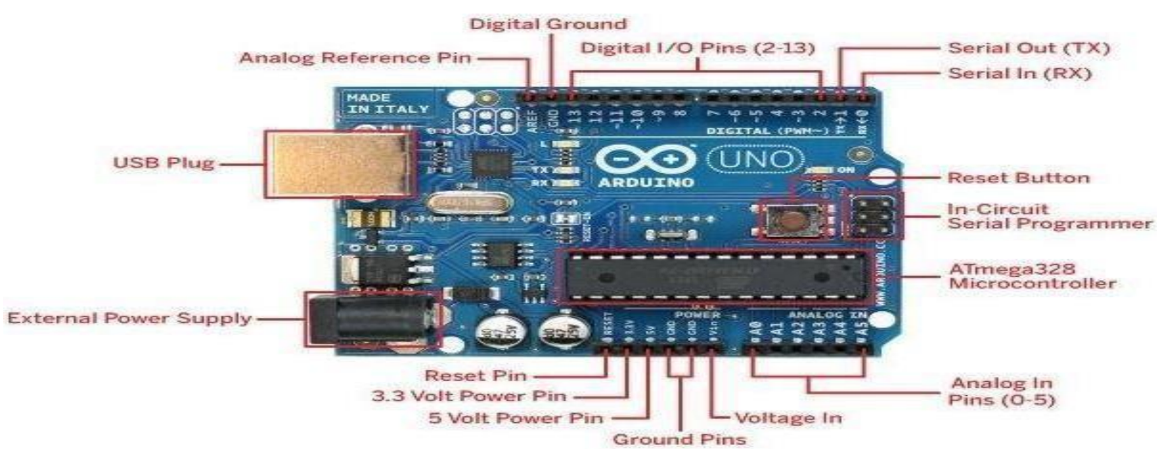


Fig.4.1.2: Arduino board description

Features of Arduino

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage	7-9V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) (0.5 KB used by bootloader)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Table 4.1.1: Features of Arduino uno

Programming

The Arduino Uno can be programmed with the Arduino Software (IDE). Select Arduino Uno from the Tools Board After your code is compiled using Arduino IDE, it should be uploaded to the main microcontroller of the Arduino UNO using a USB connection. Because the main microcontroller doesn't have a USB transceiver, you need a bridge to convert signals between the serial interface (UART interface) of the microcontroller and the host USB signals. The ATmega328 on the Arduino Uno comes pre-programmed with a boot loader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. You can also bypass the bootloader and program the microcontroller through the ICSP (In Circuit Serial Programming) header using Arduino are similar. The ATmega16U2 (or 8U2 in the rev1&rev2 boards) firmware source code is available in the Arduino repository. The ATmega16U2 or 8U2 is loaded with a DFU bootloader, which can be activated by On Rev1 boards connecting the solder jumper on the back of the board and then resetting the 8U2. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). Arduino is an opensource hardware platform that is being used by people around the globe for building electronics projects. It is an integrated platform which contains both the physical and programmable circuit otherwise known as microcontroller and a software (or IDE) that you can run on your computer to write and upload the code onto the physical board. Arduino Board is quite popular among many people who want to get started with electronics, and unlike other embedded system boards Arduino does not require any additional hardware to upload the code (generally known as programmer). The Arduino Program can be written and uploaded using the Arduino IDE that needs just an USB cable to connect. Since the interface is simple and complicacies are less Arduino is preferred by most of the aspiring engineers. Here we will try to understand about the Arduino Architecture and its functionalities. The processor of the Arduino Board uses Harvard Architecture for which the program code and program data have separate memory. The memory of it is divided into two namely program memory and data memory. The data will be stored in the data memory whereas the program code will be stored in the flash program memory. For ex: The Atmega328 microcontroller has 32kb of flash memory, 2kb of SRAM, 1kb of EPROM and operates at 16MHz clock speed. Some of the other basic functions of Arduino are:

- Digital write pin is used to write the digital value of the given pin.
- Pin mode pin is used to set the pin to I/O mode.
- Analog read pin reads and returns the value.
- Analog write pin writes the value of the pin
- Serial. Begins pin sets the beginning of serial communication by setting the rate of bit
- Digital read pin reads the digital value of given pin.

Advantages

Inexpensive -Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50.

1. Cross-platform - The Arduino software runs on Windows, Macintosh OSX, and the Linux operating systems. Most microcontroller systems are limited to Windows.

2. Simple programming environment- Here the Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino.

3. Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to

4. Open source and extensible hardware - The Arduino is based on Atmel's ATmega8 and ATmega168 microcontrollers. The plans for the modules are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even 19 relatively inexperienced users that can build the breadboard version of the module in order to understand how it works and save money.

Applications

• It is an easy USB interface. This allows interface with USB as this is like a serial device.

• The chip on the board plugs straight into your USB port and supports on your computer as a virtual serial port. The benefit of this setup is that serial communication is an extremely easy protocol which is time-tested and USB makes connection with modern computers and makes it comfortable.

• It is easy-to-find the microcontroller brain which is the ATmega328 chip. It has more number of hardware features like timers, external and internal interrupts, PWM pins and multiple sleep modes.

• It is an open source design and there is an advantage of being open source is that it has a large community of people using and troubleshooting it. This makes easy to help in debugging projects

. • It is 16MHz clock which is fast enough for most applications and does not speeds up the microcontroller.

• It is very convenient to manage power inside it and it had a feature of built-in voltage regulation. This also be powered directly off a USB port without any external power. You can connect an external power source of up to 12v and this regulates it to both 5v and 3.3v.

• 13 digital pins and 6 analog pins. This sort of pins allows you to connect hardware to your Arduino Uno board externally. These pins are used as a key for extending the computing capability of the Arduino Uno into the real world. This has an ICSP connector for bypassing the USB port and interfacing the Arduino directly as a serial device. This port is necessary to re-boot load your chip if it corrupts and can no longer used to your computer.

• It has a 32 KB of flash memory for storing your code.

• An on-board led is attached to digital pin 13 to make fast the debugging of code and to make the debug process easy finally it has the button to reset.

Power

The Arduino Uno can be powered via through USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm centre-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. the power pins are as follows.

• VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. • 5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

• 3.3V. A 3.3volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

• GND. Ground pins.

Introduction to Atmega328

ATmega328 is an eight (8) bit Microcontroller. It can handle the data sized of up to eight (8) bits. It is an AVR based micro-controller. Its built-in internal memory is around 32KB. It operates ranging from 3.3V to 5V. It has an ability to store the data even when the electrical supply is removed from its biasing terminals. it is the excellent features include the cost efficiency and low power dissipation, programming lock for security purposes, real timer counter with the separate oscillator and It is generally or normally used in Embedded Systems applications. ATmega-328 is shown in the figure given below.

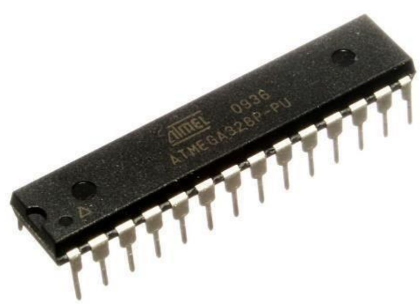


Fig.4.1.3: ATMega328 Microcontroller

Arduino UNO Atmega328

The Arduino UNO is a micro controller board based on the ATmega. UNP means one in Italian and is named to mark the upcoming release of the given Arduino 1.0 it has 14 digital input/output pins, 6 analog inputs, 16MHz ceramic resonator. A USB connection. A power jack, An ICSP header and a reset button.it contains information everything needed to support the microcontroller. simply connect it to a computer with a USB cable or power it with a ac-to-dc adapter or battery to get started. The UNO differs from all preceding boards in that it does not use the FTDI USBto serial driver chip. Instead it features the ATMega 16U2ATMega 8U2 up to version instead R2) programmed as a USB to-serial converter.

Architecture and Working of Arduino Uno

The Arduino UNO can be also powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter or battery. The adapter can connected by plugging a 2.1mm centre positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and vin pin headers of the power connector. The board can operate on an external supply of 6to 20 volts. If supplied with less than 7v however, the 5v pin may supply less than five volts and the board may be unstable. If using more than 12v.the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The processor of the Arduino board uses the Harvard architecture where the program code and program data have separate memory. It consists of two memories such as program memory and data memory. Where in the data is stored in data memory and the code is stored in the flash program memory. The Atmega328 microcontroller has 32kb of flash memory, 2kb of SRAM 1kb of EPROM and operates with a 16MHz clock speed. Architecture of Atmega328 is shown below.

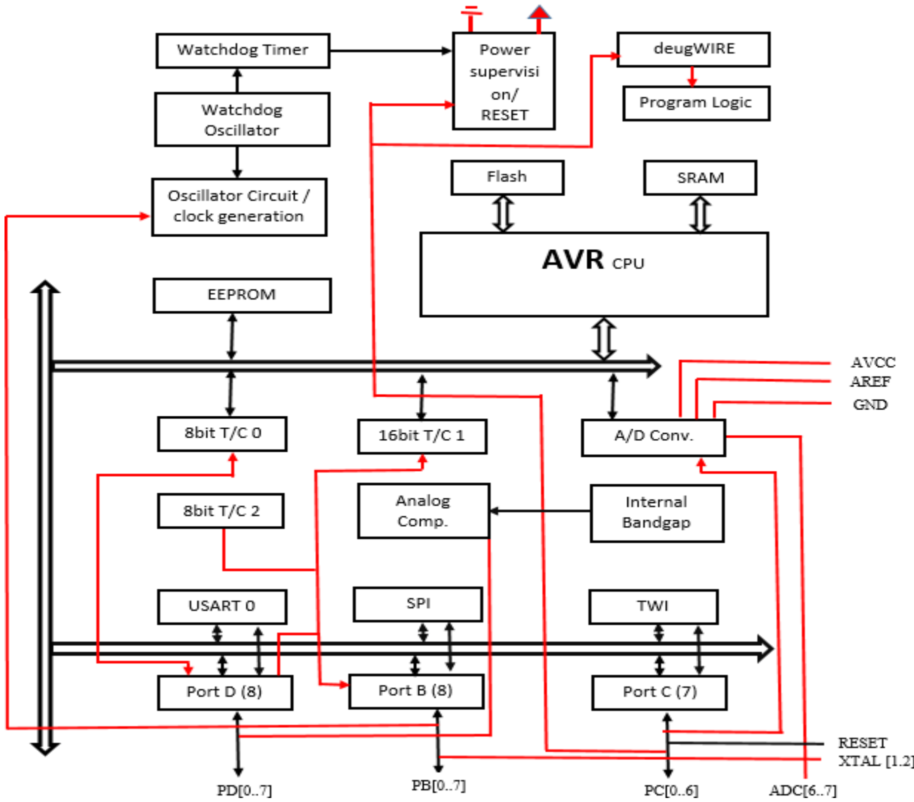


Fig.4.1.4: ATmega328P Architecture

Atmega328 Pins Description

Functions associated with the pins must be known in order to use the device appropriately. ATmega-328 pins are divided into different ports which are given in detail below.

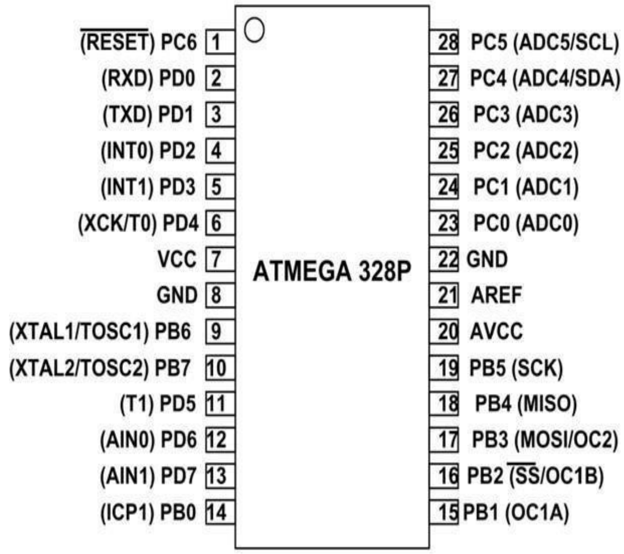


Fig.4.1.5: ATmega328P Ports

AVCC is a supply voltage pin for analog to digital converter.

VCC is a digital voltage supply GND denotes Ground and it has a 0V.

Port A consists of the pins from PA0 to PA7. These pins serve as analog input to analog to digital converters. If analog to digital converter is not used, port A acts as an eight (8) bit bidirectional input/output port.

Port B consists of the pins from PB0 to PB7. This port is an 8 bit bidirectional port having an internal pull-up resistor.

Port C consists of the pins from PC0 to PC7. The output buffers of port C

Port D consists of the pins from PD0 to PD7. It is also an 8bit input/output port having an internal pull-up resistor. All of the AVR ports are shown in figure given below.

GAS SENSOR

• Gas sensors main aim is to sense hazardous gases that evolve its surroundings

• Gas sensor detects the concentrations of combustible gas in the air and ouputs its reading as an analog voltage. The sensor can measure concentrations of flammable gas of 300 to 10,000 ppm. The sensor can operate at temperatures from -20 to 50°C and consumes less than 150 mA at 5 V.

• High sensitivity to LPG, Propane and Hydrogen.

Fig 4.1.6: MQ2 Gas Sensor

Gas sensing technologies

▪ Metal Oxide Based Gas Sensors

▪ Capacitance Based Gas Sensors

▪ Acoustic Wave Based Gas Sensors

▪ Calorimetric Gas Sensors

▪ Optical gas sensors

▪ Electrochemical gas sensors

Types

1. Metal Oxide Based Gas Sensors

▪ Metal oxide sensors are also known as chemiresistors

▪ The detection principle of resistive sensors is based on change of the resistance of a thin film upon adsorption of the gas molecules on the surface of a semiconductor.

▪ The gas-solid interactions affect the resistance of the film because of the density of electronic species in the film.

2. Capacitance Based Gas Sensors

▪ They measure the change in dielectric constant of films between the electrodes as a function of the gas concentration.

▪ The capacitive sensor relies on inter-digitated electrode structures, which correspond to the two plates of a standard capacitor, to monitor changes of the dielectric coefficient of the film. ▪ The simple theory behind it is if the dielectric constant of the film is lower than that of the analyte, the capacitance will increase and vice versa.

3. Acoustic Wave Based Gas Sensors

▪ Sound based gas sensors are known as acoustic wave based gas sensors.

▪ To launch the acoustic waves, this type of sensor uses piezoelectric material either in the thin film form or in bulk form which has one or more transducers on its surface.

4. Carbon monoxide gas sensor

▪ It can either be battery-operated or AC powered.

▪ Mostly the sensor will not sound an alarm at lower concentrations (e.g. 100 ppm). The alarm will sound within a few minutes at 400 ppm. So the function is specific to concentration-time. It shows simple carbon monoxide sensor. Carbon monoxide sensor can be of different types such as:

▪ Semiconductor sensor

▪ Electrochemical sensor

▪ Digital sensor

▪ Biomimetic sensor (chem-optical or gel cell sensor)

Power Supply

Every electrical and electronic device that we use in our day-to-day life will require a power supply. In general, we use an AC supply of 230V 50Hz, but this power has to be changed into the required form with required values or voltage range for providing power supply to different types of devices. There are various types of power electronic converters such as step-down converter, step-up converter, voltage stabilizer, AC to DC converter, DC 25 to DC converter, DC to AC converter, and so on. For example, consider the microcontrollers that are used frequently for developing many embedded systems based projects and kits used in real-time applications. These microcontrollers require a 5V DC supply.

LCD (Liquid Crystal Display)

A liquid crystal display (LCD) is a thin, flat display device made up of any number of colour or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other. A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCDs connected to the controllers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0) For a 4-bit data bus it only requires the supply lines plus 6 extra lines (RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

Pin No	Symbol	Level	Description
1	Vss	0 V	Ground
2	V_{DD}	5.0V	Supply voltage for logic
3	V o	variable	Operating voltagefor LCD
4	RS	H/L	H data/L instruction code
5	R/W	H/L	H/Read (MPU-module) L/write (MPU-module)
6	E	H, H-L	Chip enable signal
7	DB0	H/L	Data bit 0
8	DB1	H/L	Data bit 1
9	DB2	H/L	Data bit 2
10	DB3	H/L	Data bit 3
11	DB4	H/L	Data bit 4
12	DB5	H/L	Data bit 5
13	DB6	H/L	Data bit 6
14	DB 7	H/L	Data bit 7
15	A		Power supply for LED backlight (+)
16	K		Power supply for LED backlight (-)

Table 4.1.2: Pin description

Description Of 16x2:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all Parallel Ports. It however doesn't show the use of the Status Port as an input. So what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

Schematic Diagram

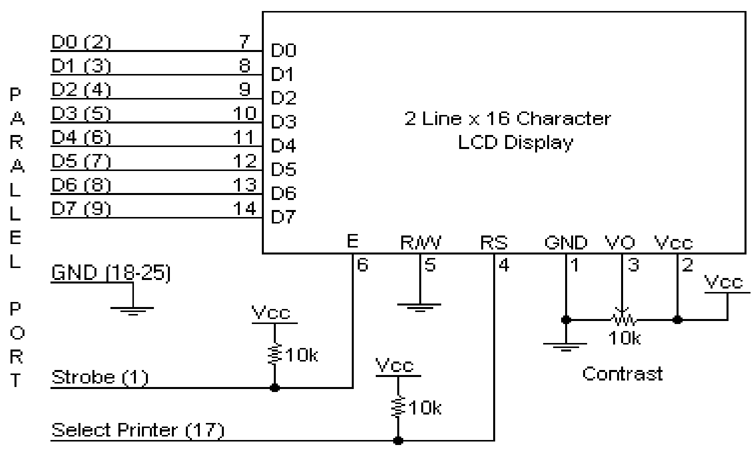


Fig.4.1.7: Schematic Diagram of LCD

• Above is the quite simple schematic. The LCD panel's Enable and register select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pullup resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

• We make no effort to place the Data bus into reverse direction. Therefore we hard wire the R/W line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

• The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.

Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left
0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

Table 4.1.3: LCD commands

SOUND SENSOR

A sound sensor is defined as a module that detects sound waves through its intensity and converting it to electrical signals. Sound detection sensor works similarly to our Ears, having diaphragm which converts vibration into signals. However, what’s different as that a sound sensor consists of an in-built capacitive microphone, peak detector and an amplifier (LM386, LM393, etc.) that’s highly sensitive to sound.

With these components, it allows for the sensor to work:

1. Sound waves propagate through air molecules

2. Such sound waves cause the diaphragm in the microphone to vibrate, resulting in capacitance change

3. Capacitance change is then amplified and digitalized for processing of sound intensity

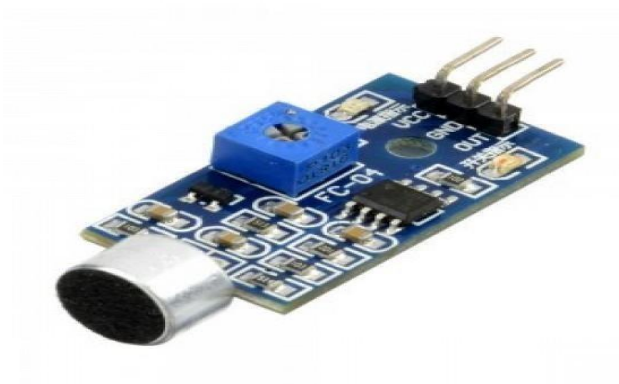


Fig 4.1.8: Sound Sensor

Applications

Apart from building various electronic projects with Arduino (covered in the later section) and more, sound sensors are used in many other day to day applications including:

- Consumer electronics such as phones, computers, music systems
- Security and Monitoring systems such as burglar alarms, door alarm, etc.
- Home automation such as lighting your house by detecting whistle/clap instead of physically turning the light switch
- Ambient sound recognition and sound level recognition

features :

- Grove compatible interface
- Seeed's own plug and play system, no soldering or jumper wires needed for pairing as compared to other sound sensor breakout boards
- Analog output signal
- Wide supply voltage range: 4V-12V
- Low quiescent current drain: 4mA
- 2.0cm x 2.0cm twig module
- Minimum external parts

WIFI ESP8266 MODULE:

- The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.
- This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existance interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

- There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support. In the Documents section below you will find many resources to aid you in using the ESP8266, even instructions on how to transforming this module into an IoT (Internet of Things) solution!

- Note: The ESP8266 Module is not capable of 5-3V logic shifting and will require an external Logic Level Converter. Please do not power it directly from your 5V dev board. This ESP8266 WiFi Module Adaptor comes with on-board logic level converter and 3.3V LDO regulator for easy interfacing to Arduino boards.

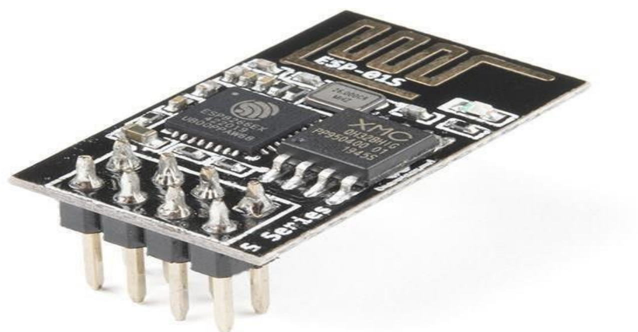


Fig 4.1.9: ESP8266 Wi-Fi Module

Features:

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval

- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- Default baudrate: 115200

Software Requirement

- Arduino software
- Programming language

Arduino software:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino 44 software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low-cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step-by-step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

• Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50

• Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

• Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

• Open source and extensible software - The Arduino software is published as opensource tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version
- Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money
- Getting Started with Arduino and Genuino products.
- Install the Arduino Software (IDE) on Windows PCs-
- This document explains how to install the Arduino Software (IDE) on Windows machines.
- Download the Arduino Software (IDE)
- Proceed with board specific instructions.

How to Download the Arduino Software (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

Installation

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.4.2.1: USB Cable A plug to B plug

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig.4.2.2: USB Cable A plug to Mini-B plug

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, 47 which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

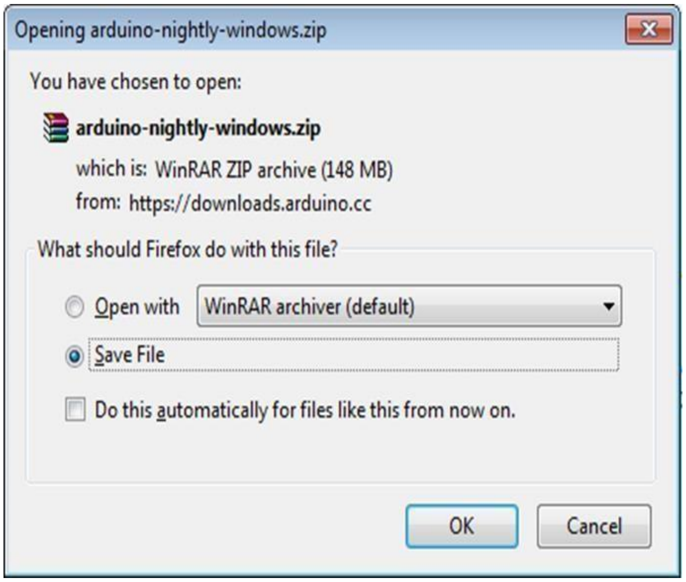


Fig.4.2.3: Unzip of Arduino IDE software

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

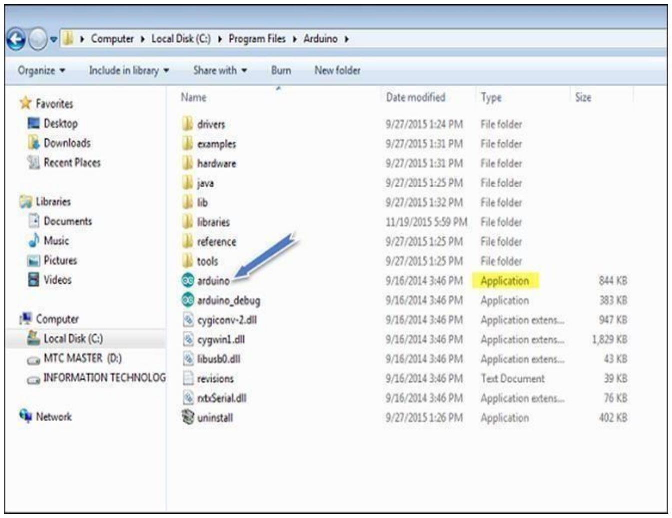


Fig.4.2.4: Open IDE software

Step 5 − Open your first project.

Once the software starts, you have two options:

• Create a new project

. • Open an existing project example.

To create a new project, select File → New.

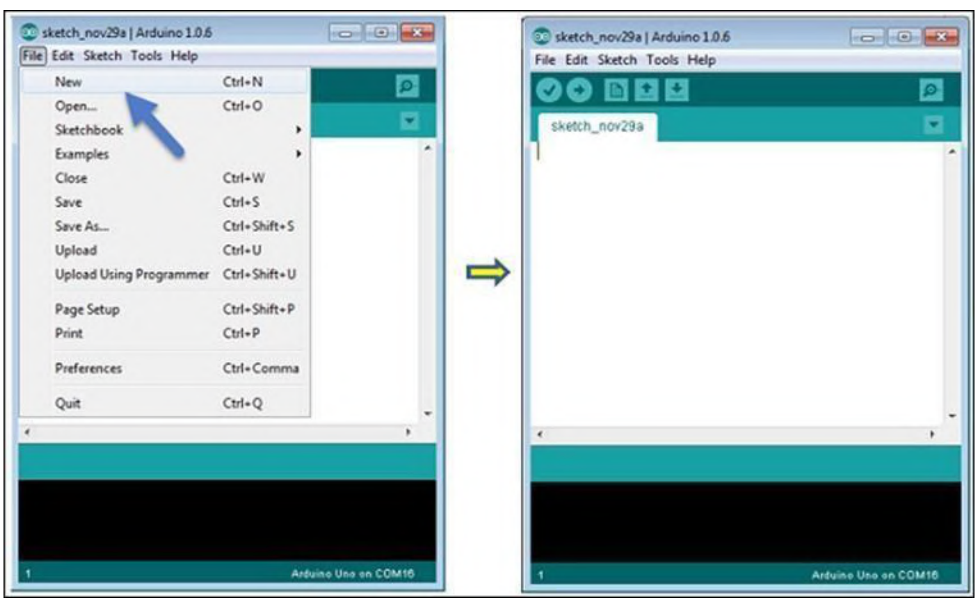


Fig.4.2.5: Open First Project

To open an existing project example, select File → Example → Basics → Blink.

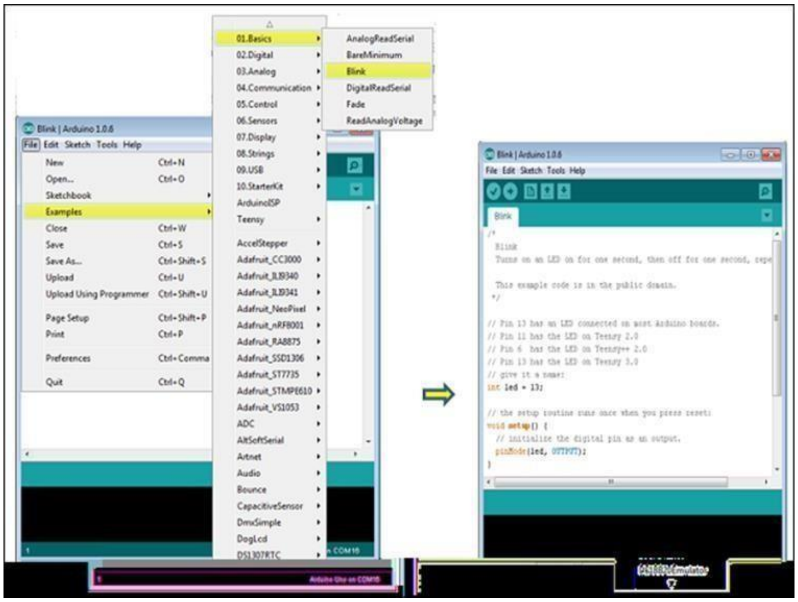


Fig.4.2.6: Open existing Project

Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board. To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer. Go to Tools → Board and select your board.

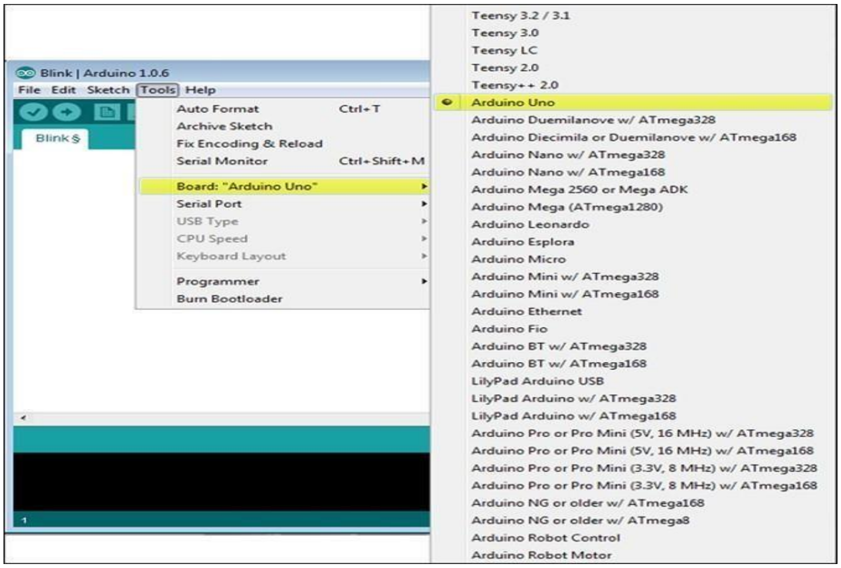


Fig.4.2.7: Selection of Arduino board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port. Select the serial device of the Arduino board. Go to Tools → Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

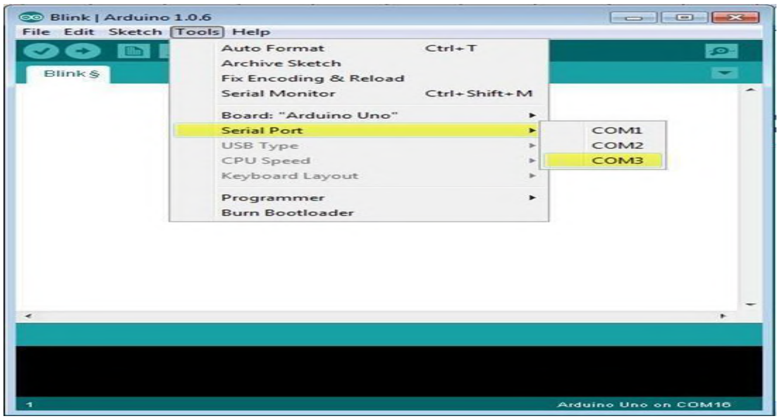


Fig.4.2.8: Selection of Serial Port

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

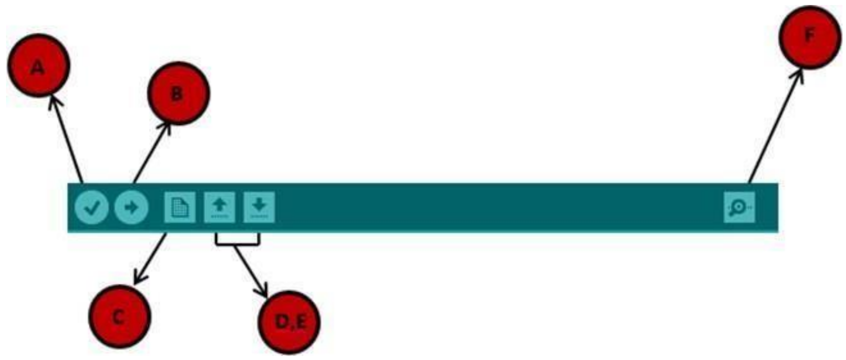


Fig.4.2.9: Uploading the program to board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Proteus

Proteus

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also has 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc. ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design

Step 1: Open ISIS software and select New design in File menu

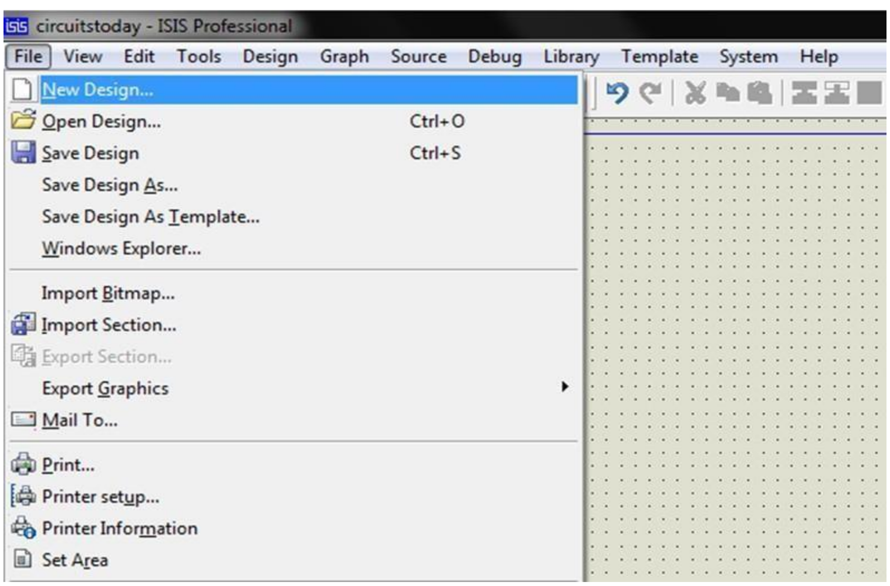


Fig.4.2.10: Proteus file menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No 57 depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

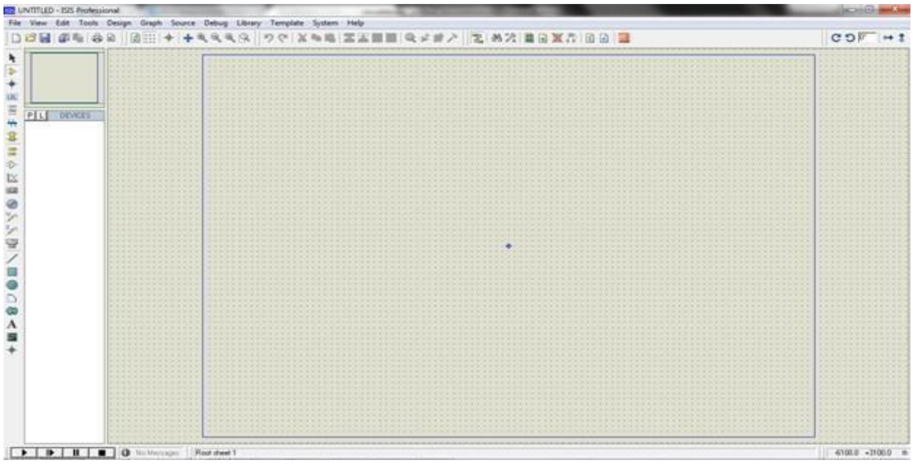


Fig.4.2.11: Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

Step 6: Select the components from categories or type the part name in Keywords text box.

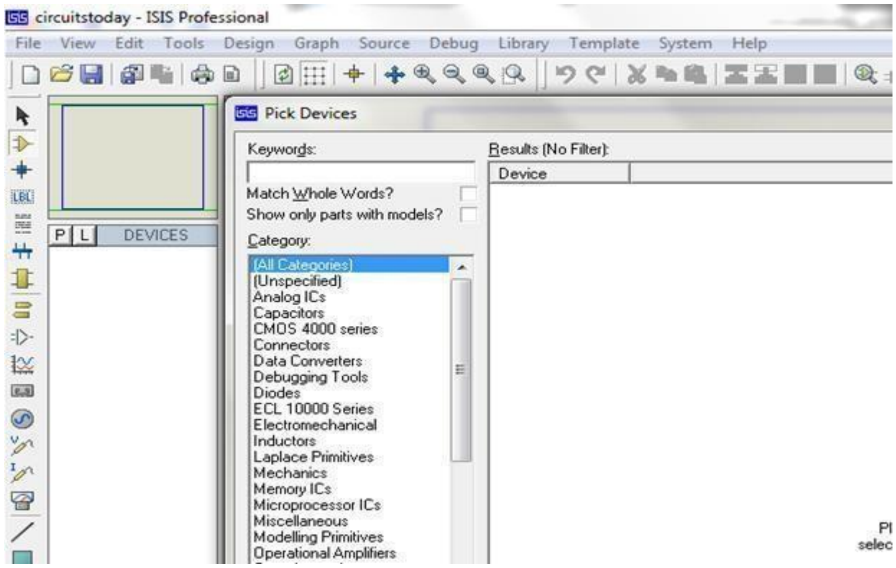


Fig.4.2.12: Key Textbox

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click. Place all the required components and route the wires i.e., make connections. Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

Step 8: After connecting the circuit, click on the play button to run the simulation.

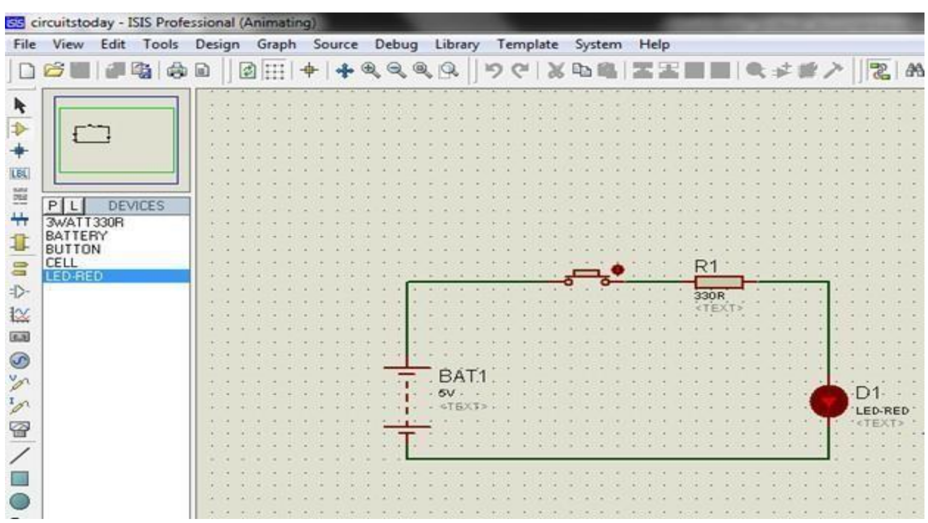


Fig.4.2.13: Stimulation Software

CHAPTER 5

SOFTWARE TESTING AND CODE IMPLEMENTATION

SOFTWARE TESTING

Software Testing is a process of executing the application with an intent to find any software bugs. It is used to check whether the application met its expectations and all the functionalities of the application is working. The final goal of testing is to check whether the application is behaving in the way it is supposed to under specified conditions. All aspects of the code are examined to check the quality of application. The primary purpose of testing is to detect software failures so that defects may be uncovered and corrected. The test cases are designed in such way that scope of finding the bugs is maximum.

TESTING LEVELS

There are various testing levels based on the specificity of test.

- Unit testing: Unit testing refers to tests conducted on a section of code in order to verify the functionality of that piece of code. This is done at the function level.
- Integration Testing: Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Its primary purpose is to expose the defects associated with the interfacing of modules.
- System Testing: System testing tests a completely integrated system to verify that the system meets its requirements.
- Acceptance testing: Acceptance testing tests the readiness of application, satisfying all requirements.

• Performance testing: Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or device such as response time or millions of instructions per second etc.

SYSTEM TEST CASES

A test case is a set of test data, preconditions, expected results and post conditions, developed for a test scenario to verify compliance against a specific requirement. We have designed and executed a few test cases to check if the project meets the functional requirements.

CODE IMPLEMENTATION

PROJECT

CODE #include

```
<SoftwareSerial.h>
SoftwareSerial
mySerial(2,3);


#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);
int gas=A4,sound=A5;
char res[130];
void upload1(unsigned char *chr,unsigned char *chr1);
void serialFlush(){
  while(Serial.available() > 0) {
    char t = Serial.read();
  }
```

```
}
```

```
void myserialFlush(){
  while(mySerial.available() > 0) {
  char t = mySerial.read();
  }
  }
char check(char* ex,int timeout)
{
  int i=0;
  int j = 0,k=0;
  while (1)
  {
    sl:
    if(mySerial.available() > 0)
    {
      res[i] = mySerial.read();
      if(res[i] == 0x0a || res[i]=='>' || i == 100)
      {
        i++;
        res[i] = 0;break;
      }
      i++;
    }
    j++;
```

```
  if(j == 30000)
  {
    k++;
    //Serial.println("kk");
    j = 0;
  }
  if(k > timeout)
  {
   // Serial.println("timeout");
    return 1;
   }
}//while 1
if(!strncmp(ex,res,strlen(ex)))
{
  //Serial.println("ok..");
  return 0;
 }
else
{
 // Serial.print("Wrong  ");
 // Serial.println(res);
  i=0;
  goto sl;
```

```
   }
}
char buff[200],k=0;
void upload1();
const char* ssid = "project";
const char* password = "project1234";
int T;
int tt;
void setup() {
int i=0;
  char ret;
  //pinMode(red1, OUTPUT);
  //pinMode(r1, INPUT);
  //digitalWrite(red1,HIGH);
  Serial.begin(9600);
  mySerial.begin(115200);
  lcd.begin(16,2);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("AIR AND NOISE POLLUTION MONITORING
SYSTEM OVER AN IOT");
 // Serial.print("WELCOME");
  delay(500);
```

```
st:
  mySerial.println("ATE0");
  Serial.println("ATE0");
//  ret  = check((char*)"OK",50);
  mySerial.println("AT");
  Serial.println("AT");
//  ret  = check((char*)"OK",50);
  if(ret != 0)
  {
    delay(1000);
   goto st;
  }
//}
  lcd.clear();lcd.setCursor(0, 0);lcd.print("CONNECTING");
  lcd.clear();
  lcd.setCursor(0, 0);
 lcd.print("CONNECTING");
 mySerial.println("AT+CWMODE=1");
 Serial.println("AT+CWMODE=1");
 ret  = check((char*)"OK",50);
 cagain:
  myserialFlush();
  Serial.print("AT+CWJAP=\"");
```

```
  mySerial.print("AT+CWJAP=\"");
  mySerial.print(ssid);
  Serial.print(ssid);
  mySerial.print("\",\"");
  Serial.print("\",\"");
  mySerial.print(password);
  Serial.print(password);
  mySerial.println("\"");
  Serial.println("\"");
  if(check((char*)"OK",300))goto cagain;
  mySerial.println("AT+CIPMUX=1");
  Serial.println("AT+CIPMUX=1");
  delay(1000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("WAITING");
delay(500);
lcd.clear();lcd.setCursor(0, 0);lcd.print("Connected");
}
void loop() {
   int gd = analogRead(gas);
   int sd = analogRead(sound);
 lcd.clear(); lcd.setCursor(0, 0);lcd.print("AIR:");lcd.print(gd);
          lcd.setCursor(8, 0);lcd.print("SOUND:");lcd.print(sd);delay(2000);
```

```
        if(gd > 600)
        {
          lcd.clear();lcd.setCursor(0,1);lcd.print("DANGER GAS");
          upload1(gd,sd);delay(2000);
        }
        if(sd > 600)
        {
          lcd.clear();lcd.setCursor(0,1);lcd.print("DANGER SOUNDS");
          upload1(gd,sd);delay(2000);
        }
}//loop
char bf2[100];
void upload1(unsigned char *chr,unsigned char *chr1)
{
  delay(2000);
  lcd.clear();lcd.setCursor(0, 1);lcd.print("UPLOADING");
myserialFlush();
mySerial.println("AT+CIPSTART=4,\"TCP\",\"api.thingspeak.com\",80");
// mySerial.println("AT+CIPSTART=4,\"TCP\",\"embeddedspot.top\",80");
   delay(8000);
```

```
sprintf(buff,"GET
https://api.thingspeak.com/update?api_key=MZP5LS9WI6404PI7&field1=%u&field2=%u\r\n\r
\n",chr);
//sprintf(buff,"GET
http://embeddedspot.top/iot/storedata.php?name=sensors010&s1=%u&s2=%u&s3=%u\r\n\r\n",
chr,chr1,chr2);
myserialFlush();
sprintf(bf2,"AT+CIPSEND=4,%u",strlen(buff));
mySerial.println(bf2)
  delay(5000);
   myserialFlush();
   mySerial.print(buff);
   Serial.print(buff);
      delay(2000);
      mySerial.println("AT+CIPCLOSE");
      Serial.println("AT+CIPCLOSE");
      lcd.setCursor(0, 1);lcd.print("UPLOADDED");  lcd.clear();


}
```

CHAPTER 6

RESULTS ANALYSIS

RESULTS:

This hardware setup gives a brief idea as to how the project “Air And Noise Pollution Monitoring System Over An IOT” works. The below figure shows the hardware setup of the proposed system and constitutes of gas sensor and sound sensor for the detection of concentration of harmful gases in the environment and level of intensity of sound in the surroundings respectively. It also constitutes of lcd display for display the information i.e., the values detected by the two sensors and a esp8266 wi-fi module for transmitting this values continuously to the online server.

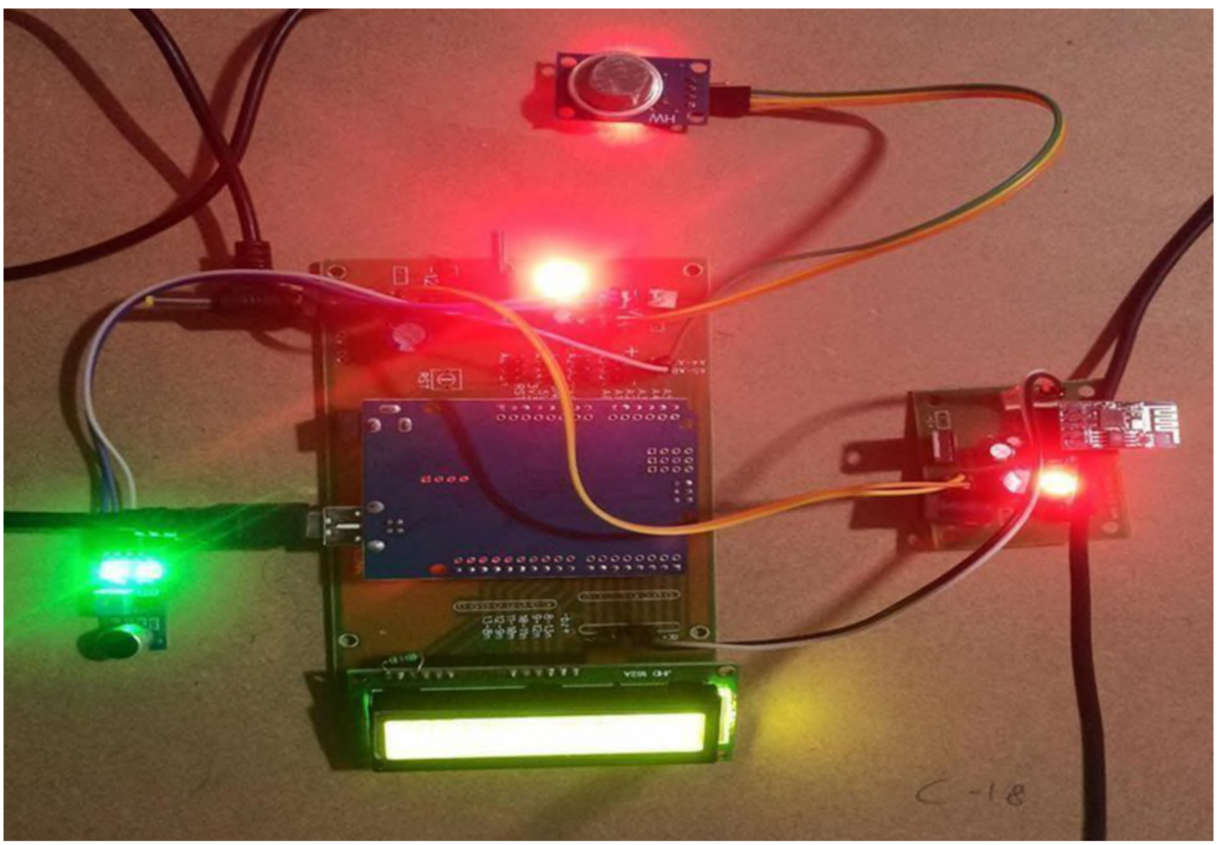


Fig 6.1:The Hardware setup of the proposed system

Displaying normal reading values on the lcd

Gas sensor and sound sensor sense the concentration of harmful gases and level of intensity of sound in the environment continuously and sends that information continuously to the Arduino uno. Now the Arduino uno which is brain of this project takes the information from the sensors and transmits that values to the lcd for displaying them. The below figure represents lcd display that displaying normal readings of the sensors and this values are normal range of concentration of harmful gases and intensity of sound that doesn't cause any harm to the living beings.

Fig 6.2: LCD display that displaying normal range values

Displaying of warning messages

Here the two sensors i.e., gas sensor and sound sensor consists of potentiometer using this we can set the threshold .When the readings from the gas sensor and sound sensor exceeds the set threshold it will display warning messages on the lcd. If the concentration of harmful gases exceeds the set threshold it will display “Danger gas” as warning message on the lcd and if the level of sound intensity exceeds the set threshold it will display “ Danger sound” warning message on the lcd.

Fig 6.3 : Displaying of warning message “Danger gas”

Fig 6.4: Displaying of warning message “Danger sound”

Monitoring readings of the sensor in the online server

Here the readings from the two sensors are continuously transmitted to the online server by the esp8266 wi-fi module and we can monitor those readings by using an open IOT platform thingspeak website .Here in this website all the readings can be viewed in graphical manner.

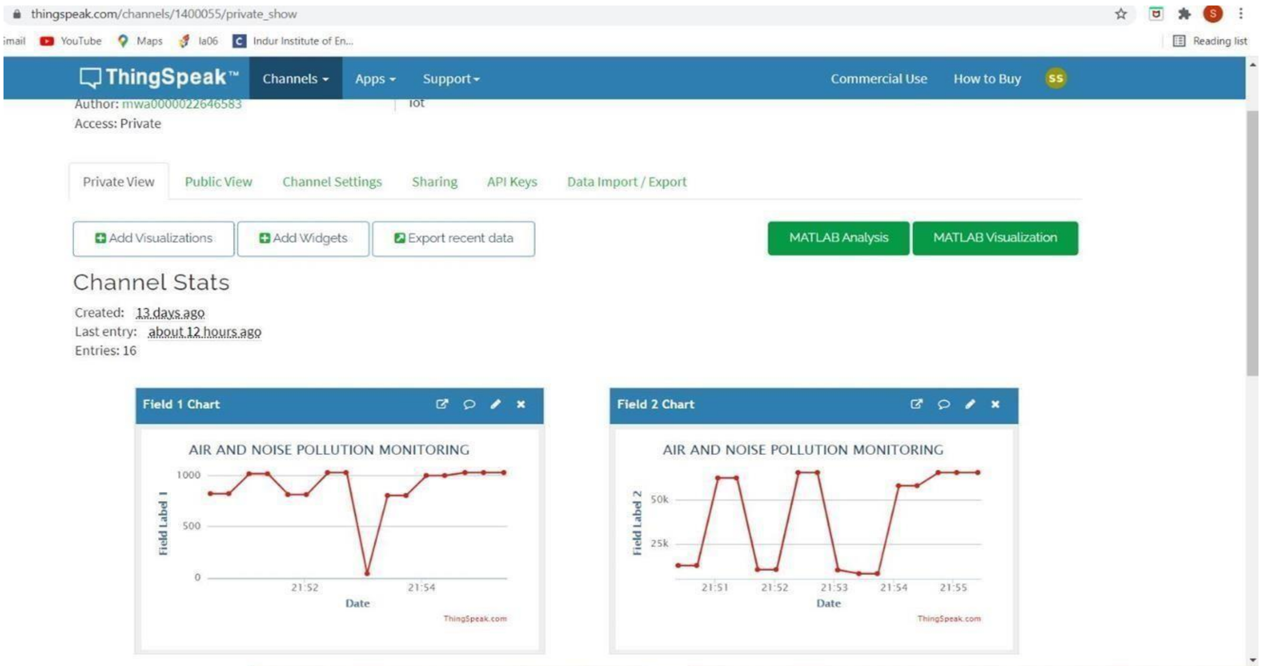


Fig 6.5 : Thingspeak website

For displaying graphical output on the website we need to create a channel in that website and we have add a particular API key to our project code. This API key unique for every project. The API key of this proposed system is given below

Fig 6.6: API key

CHAPTER 7

APPLICATIONS AND ADVANTAGES

APPLICATIONS OF THE PROJECT:

- Industrial pollution monitoring
- Public areas
- School area
- Environmental Section
- Installed in Vehicles

ADVANTAGES OF THE PROJECT:

- Remotely we can monitor pollution
- Cheap in Cost
- Data can be used to control pollution
- Small in size
- Data is useful to government Health Departments

CHAPTER-8

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION:

The **air** & **sound monitoring system** overcomes the problem of the highly-polluted areas which is a major issue. It supports the new technology and effectively supports the healthy life concept. The Automatic Air & Sound management system is a step forward to contribute a solution to the biggest threat. The air & sound monitoring system overcomes the problem of the highly-polluted areas which is a major issue. It supports the new technology and effectively supports the healthy life concept.

This system has features for the people to monitor the amount of pollution on their mobile phones using the application. So, it becomes very reliable and efficient for the Municipal officials along with the Civilians to monitor environment. Letting civilians also involved in this process adds an extra value to it. As civilians are now equally aware and curious about their environment, this concept of IOT is beneficial for the welfare of the society. And it is implemented using the latest technology

FUTURE SCOPE:

The project is intended victimization structured modeling and is ready to supply the required results. It is with success enforced as a true Time system with bound modifications. Science is discovering or making major breakthrough in varied fields, and thus technology keeps dynamic from time to time. Going more, most of the units is fictional on one in conjunction with microcontroller so creating the system compact thereby creating the present system simpler.33 To make the system applicable for real time functions parts with larger vary must be enforced. This model is any enlarged to observe the developing cities and industrial zones for pollution monitoring. To safeguard the general public health from pollution, this model provides associate economical and low price resolution for continuous observance of atmosphere.

REFERENCE

1. IOT based Air and Sound Pollution Monitoring System International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO 3297: 2007 Certified Organization) Website: www.ijareeie.com Vol. 6, Issue 3, March 2017 Copyright to IJAREEIE

2. Implementation of an Efficient Noise and Air Pollution Monitoring System Using Internet of Things (IoT) International Journal of Advanced Research in Computer and Communication Engineering ISO 3297:2007 Certified Vol. 5, Issue 7, July 2016 AnjaiahGuthi M. Tech Student, ECE, GRIET, Hyderabad, India

3. An IoT Based Automated Noise and Air Pollution Monitoring System International Journal of Advanced Research in Computer and Communication Engineering ISO 3297:2007 Certified Vol. 6, Issue 3, March 2017 Copyright to IJARCCE DOI10.17148/IJARCCE.2017.6397

4. Implementation of an Efficient Noise and Air Pollution Monitoring System Using Internet of Things (IoT) International Journal of Advanced Research in Computer and Communication Engineering ISO 3297:2007 Certified Vol. 5, Issue 7, July 2016 Copyright to IJARCCE DOI 10.17148/IJARCCE.2016.5747237 AnjaiahGuthi M. Tech Student, ECE, GRIET, Hyderabad, India

5.. International Research Journal In Advanced Engineering And Technology (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744 VOL 3 ISSUE 2 (2017) PAGES 1781 - 1785 A System For Monitoring Air And Sound Pollution Using Ardunio Controller With IOT Technology1 L.Ezhilarasi, 2 K.Sripriya, 3 A .Suganya , 4 K.Vinodhini . 1Assistant Professor, GanadipathyTulsi's Jain Engineering College, Vellore. 2,3,4 Research scholar, GanadipathyTulsi's Jain Engineering College, Vellore.

6. Industrial Air Pollution Monitoring and Analysis System.International Journal of Advanced Research in Computer Science and Software Engineering Research Paper Available online at: www.ijarcsse.com JadhavAditya S., PawarVishwajeet P., JorwekarSagar R, JadhavVidya P. Computer Engineering, SPPU, Maharashtra, India

A

PROJECT REPORT

on

SUPER-RESOLUTION BY IMAGE ENHANCEMENT USING TEXTURE TRANSFER

Submitted By

P. Bharath Chandra Reddy (17K81A04A2)

G.R. Srawan kumar Reddy (17K81A0478)

K. Revanth (17K81A0489)

Submitted in partial fulfilment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

IN

Electronics & Communication Engineering

Under the Guidance of

G Ramesh Reddy

Associate Professor

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100.

June 2021

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the major-project work entitled "**Super Resolution By Image enhancement Using Textual Transfer**" is a bonafide work carried out by **P. Bharath Chandra Reddy (17K81A04A2), G. Srawan kumar Reddy (17K81A0478), K.Revanth (17K81A0478)** in partial fulfilment of the requirements for the degree of **Bachelor of Technology in Electronics & Communication Engineering** by the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2020-2021.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree.

INTERNAL GUIDE

Mr. G. Ramesh Reddy

Associate Professor,

Department of ECE.

HEAD OF THE DEPARTMENT,

Dr. B. Hari Krishna,

Professor.

External Examiner

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **BHARATH CHANDRA REDDY** WITH ROLL NO.**17K81A04A2, G.SRAWAN KUMAR** WITH ROLL NO.**17K81A0478**, **K N V S N REVANTH** WITH ROLL NO.**17K81A0489**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**SUPER-RESOLUTION BY IMAGE ENHANCEMENT USING TEXTURE TRANSFER**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

ACKNOWLEDGEMENT

We sincerely thank the management of our college **St. MARTIN'S ENGINEERING COLLEGE** for providing required facilities during our project work. We derive our great pleasure in expressing our sincere gratitude to our principal **Dr. P. Santosh Kumar Patra** for his timely suggestions, which helped us to complete the project work successfully. It is the very auspicious moment to express our gratitude to **Dr. B. Hari Krishna**, Head of the Department, ECE for his consistent encouragement during the progress of this project. We take it as a privilege to thank our project coordinator **Mr. K. Karthik** Professor, department of ECE for the ideas that led to complete the project work and we also thank them for their continuous guidance, support and unfailing patience, throughout the course of this work. We sincerely thank our project Internal guide **Mr. G. Ramesh Reddy**, Assistant Professor, Department of ECE for guidance and encouragement in carrying out this project work.

DECLARATION

We, the students of **'Bachelor of Technology in Department of Electronics and Communication Engineering'**, session: 2017 – 2021, St. Martin's Engineering College, dullapally, kompally, secunderabad hereby declare that the project work entitled '**Super Resolution By Image enhancement Using Textual Transfer**' is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

P. Bharath Chandra Reddy	**(17K81A04A2)**
G. R Srawan kumar Reddy	**(17K81A0478)**
K. Revanth	**(17K81A0489)**

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract

Recent deep learning approaches in single image super resolution (SISR) can generate high-definition textures for super-resolved (SR) images. However, they tend to hallucinate fake textures and even produce artifacts. An alternative to SISR, reference-based SR (Ref SR) approaches use high resolution (HR) reference (Ref) images to provide HR details that are missing in the low-resolution (LR) input image. We propose a novel framework that leverages existing SISR approaches and enhances them with Ref SR. Specifically, we refine the output of SISR methods using neural texture transfer, where HR features are queried from the Ref images. The query is conducted by computing the similarity of textural and semantic features between the input image and the Ref images. The most similar HR features, patch-wise, to the LR image is used to augment the SR image through an augmentation network. In the case of dissimilar Ref images from the LR input image, we prevent performance degradation by including the similarity scores in the input features of the network. Furthermore, we use random texture patches during the training to condition our augmentation network to not always trust the queried texture features. Different from past Ref SR approaches, our method can use arbitrary Ref images and its lower-bound performance is based on the SR image. We showcase that our method drastically improves the performance of the base SISR approach

CHAPTER 1

INTRODUCTION

Introduction to the Project

The earth is associate aquatic planet and the maximum amount as eightieth of its surface is roofed by water. Moreover, there is a strong interest in knowing what lies in underwater. Present days, an image of deep waters has a scope to large investigation to explore the underwater for sea floor expedition and navigation. Enthusiasm of underwater imaging includes the inspection of plants, seabed exploration, the search for wrecks up and to the exploration of natural resources. There were several issues faced by the human in the underwater, if he dives deep into the ocean and stay there for a long time to perform experimentation. Due to the above reasons, unmanned remote vehicles are used to sea floor exploration.

Historical Development

Underwater image quality improvement approaches present a path to magnify the object recognition in underwater surrounding. A heap of research started for the upgradation of image visual quality, but a little amount of work has been carried out in this area. In the deep waters, image quality is degrading due to poor illumination conditions and the light properties differ in water compared to air.[2]. There were several parameters which decreases the quality of an image in underground waters. So, in order to remove all these effects

there are several techniques has been implemented and practiced.

Need for Pre process

Initially processing is necessary for deep water images due to their poor-quality during acquisition. Necessity for pre-processing of deep-water images is discussed below:

(i) Quality of images taken from deep water is deteriorated due to light ray attributes like scattering and absorption of light.

(ii) Specificity of surroundings such as lighting inequalities, water torridness, and blue complexion is more or less influential when vehicles move.
(iii) Video or image captured from deep waters like unknown rigid scene, and the depth of the scene and low light sensitivity due to Marine snow etc.

TRADITIONAL TECHNIQUES FOR IMAGE ENHANCEMENT

There are several techniques which are used very frequently for processing the image to improve the visual quality. Some of them are as follows:
(i) Contrast Stretching

(ii) Adaptive Histogram Equalization

Contrast stretching

The contrast stretching is a method to transform high intense region of image into brighter and less intense region into darker by using a predefined transformation function T(r) [2]. Generally, the underwater images will have fewer grey values. There are 256 grey values. ‘0’ indicates black and ‘255’ indicates white. In this method the current grey value of the image is stretched towards 255 i.e., from black to white, pixel by pixel. That means the contrast of the image is stretched, so that the quality of the image is improved for better vision.
For example:

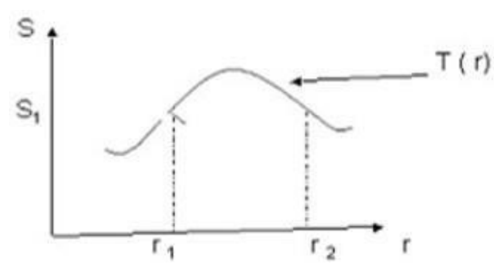


Fig.1.1 Two different gray levels look same

Here two different thresholds are considered for the entire image and the values between them are stretched to the maximum extent, so that the contrast increases. And more over by this method the entire global image contrast is enhanced.

Fig.1.2 (a) Raw image (b) Enhanced image

But the disadvantage here is that the transformation function is not unique. Depending on the application the suitable transformation function is chosen.

Adaptive histogram equalization

Adaptive histogram equalization is a PC based image processing technique which is used to improve the quality of image properties like contrast. It is similar to contrast stretching method but with a slight difference. It computes several intensities of specific gray value, each corresponding to a distinct portion of an image, and with the help of them intensities are rearranged by applying a suitable transformation function. For example, a simple transformation function such as each pixel transformed based on the histogram of a square surrounding the pixel [3]. Existing values will be mapped to new values keeping actual number of intensities in the resulting image equal or less than the original number of intensities. The transformation function applied on the histogram is proportional to the cumulative distributive function (CDF) of pixel values in the neighbourhood. Therefore it suits for enhancing the local details and enhancing the edge information of each region of an image.

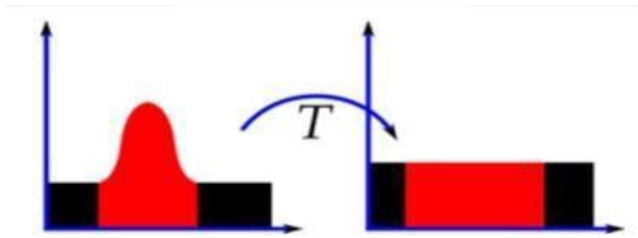


Fig 1.3. Histograms of an image before and after histogram equalization

Histogram equalization is a technique for changing the overall pixel intensities based on transformation function and contrast of an image. Histogram equalization is an effective technique which will benefit for the images with extreme contrast values. The limitation of this technique highlights the unwanted noise present in the background of an image and lead to loss in the information signal. It results in undesired effects in the resultant images.

Fig.1.4. (a) Raw image **(b) Enhanced image**

Here the noise in relatively homogeneous regions of the image is amplified which results in poor SNR. And also, only the local objects of the image are enhanced and the background is left unenhanced.

Introduction to Image Processing

1.3.1. Image:

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Fig 1.5 General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

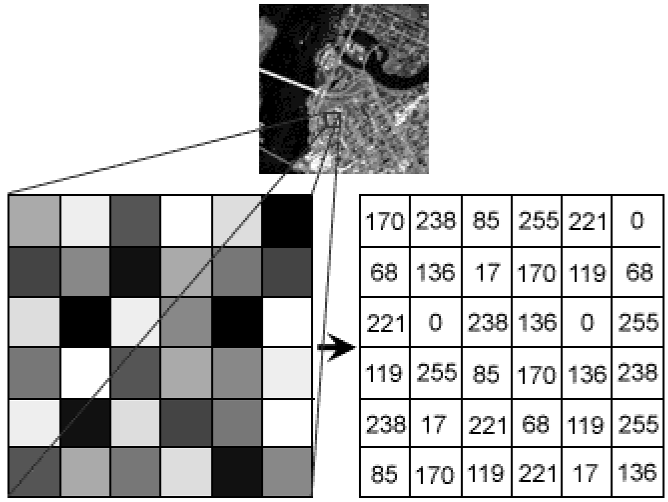


Fig 1.6 Image pixel

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

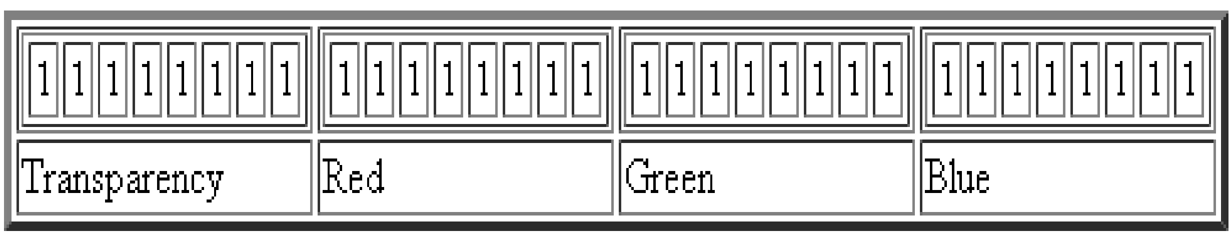


Fig 1.7 Transparency image

IMAGE FILE SIZES:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12-megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

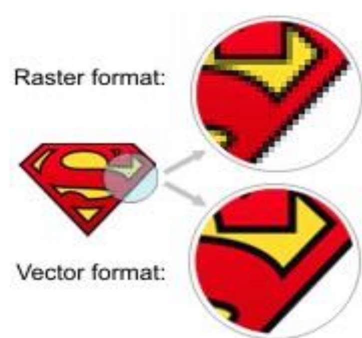


Fig 1.8 Resolution image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

RASTER FORMATS:

These formats store images as bitmaps (also known as pixmaps)

- **JPEG/JFIF:**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobes PDF files.

EXIF:

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

TIFF:

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

PNG:

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

GIF:

GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

BMP:

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM:**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

- **SVG:**

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

IMAGE PROCESSING:

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man’s ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factors combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low-cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:

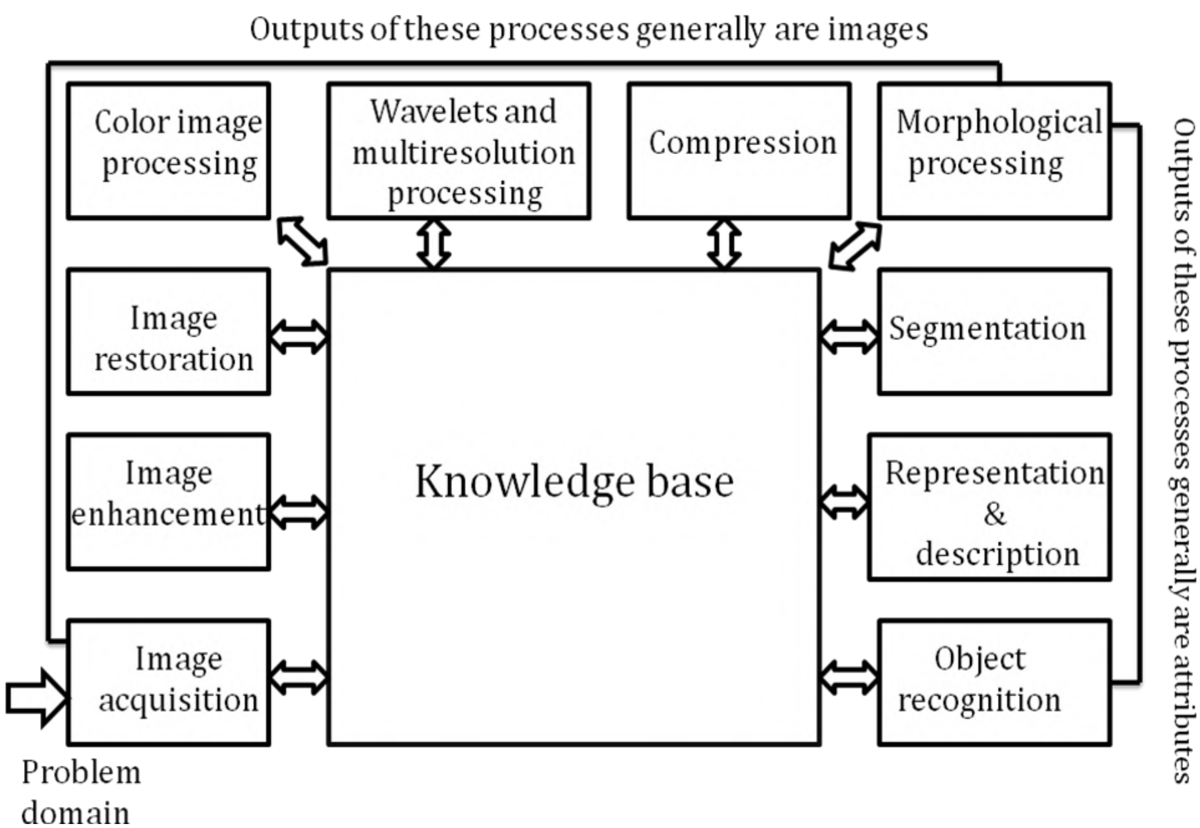


Fig 1.9 Image fundamental

Image Acquisition:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Fig 1.10 Digital camera image

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Fig 1.11 digital camera cell

Image Enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because “it looks better.” It is important to keep in mind that enhancement is a very subjective area of image processing.

Fig 1.12 Image enhancement

Image restoration:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Fig 1.13 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, whereas removal of image blur by applying a deblurring function is considered a restoration technique.

Color image processing:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Fig 1.14 Color & Gray scale image

Wavelets and multiresolution processing:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform-based image processing since the late1950's, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

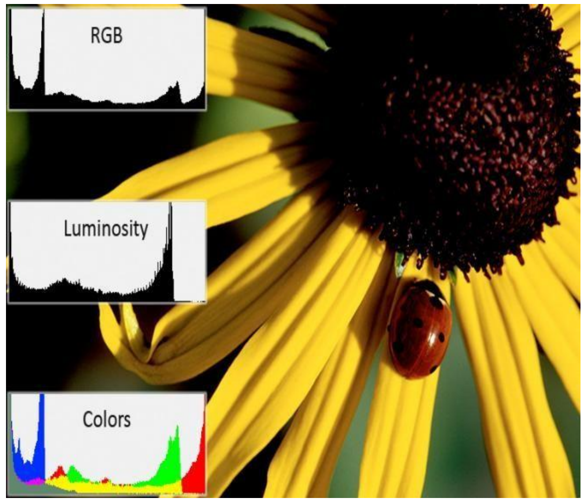


Fig 1.15 rgb histogram image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called **Multiresolution** theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

Compression:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

Morphological processing:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Fig 1.16 blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black (or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

Segmentation:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

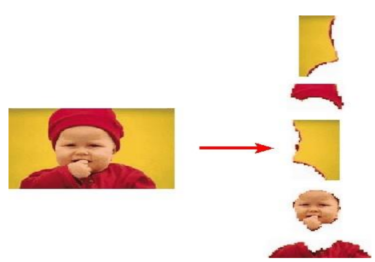


Fig 1.18 Image segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

Representation and description:

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

Object recognition:

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

Knowledgebase:

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an inter related to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

COMPONENTS OF AN IMAGE PROCESSING SYSTEM:

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

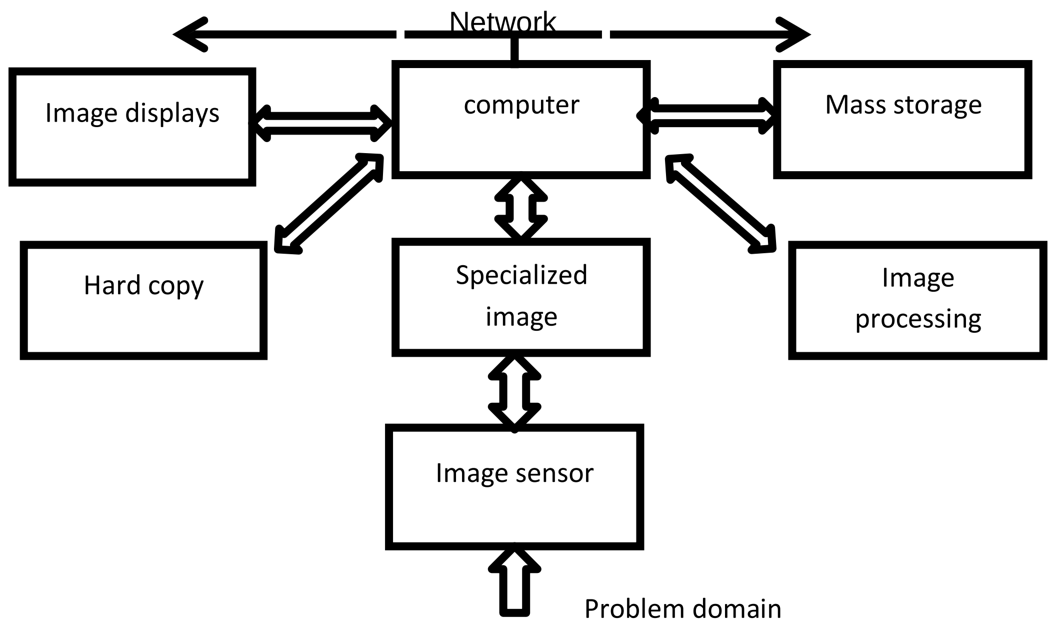


Fig 1.19 Component of image processing

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 1.24 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

- **Image sensors:**

With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

- **Specialized image processing hardware:**

Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

- **Computer:**

The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

- **Image processing software:**

Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

- **Mass storage:**

Mass storage capability is a must in image processing applications. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage for image processing applications fall into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), Bytes (meaning giga, or one billion, bytes), and Tbytes (meaning tera, or one trillion, bytes)

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig. 1.24. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in "jukeboxes" are the usual media for archival applications.

- **Image displays:**

  Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are their requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

- **Hardcopy:**

  Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

- **Network:**

  Networking is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of optical fiber and other broadband technologies.

Color and texture are two low-level features widely used for image classification, indexing and retrieval. Color is usually represented as a histogram, which is a first order statistical measure that captures global distribution of color in an image One of the main drawbacks of the histogram- based approaches is that the spatial distribution and local variations in color are ignored. Local spatial variation of pixel intensity is commonly used to capture texture information in an image. Grayscale Co-occurrence Matrix (GCM) is a well-known method for texture extraction in the spatial domain. A GCM stores the number of pixel neighborhoods in an image that have a particular grayscale combination. Let I be an image and let p and Np respectively denote any arbitrary pixel and its neighbor in a given direction. If GL denotes the total number of quantized gray levels and gl denotes the individual gray levels, where, gl ∈{0, . . .,GL _ 1}, then each component of GCM can be written as follows:

$$\mathrm{gcm}(i,j) = \mathrm{Pr}((\mathrm{gl}_{\ } \ \mathrm{gl}$$

g-cm (i, j) is the number of times the gray level of a pixel p denoted by glp equals i, and the gray level of its neighbor Np denoted by glNp equals j, as a fraction of the total number of pixels in the image. Thus, it estimates the probability that the gray level of an arbitrary pixel in an image is i, and that of its neighbor is j. One GCM matrix is generated for each possible neighborhood direction, namely, 0, 45, 90 and 135.Average and range of 14 features like Angular Second Moment, Contrast, Correlation, etc., are generated by combining all the four matrices to get a total of 28 features. In the GCM approach for texture extraction, color information is completely lost since only pixel gray levels are considered.

To incorporate spatial information along with the color of image pixels, a feature called color correlogram has recently been proposed. It is a three-dimensional matrix that represents the probability of finding pixels of any two given colors at a distance ‘d’ apart Auto correlogram is a variation of correlogram, which represents the probability of finding two pixels with the same color at a distance ‘d’ apart. This approach can effectively represent color distribution in an image. However, correlogram features do not capture intensity variation Many image databases often contain both color as well as gray scale images. The color correlogram method does not constitute a good descriptor in such databases.

Another method called Color Co-occurrence Matrix (CCM) has been proposed to capture color variation in an image. CCM is represented as a three-dimensional matrix, where color pair of the pixels p and Np are captured in the first two dimensions of the matrix and the spatial distance ‘d’ between these two pixels is captured in the third dimension. This approach is a generalization of the color correlogram and reduces to the pure color correlogram for d = 1. CCM is generated using only the Hue plane of the HSV (Hue, Saturation and Intensity Value) color space. The Hue axis is quantized into HL number of levels. If individual hue values are denoted by hl, where $\mathrm{hl} \in \{0, \ldots, \mathrm{HL} - 1\}$, then each component of CCM can be written as follows:

$$\mathrm{ccm}(i, j) = \Pr((\mathrm{hl}_{\mathrm{p}}, \mathrm{hl}_{N_{\mathrm{p}}}) = (i, j))$$

Four matrices representing neighbors at angles 0, 90, 180 and 270 are considered. This approach was further extended by separating the diagonal and the non-diagonal components of CCM to generate a Modified Color Co-occurrence Matrix (MCCM). MCCM, thus, may be written as follows: MCCM = (CCMD; CCMND)

Here, CCMD and CCMND correspond to the diagonal and off-diagonal components of CCM. The main drawback of this approach is that, like correlogram, it also captures only color information and intensity information is completely ignored.

An alternative approach is to capture intensity variation as a texture feature from an image and combine it with color features like histograms using suitable weights. One of the challenges of this approach is to determine suitable weights since these are highly application-dependent. In certain applications like Content-based Image Retrieval (CBIR), weights are often estimated from relevance feedback given by users.

While relevance feedback is sometimes effective, it makes the process of image retrieval user-dependent and iterative. There is also no guarantee on the convergence of the weight-learning algorithms. In order to overcome these problems, researchers have tried to combine color and texture features together during extraction.

proposed two approaches for capturing color and intensity variations from an image using the LUV color space. In the Single-channel Co-occurrence Matrix (SCM), variations for each color channel, namely, L, U and V are considered independently. In the Multi-channel Co-occurrence Matrix (MCM), variations are captured taking two channels at a time – UV, LU and LV. Since the LUV color space separates out chrominance (L and U) from luminance (V), SCM in effect, generates one GCM and two CCMs from each image independently. As a result, correlation between the color channels is lost

However, in MCM, the count of pair wise occurrences of the values of different channels of the color space is captured. Thus, each component of MCM can be written as follows:

mcmUV(i; j) = Pr((up; vNp) = (i; j))

mcmLU(i; j) = Pr((lp; uNp) = (i; j))

mcmLV(i; j) = Pr((lp; vNp) = (i; j))

Here, mcmUV(i, j) is the number of times the U chromaticity value of a pixel p denoted by up equals i, and the V chromaticity value of its neighbor Np denoted by vNp equals j, as a fraction of the total number of pixels in the image. Similarly, mcmLU(i, j) and mcmLV(i, j) are defined. One MCM matrix is generated for each of the four neighborhood directions, namely, 0, 45, 90 and 135.

Deng and Manjunath (2001) proposed a two-stage method called JSEG, which combines color and texture after image segmentation. In the first stage, colors are quantized to the required levels for differentiating between various regions of an image. Pixel values of the regions are then replaced by their quantized color levels to form a color map. Spatial variation of color levels between different regions in the map is viewed as a type of texture composition of the image.

Yu et al. (2002) suggested the use of color texture moments to represent both color and texture of an image. This approach is based on the calculation of Local Fourier Transformation (LFT) coefficients. Eight templates equivalent to LFT are operated over an image to generate a characteristic map of the image. Each template is a $3 \cdot 3$ filter that considers eight neighbors of the current pixel for LFT calculation. First and second order moments of the characteristic map are then used to generate a set of features.

In this project, we propose an integrated approach for capturing spatial variation of both color and intensity levels in the neighborhood of each pixel using the HSV color space. In contrast to the other methods, for each pixel and its neighbor, the amount of color and intensity variation between them is estimated using a weight function. Suitable constraints are satisfied while choosing the weight function for effectively relating visual perception of color and the HSV color space properties. The color and intensity variations are represented in a single composite feature known as Integrated Color and Intensity Co-occurrence Matrix (ICICM). While the existing schemes generally treat color and intensity separately, the proposed method provides a composite view to both color and intensity variations in the same feature. The main advantage of using ICICM is that it avoids the use of weights to combine individual color and texture features. We use ICICM feature in an image retrieval application from large image databases.

Early result on this work was reported in (Vadivel et al., 2004a). In the next section, we describe the proposed feature extraction technique after introducing some of the properties of the HSV color space. Choice of quantization levels for color and intensity axes, selection of parameter values and a brief overview of the image retrieval application

Integrated color and intensity co-occurrence matrix:

We propose to capture color and intensity variation around each pixel in a two-dimensional matrix called Integrated Color and Intensity Co-occurrence Matrix

(ICICM). This is a generalization of the Grayscale Co-occurrence Matrix and the Color Co-occurrence Matrix techniques. For each pair of neighboring pixels, we consider

their contribution to both color perception as well as gray level perception to the human eye. Some of the useful properties of the HSV color space and their relationship to human color perception are utilized for extracting this feature. In the next sub-section, we briefly explain relevant properties of the HSV color space. In the subsequent subsection, we describe how the properties can be effectively used for generating ICICM.

HSV color space:

HSV Color space: Basically, there are three properties or three dimensions of color that being hue, saturation and value HSV means Hue, Saturation and Value. It is important to look at because it describes the color based on three properties. It can create the full spectrum of colors by editing the HSV values. The first dimension is the Hue. Hue is the other name for the color or the complicated variation in the color. The quality of color as determined by its dominant wavelength. This Hue is broadly classified into three categories. They are primary Hue, Secondary Hue and Tertiary Hue. The first and the foremost is the primary Hue it consists of three colors they are red, yellow and blue. The secondary Hue is formed by the combination of the equal number of colors of the primary Hue and the colors of the secondary Hue which was formed by the primary Hue are Orange, Green and violet. The remaining one is the tertiary Hue is formed by the combination of the primary Hue and the secondary Hue. The limitless number of colors are produced by mixing the colors of the primary Hue in different amounts. Saturation is the degree or the purity of color. Then the second dimension is the saturation. Saturation just gives the intensity to the colors. The saturation and intensity drop just by mixing the colors or by adding black to the color. By adding the white to the color in spite of more intense the color becomes lighter. Then finally the third dimension is the Value. The value is the brightness of the color. When the value is zero the color space is totally black with the increase in the color there is also increase in the brightness and shows the various colors. The value describes the contrast of the color. That means it describes the lightness and darkness of the color. As similar to the saturation this value consists of the tints and shades. Tints are the colors with the added white and shades are the colors with the added black.

Properties of the HSV color space:

Sensing of light from an image in the layers of human retina is a complex process with rod cells contributing to scotopic or dim-light vision and cone cells to photopic or bright-light vision (Gonzalez and Woods, 2002). At low levels of illumination, only the rod cells are excited so that only gray shades are perceived. As the illumination level increases, more and more cone cells are excited, resulting in increased color perception. Various color spaces have been introduced to represent and specify colors in a way suitable for storage, processing or transmission of color information in images. Out of these, HSV is one of the models that separate out the luminance component (Intensity) of a pixel color from its chrominance components (Hue and Saturation). Hue represents pure color, which is perceived when incident light is of sufficient illumination and contains a single wavelength. Saturation gives a measure of the degree by which a pure color is diluted by white light. For light with low illumination, corresponding intensity value in the HSV color space is also low.

The HSV color space can be represented as a Hexa cone, with the central vertical axis denoting the luminance component, I (often denoted by V for Intensity Value). Hue, is a chrominance component defined as an angle in the range [0,2p] relative to the red axis with red at angle 0, green at 2p/3, blue at 4p/3 and red again at 2p. Saturation, S, is the other chrominance component, measured as a radial distance from the central axis of the hexacone with value between 0 at the center to 1 at the outer surface. For zero saturation, as the intensity is increased, we move from black to white through various shades of gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the color represented by its hue. When saturation is near 0, all the pixels in an image look alike even though their hue values are different.

As we increase saturation towards 1, the colors get separated out and are visually perceived as the true colors represented by their hues. Low saturation implies presence of a large number of spectral components in the incident light, causing loss of color information even though the illumination level is sufficiently high. Thus, for low values of saturation or intensity, we can approximate a pixel color by a gray level while for higher saturation and intensity, the pixel color can be approximated by its hue. For low intensities, even for a high saturation, a pixel color is close to its gray value. Similarly, for low saturation even for a high value of intensity, a pixel is perceived as gray. We use these properties to estimate the degree by which a pixel contributes to color perception and gray level perception.

One possible way of capturing color perception of a pixel is to choose suitable thresholds on the intensity and saturation. If the saturation and the intensity are above their respective thresholds, we may consider the pixel to have color dominance; else, it has gray level dominance. However, such a hard thresholding does not properly capture color perception near the threshold values. This is due to the fact that there is no fixed level of illumination above which the cone cells get excited. Instead, there is a gradual transition from scotopic to photopic vision. Similarly, there is no fixed threshold for the saturation of cone cells that leads to loss of chromatic information at higher levels of illumination caused by color dilution. We, therefore, use suitable weights that vary smoothly with saturation and intensity to represent both color and gray scale perception for each pixel.

NON-INTERVAL QUANTIZATION:

Due to the large range for each component by directly calculating the characteristics for the retrieval then the computation will be very difficult to ensure rapid retrieval. It is essential to quantify HSV space component to reduce computation and improve efficiency. At the same time, because the human eye to distinguish colors is limited, do not need to calculate all segments. Unequal interval quantization according the human color perception has been applied on H , S ,V components.

Based on the color model of substantial analysis, we divide color into eight parts. Saturation and intensity is divided into three parts separately in accordance with the human eyes to distinguish. In accordance with the different colors and subjective color perception quantification, quantified hue(H), saturation(S) and value(V)

In accordance with the quantization level above, the H, S, V three-dimensional feature vector for different values of with different weights to form one dimensional feature vector and is given by the following equation:

$$G = Qs*Qv*H+Qv*s+V$$

Where Qs is the quantized series of S and Qv is the quantized series of V. And now by setting Qs = Qv = 3, Then G = 9H+3S+V

$$H = \begin{cases} 0 & if \quad h \in [316,20] \\ 1 & if \quad h \in [21,40] \\ 2 & if \quad h \in [41,75] \\ 3 & if \quad h \in [76,155] \\ 4 & if \quad h \in [156,190] \\ 5 & if \quad h \in [191,270] \\ 6 & if \quad h \in [271,295] \\ 7 & if \quad h \in [296,315] \end{cases}$$

$$S = \begin{cases} 0 & if \quad s \in [0,0.2) \\ 1 & if \quad s \in [0.2,0.7) \\ 2 & if \quad s \in [0.7,1) \end{cases}$$

$$V = \begin{cases} 0 & if \quad v \in [0,0.2) \\ 1 & if \quad v \in [0.2,0.7) \\ 2 & if \quad v \in [0.7,1) \end{cases}$$

In this way three component vector of the HSV from one dimensional vector, Which quantize the whole color space for the 72 kinds of the main colors. So we can handle 72 bins of one dimensional histogram. This qualification is effective in reducing the images by the effect of the light intensity, but also reducing the computational time and complexity.

IMAGE RETRIEVAL:

Image retrieval is nothing but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords or the descriptions to the images so that the retrieval can be performed. Manual image annotation is time consuming, expensive and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc.,

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others. one can today generate volumes of images with content as diverse as family get-togethers and national park visits. Low-cost storage and easy Web hosting has fueled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchable image data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real-world image search system designers.

As far as technological advances are concerned, growth in Content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real-world implications, applications, and constraints of the technology. Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future.

An image retrieval system designed to serve a personal collection should focus on features such as personalization, flexibility of browsing, and display methodology. For example, Google's Picasa system [Picasa 2004] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods. The notable few applications include an FPGA implementation of a color-histogram-based image retrieval system [Kotoulas and Andreadis 2003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 2003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 2002].

Discussion. Regardless of the nature of the collection, as the expected user-base grows, factors such as concurrent query support, efficient caching, and parallel and distributed processing of requests become critical. For future real-world image retrieval systems, both software and hardware approaches to address these issues are essential. More realistically, dedicated specialized servers, optimized memory and storage support, and highly parallelizable image search algorithms to exploit cluster computing powers are where the future of large-scale image search hardware support lies.

OVERVIEW OF TEXTURE:

We all know about the term Texture but for defining it is a hard time. One can differentiate the two different Textures by recognizing the similarities and differences. Commonly there are three ways for the usage of the Textures:

Based on the Textures the images can be segmented To differentiate between already segmented regions or to classify them. We can reproduce Textures by producing the descriptions. The texture can be analyzed in three different ways. They are Spectral, Structural and Statistical:

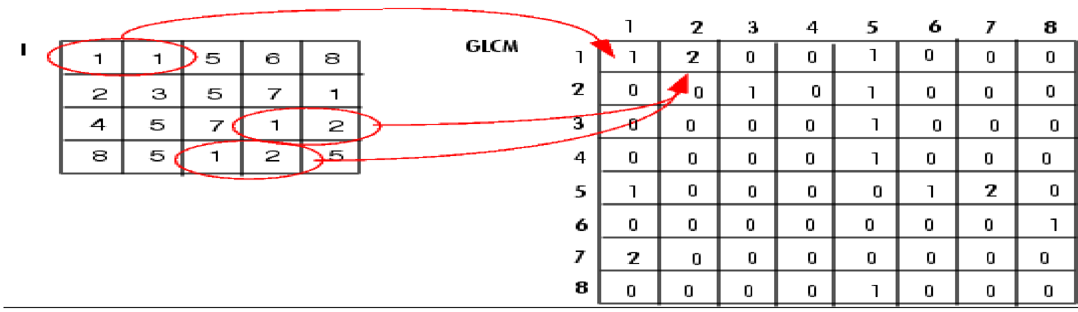


Fig 3.7 Texture enhancement

Digital image processing

Background:

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An important characteristic underlying the design of image processing systems is the significant level of testing & experimentation that normally is required before arriving at an acceptable solution. This characteristic implies that the ability to formulate approaches &quickly prototype candidate solutions generally plays a major role in reducing the cost & time required to arrive at a viable system implementation.

What is DIP

An image may be defined as a two-dimensional function f(x, y), where x & y are spatial coordinates, & the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital image by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensor, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis& computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting & somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low-level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low- level process is characterized by the fact that both its inputs & outputs are images.

Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images. Finally higher- level processing involves “Making sense” of an ensemble of recognized objects, as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

What is an image?

An image is represented as a two dimensional function f(x, y) where x and y are spatial co-ordinates and the amplitude of 'f' at any pair of coordinates (x, y) is called the intensity of the image at that point.

Gray scale image:

A grayscale image is a function I (xylem) of the two spatial coordinates of the image plane.

I(x, y) is the intensity of the image at the point (x, y) on the image plane.

I (xylem) takes non-negative values assume the image is bounded by a rectangle [0, a] ×[0, b]I: [0, a] × [0, b] → [0, info)

Color image:

It can be represented by three functions, R (xylem) for red, G (xylem) for green *and* B (xylem) for blue.

An image may be continuous with respect to the x and y coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate's values is called sampling. Digitizing the amplitude values is called quantization.

Coordinate convention:

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image f(x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size M X N. The values of the coordinates (xylem) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates.

In many image processing books, the image origin is defined to be at (xylem)=(0,0).The next coordinate values along the first row of the image are (xylem)=(0,1).It is important to keep in mind that the notation (0,1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Following figure shows the coordinate convention. Note that x ranges from 0 to M-1 and y from 0 to N-1 in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (xylem) the toolbox uses the notation (race) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate topples, (alb), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentation refers to the coordinates. Less frequently the toolbox also employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refers to rows. This is the opposite of our use of variables x and y.

Image as Matrices:

The preceding discussion leads to the following representation for a digitized image function:

$$f\,(xylem)= \begin{matrix} f(0,0) & f(0,1) & \ldots\ldots\ldots.. & f(0,N-1) \\ f(1,0) & f(1,1) & \ldots\ldots\ldots\ldots & f(1,N-1) \\ . & . & & . \\ . & . & & . \\ f(M-1,0) & f(M-1,1) & \ldots\ldots\ldots\ldots\ldots & f(M-1,N-1) \end{matrix}$$

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements.

A digital image can be represented naturally as a MATLAB matrix:

$$
f = \begin{matrix} f(1,1) & f(1,2) \ldots\ldots & f(1,N) \\ f(2,1) & f(2,2)\ldots\ldots\ldots & f(2,N) \\ . & . & . \\ . & . & . \\ f(M,1) & f(M,2)\ldots\ldots\ldots & f(M,N) \end{matrix}
$$

Where f (1,1) = f(0,0) (note the use of a monoscope font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation f(p ,q) denotes the element located in row p and the column q. For example f(6,2) is the element in the sixth row and second column of the matrix f. Typically we use the letters M and N respectively to denote the number of rows and columns in a matrix. A 1xN matrix is called a row vector whereas an Mx1 matrix is called a column vector. A 1x1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using mono-scope characters. We use conventional Roman, italic notation such as f(x ,y), for mathematical expressions

Reading Images:

Images are read into the MATLAB environment using function imread whose syntax Imread (‘filename’)

Format name	Description	recognized extension
BMP	Windows Bitmap	.bmp
GIF	Graphics Interchange Format	.gif
JPEG	Joint Photograph Experts Group	.jpg, .jpeg
PNG	Portable Network Graphics	.png
TIFF	Tagged Image File Format	.tif, .tiff
XWD	X Window Dump	.xwd

Table 1.1 Syntax for Imread

Here filename is a spring containing the complete of the image file (including any applicable extension).For example the command line

>> f = imread (‘8. jpg’);

Reads the JPEG (above table) image chest ray into image array f. Note the use of single quotes (‘) to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output If a semicolon is not included. MATLAB displays the results of the operation(s) specified in that line. The prompt symbol (>>) designates the beginning of a command line, as it appears in the MATLAB command window.

Data Classes:

Although we work with integers coordinates the values of pixels themselves are not restricted to be integers in MATLAB. Table above list various data classes supported by MATLAB and IPT are representing pixels values. The first eight entries in the table are refers to as numeric data classes. The ninth entry is the char class and, as shown, the last entry is referred to as logical data class.

All numeric computations in MATLAB are done in double quantities, so this is also a frequent data class encounter in image processing applications. Class unit 8 also is encountered frequently, especially when reading data from storages devices, as 8 bit images are most common representations found in practice. These two data classes, classes logical, and, to a lesser degree, class unit 16 constitute the primary data classes on which we focus. Many ipt functions however support all the data classes listed in table. Data class double requires 8 bytes to represent a number uint8 and int 8 require one byte each, uint16 and int16 requires 2bytes and unit 32.

Name **Description**

Double Double _ precision, floating_ point numbers the Approximate.

Uint8 unsigned 8_bit integers in the range [0,255] (1byte per Element).

Uint16 unsigned 16_bit integers in the range [0, 65535] (2byte per element).

Uint 32 unsigned 32_bit integers in the range [0, 4294967295](4 bytes per element). Int8 signed 8_bit integers in the range [-128,127] 1 byte per element)

Int 16 signed 16_byte integers in the range [32768, 32767] (2 bytes per element).

Int 32 Signed 32_byte integers in the range [-2147483648, 21474833647] (4 byte per element).

Single single _precision floating _point numbers with values

In the approximate range (4 bytes per elements)

Char characters (2 bytes per elements).

Logical values are 0 to 1 (1byte per element).

Int 32 and single required 4 bytes each. The char data class holds characters in Unicode representation. A character string is merely a 1*n array of characters logical array contains only the values 0 to 1,with each element being stored in memory using function logical or by using relational operators.

Image Types:

The toolbox supports four types of images:

1. Intensity images;

2. Binary images;

3. Indexed images;

4. R G B images.

Most monochrome image processing operations are carried out using binary or intensity images, so our initial focus is on these two image types. Indexed and RGB colour images.

Intensity Images:

An intensity image is a data matrix whose values have been scaled to represent intentions. When the elements of an intensity image are of class unit8, or class unit 16, they have integer values in the range [0,255] and [0, 65535], respectively. If the image is of class double, the values are floating point numbers. Values of scaled, double intensity images are in the range [0, 1] by convention.

Binary Images:

Binary images have a very specific meaning in MATLAB.A binary image is a logical array 0s and1s.Thus, an array of 0s and 1s whose values are of data class, say unit8, is not considered as a binary image in MATLAB. A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, we create an array B using the statement.

B=logical (A)

If A contains elements other than 0s and 1s.Use of the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.

Using relational and logical operators also creates logical arrays.

To test if an array is logical we use the I logical function: is logical(c).

If c is a logical array, this function returns a 1. Otherwise returns a 0. Logical array can be converted to numeric arrays using the data class conversion functions.

Indexed Images:

An indexed image has two components:

A data matrix integer, x

A color map matrix, map

Matrix map is an m*3 arrays of class double containing floating point values in the range [0, 1]. The length m of the map are equal to the number of colors it defines. Each row of map specifies the red, green and blue components of a single color. An indexed images uses “direct mapping” of pixel intensity values color map values. The color of each pixel is determined by using the corresponding value the integer matrix x as a pointer in to map. If x is of class double, then all of its components with values less than or equal to 1 point to the first row in map, all components with value 2 point to the second row and so on. If x is of class units or unit 16, then all components value 0 point to the first row in map, all components with value 1 point to the second and so on.

RGB Image:

An RGB color image is an M*N*3 array of color pixels where each color pixel is triplet corresponding to the red, green and blue components of an RGB image, at a specific spatial location. An RGB image may be viewed as “stack” of three gray scale images that when fed in to the red, green and blue inputs of a color monitor

Produce a color image on the screen. Convention the three images forming an RGB color image are referred to as the red, green and blue components images. The data class of the components images determines their range of values. If an RGB image is of class double the range of values is [0, 1].

Similarly, the range of values is [0,255] or [0, 65535]. For RGB images of class units or unit 16 respectively. The number of bits use to represents the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8bit image, the corresponding RGB image is said to be 24 bits deep.

Generally, the number of bits in all component images is the same. In this case the number of possible colors in an RGB image is (2^b) ^3, where b is a number of bits in each component image. For the 8bit case the number is 16,777,216 colors

CHAPTER 2

LITERATURE SURVEY

2.1 Literature survey

Image generation tasks such as deblurring, DE mosaicking, and super-resolution are difficult tasks because there are multiple viable image outputs. In super resolution, recent advancements allow the domain of possible outputs to be constricted using perceptual loss and adversarial loss. State-of-the-art methods use these losses to generate realistic high-resolution (HR) images. These methods are single-image super resolution (SISR) approaches that attempts to recover a high-resolution (HR) image from its low-resolution (LR) observation. However, using single images for super-resolution is not practical for recovering textural details when using large upscaling factors due to the extreme information loss. For example, take an image patch of a green cloth with grid patterns, after image down sampling image, the image patch will become a blob of green and the information about the grid patterns is lost. Different from SISR, reference-based super-resolution (Ref SR) solutions utilize HR reference images to compensate lost details in LR images. Previous works in Ref SR have good performance by using image statistics of the reference (Ref) images to enhance the LR input image. However, these approaches assume the Ref images have similar content and/or have good alignment with the LR image; otherwise, their performance would degrade and even become worse than SISR methods. Recent works also use texture transfer; however, the Ref images are assumed to be homogeneous texture images and there is still performance reduction if the Ref images are different from the LR input. Ideally, a robust Ref SR algorithm should outperform SISR when good Ref images are given, and achieve comparable performance as SISR when Ref images are not provided or do not possess relevant texture at all.

CHAPTER 3

PROPOSED METHOD

Block Diagram

Introduction

Our proposed method, Aug TT (Augmented Texture Transfer), is a robust Ref SR approach that improves upon existing SISR algorithms. Aug TT leverages the performance of existing SISR approaches by refining their super-resolved (SR) outputs rather than building an SR image from ground up. We enhance the SISR output by performing neural texture transfer, where we query the Ref images for the best matching texture features then use the queried features to augment the SR image. The query is performed by computing the similarity of the texture and semantic features between the patches of the LR image and LR counterpart of the Ref Image. The most similar features are used to enhance the SR output of the SISR baselines through an augmentation network. Previous Ref SR approaches have performance degradation when Ref images have dissimilar content with the LR image. To prevent this problem, we use the similarity scores, which was computed in the texture query, as supplementary input to our network. Furthermore, during network training, we randomly select patch features from the Ref image to condition our augmentation network to not immediately trust the queried texture features. This ensures that our method has at least comparable results with the SISR methods and the lower-bound performance is based on the SISR method.

We proposed a Ref SR method that enhances image out-puts of existing SISR methods using neural texture transfer. We implemented a texture query method that takes account of texture and semantic details. We ensured that our method is robust even if using dissimilar Ref images by utilizing similarity scores in the input features and performing random feature queries during the training. We present that our method drastically improves existing SISR approaches.

The proposed method, Aug TT, is a Ref SR approach that attempts to reconstruct the HR image IHR given the input LR image ILR with the help of Ref images IRef . Different from other Ref SR approaches that attempts to generate IHR by itself, Aug TT leverages existing SISR approaches. Specifically, the SISR method SISR(·), will attempt to generate an SR image ˆISR given ILR. In this paper, we use 4 × magnification factor for the ILR.

$$\hat{I}_{SR} = SISR(I_{LR}) \qquad (1)$$

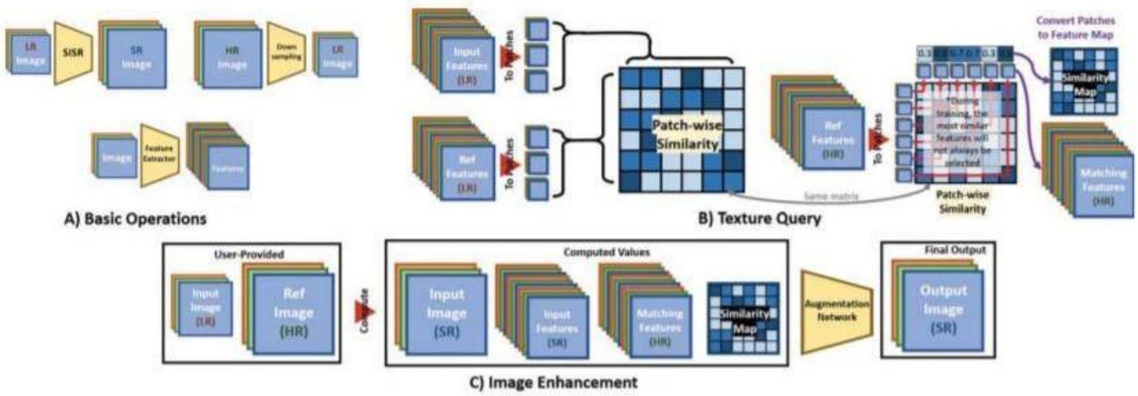


Fig 3.1 Block diagram

Our method, Aug TT, attempts to refine ˆISR using Ref images IRef. Note that there could be more than one Ref images but for simplicity's sake IRef is represented as if it is a single

$$I_{SR} = AugTT(\hat{I}_{SR}, I_{Ref}) \qquad (2)$$

entity

Aug TT stands for Augmented Texture Transfer, where the Augmented keyword represents its refinement of existing SISR methods, while the Texture Transfer is the refinement methodology. In the following subsections we will explain

Texture Query

To enhance the SR image ˆISR, we query the Ref images IRef for relevant texture features. The query is performed by finding the most similar patches to the input image from the Ref image. However, the input and Ref images are in different domains, as such, we match them. The Ref image is in the natural HR domain while the input image could be in the LR domain (ILR) or SR domain (ˆISR). We do not have access to the natural HR domain while computing in the SR domain is too memory-intensive. Therefore, we down sample the Ref image to the LR domain using bicubic interpolation producing I↓ Ref . We extract the texture and semantic features of both ILR and I↓ Ref using a pre-trained VGG19 network [17]. The intermediate outputs of the layers in a VGG network can extract different levels of details [19], with shallow layers extracting low-level features (e.g., textures, edges) and deeper layers extracting high-level features (e.g., shapes, objects). The texture and semantic features of ILR and I↓ Ref are from the intermediate outputs of the VGG network, specifically, we use the layers relu1 1, relu2 1, and relu3 1. Then, we convert the features into patches, with stride = 1 and patch size = 3. The patches are acquired in a sliding window fashion similar to the convolution operation. Since we view the features as patches, we could use arbitrary number of Ref images, with each additional Ref image appending more patches in the patch list.

For each intermediate features (e.g., relu1 1), we compute a similarity matrix between the two sets of patches using cosine distance. The final similarity matrix S is the weighted sum of the similarity matrix for each intermediate features. We set the shallow layers (i.e., relu1 1) of VGG [17] to have the highest weights. We will use S for querying the HR Ref features for the best matching patches

We extract the features from the original Ref image IRef, specifically, relu1 1. We selected only this layer since it has sufficient information to reconstruct the entire image [19]. Then we convert the features into patches and use matrix S to query the best matching patches. The patches are converted into feature map in a reverse sliding window fashion. The feature maps has cell overlaps so we divide each cell with its corresponding number of overlaps.

In case of Ref images that has dissimilar content with the input image, we include the similarity score in the input of our augmentation network. The similarity scores of the best patches is converted into a feature map using the similar method performed on the feature patches.

Image Enhancement

We will use the queried texture features to enhance the SR image through an augmentation network defined in Figure 2. The network has four input maps that will be concatenated channel-wise. The (1) similarity map and (2) matching feature map from the texture query. The (3) SR image from the SISR method, and (4) its features. The SR features and the matching features are generated from the same layer, so that processing the queried features will be easier. The augmentation network is a feed-forward network with stacks of CNN but there is a residual connection from the SR image and before the refined SR output. We want the network to learn the modifications on the input SR to generate the refined SR, rather than build the final output from ground up.

Training Details

During training, the best matching features is not always selected for image enhancements. This is to ensure that the network can still function despite the sub-optimal texture guidance. Furthermore, the augmentation network is initially trained with Bicubic interpolation as the SISR method. The network is optimized for different SISR approaches after the initial training. To train our augmentation network, we use the following loss functions. Reconstruction loss is computed using mean -norm in the pixel space (i.e. RGB) between ISR and IHR. Perceptual loss is computed with MSE in the feature space using a VGG network [17] to extract the features. Adversarial loss is computed using the formula provided by WGANGP [20] and it is, basically, a loss function that could learn the image statistics of real HR images. The dataset we used for training our augmentation network is CUFED5 [16], which contains HR target image, Ref image, and LR input image with sizes 160×160, 160×160, and 40×40, respectively.

CHAPTER 4

SOFTWARE INTRODUCTION:

Introduction to MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

The MATLAB system:

The MATLAB system consists of five main parts

- **Development Environment**:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library**:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language**:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

- **Graphics:**

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

GRAPHICAL USER INTERFACE (GUI):

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.) and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension. fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.

- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted,

GUIDE opens a new (i.e., blank) window.

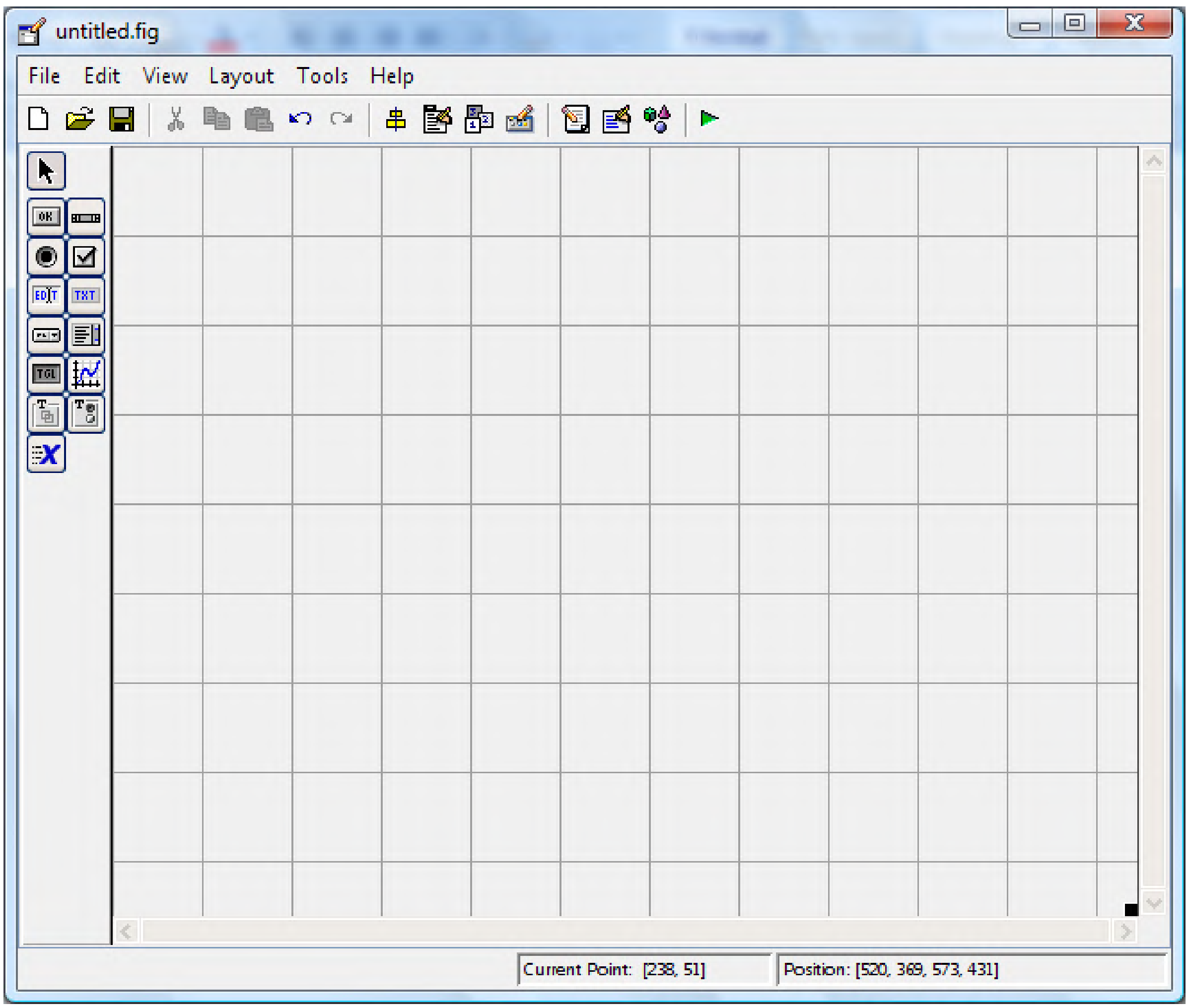


Fig 4.1 Graphical user interface

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

Getting Started

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

Introduction - describes the components of the MATLAB system.

Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

DEVELOPMENT ENVIRONMENT

4.5.1Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with

MATLAB. The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example, emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type help browser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Content’s tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorite's tab - View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace as from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a. mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to run MATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

MANIPULATING MATRICES

Entering Matrices

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Durer’s matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Durer’s matrix, simply type in the Command Window

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

MATLAB displays the matrix you just entered.

```
A =

   16    3    2   13
```

5	10	11	8
9	6	7	12
4	15	14	1

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3 -99 0.0001

9.6397238 1.60210e-20 6.02252e23

1i -3.14159j 3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

Operators

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power

'	Complex conjugate transpose
( )	Specify evaluation order

Table: 4.1 Operators

Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.14159265...
I	Imaginary unit, √-1
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2- \varepsilon)2^{1023}$
Inf	Infinity

NaN	Not-a-number

Table 4.2 Constants

GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

GUI Development Environment

The process of implementing a GUI involves two basic tasks.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of its children (uncontrol's and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the callbacks, which are defined as subfunctions. This M-file is referred to as the application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

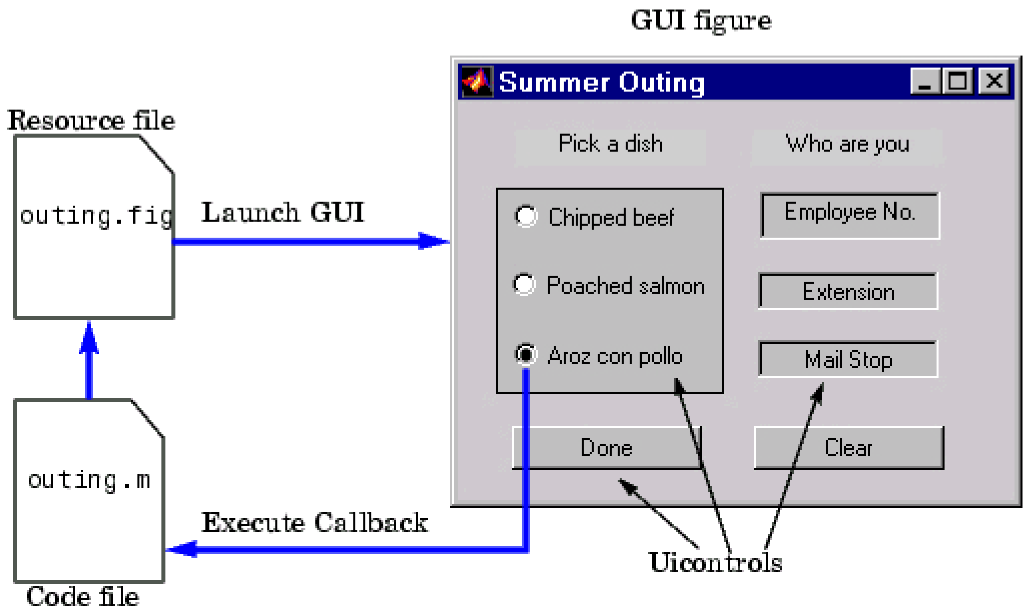


FIG 4.2 Graphical user blocks

Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uncontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques used in the application M-file.

Application Examples - a collection of examples that illustrate techniques which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uncontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- List boxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)

button_state = get(h,'Value');

if button_state == get(h,'Max')

   % Toggle button is pressed

elseif button_state == get(h,'Min')

   % toggle button is not pressed

end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 true color image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand (16,128);

a(:,:,2) = rand(16,128);

a(:,:,3) = rand(16,128);

set(h,'CData',a)
```

Radio Buttons

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)

set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)

off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];

mutual_exclude(off)

% Continue with callback

 .

 .

 .
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)

if (get(h,'Value') == get(h,'Max'))

   % then checkbox is checked-take approriate action

else

   % checkbox is not checked-take approriate action

end
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)

user_string = get(h,'string');

% proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)

user_entry = str2double(get(h,'string'));

if isnan(user_entry)

   errordlg('You must enter a numeric value','Bad Input','modal')

end

% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menu bar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menu bar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uncontrol's can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Non integer values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

List boxes differentiate between single and double clicks on an item and set the figure SelectionType property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure SelectionType property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

switch val

case 1

% The user selected the first item

case 2

% The user selected the second item

% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

string_list = get(h,'String');

selected_string = string_list{val}; % convert from cell array to string

% etc.
```

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's Button-down callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uncontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes Button-down property to define the callback.

Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

axes(handles. axes1)

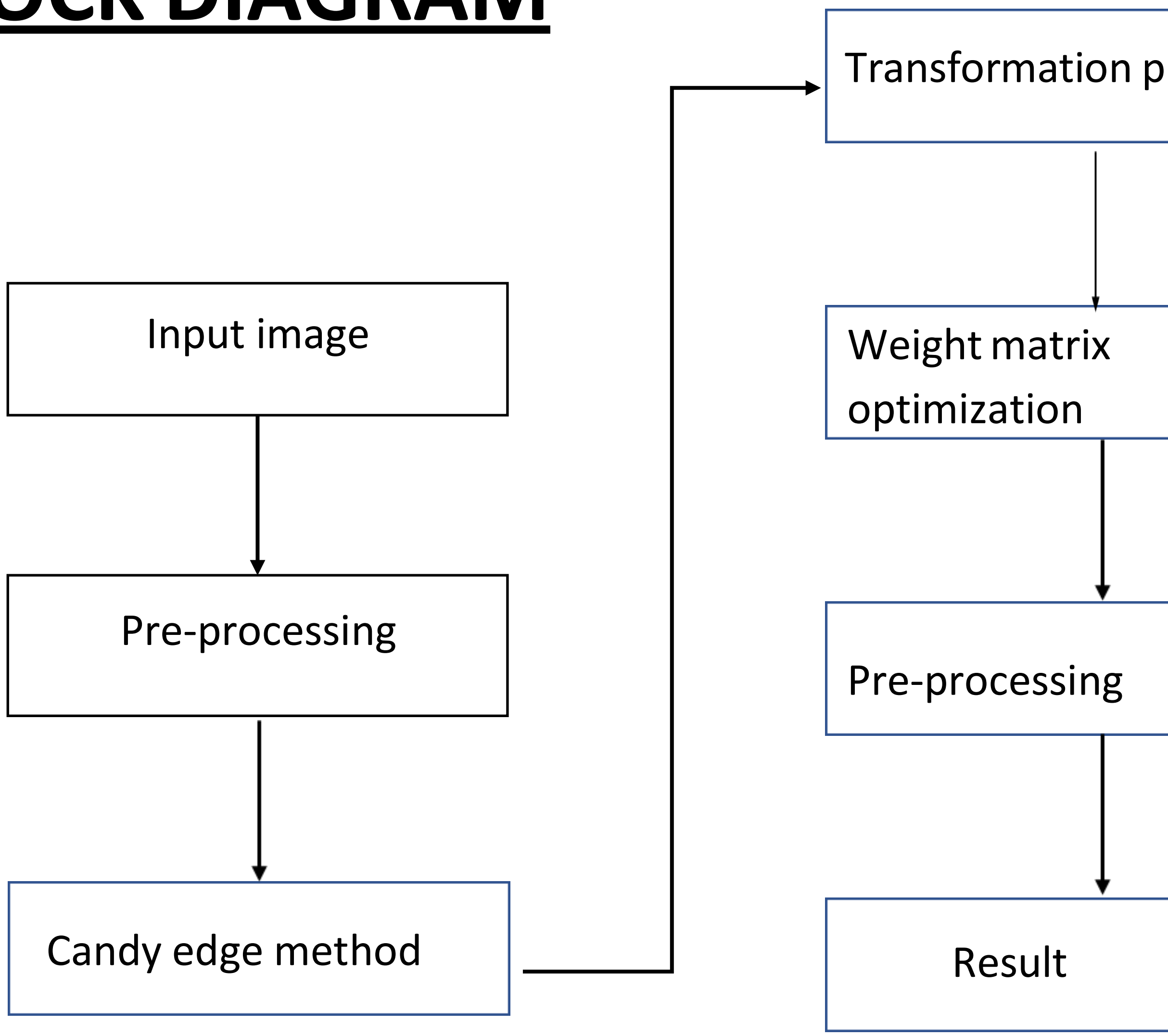
makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axis. See GUI with Multiple Axes for an example that uses two axes.

Figure

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control.

CHAPTER 5:

BLOCK DIAGRAM



Recent deep learning approaches in single image super resolution (SISR) can generate high-definition textures for super-resolved (SR) images. However, they tend to hallucinate fake textures and even produce artifacts. An alternative to SISR, reference-based SR (Ref SR) approaches use high resolution (HR) reference (Ref) images to provide HR details that are missing in the low-resolution (LR) input image. We propose a novel framework that leverages existing SISR approaches and enhances them with Ref SR. Specifically, we refine the output of SISR methods using neural texture transfer, where HR features are queried from the Ref images. The query is conducted by computing the similarity of textural and semantic features between the input image and the Ref images. The most similar HR features, patch-wise, to the LR image is used to augment the SR image through an augmentation network. In the case of dissimilar Ref images from the LR input image, we prevent performance degradation by including the similarity scores in the input features of the network. Furthermore, we use random texture patches during the training to condition our augmentation network to not always trust the queried texture features. Different from past Ref SR approaches, our method can use arbitrary Ref images and its lower-bound performance is based on the SR image. We showcase that our method drastically improves the performance of the base SISR approach

CHAPTER 7

EXPERIMENT RESULTS

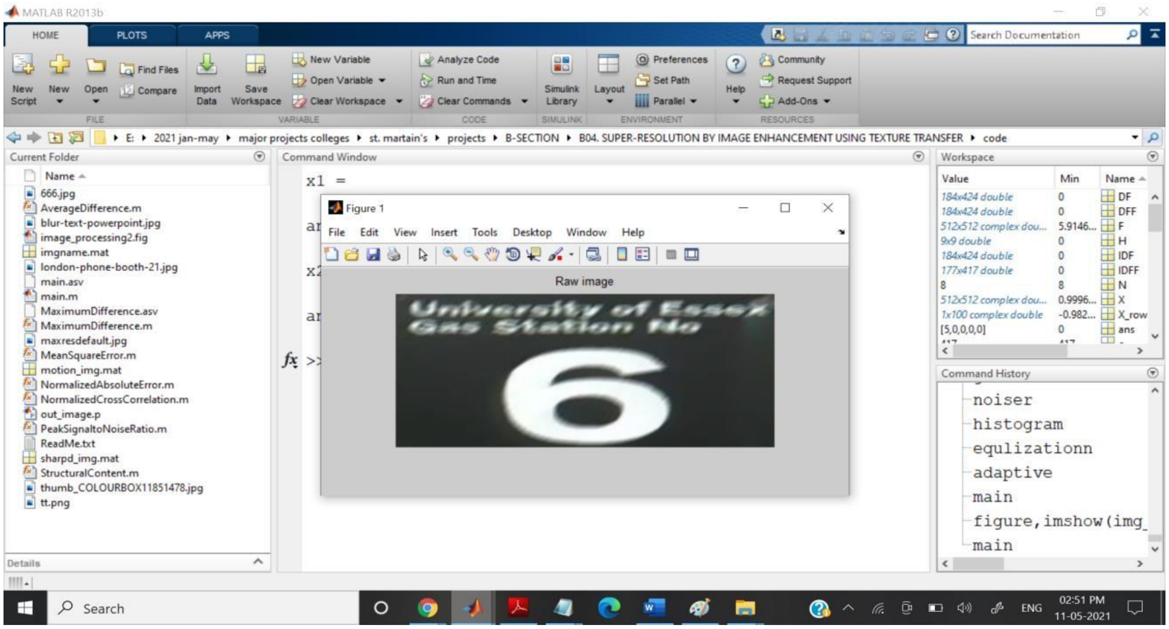


Fig 7.1 input image

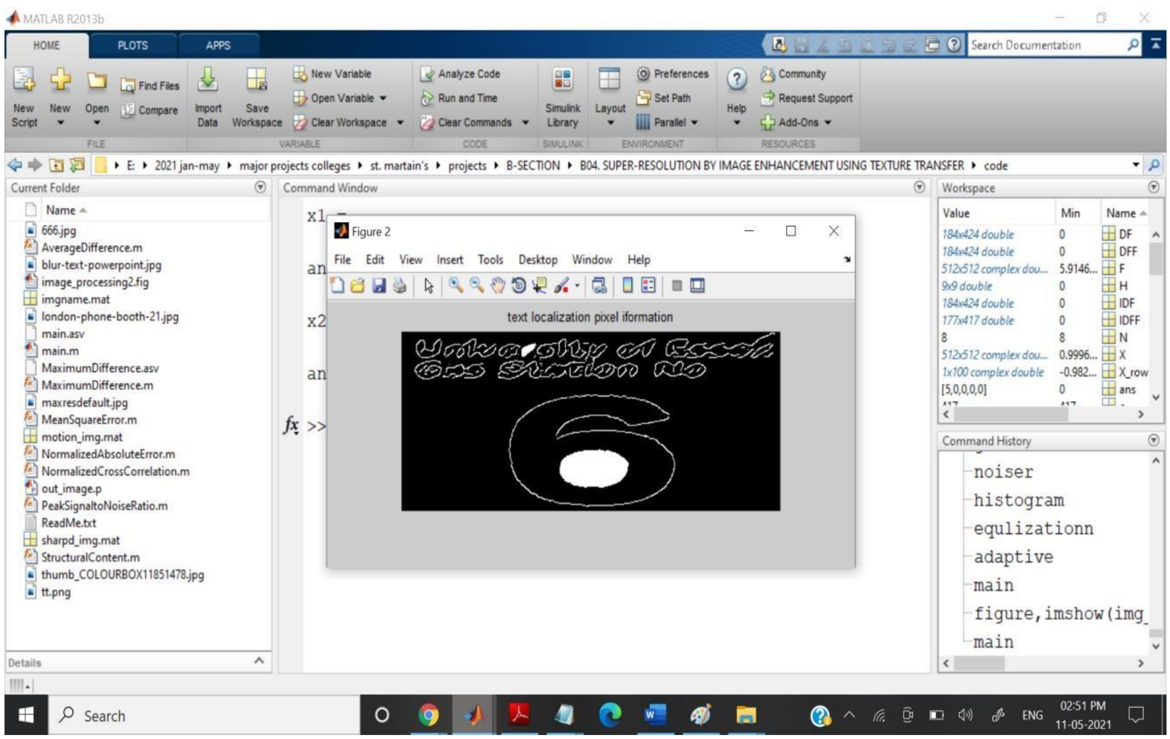


Fig 7.2 Gradient Image of Input

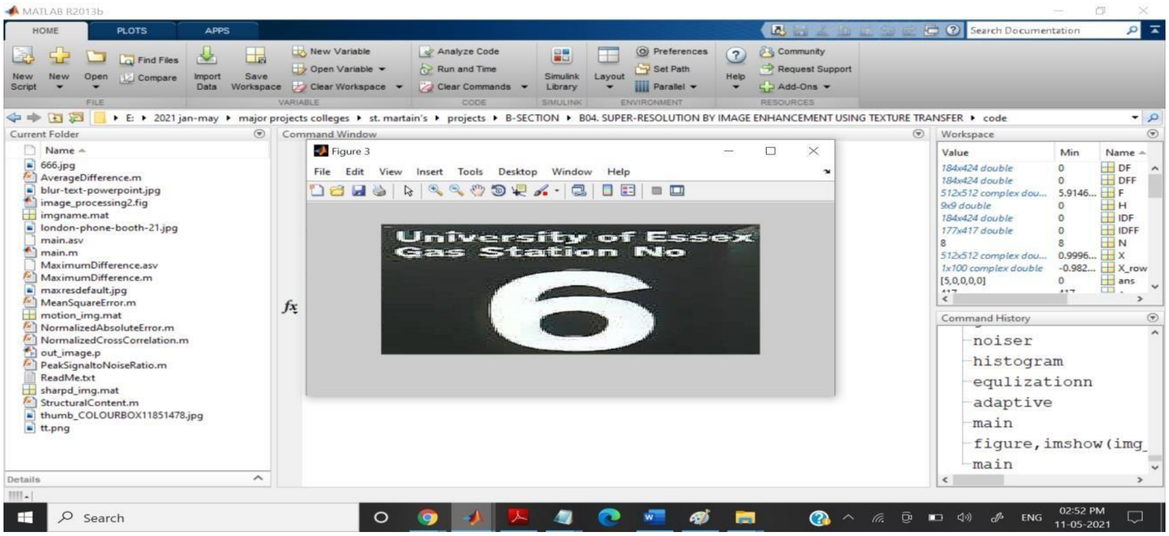


Fig 7.3 Output

CHAPTER 8

CONCLUSIONS AND FUTURE ENCHANCEMENT

Conclusion

In this paper, we proposed the RefSR method AugTT, which augments the output of existing SISR methods using Ref images. The Ref images provide the missing texture details that cannot be retrieved from the LR image due to extreme down sampling scale (e.g. 4 ×). We query the Ref images for the best matching texture using textural and semantic features extracted using a pre-trained VGG [17] network. The best matching texture features is used to enhance the image output of the SISR baselines. To prevent degradation due to Ref-LR content dissimilarity, we use the similarity scores in the input features of our augmentation network. Furthermore, during training, we don't always use the best matching features so that we condition our network to not always trust the provided features. Our method sets the SR input image for its lower-bound performance. We showcase that AugTT could drastically enhance the SR image and our method is robust even when the Ref images does not provide relevant textures.

Future Enhancement

REFERENCES

[1] Jose Jaena Mari Ople, Pin-Yi Yeh, Shih-Wei Sun, I-Te Tsai, and Kai-Lung Hua, "Multi-scale neural network with dilated convolutions for image deblurring," IEEE Access, vol. 8, pp. 53942–53952, 2020.

[2] Daniel Stanley Tan, Wei-Yang Chen, and Kai-Lung Hua, "Deepdemosaicking: Adaptive image demosaicking via multiple deep fully convolutional networks," IEEE Transactions on Image Processing, vol. 27, no. 5, pp. 2408–2419, 2018.

[3] Kai-Lung Hua, Shintami Chusnul Hidayati, Dangling He, Chia-Po Wei, and Yu-Chiang Frank Wang, "Context-aware joint dictionary learning for color image DE mosaicking," Journal of Visual Communication and Image Representation, vol. 38, pp. 230–245, 2016.

[4] Fang-Lin He, Yu-Chiang Frank Wang, and Kai-Lung Hua, "Self-learning approach to color DE mosaicking via support vector regression," in 2012 19th IEEE International Conference on Image Processing. IEEE, 2012, pp. 2765–2768.

[5] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Ca- ´ ballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al., "Photo-realistic single image super resolution using a generative adversarial network," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4681–4690.

[6] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1646–1654.

[7] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Ca- ´ ballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al.,

"Photo-realistic single image super resolution using a generative adversarial network," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4681–4690.

[8] Mehdi SM Sajjadi, Bernhard Scholkopf, and Michael Hirsch, "Enhancement: Single image super-resolution through automated texture synthesis," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4491–4500.

[9] Justin Johnson, Alexandre Alahi, and Li Fei-Fei, "Perceptual losses for real-time style transfer and super resolution," in European conference on computer vision. Springer, 2016, pp. 694–711.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in Advances in neural information processing systems, 2014, pp. 2672–2680.

[11] Manri Cheon, Jun-Hyuk Kim, Jun-Ho Choi, and JongSeok Lee, "Generative adversarial network-based image super-resolution using perceptual content losses," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 0–0.

[12] Assaf Shocher, Nadav Cohen, and Michal Irani, ""zero shot" super-resolution using deep internal learning," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3118–3126.

[13] Radu Timofte, Vincent De Smet, and Luc Van Gool, "Anchored neighborhood regression for fast example based super-resolution," in Proceedings of the IEEE international conference on computer vision, 2013, pp. 1920–1927.

[14] Libin Sun and James Hays, "Super-resolution from internet-scale scene matching," in 2012 IEEE International Conference on Computational Photography (ICCP). IEEE, 2012, pp. 1–12

. [15] Il Jun Ahn, Nam Nam, and Woo Hyun, “Texture enhancement via high-resolution style transfer for single image super-resolution,” Electronic Imaging, vol. 2018, no. 5, pp. 245–1, 2018.

A

PROJECT REPORT

On

SIMPLE AND SECURE IMAGE STEGANOGRAPHY USING LSB AND TRIPLE XOR OPERATION ON MSB

Submitted by

1) Mr. M Snehan Reddy (17K81A0493)

2) Ms. S Ashwitha Reddy (17K81A04B0)

3) Mr. D Manish Reddy (17K81A0472)

in partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

IN

Electronics and Communication Engineering

Under The Guidance of

Mr. K. Karthik

(B.Tech., M.Tech)

DEPARTMENT OF ECE

ST.MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

ST.MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad – 500 100

NBA & NAAC A+ Accredited

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**SIMPLE AND SECURE IMAGE STEGANOGRAPHY USING LSB AND TRIPLE XOR OPERATION ON MSB**" is being submitted by **1.Mr. M.Snehan Reddy** (17K81A0493), **2.Ms.S.Ashwitha Reddy** (17K81A04B0), **3.Mr. D.Manish Reddy** (17K81A0472) in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology** in **Electronics and Communication Engineering** is recorded of bonafide work carried out by them. The results embodied in this report have been verified and found satisfactory.

Mr. K.Karthik
Department of Electronics and
Communication Engineering.

Head of the Department
Dr.B. Hari krishna ,
Department of Electronics and
Communication Engineering.

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **D.MANISH REDDY** WITH ROLL NO.**17K81A0472, M.SNEHAN REDDY** WITH ROLL NO.**17K81A0493**, **S.ASHWITHA REDDY** WITH ROLL NO.**17K81A04B0**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**SIMPLE AND SECURE IMAGE STEGANOGRAPHY USING LSB AND TRIPLE XOR OPERATION ON MSB**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014 Email : contact@lasyainfotech.com, o v@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of **Bachelor of Technology** in Department of Electronics and Communication Engineering, session: <2017 - 2021>, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work Entitled **"SIMPLE AND SECURE IMAGE STEGANOGRAPHY USING LSB AND TRIPLE XOR OPERATION ON MSB"** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

M Snehan Reddy 17K81A0493
S Ashwitha Reddy 17K81A04B0
D Manish Reddy 17K81A0472

ACKNOWLEDGEMENTS

While bringing out this project report to its final form, we came across a number of people whose contributions in various ways helped my field of research and they deserve special thanks. It is a pleasure to convey our gratitude to all of them.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our college management** for their kind support and permission to use the facilities available in the institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.Santosh Kumar Patra**, our **beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna, Head & professor,** the **Dept. of Electronics and Communication Engineering** for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our **Dept. Project Coordinator** Dr. **A. Anand, Assistant Professor, Dept.of ECE** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project guide, **Mr.K.Karthik, Assistant Professor,** Department of ECE and our class project coordinator **Mr. K. Karthik, Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level project coordinators **Dr.Malayadri, Professor in IT & Dr.R.Santhosh Kumar, Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and cooperation that is difficult to express in words.

1.M Snehan Reddy

2.S Ashwitha Reddy

3.D Manish Reddy

ABSTRACT

Least Significant Bit (LSB) is a very popular method in the spatial domain of steganographic images. This method is widely used and continues to be developed to date, because of its advantages in steganographic image quality. However, the traditional LSB method is very simple and predictable. It needs a way to improve the security of hidden messages in this way. This research proposes a simple and safe way to hide messages in LSB techniques. Three times the XOR operation is done to encrypt the message before it is embedded on the LSB. To facilitate the process of encryption and decryption of messages, three MSB bits are used as keys in XOR operations. The results of this study prove that this method provides security to messages with very simple operation. The imperceptibility quality of the stego image is also excellentwith a PSNR value above 50 dB.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

Introduction to the project

Data hiding is of importance in many applications. For hobbyists, secretive data transmission, for privacy of users etc. the basic methods are: Steganography and Cryptography. Steganography is a simple security method. Generally there are three different methods used for hiding information: steganography, cryptography, watermarking. In cryptography, the information to be hidden is encoded using certain techniques; this information is generally understood to be coded as the data appears nonsensical. Steganography is hiding information; this generally cannot be identified because the coded information doesn't appear to be abnormal i.e. its presence is undetectable by sight. Detection of steganography is called Steganalysis.

IMAGE STEGANOGRAPHY

Image compression is a technique which is widely used in Steganography. It is of two types- lossy compression and lossless compression. Lossy compression may not preserve the integrity of original image where as Lossless compression preserves the original image data correctly. Hence lossless compression is chosen. Examples of Lossless compression formats are GIF [84], BMP and PNG formats. JPEG format is the example for Lossy compression format.

Types of Steganography

1. Text steganography
2. Image steganography
3. Audio steganography
4. Video steganography

In all of these methods, the basic principle of steganography is that a secret message is to be

embedded in another cover object which may not be of any significance in such a way that the encrypted data would finally display only the cover data. So it cannot be detected easily to be containing hidden information unless proper decryption is used.

Cryptography, where the goal is to secure communications from an eaves- dropper, steganographic techniques strive to hide the very presence of the message itself from an observer. The general idea of hiding some information in digital content has a wider class of applications that go beyond steganography, The techniques involved in such applications are collectively referred to as information hiding. For example, an image printed on a document could be annotated by metadata that could lead a user to its high resolution version.

In general, metadata provides additional information about an image. Although metadata can also be stored in the file header of a digital image, this approach has many limitations. Usually, when a file is transformed to another format (e.g., from TIFF to JPEG or to BMP), the metadata is lost. Similarly, cropping or any other form of image manipulation destroys the metadata. Finally, metadata can only be attached to an image as long as the image exists in the digital form and is lost once the image is printed. Information hiding allows the metadata to travel with the image regardless of the file format and image state (digital or analog).

A special case of information hiding is digital watermarking. Digital watermarking is the process of embedding information into digital multimedia content such that the information (the watermark) can later be extracted or detected for a variety of purposes including copy prevention and control. Digital watermarking has become an active and important area of research, and development and commercialization of watermarking techniques is being deemed essential to help address some of the challenges faced by the rapid proliferation of digital content.

The key difference between information hiding and watermarking is the absence of an active adversary. In watermarking applications like copyright protection and authentication, there is an active adversary that would attempt to remove, invalidate or forge watermarks. In information hiding there is no such active adversary as there is no value associated with the act of removing the information hidden in the content. Nevertheless, information hiding techniques need to be robust against accidental distortions. Unlike information hiding and digital watermarking, the main goal of steganography is to communicate securely in a completely undetectable manner. Although steganography is an ancient art, first used against the persian by the romans, it has evolved much through the years.

Image Steganography has many applications, especially in today's modern, hightech world. Privacy and anonymity is a concern for most people on the internet. Image Steganography allows for two parties to communicate secretly and covertly. It allows for some morally-conscious people to safely whistle blow on internal actions; it allows for copyright protection on digital files using the message as a digital watermark. One of the other main uses for Image Steganography is for the transportation of high-level or top-secret documents between international governments. While Image Steganography has many legitimate uses, it can also be quite nefarious. It can be used by hackers to send viruses and trojans to compromise machines, and also by terrorists and other organizations that rely on covert operations to communicate secretly and safely.

Types of Steganography messages in text

Secret messages can be hidden in text format by reframing the text of the carrier file, while maintaining the context. On e form of steganography is a program called Spam Mimic. Based on a set of rules called a mimic engine by Peter Wayner, it encodes your message into what looks like your typical, quickly deleted Spam message. However, hiding a message in plain text is a thing of past, as people are suspicious of irrelevant text. MESSAGES IN STILL IMAGES most popular tool is outguess. MESSAGES IN AUDIO data is hidden in layer III of encoding process of MP3 file. Messages in audio are always sent along with ambient noise. The data is hidden in the heart of the layer III encoding process of MP3 file, namely the inner loop during compression.

The inner loop limits the input data and increases the step size until the data can be coded with the available number of bits. The data is compressed, encrypted and then hidden in MP3 bit stream. MESSAGES IN VIDEO embedding information into multimedia data has gained increasing attention lately. The method of encryption is the same as in audio steganography. Video files are generally very good carrier files since they have a lot of irrelevant bits. AN EXAMPLE Fishing freshwater bends and saltwater coasts rewards anyone feeling stressed. Resourceful anglers usually find masterful leapers fun and admit swordfish rank overwhelming any day. “Send lawyers guns and money” Steganography is closely related to the problem of "hidden channels" in secure operating system design, a term which refers to all communication paths that can not easily be restricted by access control mechanisms (e.g. two processes that communicate by modulating and measuring the CPU load).

Steganography is also closely related to spread spectrum radio transmission, a technique that allows to receive radio signals that are over 100 times weaker than the atmospheric background noise, as well as TEMPEST, techniques which analyze RF transmissions of computer and communication equipment in order to get access to secret information handled by these systems. Most communication channels like telephone lines and radio broadcasts transmit signals which are always accompanied by some kind of noise. This noise can be replaced by a secret signal that has been transformed into a form that is indistinguishable from noise without knowledge of a secret key and this way, the secret signal can be transmitted.

Implementation:

There are currently three effective methods in applying Image Steganography: LSB Substitution, Blocking, and Palette Modification. LSB (Least Significant Bit) Substitution is the process of modifying the least significant bit of the pixels of the carrier image. Blocking works by breaking up an image into “blocks” and using Discrete Cosine Transforms (DCT). Each block is broken into 64 DCT coefficients that approximate luminance and color—the values of which are modified for hiding messages. Palette Modification replaces the unused colors within an image’s color palette with colors that represent the hidden message. With LSB Substitution I could easily change from Image Steganography to Audio Steganography and hide a zip archive instead of a text message. LSB Substitution lends itself to become a very powerful Steganographic method with few limitations. LSB Substitution works by iterating through the pixels of an image and extracting the ARGB values. It then separates the color channels and gets the least significant bit. Meanwhile, it also iterates through the characters of the message setting .

Steganography can be viewed as akin to cryptography. Both have been used throughout recorded history as means to protect information. At times these two technologies seem to converge while the objectives of the two differ. Cryptographic techniques "scramble" messages so if intercepted, the messages cannot be understood. Steganography, an essence, "camouflages" a message to hide its existence and make it seem "invisible" thus concealing the fact that a message is being sent altogether. An encrypted message may draw suspicion while an invisible message will not. In an ideal world we would all be able to openly send encrypted email or files to each other with no fear of reprisals. However there are often cases when this is not possible, either because you are working for a company that does not allow encrypted email or perhaps the local government does not approve of encrypted communication (a reality in some parts of the world). This is where steganography can come into play.

A good steganography system should fulfill the same requirements posed by the "Kirchhoff principle" in cryptography. This means that the security of the system has to be based on the assumption that the "enemy" has full knowledge of the design and implementation details of the steganographic system. The only missing information for the "enemy" is a short easily exchangeable random number sequence, the secret key, and without the secret key, the "enemy" should not have the slightest chance of even becoming suspicious that on an observed communication channel hidden communication might take place. Steganography cannot be detected. Therefore, it is used when encryption is not permitted. Or, more commonly, steganography is used to supplement encryption. An encrypted file may still hide information using steganography, so even if the encrypted file is deciphered, the hidden message is not seen.

Evolution of Steganography Code Breakers :

David Kahn's The Code breakers and Bruce Norman’s Secret Warfare: The Battle of Codes and Ciphers recounts numerous tales of steganography. INVISIBLE INK : An innocent letter may contain a very different message written between the lines with invisible ink. Common sources for invisible inks are milk, vinegar, fruit juices and urine. All of these darken when heated. Later on, more sophisticated inks were developed which react to various chemicals. MICRODOTS: The Germans developed microdot technology.

Microdots are photographs the size of a printed period having the clarity of standard-sized typewritten pages. The first microdots were discovered masquerading as a period on a typed envelope carried by a German agent in 1941. Besides being so small, microdots permitted the transmission of large amounts of data including drawings and photographs.

DISSECTING STEGANOGRAPHY

Steganography is a term used for hiding messages within an image. Any colour pixel is made of a combination of red –green-blue mode (RGB) wherein each RGB component consist of 8 bits. If letters in ASCII are to be represented within the colour pixels, the rightmost digit, called the least significant bit (LSB), can be altered. Any variation in the value of this bit leads to very minimal variation in colour. If we have to hide the word ‘digit’ in the image, we take the LSB of every colour and hide each bit of the word in its RGB combination. To insert the letter ‘D’ we modify three colour pixels with three bits in each colour pixel.

Steps for hiding an image using Steganography

1. Start s-tool and window explorer using the later as drag and drop interface for the software. 2. Drag and drop the image to be used as the carrier file from the explorer onto the actions window in s-tool.

3. Drag and drop the data file on the carrier file.

4. Give pass phrase and encryption algorithm when prompted. Pass these to receiver too.

5. The hidden file is ready. Receiver has to click on the “reveal” button to extract the data.

Steganography simply takes one piece of information and hides it within another. Computer files (images, sounds recordings, even disks) contain unused or insignificant areas of data. Steganography takes advantage of these areas, replacing them with information (encrypted mail, for instance). The files can then be exchanged without anyone knowing what really lies inside of them. An image of the space shuttle landing might contain a private letter to a friend. A recording of a short sentence might contain your company's plans for a secret new product. Steganography can also be used to place a hidden "trademark" in images, music, and software, a technique referred to as watermarking

There are two different methods for image steganography:

1. Spatial methods
2. Transform methods

In spatial method, the most common method used is LSB substitution method. Least significant bit (LSB) method is a common, simple approach to embedding information
in a cover file. In steganography, LSB substitution method is used. i.e. since every image has three components (RGB). This pixel information is stored in encoded format in one byte. The first bits containing this information for every pixel can be modified to store the hidden text. For this, the preliminary condition is that the text to be stored has to be smaller or of equal size to the image used to hide the text.

LSB based method is a spatial domain method. But this is vulnerable to cropping and noise. In this method, the MSB (most significant bits) of the message image to be hidden are stored in the LSB (least significant bits) of the image used as the cover image. It is known that the pixels in an image are stored in the form of bits.

In a grayscale image, the intensity of each pixel is stored in 8 bits (1byte). Similarly for a colour (RGB-red, green, blue) image, each pixel requires 24 bits (8bits for each layer). The Human visual system (HVS) cannot detect changes in the colour or intensity of a pixel when the LSB bit is modified. This is psycho-visual redundancy since this can be used as an advantage to store information in these bits and yet notice no major difference in the image.

Steps used in LSB steganography:

a. Steps for hiding message image:

1. Read the image to be used as cover image. Noise is added to make it easier to disguise
changes due to embedding the message image.
2. Read the image to be sued as message image.
3. Separate the bit planes of each image.

As it is known that the LSB (least significant bit) plane contains the least information associated with any image, and the MSB (most significant bit) plane contains most of the shape, colour information of an image. It is generally ideal to replace up to 4 least bitplanes of the cover image, with the upper 4 bit-planes without revealing changes in the resultant image. Lesser number of bit-planes from the message image could be used, but the retrieved image would become distorted and loses information.

4. Replace the least 4 bit-planes of cover image with the 4 most significant bit- planes from
message image.

5. Get the resultant Steganographic image by recombining these bit-planes.

b. Retrieving message image:

1. Read the Steganographic image.
2. Extract the required number of bit-planes of the image.
3. Recombining the lower four bit-planes would give the retrieved message image.

Steganographic Security:

In steganography, unlike other forms of communications, one's awareness of the underlying communication between the sender and receiver defeats the whole purpose. Therefore, the first requirement of a steganographic system is its undetectability. In other words, a steganographic system is considered to be insecure, if the warden Wendy is able to differentiate between cover objects and stego-objects. There have been various approaches in defining and evaluating the security of a steganographic system. Zollner et al. were among the first to address the undetectability aspect of stegano graphical systems.

They provide an analysis to show that information theoretically secure steganography is possible if embedding operation has a random nature and the embedded message is independent from both the cover-object and stego-object. These conditions, however, ensure undetectability against an attacker who knows the stego- object but has no information available about the indeterministic embedding operation. That is, Wendy has no access to the statistics, distribution, or conditional distribution of the cover-object. On the other hand, approached steganographic security from a complexity theoretic point of view. Based on cryptographic principles, they propose the design of encryption-decryption functions for steganographic embedding and detection.

In this setting, the underlying distribution of the cover-objects is known by the attacker, and undetectability is defined in a conditional sense as the inability of a polynomial-time attacker (Wendy) to distinguish the stego-object from a cover-object. This model assumes that stego-object is a distorted version of the cover-object, however, it does not attempt to probabilistically characterize the stego object. Cachin defined the first steganographic security measure that quantifies the information theoretic security of a stegosystem. His model assigns probability distributions to cover-object and stego-object under which they are produced. Then, the task of Wendy is to decide whether the observed object is produced according to known cover-object distribution or not. In the best case scenario, Wendy also knows the distribution of stego-object and makes a decision by performing a binary hypothesis test. Consequently, the detectability of a stegosystem is based on relative entropy between the probability distributions of the cover-object and stego-object, denoted by Pc and Ps, respectively, i.e., D(Pc||Ps) = Z Pc log Pc Ps . (1)

From this equation, we note that D(Pc||Ps) increases with the ratio P c P s which in turn means that the reliability of steganalysis detector will also increase. Accordingly, a stego technique is said to be perfectly secure if D(Pc||Ps) = 0 (Pc and Ps are equal), and ²-secure if the relative entropy between Pc and Ps is at most ², D(Pc||Ps) ≤ ². Perfectly secure algorithms are shown to exist, although they are impractical.

However, it should be noted that this definition of security is based on the assumption that the cover-object and stego-object are independent, identically distributed (i.i.d.) vectors of random variables. Since Wendy uses hypothesis testing in distinguishing between stegoobjects and cover-objects, she will make two types of errors, namely, typeI and type-II errors.

A type-I error, with probability α occurs, when a cover-object is mistaken for a stego-object (false alarm rate), and a typeII error, with probability β, occurs when a stego-object is mistaken for a cover-object (miss rate). Thus bounds on these error probabilities can be computed using relative entropy, thereby relating steganographic security to detection error probabilities. Cachin obtains these bounds utilizing the facts that deterministic processing can not increase the relative entropy between two distributions, say, Pc and Ps, and hypothesis testing is a form of processing by a binary function that yields α (P(detect message present | message absent)) and β (P(detect message absent | message present)). Then, the relative entropy between distributions Pc and Ps and binary relative entropy of two distributions with parameters $(\alpha, 1 - \alpha)$ and $(\beta, 1 - \beta)$ need to satisfy $d(\alpha, \beta) \leq D(Pc||Ps)$, (2) where $d(\alpha, \beta)$ is expressed as $d(\alpha, \beta) = \alpha \log \alpha\ 1 - \beta + (1 - \alpha) \log 1 - \alpha\ \beta$. (3) Then, for an ²- secure stegosystem we have $d(\alpha, \beta) \leq$ ². (4) Consequently, when the false alarm rate is set to zero ($\alpha = 0$), the miss rate is lower bounded as $\beta \geq 2$ −² . It should be noted that the probability of detection error for . If we approximate the probability distribution functions using histograms, then, examples such a Wendy is defined as Pe = αP(message absent) + βP(message present). (5) Based on above equations, for a perfectly secure stegosystem, $\alpha + \beta = 1$, and when a cover-object is equally likely to undergo embedding operation, then Pe = 1 2 . Hence, Wendy's decisions are unreliable. As one can observe, there are several shortcomings in the above definition of security.

This is because, real life cover-objects have a rich statistical structure in terms of correlation, higher-order dependence, etc. By exploiting these structures, it is possible to design good steganalysis detectors even if the first order probability distribution is preserved (i.e., ² = 0) during the embedding process. If we approximate the probability distribution functions using histograms, then, examples such as show that it is possible to design good steganalysis detectors even if the histograms of the cover image is and the stego image are the same.

Techniques for Image Steganography :

Given the proliferation of digital images, and given the high degree of redundancy present in a digital representation of an image (despite compression), there has been an increased interest in using digital images as cover-objects for the purpose of steganography. Therefore we have limited our discussion to the case of images for the rest of this tutorial. We should also note that there have been much more work on embedding techniques which make use of the transform domain or more specifically JPEG images due to their wide popularity. Thus to an attacker the fact that an image other that that of JPEG format is being transferred between two entities could hint of suspicious activity. There have been a number of image steganography algorithm proposed, these algorithm could be categorized in a number of ways:

• Spatial or Transform, depending on redundancies used from either domain for the embedding process.

• Model based or ad-hoc, if the algorithm models statistical properties before embedding and preserves them, or otherwise.

• Active or Passive Warden, based on whether the design of embedder- detector pair takes into account the presence of an active attacker.

Another category for embedding techniques for which a number of algorithms have been proposed is the transform domain embedding category. Most of the work in this category has been concentrated on making use of redundancies in the DCT (discrete cosine transform) domain, which is used in JPEG compression. But there has been other algorithms which make use of other transform domains such as the frequency domain [30]. Embedding in DCT domain is simply done by altering the DCT coefficients, for example by changing the least significant bit of each coefficient. One of the constraints of embedding in DCT domain is that many of the 64 coefficients are equal to zero, and changing two many zeros to non-zeros values will have an effect on the compression rate. That is why the number of bit one could embed in DCT domain, is less that the number of bits one could embed by the LSB method. Also the embedding capacity becomes dependent on the image type used in the case of DCT embedding, since depending on the texture of image the number of non-zero DCT coefficients will vary. Although changing the DCT coefficients will cause unnoticeable visual artifices, they do cause detectable statistical changes.

As mentioned before, another transform domain which has been used for embedding is the frequency domain. Alturki et al. propose quantizing the coefficients in the frequency domain in order to embed messages. They first decorrelate the image by scrambling the pixels randomly, which in effect whitens the frequency domain of the image and increases the number of transform coefficients in the frequency domain thus increasing the embedding capacity. The frequency coefficients are then quantized to even or odd multiples of the quantization step size to embed zeros or ones. Then the inverse FFT of the signal is taken and descrambled. The resulting image would be visually incomparable to the original image. But statistically the image changes and as the authors show in their work, the result of the embedding operation is the addition of a gaussian noise to the image.

Nowadays and because of unsecure networks and internet that can be accessed by anyone it became very risky to send important messages and files without any security measures. Before sending an important message we have to make sure that it will arrive to the destination in a secure way without being seen or modified by an intruder. Because of the threats mentioned before, steganography was discovered. Steganography in brief is the art of hiding information by other information. Usually we need to hide a text beneath any type of media. The most type of media used in hiding text is image. Steganography can be considered a new technology that still needs work and improvements. We are interesting in image steganography base type because it is very good idea according to a security method to send an image that doesn‟t draw any attention or suspicions about containing an important message.

Another important reason for our choice was the use of this type of security by terrorists to hide their plans so everyone should read and be aware of such subjects. Steganography is one of the newest technologies in sending secure data between sender and receiver. After the large spread of cryptography a new concept was added in order to add a new security level that prevents the detection of hidden information under a cover of any digital media. Nowadays, the internet became a robust and enormous world that offers a diversity of facilities and needs, because of these different functionalities it became easier to find diversity of ways to send messages to people without anyone noticing or even knowing that the message exist. Steganography is the procedure in which one type of communication (text, sound or image beneath another type.

Steganography is described by Neil F. Johnson and Sushil Jajodia in their paper „Steganalysis: The Investigation of Hidden Information" as „The goal of Steganography is to avoid drawing suspicion to the transmission of a hidden message. If suspicion is raised, then this goal is defeated." This technology is used widely nowadays, and a lot of people started hiding data, images video and audio by using it. For this reason even if a message seems normal and innocent it might be holding within it a secret message with high important data. The source of the word Steganography comes from the Greek word steganos which means covered or secret and the word graphy which means writing or drawing. So the whole meaning is secret writing .

To clarify the idea of steganography, the three famous characters named Alice, Bob and Ward are used. Alice (A) wants to send a secret message (M) to Bob (B). Bob must receive it safely without raising suspicion. To do that, Alice changes the message (M) into a steganography object (stego-object, i.e. new file carrying the embedded-object) (S). Stego-object is created by covering the message (M) with another random harmless message to produce a cover (C, i.e. data file that will hold the secret message). Covering the massage (M) with message (C) happens by using a secret key (stego-key) (K). Now Alice should be able to send the stego-object (S) to Bob without being detected by Ward. When Bob receives (S) he will use the stego- key (K) which he already knows to reproduce secret message (M) from the cover message (C) and be able to read it. Steganography have to guarantee these requirements :

Robustness: information is robust when it is embedded inside an image and although it disappeared behind it but it is not destroyed, it is present, but is only detected with reliability after modifying the image.

Undetectability: the data hidden under an image cannot be detected as long as the cover image is not doubtable or suspicious and looks unchanged.

Perceptual transparency: this requirement depends on human visual andaudio system. If the hidden data didn't raise the attention of human systems and no one could distinguish whether the cover contains secret data then this requirement is guaranteed.

Security: as long as no one other than the legal receiver can remove theembedded data from behind cover, the embedding algorithm is said to be secure. This requirement assures that no targeted attacks can detect or view the hidden message unless they have a full knowledge of the embedding algorithm.

There may be different approaches for steganography methods classification. They may be grouped according to the type of cover file or according to the different modifications each method may apply during embedding process. Our proposed technique belongs to the second classification type. We can find more than six different methods concerning the second classification, noting that in some cases it is very difficult to classify a specific method one clear category. Here are the six most used methods: Substitution, Injection (or insertion), Distortion, Generation, Transform domain techniques, Statistical steganography.

Internet technology provides many benefits for humans, especially in getting or exchanging information, learning, working, and others. One of the problems on the internet is security and data privacy. Many methods have been applied in providing security such as cryptography, steganography, watermarking, and digital signatures . Steganography and watermarking are branches of the science of hiding data on other media called cover . The difference is in its function, where steganography is used to hide messages while watermarking is used to preserve copyright. Data hiding in the image is divided into two domains, namely spatial domain and frequency domain. In the frequency domain, the message is hidden by first transforming the cover image. Transformations that have been widely used are, DCT, DWT, and SVD .

This chapter applies LSB Steganography technique for various lossless file formats such as BMP, GIF and PNG. The science which deals with the hidden communication is called Steganography. There are different kinds of steganographic techniques which are complex and which have strong and weak points in hiding the invisible information in various file formats. The innocent carriers are the possible cover carriers which will hold the hidden communication.

A Steganography method is admirably secure only when the statistics of the cover information and the stego information are similar with each other.

In other words it conveys the meaning that the relative entropy between the cover information and the stego information is zero. The LSB embedding technique suggests that data can be hidden in such a way that even the naked eye is unable to identify the hidden information in the LSBs of the cover file.

An encryption and decryption technique on the data to be hidden into the image file is performed to provide additional security to the data. 47 Steganography is an alternative method for privacy and security. Instead of encrypting, we can hide the messages in other innocuous looking medium (carrier) so that their existence is not revealed. Among the several advantages for employing the Steganography, secretly transmitting the secret information from source to destination is one. In this chapter, different approaches towards implementation of image Steganography have been thoroughly and clearly discussed. Among several techniques, Masking and Filtering, Algorithms and Transformations and LSB insertion are some of the methods to achieve Steganography. Among these techniques, LSB insertion is a very simple and commonly applied technique for embedding data in a cover file.

CHARCTERIZING DATA HIDING TECHNIQUES

An image is a picture that has been created or copied and stored in electronic form. An image can be described in terms of vector graphics or raster graphics. An image stored in raster form is sometimes called a bitmap. An image map is a file containing information that associates different locations on a specified image with hypertext links. An image is a collection of numbers that constitute different light intensities in different areas of the image.

This numeric representation forms a grid and the individual points are referred to as pixels (picture element).Greyscale images use 8 bits for each pixel and are able to display 256 different colours or shades of grey. Digital colour images are typically stored in 24-bit files and use the RGB colour model, also known as true colour. All colour variations for the pixels of a 24-bit image are derived from three primary colours: red, green and blue, and each primary colour is represented by 8 bits. Thus in one given pixel, there can be 256 different quantities of red, green and blue.

Steganography is a kind of technique which can embed a message inside a cover object. There are a number of features that characterizes the merits and demerits of the embedding techniques. The way they are applied decides the importance of each and every feature. A set of criteria are proposed to define the invisibility of an algorithm. The criterias are as follows: Invisibility The imperceptibility of a Steganography technique is the most important necessity, since the quality of Steganography lies in its capacity to be unseen by the naked eyes.

Payload Capacity Steganography techniques used aim at hiding the embedded secret data and also maximize the amount of information embedded. The amount of information that is hidden is called payload capacity. Hiding Capacity Concealing capacity is nothing but the size of data that could be concealed with respect to the size of the cover object. A vast concealing capacity permits the use of smaller cover images and thus decreases the data transmission needed to broadcast the stego image. Perceptual Transparency The inability of an eavesdropper to detect hidden data is referred by Perceptual transparency.

BINARY REPRESENTATION OF AN RGB COLOR IMAGE

For a 24-bit RGB image, every RGB component requires 8 bits of memory. The range of every RGB component value is in between 0 to 255 where 255 represent brightest shade of the color and 0 represents darkest shade of the color. All different colors could be produced with the combination of these ranges. Subsequently, the test image is represented by integer matrix. Every pixel is a mix of RGB values.

CHAPTER 2
LITERATURE SURVEY

Discrete Cosine Transform (DCT) method:

When information is embedded in spatial domain, losses can occur such as when the image is cropped etc. To overcome this problem the information is embedded in frequency domain in such a way that we embed the secret information in the significant frequency values and omit the higher frequency part. First the required transformations are applied and then accordingly to hide the secret message, the transform coefficients are changed. Like in other transforms, de-correlation of the image data is required after applying discrete cosine transform (DCT). And encoding can be then done independently for each coefficient. Hence, compression efficiency is not lost.

In blocking method, blocks of the image are considered and DCT (discrete cosine transform) is done in order to break them. Each block is then subdivided into

64 parts (DCT coefficients). These coefficients are modified i.e. the colour gets modified a little by storing some text or another image in it. Embedding the secret data in the carrier image is generally done for the DCT coefficients that are lower than the chosen threshold value. But embedding information in DCT coefficient value 0 is avoided as this may lead to visual distortion of the cover image.

Palette modification:

In palette modification, the unused colours in an image’s colour palette are replaced with colours to represent hidden message. Palette Modification replaces the unused colours within an image’s colour palette with colours that represent the hidden message. For example, we have an image containing 6shades of blue and 5 shades of brown. By modifying the bits, it is possible to generate a completely new palette of colours that were
originally absent in the previous image. This changed colour palette may not be detected easily by human eye (HVS) and hence can be used to store other data or information.

[1] R. D. Ardy, O. R. Indriani, C. A. Sari, D. R. I. M. Setiadi and E. H. Rachmawanto, "Digital Image Signature using Triple Protection Cryptosystem (RSA, Vigenere, and MD5)," in International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICONSONICS), Yogyakarta, 2017.
[2] A. Winarno, D. R. I. M. Setiadi, A. A. Arrasyid, C. A. Sari and E. H. Rachmawanto, "Image Watermarking using Low Wavelet Subband based on 8×8 Sub-block DCT," in International Seminar on Application for Technology of Information and Communication (iSemantic), Semarang, 2017.

CHAPTER 3

PROPOSED SYSTEM

ALGORITHMS USED IN PROPOSED METHOD

In the proposed method, Steganography is combined with Cryptography. It changes the meaning of the information as well as it hides the presence of information from the hacker. The LZ algorithm for compression and RSA algorithm for encryption and decryption are used in this chapter.

LSB The easiest way to embed secret information within the cover file is called LSB insertion. In this technique, the binary representations of the secret data have been taken and the LSB of each byte is overwritten within the image. If 24-bit color images are used, then the quantity of modification will be small. As an example, supposing that we have three neighbouring pixels (nine bytes) with the following RGB encoding:

50 01101010

11110010 00110110

01101001 11110000

00110101 01100000

11101111 00110100

Now if we wish to embed the following 9 bits of compressed secret information: 010010011. If we insert these 9 bits over the LSB of the 9 bytes above, we get the following sequence of bits (where bits in red color have been modified):

01101010 11110011

00110110 01101000

11110001 00110100

01100000 11101111

00110101

Note that we have successfully hidden 9 bits but at a cost of only modifying 5, or roughly 50% of the LSB bits. 3.4.2 LZ Compression Algorithm:

Step-1: Read the original file.

Step-2: Count the total number of words, alphabets, special characters and digits.

Step-3: Find out the repeated words in the file.

Step-4: Prepare the word dictionary for the original file context.

Step-5: Create compressed file. In the compressed file place the word's number instead of actual words.

Step-6: Add dictionary to compressed file.

Step-7: Save the compressed file along with the dictionary.

RSA Algorithm

In Cryptography, RSA is an algorithm for public-key Cryptography. The RSA algorithm involves three steps: Key generation, encryption and decryption. Key Generation: The keys for the RSA algorithm are generated in the following way:

Step-1: Choose two different random prime numbers p and q.

Step-2: Compute n = p*q. n is used as the modulus for both the private and public Keys.

Step-3: Compute φ (n) = (p-1) (q-1). (φ is Euler's totient function).

Step-4: Choose an integer e such that 1 < e < φ (pq), and gcd (e, φ(n))=1 Step-5: Compute d =e-1 mod [φ (n)]

Step-6: Publish the public encryption key: (e; n)

Step-7: Keep secret private decryption key: (d; n) 53

Encryption: The steps required to encrypt information at sender are as follows: Step-1: Obtain public key of recipient (e; n)

Step-2: Represent the information as an integer m in [0, n-1] Step-3: Compute c = me mod n Decryption:

The steps required to decrypt information at receiver side are as follows: Step-1: use private key (d; n)

Step-2: compute m = cd mod n

IMPLEMENTATION LSB :

Insertion is the easiest way to embed secret data in an image. By replacing the LSB of each sampling bit with a binary information, LSB insertion permits for a huge amount of secret information to be embedded. During hiding and unhiding procedure, the content of the secret information should not be modified.

LSB Encoding Algorithm First the original image, and the compressed encrypted secret message are taken. Then the encrypted secret data has to be converted into binary format. Binary conversion is done by taking the American Standard Code of Information Interchange (ASCII) values of the character and converting them into binary format and generating stream of bits. Similarly, in cover image, bytes representing the pixels are taken in single array and byte stream is generated. Message bits are taken sequentially and then are placed in LSB bit of image byte.

Same procedure is followed till all the message bits are placed in image bytes. Image generated is called 'Stego-Image'. It is ready for transmission through the Internet. Algorithm for hiding secret data in Cover image:

Step-1: Read the cover image and secret text information which is to be embedded in to the cover image.

Step-2: Compress the secret information.

Step-3: Convert the compressed secret information into cipher text by using secret key shared by receiver and sender.

Step-4: Convert compressed encrypted text message into binary form.

Step-5: Find LSBs of each RGB pixels of the cover image.

Step-6: Embed the bits of the secret information into bits of LSB of RGB pixels of the cover image.

Step-7: Continue the procedure until the secret information is fully hidden into cover file.

LSB Decoding Algorithm First, 'Stego-Image' is taken and single array of bytes are generated as it was done at the time of encoding. The total number of bits of encrypted secret information and the bytes representing the pixels of stego-image are taken. Counter is initially set to 1, which in turn gives the index number of the pixel byte where secret message bit is available in LSB.

The process is continued till final count of secret message bit is reached. After this, the bit stream of the message shall be generated. Available bits are grouped to form bytes such that each byte represents single ASCII character. Characters are stored in text file which represents the encrypted embedded message. After that the decryption and decompression are to be performed.

Algorithm for un hiding secret data from Stego image:

Step-1: Read the stego image.

Step-2: Find LSBs of each RGB pixel of the stego image.

Step-3: Find and retrieve the LSBs of each RGB pixel of the stego image.

Step-4: Continue the process until the message is fully extracted from stego image.

Step-5: Decompress the extracted secret data.

Step-6: Using shared key, decrypt secret information to get original information.

Step-7: Reconstruct the secret information.

Comparing cover image with stego image needs an image quality measure. Commonly used measures are MSE, PSNR, Correlation and Histogram.

MSE In statistics, MSE is quantifying the difference between values implied by an estimator and the true values of the quantity being estimated. MSE measures the average of the squares of the "errors.

PSNR scales the MSE according to the image range. The PSNR between cover image and stego image. A higher PSNR indicates that the quality of the stego image is similar to the cover image.

Correlation, a best-known method, not only evaluates the degree of closeness between two functions but also determines the extent to which the cover image and the stego image are close to each other even after embedding data.

The MSE, PSNR and Correlation values for various image file formats. Quality Metrics for various Image File Formats Cover Image Name Stego Image Name MSE PSNR (dB) Correlation Cover_Leena.bmp Stego_Leena.bmp 3.6791
46.19 0.9984 Cover_Barbara.bmp Stego_Barbara.bmp 4.0184 49.42 0.9986 Cover_Baboon.gif Stego_Baboon.gif 2.1883 44.72 0.9993 Cover_House.gif Stego_House.gif 3.8426 47.49.0. The results show that MSE, PSNR and correlation are in comparison with already existing results. The MSE obtained for different file formats varies from 2.6 to 4.0 and PSNR varies from 42.15 to 49.42 dB and 99.98% of correlation is obtained with the proposed technique.

According to Steganography, the secret message which is hidden may result in a distortion less image. At the same time this distortion will be perceptible to the naked eye. The quantity of information invisibly hidden in the image resulting in a distortion less image plays a pivotal role and this is decided by algorithm. The required characteristics are assessed while chosing a specific file format for Steganography.

In the process of Steganography, the message which is hidden is invisible. An attempt has been made to implement encryption and decryption techniques on the data to be hidden into the carrier files, so that this will provide additional security to the data. The sender and receiver only know how to hide and 61 unhide the data into the carrier files. No other intermediate person will even know that there is a second the carrier file. The sender and receiver only know the commands to hide and unhide. Since BMP utilizes lossless compression, LSB makes utilization of BMP image.

To have the capacity to conceal secret information within a BMP image, one requires a substantial cover medium. The advantage of LSB hiding is its simplicity. LSB embedding technique also allows high perceptual transparency. The data hiding capacity of LSB technique is high and more secure. Embedding secret information with Steganography technique decreases the probability of secret information being detected. LSB insertion method to image Steganography works effectively for 24 BMP, GIF and PNG image file formats. Using this embedding and extracting algorithms, one can extract the secret message exactly as original message without changing the cover image.

Hiding information into a medium requires following elements

1. The cover medium(C) that will hold the secrat message.
2. The secret message (M), may be plain text, digital image file or any type of data.
3. The stegonographic techniques
4. A stego-key (K) may be used to hide and unhide the message.

In modern approach, depending on the cover medium, steganography can be divided into five types:

1. Text Steganography
2. Image Steganography
3. Audio Steganography
4. Video Steganography
5. Protocol Steganography

• **Text steganography** Hiding information in text file is the most common method of steganography. The method was to hide a secret message into a text message. After coming of Internet and different type of digital file formats it has decreased in importance. Text stenography using digital files is not used very often because the text files have a very small amount of excess data.

• **Image steganography** Images are used as the popular cover medium for steganography. A message is embedded in a digital image using an embedding algorithm, using the secret key. The resulting stego-image is send to the receiver. On the other side, it is processed by the extraction algorithm Contents using the same key. During the transmission of stego- image unauthenticated persons can only notice the transmission of an image but cant see the existence of the hidden message.

• **Audio steganography** Audio steganography is concerned with embedding information in an innocuous cover speech in a secure and robust manner.

Communication andtransmission security and robustness are essential for transmitting vital information to intended sources while denying access to unauthorized persons. An audible, sound can be inaudible in the presence of another louder audible sound

.This property allows to select the channel in which to hide information . Existing audio steganography software can embed messages in WAV and MP3 sound files.

The list of methods that are commonly used for audio steganography are listed and discussed below.

• LSB coding
• Parity coding
• Phase coding
• Spread spectrum
• Echo hiding
• Video steganography

Video Steganography is a technique to hide any kind of files in any extension into a carrrying Video file.

• Protocol steganography The term protocol steganography is to embedding information within network protocols such as TCP/IP. We hide information in the header of a TCP/IP packet in some fields that can be either optional or are never used.

Applications of

Steganography

(i)Secret

Communications

The use steganography does not advertise secret communication and therefore avoids scrutiny of the sender, message, Contents and recipient. A trade secret, blueprint, or other sensitive information can be transmitted without alerting potential attackers.

(ii) Feature Tagging Elements can be embedded inside an image, such as the names of individuals in a photo or locations in a map. Copying the stego-image also copies all of the embedded features and only parties who possess the decoding stego- key will be able to extract and view the features.

(iii) Copyright Protection Copy protection mechanisms that prevent data, usually digital data, from being copied.

IMAGE STEGNOGRAPHY TECHNIQUES

There are several Steganographic techniques for image file format which are as follows:

Spatial Domain Technique:

There are many versions of spatial steganography, all directly change some bits in the image pixel values in hiding data.

Least significant bit (LSB) based steganography is one of the simplest techniques that hides a secret message in the LSBs of pixel values without perceptible distortions. To our human eye, changes in the value of the LSB are imperceptible. Least Significant Bit (LSB) replacement technique, Matrix embedding, are some of the spatial domain techniques. Advantages of spatial domain LSB technique are:

1. Degradation of the original image is not easy.
2. Hiding capacity is more i.e. more information can be stored in an image.

Disadvantages of LSB technique are:

i. Robustness is low

ii. Hidden data can be destroyed by simple attacks.

Masking and Filtering Masking and Filtering:

It is a steganography technique which can be used on gray-scale images. Masking and filtering is similar to placing watermarks on a printed image. These techniques embed the information in the more significant areas than just hiding it into the noise level. Watermarking techniques can be applied without the fear of image destruction due to lossy compression as they are more integrated into the image.

Advantages of Masking and filtering Techniques: This method is much more robust than LSB replacement with respect to compression. Disadvantages: Techniques can be applied only to gray scale images and restricted to 24 bits.

Transform Domain Technique:

The Frequency domain the message is inserted into transformed coefficients of image giving more information hiding capacity and more robustness against attacks. Transform domain embedding can be termed as a domain of embedding techniques for which a number of algorithms have been suggested . Most of the strong steganographic systems today operate within the transform domain Transform domain techniques have an advantage over LSB techniques as they hide information in areas of the image that are less exposed to compression, cropping, and image processing.

Some transform domain techniques do not seem dependent on the image format and they may outrun lossless and lossy format conversions. Transform domain techniques are of different types[3]:

1. Discrete Fourier transformation technique (DFT).
2. Discrete cosine transformation technique (DCT).
3. Discrete Wavelet transformation technique (DWT).

Distortion Techniques:

In this technique, store information by signal distortion and measure the deviation from the original cover in the decoding process. Distortion techniques need knowledge of the original cover image during the decoding process where the decoder functions to check for differences between the original cover image and the distorted cover image in order to restore the secret message. In this technique, a stego-image is created by applying a sequence of modifications to the cover image.

This sequence of modifications is use to match the secret message required to transmit. The message is encoded at pseudo-randomly chosen pixels. If the stego image is different from the cover image at the given message pixel, the message bit is a 1. Otherwise, the message bit is a 0. The encoder can modify the 1 value pixels in such a manner that the statistical properties of the image are not affected. If an attacker interfere with the stego-image by cropping, scaling or rotating, the receiver can easily detect it.

Characteristics feature of Data Hiding Techniques:

Perceptibility does embedding message distort cover medium to a visually unacceptable level.

Capacity how much information can be hidden with relative to the change in perceptibility.

Robustness to attacks can embedded data exist manipulation of the stego medium in an effort to destroy, or change the embedded data.

Tamper Resistance Beyond robustness to destruction, tamper-resistance refers to the difficulty for an attacker to alter a message once it has been embedded in a stego- image.

Image Steganalysis :

Steganalysis is the breaking of steganography and is the science of detecting hidden information . The main objective of steganalysis is to break steganography and the detection of stego image. Almost all steganalysis algorithms depend on steganographic algorithms introducing statistical differences between cover and stego image. Steganalysis are of three different types:

Visual attacks it discovered the hidden information, which helps to separate the image into bit planes for further more analysis.

Statistical attack:

Statistical attacks may be passive or active. Passive attacks involves with identifying presence or absence of a secret message or embedding algorithm used. Active attacks is used to investigate embedded message length or hidden message location or secret key used in embedding.

Structural attacks The format of the data files changes as the data to be hidden is embedded, identifying this characteristic structure changes can help us to find the presence of image/text file.

Steganalytic tools :

There are several steganalytic tools available in market like PhotoTitle, 2Mosaic and StirMark Benchmark etc. These three steganalytic tools can remove steganographic content from any image. This is achieved by destroying secret message by two techniques: break apart and resample.

Least-Significant Bit (LSB) Technique

The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an image is changed to a bit of the secret message. Digital images are mainly of two types

(i) 24 bit images and

(ii) 8 bit images.

In 24 bit images we can embed three bits of information in each pixel, one in each LSB position of the three eight bit values. Increasing or decreasing the value by changing the LSB does not change the appearance of the image; much so the resultant stego image looks almost same as the cover image.

Cover and Hidden Images:

The hidden image is extracted from the stego-image by applying the reverse process. If the LSB of the pixel value of cover image C(i,j) is equal to the message bit m of secret massage to be embedded, C(i,j) remain unchanged; if not, set the LSB of C(i, j) to m. The message embedding procedure is given below

S(i,j) = C(i,j) - 1, if LSB(C(i,j)) = 1 and m = 0

S(i.j) = C(i,j), if LSB(C(i,j)) = m

S(i,j) = C(i,j) + 1, if LSB(C(i,j)) = 0 and m = 1 where LSB(C(i, j)) stands for the LSB of cover image C(i,j) and m is the next message bit to be embedded. S(i,j) is the stego image As we already know each pixel is made up of three bytes consisting of either a 1 or a 0. For example, suppose one can hide a message in three pixels of an image (24-bit colors). Suppose the original 3 pixels are:

(11101010 11101000 11001011)

(01100110 11001010 11101000)

(11001001 00100101 11101001)

A steganographic program could hide the letter ”J” which has a position 74 into ASCII character set and have a binary representation ”01001010”, by altering the channel bits of pixels.

(11101010 11101001 11001010)

(01100110 11001011 11101000)

(11001001 00100100 11101001)

In this case, only four bits needed to be changed to insert the character successfully. The resulting changes that are made to the least significant bits are too small to be recognised by the human eye, so the message is effectively hidden. The advantage of LSB embedding is its simplicity and many techniques use these methods

. LSB embedding also allows high perceptual transparency.

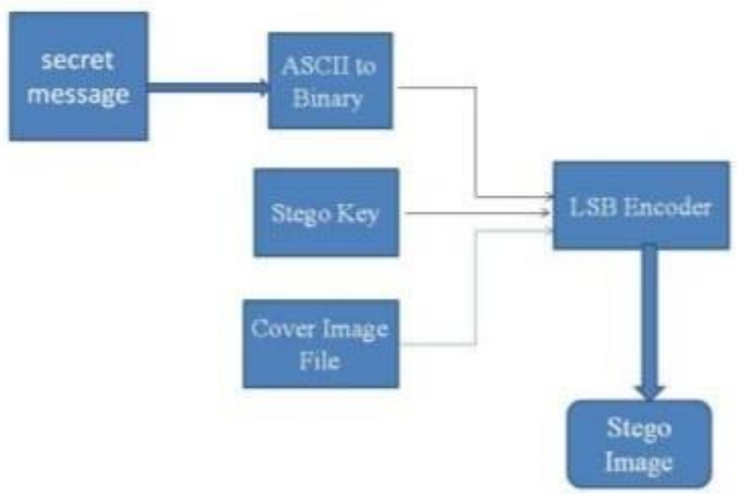


Fig3.2 LSB insertion technique

Data Embedding :

The embedding process is as follows.

Inputs Cover image, stego-key and the

text file Output stego image Procedure

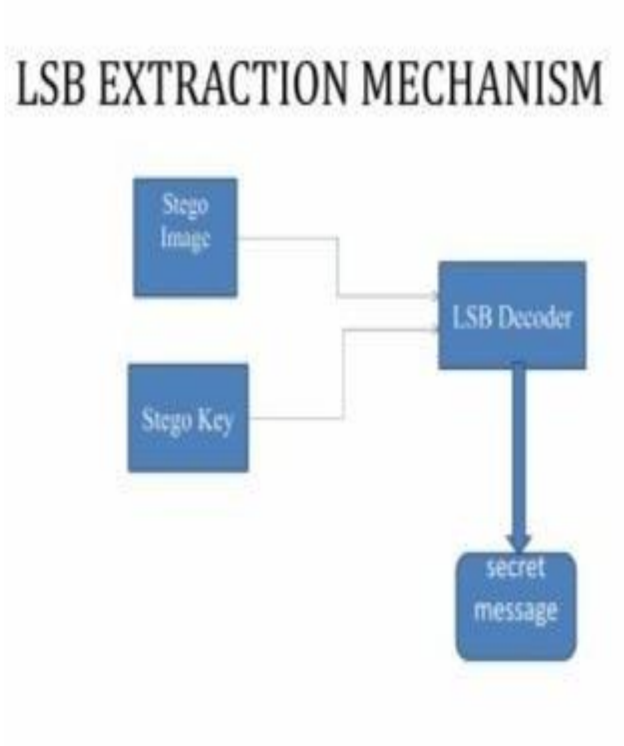


Fig3.3 LSB extraction

mechanism Step 1: Extract the pixels of the cover image.

Step 2: Extract the characters of the text file. Step 3: Extract the characters from the Stego key.

Step 4: Choose first pixel and pick characters of the Stego key and place it in first component of pixel.

Step 5: Place some terminating symbol to indicate end of the key. 0 has been used as a terminating symbol in this algorithm.

Step 6: Insert characters of text file in each first component of next pixels by replacing it.

Step 7: Repeat step 6 till all the characters has been embedded.

Step 8: Again place some terminating symbol to indicate end of data. Step 9: Obtained stego image.

Data Extraction

The extraction process is as follows. Inputs Stego-image file, stego-key Output Secret text message. Procedure:

Step 1: Extract the pixels of the stego image.

Step 2: Now, start from first pixel and extract stego key characters from first component of the pixels. Follow

Step3: up to terminating symbol, otherwise follow step 4.

Step 4: If this extracted key matches with the key entered by the receiver, then follow Step 5, otherwise terminate the program.

Step 5: If the key is correct, then go to next pixels and extract secret message characters from first component of next pixels. Follow Step 5 till up to terminating symbol, otherwise follow step 6.

Step 6: Extract secret message.

Image Encoding Algorithm

Inputs Image file, stego key and image file Output Stego image.

1. The cover and secret images are read and converted into the unit8 type.
2. The numbers in secret image matrix are conveyed to 8-bit binary. Then the matrix is reshaped to a new matrix a.
3. The matrix of the cover image is also reshaped to matrix b
4. Perform the LSB technique described above
5. The stego-image, which is very similar to the original cover image, is achieved by reshaping matrix b.
6. While extracting the data, the LSB of the stego image is collected and they are reconstructed into the decimal numbers. The decimal numbers are reshaped to the secret image.

Pseudo-Random Encoding Technique

In this technique, A random key is used to choose the pixels randomly and embed the message. This will make the message bits more difficult to find and hopefully reduce the realization of patterns. Data can be hidden in the LSB of a particular colour plane (Red plane) of the randomly selected pixel in the RGB colour space. This will make the message bits more difficult to find and hopefully reduce the realization of patterns.

Embedding Algorithm

In this process of encoding method, a random key is used to randomised the cover image and then hide the bits of a secret message into the least significant bit of the pixels within a cover image. The transmitting and receiving end share the stego key and random-key. The random-key is usually used to seed a pseudo-random number generator to select pixel locations in an image for embedding the secret message.

Inputs Cover image, stego-key and the message Output stego image

1) Read character from text file that is to be hidden and convert the ASCII value of the character into equivalent binary value into an 8 bit integer array.

2) Read the RGB colour image(cover image) into which the message is to be embedded.

3) Read the last bit of red pixel.

4) Initialize the random key and Randomly permute the pixels of cover image and reshape into a matrix.

5) Initialize the stego-key and XOR with text file to be hidden and give message. 6) Insert the bits of the secret message to the LSB of the Red plane's pixels. 7) Write the above pixel to Stego Image File.

Extraction of Hidden Message:

In this process of extraction, the process first takes the key and then random- key. These keys takes out the points of the LSB where the secret message is randomly distributed . Decoding process searches the hidden bits of a secret message into the least significant bit of the pixels within a cover image using the random key. In decoding algorithm the random-key must match i.e. the random-key which was used in encoding should match because the random key sets the hiding points of the message in case of encoding. Then receiver can extract the embedded messages exactly using only the stego-key.

Message extraction algorithm:

Inputs Stego-image file, stego-key, random key. Output Secret message.

1) Open the Stego image file in read mode and from the Image file, read the RGB colour of each pixel.

2) Extract the red component of the host image.

3) Read the last bit of each pixel.

4) Initialize the random-key that gives the position of the message bits in the red pixel that are embedded randomnly.

5) For decoding, select the pixels and Extract the LSB value of red pixels.

6) Read each of pixels then content of the array converts into decimal value that is actually ASCII value of hidden character.

7) ASCII values got from above is XOR with stego-key and gives message file, which we hide inside the cover image.

GRAY SCALE

The following chart displays all 256 Gray-scale colours. The gray-scale colour naming scheme uses a two digit hex value to define up to 256 shades of gray. In photography and computing, a gray scale digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest.

Gray scale images are distinct from one-bit bi-tonal black-and-white images, which in the context of computer imaging are images with only the two colours, black (also called bi-level or binary images). Gray scale images have many shades of gray in between. Gray scale images are often the result of measuring the intensity of light at each pixel in a single band of the electromagnetic spectrum (e.g. infrared, visible light, ultraviolet, etc.), and in such cases they are monochromatic proper when only a given frequency is captured. Gray scale Shading Strengths (0=no colour; 15=full colour.

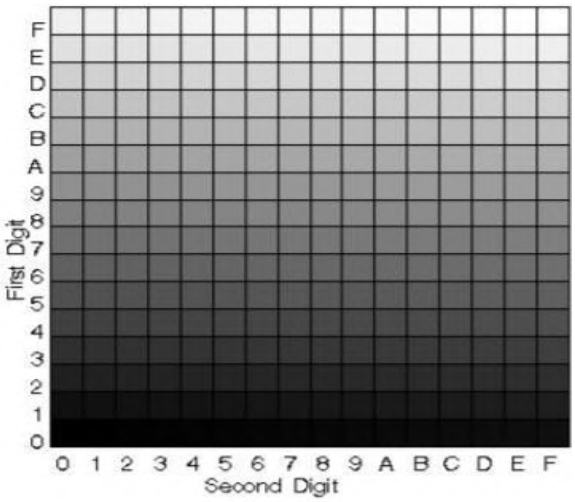


Figure 1: Gray Scale color

A message is embedded into the image by the stego system encoder via a secret key or password. This password or secret key should be kept secret. The resulting stego image is transmitted over a channel to the receiver. The stego system at the decoder end, using the same key or password, will decode the stego image.

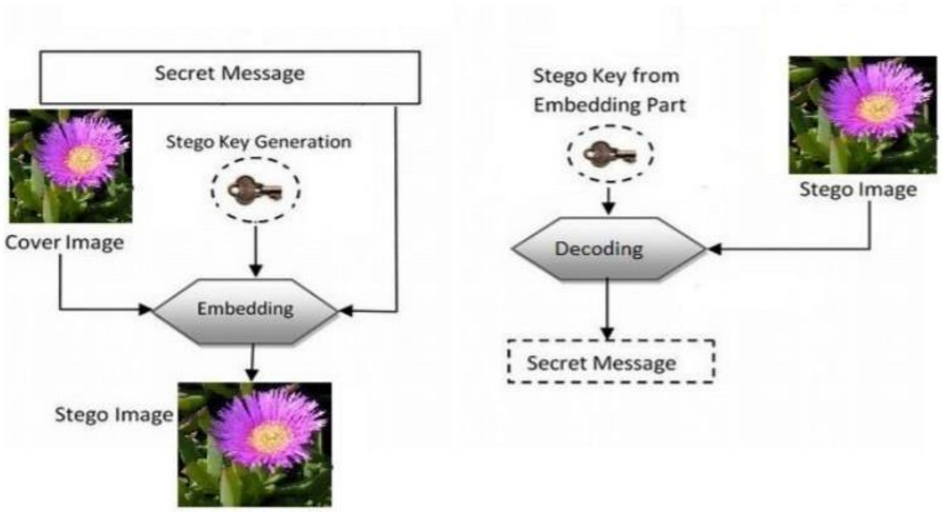


Fig 3.4 stegnography block diagram

There are two important components, cover image and hiding data, in data hiding technique. The cover image I is an 8-bit gray scale image. The size of cover image is m×n. The hiding data H embedded in I is g-bits bit stream. We use the equation below to express image C, data D and each pixel separately.I= { } H= One of the simplest systems for embedding digital data into a digital cover is the Least Significant Bit method.

Consider an N×M image in which each pixel value is represented by a decimal number in the range determined by the number of bits used. In a gray-scale image, with 8 bit precision per pixel, each pixel assumes a value between [0, 255] and each positive number P can be represented by: P= + = This property allows the decomposition of an image into a collection of binary images by separating the into n bit planes.

In the classical LSB embedding methods, the secret message is inserted into the least-significant bit plane of the cover image either by directly replacing those bits. The amount of data to be embedded may also be fixed or variable in size depending on the number of pixels selected. The main advantage of such a technique is that the modification of the LSB plane does not affect the human perception of the overall image quality as the amplitude variation of the pixel values is bounded by ±1. The masking properties of the Human Visual System allow significant amounts of embedded information to be unnoticed by imperceptible by the average observer under normal viewing conditions. “Masking” refers to the phenomenon where a signal can be imperceptible to an observer in the presence of another signal. A detailed review of these techniques is given in. Other advantages of LSB data hiding included high embedding capacity and low computational complexity. The main disadvantages are the weaknesses with respect to robustness, tampering, geometric attacks, filtering, and compression.

In LSB steganography, the least significant bits of the cover media’s digital data are used to conceal the message. The simplest of the LSB steganography techniques is LSB replacement. LSB replacement steganography changes the last bit of each of the pixel values to reflect the message that needs to be hidden. Consider an 8- bit gray scale bitmap image where each pixel is stored as a byte representing a gray scale color value. Suppose the first eight pixels of the original image have the following gray color values:

01010010

01001010

10010111

11001100

11010101

01010111

00100110

01000011

To hide the letter Z whose binary value of ASCII [11] code is 10110101, we would replace the LSBs of these pixels to have the following new values:

01010011

01001010

10010111

11001101

Note that, on average, only half the LSBs need to change. The difference between the cover (i.e. original) image and the stego image will be hardly noticeable to the human eye. However, one of its major limitations is small size of data which can be embedded in such type of images using only LSB. LSB is extremely vulnerable to attacks. LSB techniques implemented to 24 bit formats for the color image are difficult to detect contrary to 8 bit format.

LEAST SIGNIFICANT BIT TECHNIQUE

Least significant bit (LSB) insertion is a common and simple approach to embed information in an image file. In this method the LSB of a byte is replaced with an M"s bit. This technique works good for image steganography. To the human eye the stego image will look identical to the carrier image. For hiding information inside the images, the LSB (Least Significant Byte) method is usually used. To a computer an image file is simply a file that shows different colors and intensities of light on different areas of an image.

The best type of image file to hide information inside is a 24 Bit BMP (Bitmap) image. When an image is of high quality and resolution it is a easier to hide information inside image. Although 24 Bit images are best for hiding information due to their size. Some people may choose 8 Bit BMP"s or possibly another image format such as GIF. The reason being is that posting of large images on the internet may arouse suspicion. The least significant bit i.e. the eighth bit is used to change to a bit of the secret message. When using a 24-bit image, one can store 3 bits in each pixel by changing a bit of each of the red, green and blue color components.

Suppose that we have three adjacent pixels (9 bytes) with the RGB encoding

[3] [3]

10010101

00001101

11001001

10010110

00001111

11001011

10011111

00010000

When the number 300, can be which binary representation is 100101100 embedded into the least significant bits of this part of the image. If we overlay these 9 bits over the LSB of the 9 bytes above we get the following (where bits in bold have been changed) 10010101 00001100 11001000
10010111 00001110
11001011 10011111
00010000 11001010

Here the number 300 was embedded into the grid, only the 5 bits needed to be changed according to the embedded message. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size.

1. Select a cover image of size M*N as an input.
2. The message to be hidden is embedded in RGB component only of an image.
3. Use a pixel selection filter to obtain the best areas to hide information in the cover image to obtain a better rate. The filter is applied to Least Significant Bit (LSB) of every pixel to hide information, leaving most significant bits (MSB).
4. After that Message is hidden using Bit Replacement method.

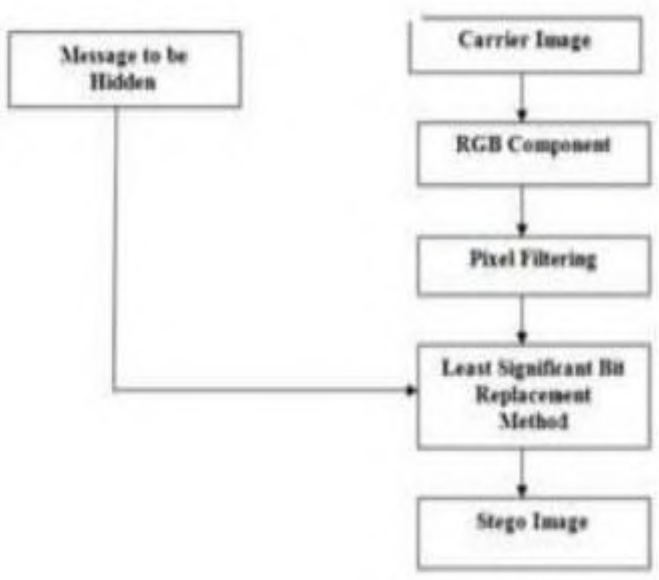


Fig 3.5 Algorithm of Least Significant Bit

CHAPTER-4

SOFTWARE REQUIREMENTS

Introduction to MATLAB

What is MATLAB?

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modelling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building.

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for *matrix laboratory*. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called *toolboxes*. Very important to most users of MATLAB, toolboxes allow you to *learn* and *apply* specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

LOCAL ENVIRONMENT SET UP :

Setting up MATLAB environment is a matter of few clicks. The installer can be downloaded from http://in.mathworks.com/downloads/web_downloads: MathWorks provides the licensed product, a trial version and a student version as well. You need to log into the site and wait a little for their approval. After downloading the installer the software can be installed through few clicks.

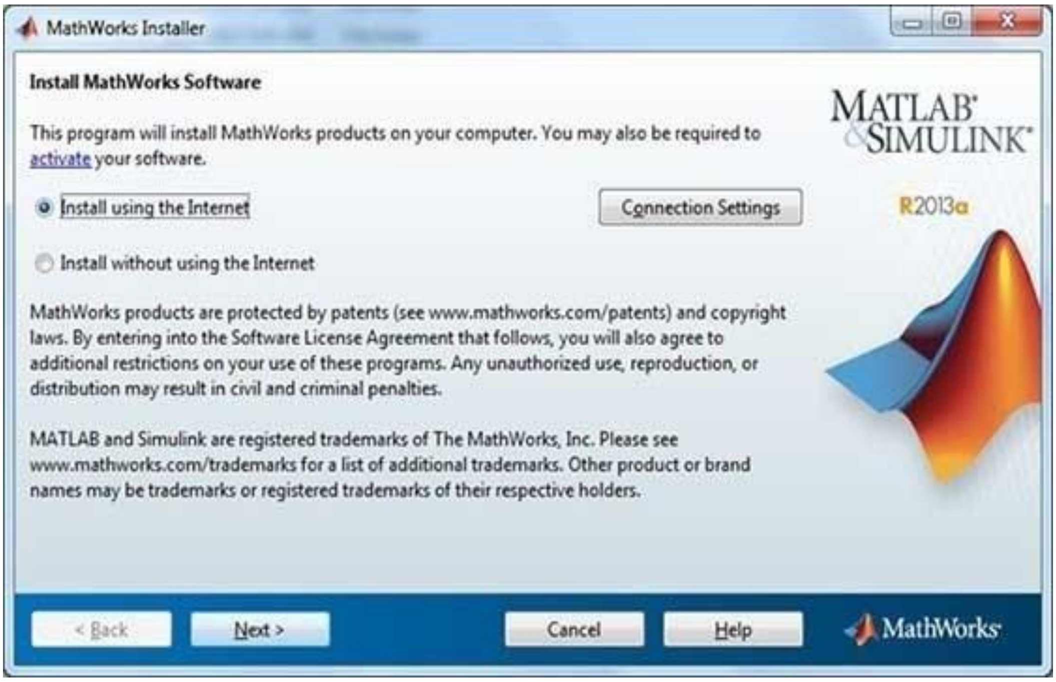


Figure 4.1: local environment setup

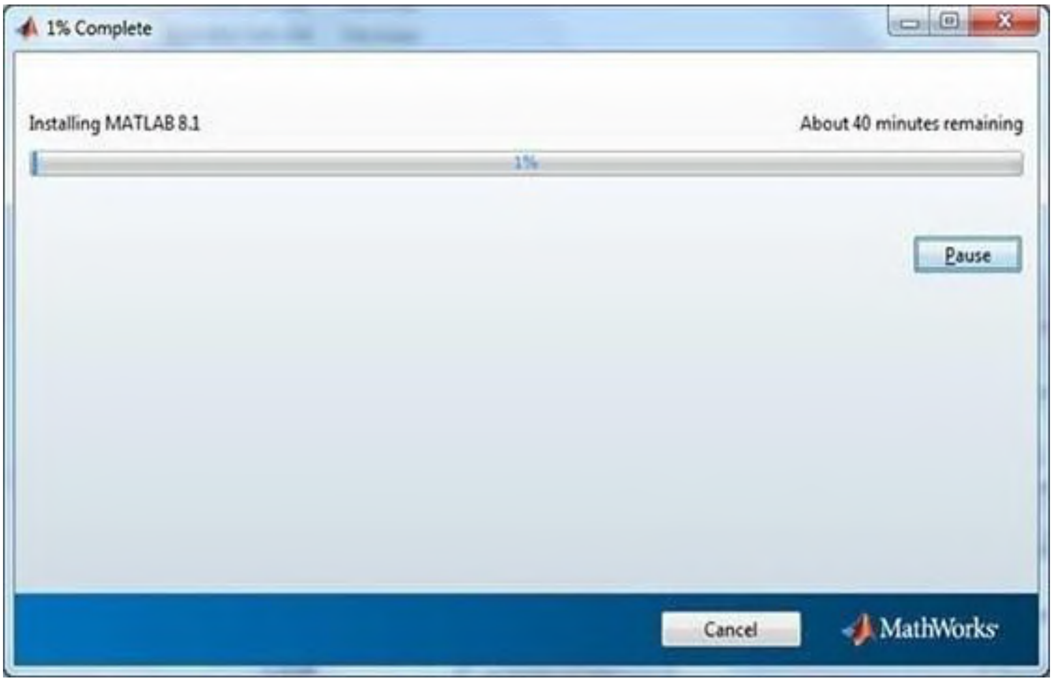


Figure4.2: Installing matlab software

THE MATLAB SYSTEM :

The MATLAB system consists of five main parts:

Development Environment:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

MATLAB Mathematical Function:

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigen values, Bessel functions, and fast Fourier transforms.

MATLAB Language:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

Graphics:

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

MATLAB Application Program Interface (API):

This is a library that allows you to write C and Fortran programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

MATLAB WORKING ENVIRONMENT:

Matlab desktop :

Matlab Desktop is the main Matlab application window. The desktop contains five sub windows, the command window, the workspace browser, the current directory window, the command history window, and one or more figure windows, which are shown only when the user displays a graphic.

The command window is where the user types MATLAB commands and expressions at the prompt (>>) and where the output of those commands is displayed. MATLAB defines the workspace as the set of variables that the use session. The workspace browser shows these variables and some information about them. Double clicking on a variable in the workspace browser launches the Array Editor, which can be used to obtain information and income instances edit certain properties of the variable.

The current Directory tab above the workspace tab shows the contents of the current directory, whose path is shown in the current directory window. For example, in the windows operating system the path might be as follows: C:\MATLAB\Work, indicating that directory “work” is a subdirectory of the main directory “MATLAB”; WHICH IS INSTALLED IN DRIVE C. clicking on the arrow in the current directory window shows a list of recently used paths. Clicking on the button to the right of the window allows the user to change the current directory.

MATLAB uses a search path to find M-files and other MATLAB related files, which are organize in directories in the computer file system. Any file run in MATLAB must reside in the current directory or in a directory that is on search path. By default, the files supplied with MATLAB and math works toolboxes are included in the search path. The easiest way to see which directories are on the search path. The easiest way to see which directories are soon the search path, or to add or modify a search path, is to select set path from the File menu the desktop, and then use the set path dialog box. It is good practice to add any commonly used directories to the search path to avoid repeatedly having the change the current directory.

The Command History Window contains a record of the commands a user has entered in the command window, including both current and previous MATLAB sessions. Previously entered MATLAB commands can be selected and re-executed from the command history window by right clicking on a command or sequence of commands. This action launches a menu from which to select various options in addition to executing the commands. This is useful to select various options in addition to executing the commands. This is a useful feature when experimenting with various commands in a work session. The Command History Window contains a record of the commands a user.

Using the MATLAB Editor to create M-Files:

The MATLAB editor is both a text editor specialized for creating M-files and a graphical MATLAB debugger. The editor can appear in a window by itself, or it can be a sub window in the desktop. M-files are denoted by the extension .m, as in pixel up.m. The MATLAB editor window has numerous pull-down menus for tasks such as saving, viewing, and debugging files. Because it performs some simple checks and also uses color to differentiate between various elements of code, this text editor is recommended as the tool of choice for writing and editing M-functions. To open the editor , type edit at the prompt opens the M-file filename.m in an editor window, ready for editing. As noted earlier, the file must be in the current directory, or in a directory in the search path.

Getting Help:

The principal way to get help online is to use the MATLAB help browser, opened as a separate window either by clicking on the question mark symbol on the desktop toolbar, or by typing help browser at the prompt in the command window. The help Browser is a web browser integrated into the MATLAB desktop that displays a Hypertext Markup Language(HTML) documents. The Help Browser consists of two panes, the help navigator pane, used to find information, and the display pane, used to view the information. Self-explanatory tabs other than navigator pane are used to perform a search.

HISTORY:

Origins

MATLAB was invented by mathematician and computer programmer Cleve Moler. The idea for MATLAB was based on his 1960s PhD thesis. Moler became a math professor at the University of New Mexico and started developing MATLAB for his students as a hobby. He developed MATLAB's initial linear algebra programming in with his one-time thesis advisor , George Forsythe . This was followed by Fortan code for linear equations in 1971.

The first early version of MATLAB was completed in the late 1970s. The software was disclosed to the public for the first time in February 1979 at the Naval Postgraduate school in California. Early versions of MATLAB were simple matrix calculators with 71 pre-built functions. At the time, MATLAB was distributed for free to universities. Moler would leave copies at the universities he visited and the software developed a strong following in the math departments of university campuses.

In the 1980s, Cleve Moler met John N. Little. They decided to reprogram MATLAB in C and market it for the IBM desktops that were replacing mainframe computers at the time. John Little and programmer Steve Bangert re-programmed MATLAB in C, created the MATLAB programming language , and developed features for toolboxes.

Commercial Development

MATLAB was first released as a commercial product in 1984 at the Automatic Control Conference in Las Vegas. MathWorks, Inc. was founded to develop the software and the MATLAB programming language was released. The first MATLAB sale was the following year, when Nick Trefethen from the Massachusetts Institute of Technology bought ten copies.

By the end of the 1980s, several hundred copies of MATLAB had been sold to universities for student use. The software was popularized largely thanks to toolboxes created by experts in various fields for performing specialized mathematical tasks. Many of the toolboxes were developed as a result of standard students that used MATLAB in academia, then brought the software with them to the private sector.

Over time, MATLAB was re-written for early operating systems created by Digital Equipment corporation, VAX, Sun Microsystems and for Unix PCs.Version 3 was released in 1987. The first MATLAB compiler was developed by Stephen c. Jhonson in the 1990s.

In 2000, MathWorks added a Fortran-based library for linear algebra in MATLAB 6, replacing the software's original LINPACK and EISPACK subroutines that were in C. MATLAB's Parallel Computing Toolbox was released at the 2004 Supercomputing Conference and support for graphics processing units (GPUs) was added to it in 2010.

Recent history

Some especially large changes to the software were made with version 8 in 2012.The user interface was reworked:517 and simulink's functionality was expanded. By 2016, MATLAB had introduced several technical and user interface improvements, including the MATLAB Live Editor notebook, and other feature.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

4.1 **Table: Operators**

Syntax

The MATLAB application is built around the MATLAB programming language. Common usage of the MATLAB application involves using the "Command Window" as an interactive mathematical shell or executing text files containing MATLAB code.

Variables

Variables are defined using the assignment operator, =. MATLAB is a weakly function language because variables can be assigned without declaring their type, except if they are to be treated as symbolic objects, and that their type can change.

Values can come from constants from computation involving values of other variables, or from the output of a function.

Structures

MATLAB supports structure data types. Since all variables in MATLAB are arrays, a more adequate name is "structure array", where each element of the array has the same field names. In addition, MATLAB supports dynamic field names (field look-ups by name, field manipulations, etc.)

Functions

MATLAB supports elements of lambda function by introducing function handles, or function references, which are implemented either in .m files or anonymous nested functions.

Classes and object-oriented programming

MATLAB supports object oriented programming including classes, inheritance, virtual dispatch, packages, pass-by-value semantics, and pass-by-reference semantics. However, the syntax and calling conventions are significantly different from other languages. MATLAB has value classes and reference classes, depending on whether the class has *handle* as a super-class (for reference classes) or not (for value classes).

Method call behaviour is different between value and reference classes. For example, a call to a method:

object.method();

can alter any member of *object* only if *object* is an instance of a reference class, otherwise value class methods must return a new instance if it needs to modify the object.

Interfacing with other languages

MATLAB can call functions and subroutines written in the programming languages C or Fortron. A wrapper function is created allowing MATLAB data types to be passed and returned. Mex files (MATLAB executables) are the dynamically loadable object files created by compiling such functions. Since 2014 increasing two- way interfacing with was python being added.

Libraries written in perl, Java, ActiveX or .Net can be directly called from MATLAB, and many MATLAB libraries (for example XML or SQL support) are implemented as wrappers around Java or ActiveX libraries. Calling MATLAB from Java is more complicated, but can be done with a MATLAB toolbox which is sold separately by Mathworks, or using an undocumented mechanism called JMI (Java-to- MATLAB Interface), (which should not be confused with the unrelated Java meta interface that is also called JMI). Official MATLAB API for Java was added in 2016.

As alternatives to the MuPAD based Symbolic Math Toolbox available from MathWorks, MATLAB can be connected to Maple or Mathematica.

Libraries also exist to import and export MathML.

MATLAB Windows

The MATLAB **Command Window** is the main window where you type commands directly to the MATLAB interpreter. The MATLAB **Editor Window** is a simple text editor where you can load, edit and save complete MATLAB programs. The Editor window also has a menu command (Debug/Run) which allows you to submit the program to the command window. The MATLAB **Help Window** gives you access to a great deal of useful information about the MATLAB language and MATLAB computing environment. It also has a number of example programs and tutorials.

COMMAND WINDOW:

This is the main area where commands can be entered at the command line. It is indicated by the command prompt (>>).

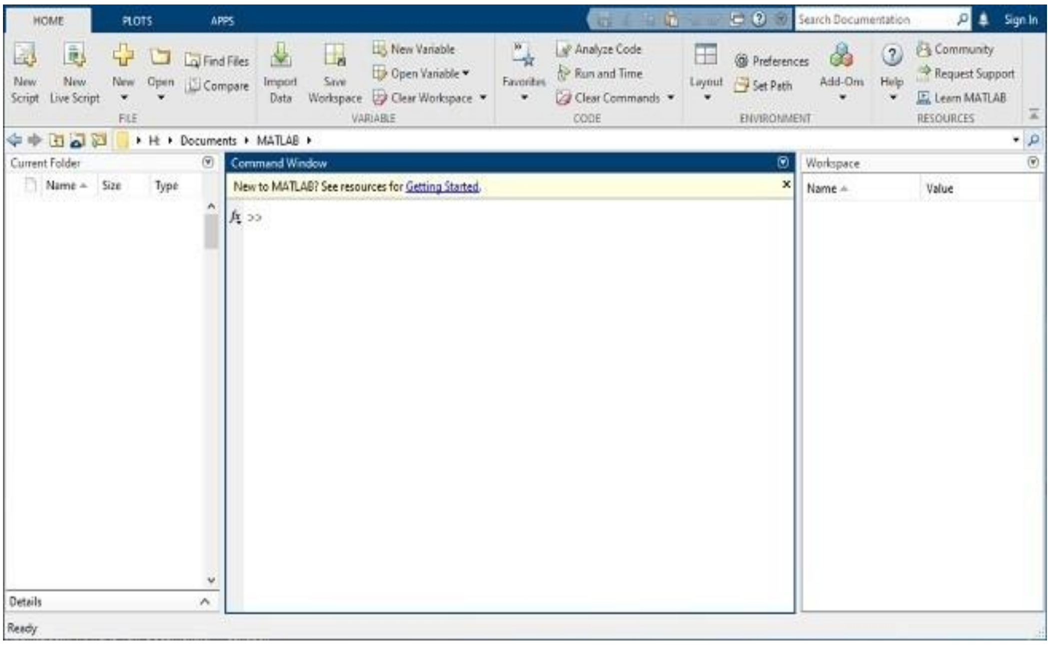


Figure 4.3: Command window

MATLAB Workspace

The workspace contains variables that you create or import into MATLAB from data files or other programs. You can view and edit the contents of the workspace in the Workspace browser or in the Command Window. For more information, see [Create and Edit Variables](#).

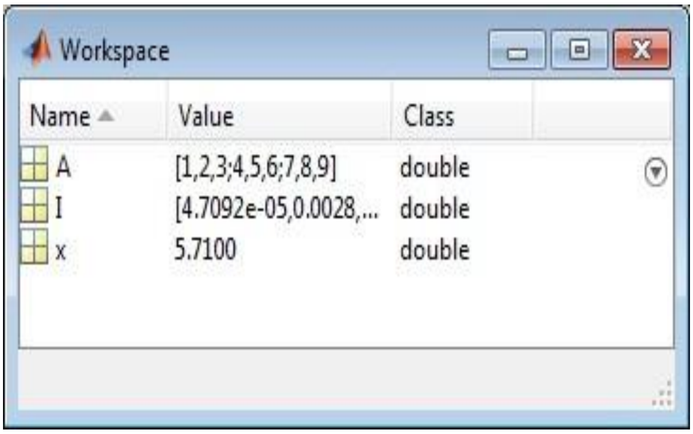


Figure 4.4: MATLAB workspace

Figure Window

Figure creates figure graphics objects. figure objects are the individual windows on the screen in which MATLAB displays graphical output.

`Figure ('PropertyName', PropertyValue,...)` creates a new figure object using the values of the properties specified. MATLAB uses default values for any properties that you do not explicitly define as arguments.

figure(h) does one of two things, depending on whether or not a figure with handle h
exists. but is an integer, figure(h) creates a figure, and assigns it the handle h. figure(h) where h is not the handle to a figure, and is not an integer, is an error.

h = figure(...) returns the handle to the figure object.

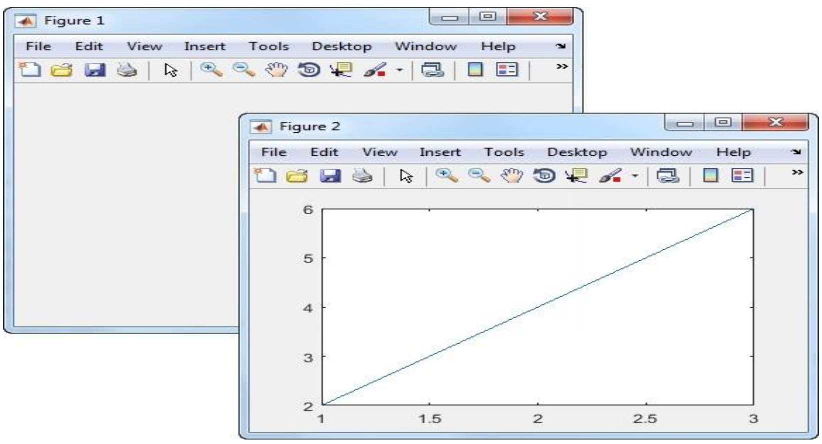


Figure 4.5: Matlab figure window

CURRENT FOLDER:

This panel allows you to access the project folders and files.

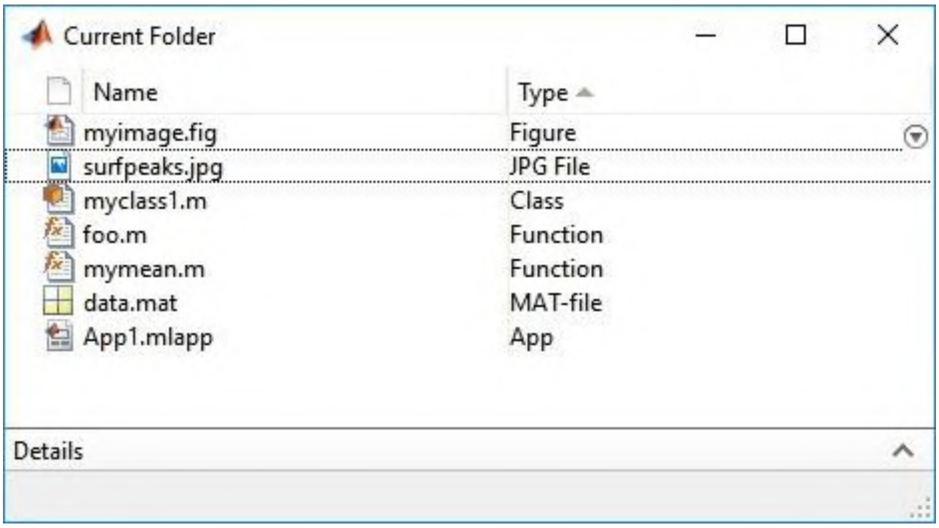


Figure4.6: Current folder

COMMAND HISTORY:

This panel shows or rerun commands that are entered at the command line.

Figure 4.7: Command history

FUNCTIONS:

Reading images

Images are read in MATLAB environment using the function ‘imread.’ Syntax of imread is: imread(‘filename’); where ‘filename’ is a string having the complete name of the image, including its extension. For example,

F=imread(‘Penguins_grey.jpg’);

Please note that when no path information is included in ‘filename,’ ‘imread’ reads the file from the current directory. When an image from another directory has to be read, the path of the image has to be specified.

Semicolon (;) at the end of a statement is used to suppress the output. If it is not included, MATLAB displays on the screen the result of the operation specified in that line.

‘>>’ indicates the beginning of a command line as it appears in the MATLAB command window. imread, imshow and imwrite functions in MATLAB are used to read images in MATLAB environment, display them on MATLAB desktop and write them to the current directory, respectively.

Figure 4.8: Grayscale image of penguins

In case of grayscale images, the resultant matrix of the imread statement comprises 256×256 or 65,536 elements. The first statement takes grey values of all the pixels in the grayscale image and puts them into a matrix F(256×256 elements), which is now a MATLAB variable on which various matrix operations can be performed.

Image display

Images are displayed on the MATLAB desktop using the function imshow, which has the basic syntax: imshow(f);

where ‘f’ is an image array of data type ‘uint8’ or double. Data type ‘uint8’ restricts the values of integers between 0 and 255.

It should be remembered that for a matrix of type ‘double,’ the imshow function expects the values to be between 0 and 1, where 0 is displayed as black and 1 as white., and a value less than zero is displayed as black. To get the values within this range, a division factor can be used. The larger the division factor, the darker will be the image.

As an example, if you give the command imshow(F), the image shown on the desktop is given in Fig. 5.3 ‘imshow(f, [low high])’ displays as black all values less than or equal to ‘low’ and as white all values equal to or greater than ‘high.’ The values in between are displayed as intermediate intensity values.

Figure 4.9: Image obtained with imshow(F) command

‘imshow(f, [ ])’ sets variable low to the minimum value of array ‘f’ and high to its maximum value. This helps in improving the contrast of images having a low dynamic range.

Multiple images can be displayed within one figure using the subplot function. This function has three parameters within the brackets, where the first two parameters specify the number of rows and columns to divide the figure. The third parameter specifies which subdivision to use. For example, subplot(3,2,3) tells MATLAB to divide the figure into three rows and two columns and set the third cell as active. To display the images Penguins_RGB.jpg and Penguins_grey.jpg together in a single figure, you need to give the following commands:

A=imread(‘Penguin_grayscal

e.jpg’);

B=imread(‘Penguin_RGB.jp

g’):

Figure

subplot(1,2,1),imsh

ow(A);

subplot(1,2,2),imshow(B);

MEDIAN FILTER:

2dimentional median filter is represented as

medfilt2 Syntax:

J=medfilt2(I)

J=medfilt2(I,[

m,n])

J=medfilt2(,

padopt)

Description:
J = medfilt2(I) performs median filtering of the image I in two dimensions. Each output pixel contains the median value in a 3-by-3 neighborhood around the corresponding pixel in the input image.

J = medfilt2(I,[m n]) performs median filtering, where each output pixel contains the median value in the m-by-n neighborhood around the corresponding pixel in the input image.

J = medfilt2(___,padopt) controls how medfilt2 pads the image boundaries.

UIGETFILE:

Uigetfile command is used to open file selection dialog box.

Syntax:

file =
uigetfile
[file,path] =
uigetfile
[file,path,indx] = uigetfile
___= uigetfile(filter)
___= uigetfilter(filter,title)
___= uigetfile(filter ,title, defname)
___= uigetfile(,'multiselect',mode)

Description ***:***

file = uigetfile opens a modal dialog box that lists files in the current folder. It enables a user to select or enter the name of a file. If the file exists and is

valid, uigetfile returns the file name when the user clicks **Open**. If the user clicks **Cancel** or the window close button (X), uigetfile returns 0.

[file,path] = uigetfile returns the file name and path to the file when the user clicks **Open**. If the user clicks **Cancel** or the window close button (X),
then uigetfile returns 0 for both of the output arguments.

[file,path,indx] = uigetfile returns the index of the filter selected in the dialog box when the user clicks **Open.**

____= uigetfile(filter) specifies a file extension by which files displayed in the dialog box are filtered. Use this syntax with any of the output argument combinations in the previous syntaxes.

Typically, only files with a matching file extension are displayed. On some platforms, uigetfile displays files that do not match the filter, but dims those file names. If the filter is missing or empty, uigetfile uses the default list of file types .

____= uigetfile(filter,title) specifies a dialog box title. To filter using the default file filter, but specify a custom title, use empty quotes for the filter value. For example:

file = uigetfile('','Select a File')

____= uigetfile(filter,title,defname) specifies a default file name for the **File name** field.

____= uigetfile(_,'MultiSelect',mode) specifies whether a user can select multiple files. Set the mode to 'on' to enable multifile selection. By default it is set to 'off'.

Windows® libraries can span multiple folders.

CORRELATION COEFFICIENT:

2Dimentional correlation coefficient is represented as corr2.

Syntax:

R = corr2(A, B)

Description:

R = corr2(A,B) returns the 2-D correlation coefficient R between arrays A and B.

MATRIX TO GRAYSCALE IMAGE:

2Dimensional matrix to grayscale command is represented as mat2gray. Syntax:

I=mat2gray(A,[amin, amax]) I=mat2gray(A)

Description:

`I = mat2gray(A,[amin amax])` converts the matrix `A` to a grayscale image `I` that contains values in the range 0 (black) to 1 (white). `amin` and `amax` are the values in `A` that correspond to 0 and 1 in `I`. Values less than `amin` are clipped to 0, and values greater than `amax` are clipped to 1.

`I = mat2gray(A)` sets the values of `amin` and `amax` to the minimum and maximum values in `A`.

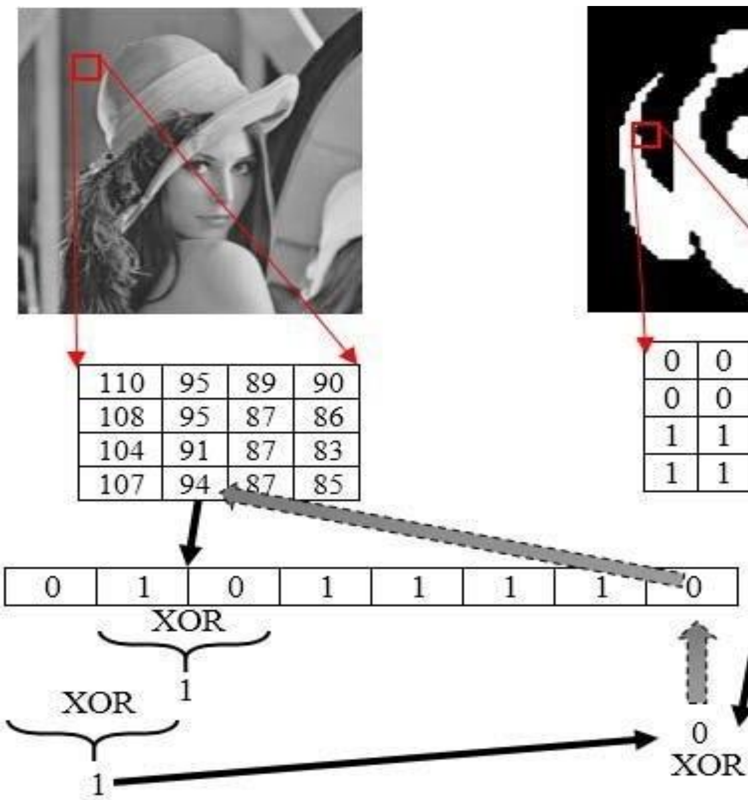
CHAPTER -5

DESIGN AND IMPLEMENTATION

In the method we propose, there are two main schemes, namely message embedding and message extraction. For more details can be seen in the sub-section below.

5.1. Embedding Scheme

In the embedding, scheme required input in the form of gray-scale cover image and image of the message in the form of a binary image with the exact same size. While the output obtained is a stego image. For more details can see the visualization in Figure below



.

Fig 5.1 embedded scheme

Here are the steps for the embedding process:

Step 1: Read the cover image (A) and message

image (B). Step 2: Change the pixel value to binary.

Step 3: Perform XOR operations on the 7th and on the 6th bit.

Step 4: Perform XOR operation on bit 8th with XOR operation result on the 7th and on the 6th bits

Step 5: Perform XOR operations on message bits with three MSB bits (8th, 7th, and 6th bits).

Step 6: Save the XOR operation result in the message bit, then convert again to uint8, the result of this conversion being the stego image pixel value.

Extraction Scheme

In the extraction scheme only required input in the form of stego image. While the output of the recovered message in the form of a binary image.

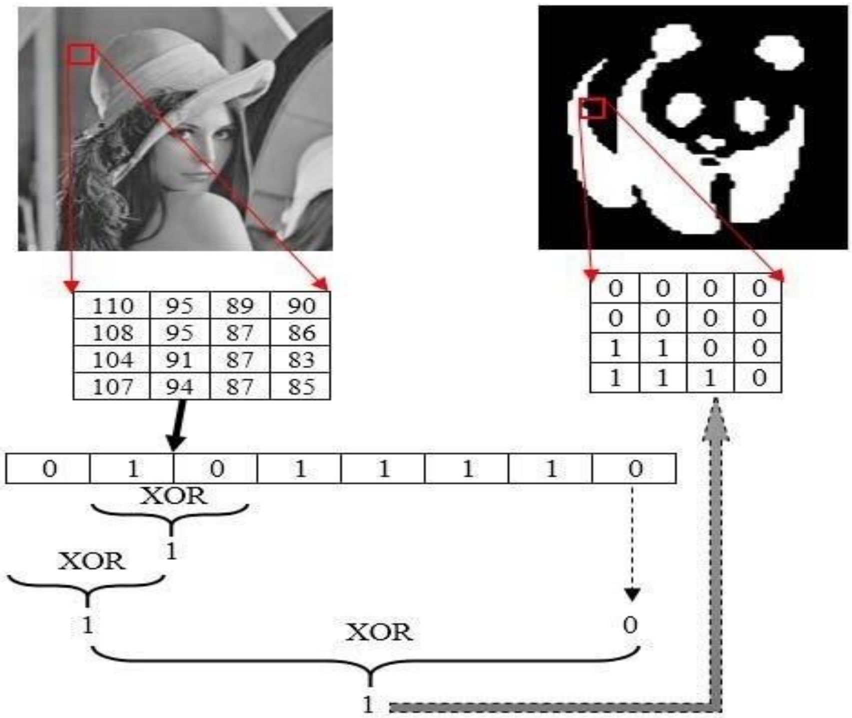


Fig 5.2 Extraction Scheme

Figure 2 shows the visualization of the extraction process, in detail, the steps in the extraction process are as follows:

Step 1: Read the stego image (S).

Step 2: Change the pixel value to binary.

Step 3: Perform XOR operations on the 7th and 6th bits.

Step 4: Perform XOR operation on the 8th bit with XOR operation result on the 7th and on the 6th bit.

Step 5: do the XOR operation on the LSB with three bits MSB (bits 8th, 7th, and 6th). Step 6: Save and collect the results of the XOR operation on the LSB, then convert again to uint8, the result of this conversion being a recovery .

FLOW CHART

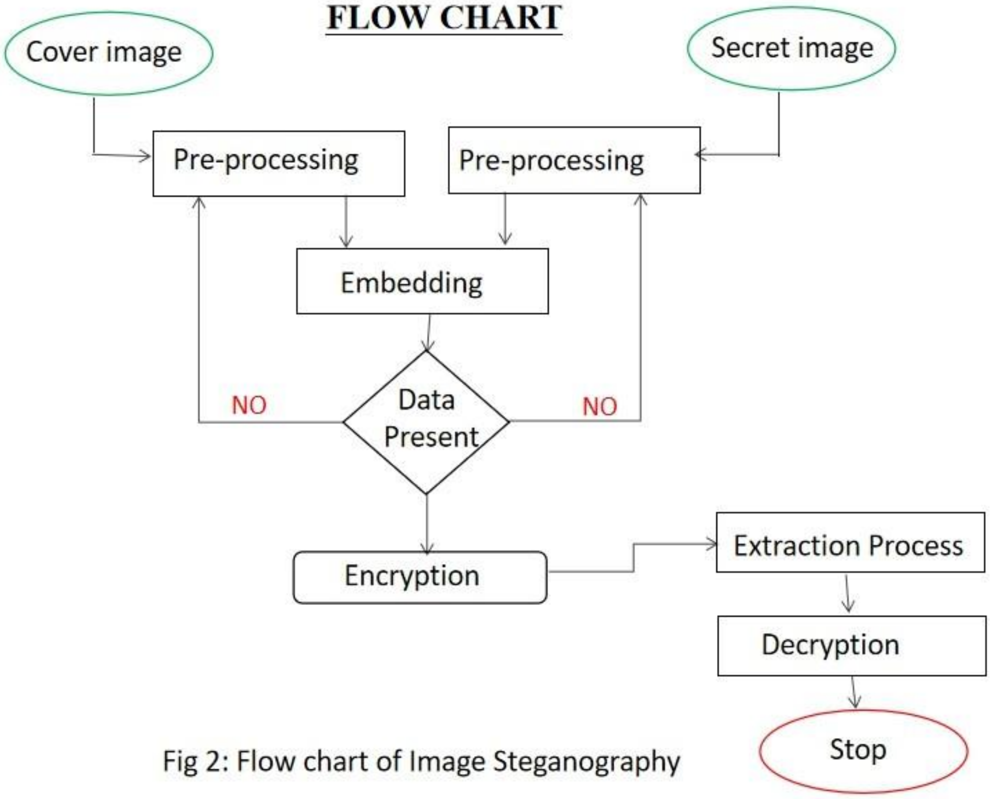


Fig 2: Flow chart of Image Steganography

Applications

1) In Military applications.

2) To communicate between two international governments.

3) Database managements.

CHAPTER-6
EXPERIMENT AND RESULTS

Experiments conducted in this research using two images, one as the cover image and another image as a secret image. The cover image and message image have the same size, 256 * 256 in size. Here is a cover image used in this study:

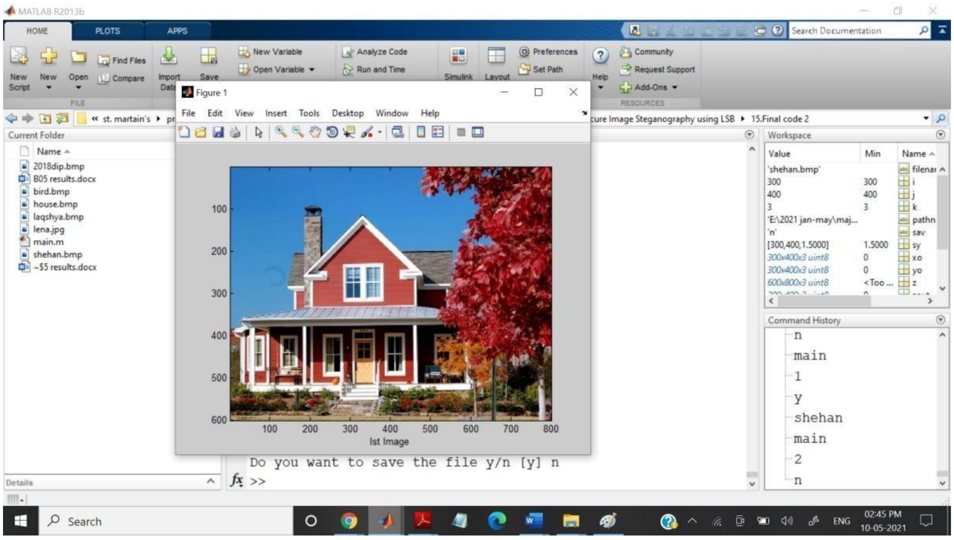


Fig 6.1: Cover Image used in the project

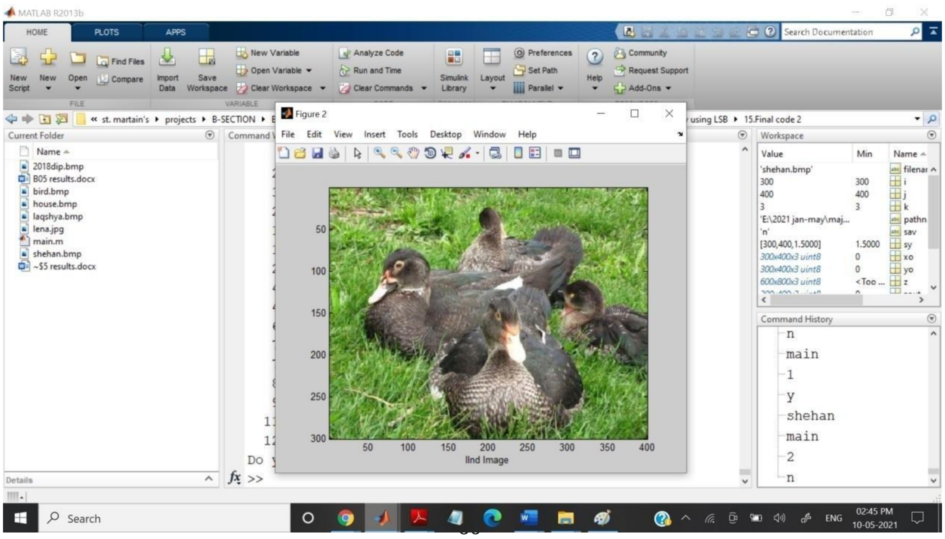


Fig 6.2: Secret Image used in the project

cover image and the message image have same pixel size. Then one pixel of message image embedded on one cover image pixel. The number of message bits embedded in the cover image is 65536 bits. Furthermore, an experiment of embedding the message using the method proposed above.

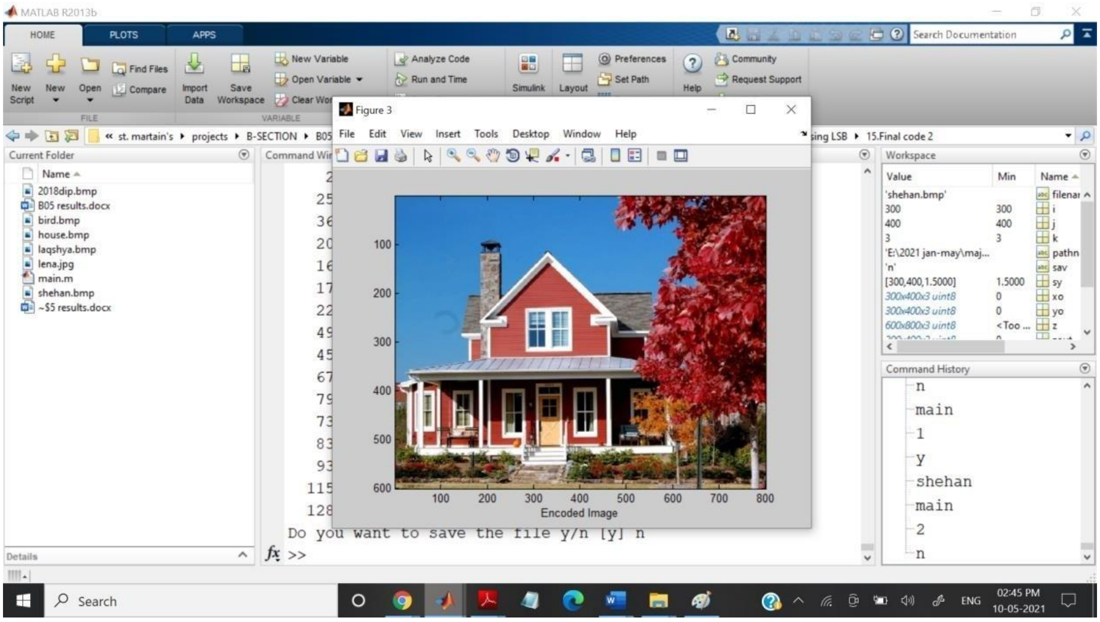


Fig 6.3: Embedded Image

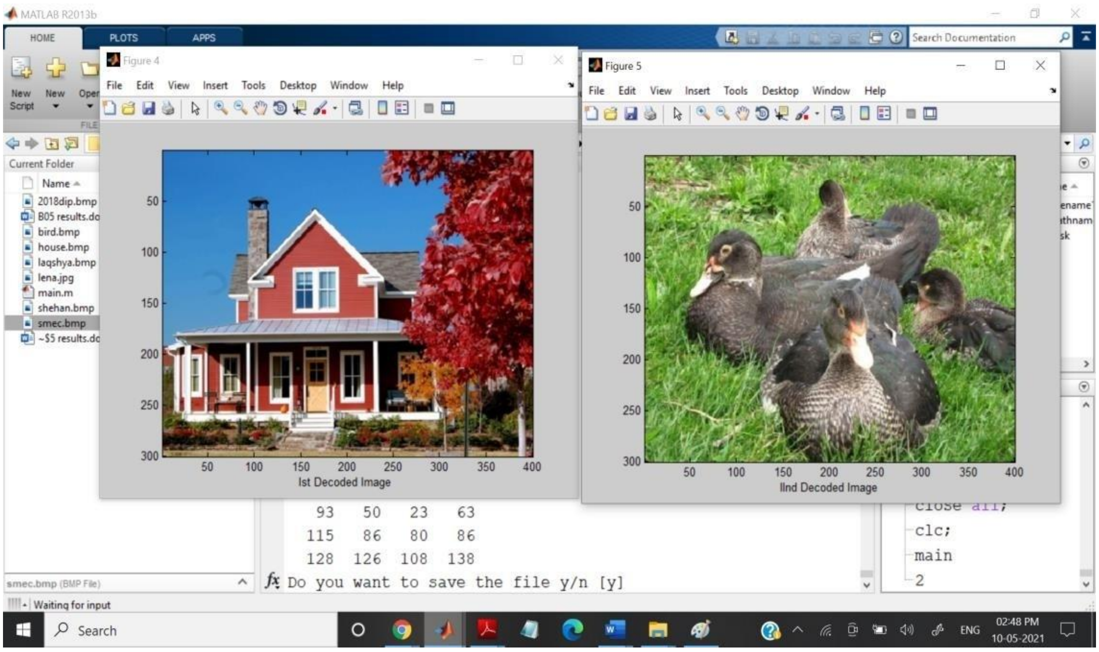


Fig 6.4: Extracted Image(Decoded Image)

In plain view, humans cannot distinguish the difference between the stego image and the cover image. Then, does this mean the stego image is good? Obviously needed a tool to measure the quality of stego image above. In this study used PSNR, MSE and image histogram to measure stego image quality. The measurement is done by comparing the cover image with stego image. To calculate MSE used Formula 1 and to calculate PSNR used Formula

$$MSE = \sum_{h=1}^{H-1}\sum_{g=1}^{G-1}\left\|A_f(h,g) - S_f(h,g)\right\| \quad (1)$$

$$PSNR = 10log_{10}\left(\frac{256-1}{MSE}\right) \quad (2)$$

Furthermore, histogram measurement is done by using the 'imhist' function in MATLAB.

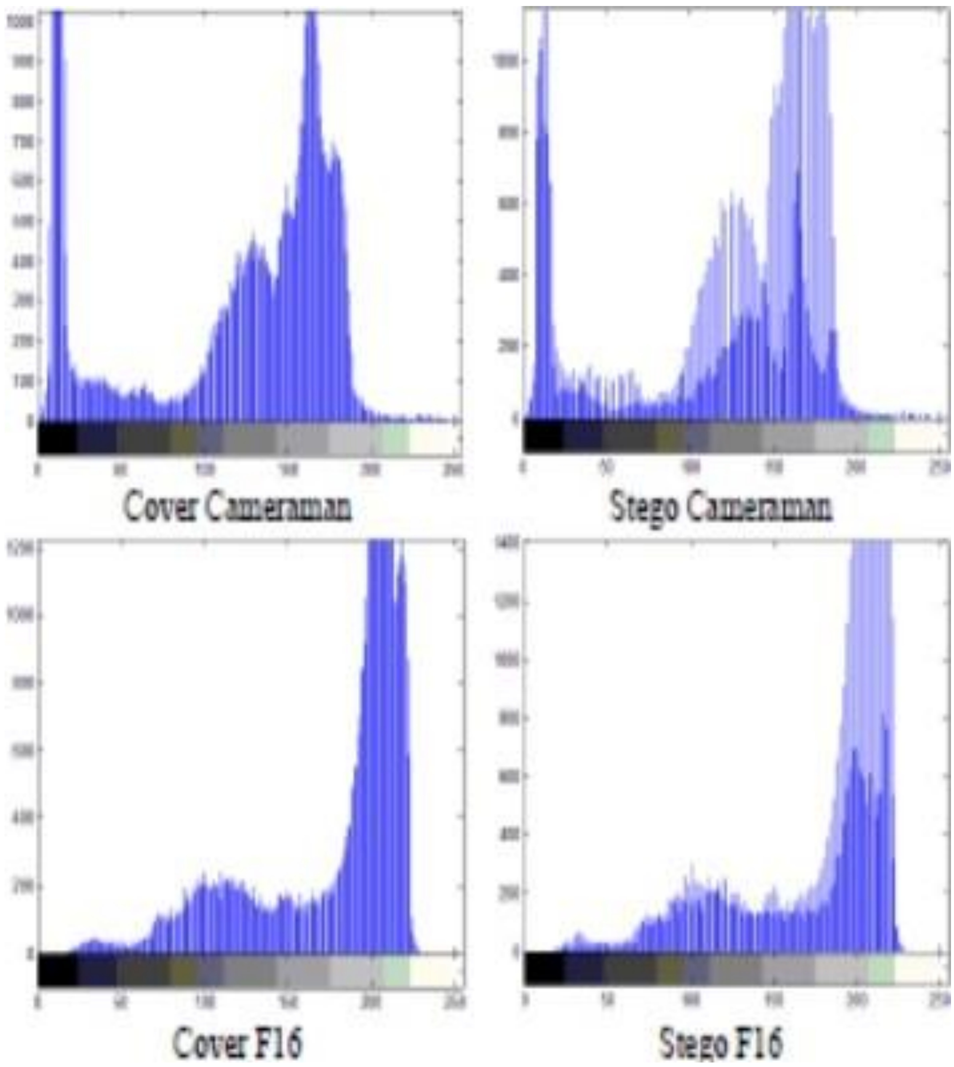


Fig. 6.5: The histogram of the cover image (left) and the stego image (right)

If Figure 6.5 observed carefully there is a difference between histogram image cover and stego image. The histogram on the stego image has a more split pattern than the cover image. As for the measurement of PSNR and

MSE values measurements are shown in Table .

TABLE I. PSNR AND MSE RESULTS

Image	PSNR	MSE
Lena	54.616	0.225
Barbara	54.037	0.257
cameraman	54.665	0.222
F16	56.035	0.162
Pentagon	53.714	0.277
peppers	54.307	0.241

To know the performance of extraction results. Recover message image is also measured using correlation coefficient (CC). A CC value equal to 1 indicates that the extraction processcan be done perfectly. If the value of cc between 0 to 1 then there are bits of messages that can not be extracted properly. To compute CC used Formula 3.

$$cc = \frac{\sum_h \sum_g (I_o - \bar{I}_o)(I_r - \bar{I}_r)}{\sqrt{\sum_h \sum_g (I_o - \bar{I}_o) \sum_h \sum_g (I_r - \bar{I}_r)}} \qquad (3)$$

CHAPTER 7
CONCLUSION AND FUTURE ENHANCEMENT

Conclusion

The method proposed in this study has an advantage in the aspect of imperceptibility as evidenced by the excellent value of PSNR and MSE. Where all PSNR values are more than 50dB, so does the MSE value not more than 0.3. This method is also very simple and safe because with XOR operation steganography process can be done quickly and easily. With the XOR operator, the embedded bits cannot be directly guessed. Moreover, there are three keys used, with three times the XOR operation. The use of an integrated key in the cover image also keeps the stego file the same size, and no key delivery is required to the receiver so it can speed up the messaging process as the file size is maintained. However, based on histogram analysis there is a distinct pattern difference between the cover image and stego image.

Future Enhancement

This model can be further implemented to send multiple objects at a time and can also be used to send videos and document files.

CODE

```
% Simple and Secure Image Steganography using LSB
% and Triple XOR Operation on MSB

clc; % clear the command window
clear; % clear the workspace
% close all

disp   ('       ');

disp (' ***** WELCOME TO IMAGE HIDER *****');

disp   ('       ');

disp  (' ******* Enter ur choice **********');



task =input('****---Encode :- 1 \n****---Decode :- 2\n Enter your task:');

% select task
if isempty  (task)
task=  1;
end
if task ==  1
% reads two image files

[filename1,pathname]=uigetfile('*.jpg','select cover the image');
x=imread(num2str(filename1))



[filename1,pathname]=uigetfile('*.jpg','select hiding the image');
y=imread(num2str(filename1))
```

```
%checkcompatibility
sx =  size  (x);
sy =   size  (y);

x= imresize  (x, [2*sy(1),2*sy(2)]);

%
% Applying shifting
x1  =  bitand  (x,uint8(252));
y1 =  bitshift  (y,-4);
y1_=   bitand   (y1,12);
y1_=  bitshift  (y1_,-2);
y1 =  bitand  (y1,3);
y_lsb1  =  bitshift  (bitand(y,12),-2);
y_lsb2 =  bitand  (y,3);
z= x1;
for j=1:sy(2)
for i=1:sy(1)
for k=1:3

z (i ,j ,k) = bitor (x1(i,j,k), y1_(i,j,k));
z (i+sy(1) ,j+sy(2),k) = bitor (x1(i+sy(1) ,j+sy(2),k), y1(i,j,k));
z (i+sy(1) ,j,k) = bitor (x1(i+sy(1),j ,k), y_lsb1(i,j,k));
z (i,j+sy(2) ,k) = bitor (x1(i,j+sy(2) ,k), y_lsb2(i,j,k));
end
end
end
z=z
% the  first  image
figure(1)
image (x);
xlabel(' Ist Image ');
%  IInd image
figure(2);
```

```
image (y);
xlabel(' IInd Image ');
%  encoded image
figure(3);
image (z);
xlabel(' Encoded Image ');
% saving image file
sav= input('Do you want to save the file y/n [y] ','s');
if isempty(sav)
sav='y';
end
if sav == 'y'
name= input('Enter a name for the encoded image: ','s');
if isempty (sav)
name= 'encoded_temp';
end
name=[name,'.bmp']; % concatination
imwrite (z, name,'bmp');
end
else
% Decoding encoded image
clear;
[filename1,pathname]= uigetfile('*.bmp','select cover the image');
z=imread (num2str(filename1));
sy = size(z)/2; % take the size of input file

% shifting
xo= bitand (z,uint8(252));
xo= imresize (xo,[sy(1),sy(2)]) % reduce the resolution to half so

%that it becoms the original image's resolution
for j=1:sy(2) % y variation

for i=1:sy(1) % x variation
```

```
for k=1:3
 zout1(i,j,k) =  bitshift (bitand(z(i,j,k),uint8(3)),2);

 zout2(i,j,k) =  bitand (z(i+sy(1),j+sy(2),k), uint8(3));

 zout3(i,j,k) =  bitshift (bitand(z(i+sy(1),j,k),uint8(3)),2);

 zout4(i,j,k) =  bitand  (z(i,j+sy(2),k),uint8(3));

end
end
end
 zout = bitshift ((zout1+zout2),4)+zout3+zout4;

 yo =  zout;

 % display Ist & IInd image from encoded image

 figure(4);

 image(xo);

 xlabel (' encryption Image ');

 figure (5);

image (yo);

 xlabel (' Decoded Image');

% saving file
 sav= input ('Do you want to save the file y/n [y] ','s');
```

```
 if isempty ( sav)

 sav='y';
 end
if sav == 'y'
name1 = input('Enter a name for the first image: ','s');

name2 = input('Enter a name for the second image: ','s');

if isempty(name1)

 name1 = 'Ist_temp';

 end
 if isempty(name2)

 name2 = 'IInd_temp';
 end
 name1 = [name1, '.bmp']

 name2 = [name2, '.bmp']

 imwrite(xo,name1, 'bmp')

 imwrite(yo,name2, 'bmp')

end
end
```

REFERENCES

[1] W. S. Sari, E. H. Rachmawanto, D. R. I. M. Setiadi and h. A. Sari, "A Good Performance OTP Encryption Image based on DCT-DWT Steganography," TELKOMNIKA (Telecommunication Computing Electronics and Control), vol. 15, no. 4, pp. 1987-1995, 2017.

[2] R. D. Ardy, O. R. Indriani, C. A. Sari, D. R. I. M. Setiadi and E. H. Rachmawanto, "Digital Image Signature using Triple Protection Cryptosystem (RSA, Vigenere, and MD5)," in International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICONSONICS), Yogyakarta, 2017.

[3] A. Winarno, D. R. I. M. Setiadi, A. A. Arrasyid, C. A. Sari and E. H. Rachmawanto, "Image Watermarking using Low Wavelet Subband based on 8×8 Sub-block DCT," in International Seminar on Application for Technology of Information and Communication (iSemantic), Semarang, 2017.

[4] G. Ardiansyah, C. A. Sari, D. R. I. M. Setiadi and E. H. Rachmawanto, "Hybrid Method using 3-DES, DWT and LSB for Secure Image Steganography Algorithm," in International Conference on Information Technology, Information System, and Electrical Engineering
(ICITISEE), Yogyakarta, 2017.

[5] A. Setyono, D. R. I. M. Setiadi and Muljono, "StegoCrypt Method using Wavelet Transform and One-Time Pad for Secret Image Delivery," in International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), Semarang, 2017.

[6] A. U. Islam, F. Khalid, M. Shah, Z. Khan, T. Mahmood, A. Khan, U. Ali and M. Naeem, "An Improved Image Steganography Technique based on MSB using Bit Differencing," in International Conference on Innovative Computing Technology (INTECH), Dublin, 2016.

[7] E. H. Rachmawanto, R. S. Amin, D. R. I. M. Setiadi and C. A. Sari, "A Performance Analysis StegoCrypt Algorithm based on LSB-AES 128 bit in Various Image Size," in International Seminar on Application for Technology of Information and Communication (ISemantic), Semarang, 2017.

[8] E. J. Kusuma, O. R. Indriani, C. A. Sari, E. H. Rachmawanto and D. R. I. M. Setiadi, "An Imperceptible LSB Image Hiding on Edge Region Using DES Encryption," in International Conference on Innovative and Creative (ICITech), Salatiga, 2017.
[9] K. Joshi, P. Dhankhar and R. Yadav, "A New Image Steganography Method in Spatial Domain Using XOR," in Annual IEEE India Conference (INDICON), New Delhi, 2015.
[10] P. S. Sapra and H. Mittal, "Secured LSB Modification using Dual Randomness," in International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, 2016.
[11] C. Irawan, D. R. I. M. Setiadi, C. A. Sari and E. H. Rachmawanto, "Hiding and Securing Message on Edge Areas of Image using LSB Steganography and OTP Encryption," in International Conference on Informatics and Computational Sciences (ICICoS), Semarang, 2017.
[12] K. Joshi and R. Yadav, "New Approach Toward Data Hiding using XOR for Image Steganography," in International Conference on Contemporary Computing (IC3), Noida, 2016

A

PROJECT REPORT

On

IOT BASED ANTENNA POSITIONING SYSTEM

Submitted by

1) Ms. A.S.S. Yasaswini (17K81A0461)

2) Mr. N. Sujeeth rao (17K81A0499)

3) Ms. D. Jyothi (18K85A0418)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATIONS ENGINEERING

Under The Guidance of

Ms. E. Parvathi

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **A.S.S. YASASWINI** WITH ROLL NO.**17K81A0461, D. JYOTHI** WITH ROLL NO.**18K85A0418**, **N.SUJEETH RAO** WITH ROLL NO.**17K81A0499**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**IOT BASED ANTENNA POSITIONING SYSTEM**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the project entitled "IOT Based antenna positioning system'', is being submitted by **Ms. A.S.S. Yasaswini (17K81A0461), Mr. N. Sujeeth rao (17K81A0499), Ms. D. Jyothi (18K85A0418)** in partial fulfilment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Head of the Department

Ms. E.Parvathi	Dr. B. Hari Krishna
Department of ECE	Department of ECE

Internal Examiner	External Examiner

Place:

Date:

DECLARATION

We, the student of **Bachelor of Technology** in Department of Electronics and Communication Engineering, session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled IOT Based antenna positioning system is theoutcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

A.S.S. Yasaswini (17K81A0461)

N. Sujeeth rao (17K81A0499)

D. Jyothi(18K851A0418)

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to our College Management for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr. P. Santosh Kumar Patra**, our beloved **Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Aboveall, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr. B. Hari Krishna**, **Head & Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, andthe vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our **dept. Project coordinator Dr. A. Anand** for his involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time to come.

We sincerely thank our project internal guide, **Ms. E.parvathi, Assistant professor**, department of ECE and our class project coordinator, **Mr. K.Karthik, Assistant professor**, department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level Project Guide **Dr. Maalyadri**, **Professor in IT** and **Dr. R. Santhosh Kumar**, **Associate Professor in CSE** advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

1. A.S.S. Yasaswini
2. N. Sujeeth rao
3. D. Jyothi

ABSTRACT

Wireless communication systems rely on antennas for signal reception. Proper positioning of antennas is necessary according to satellites/transmitters to achieve effective wireless communication. Manual antenna changes are often risky in a way that usually results in unexpected accidents and can lead to death.

So here, we propose an IOT based antenna positioning system that allows for remotely positioning of antennas based over IOT. To determine its position, the sensors are mounted on the antenna, and the motors change their position. Here, sensor-based system with motor on each antenna using antenna to check its facing direction that is transmitted and its direction will be changed by motors using IOT. If the direction of a satellite or transmitting station changes over time, the antenna direction must also be changed accordingly.

The receiving antennas may be placed far apart from each other across the globe. So, our system allows for antenna positioning over very long distances. The antenna positions are visible over internet to controlling operator on the IOT GUI. We here use IOT Gecko to develop the antenna monitoring GUI system. Our system allows for monitoring antenna direction as well as transmitting new coordinates to position the antenna and motor appropriately positions the antenna accordingly.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

Objective of the project

There are many commercial communications satellites in the geosynchronous orbit. Geostationary satellites are located in orbit directly above the equator and stay in the same place in the sky since they go around the earth at the same angular speed as that of the earth as it rotates. Satellite locations may thus be defined by longitude only. Geo Orbit position is the longitude position around the geostationary orbit. The most common issue anyone will come across when aligning the dish is aiming at the correct satellite for the broadcasts and receiving, they require. Satellite receivers do have details on them regarding the satellites to which they are communicating. They cannot determine whether you are aligned to the correct satellite and rely on the user to align the dish in the correct direction. The alignment of satellite dishes is currently performed manually wasting many hours to perform this process. Basically, for the manual, we use a spectrum analyser to monitor the signal during movement of the antenna until the desired satellite is found. Also, to look for the location of a satellite in order to achieve the best reception is an exhaustive process. Due to manual installation of the system, there is possibility of human errors i.e., misalignment of the antenna. Since each antenna is dedicated to a satellite, it is difficult to point it. So, it is necessary for an automated system in order to achieve the optimum positioning for reception of the signal.

IOT is one of the latest and emerging technology with the ability to transform the beautiful world like industries with smart machines, smarter cities etc. However, the importance of technology like IOT in communication systems has very great impact. In wireless communication concept antennas important for receiving signals. For very effective results of wireless communication the proper direction of antennas matters. So, we proposed a model for antenna positioning based on IOT technology. Sensors will be mounted on the antenna to detect the proper direction and the interesting thing is the motors using IOT concept will change the direction from anywhere across the world. The direction of transmitting stations changes with time means when the transmitting station changes over time the antenna direction need to be automatically changes accordingly. In simple words this model will help us in monitoring the direction of antennas and transmitting new coordinates to properly position the antennas. In IoT (Internet of things), “things” refer to a wide range of devices such as heart monitoring implants, remotely handling home appliances, biochip transponders used on farm animals, cameras that are streaming live feeds of wild animals in coastal waters etc.

Thus, we can say that “things” are a “mixture of hardware, software, data and services”. System engineering of a satellite-based data communication baseline concept is presented to achieve terabit per second throughput if sensors and actuators are augmented, the technology becomes as more general class of cyber physical systems which can also encompass the technologies.

Project overview

Embedded systems

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer. An embedded system is a dedicated computer system designed for one or two specific functions. This system is embedded as a part of a complete device system that includes, such as electrical and mechanical components [1].

The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes. The applications software on such processors is sometimes referred to as firmware Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory [4].

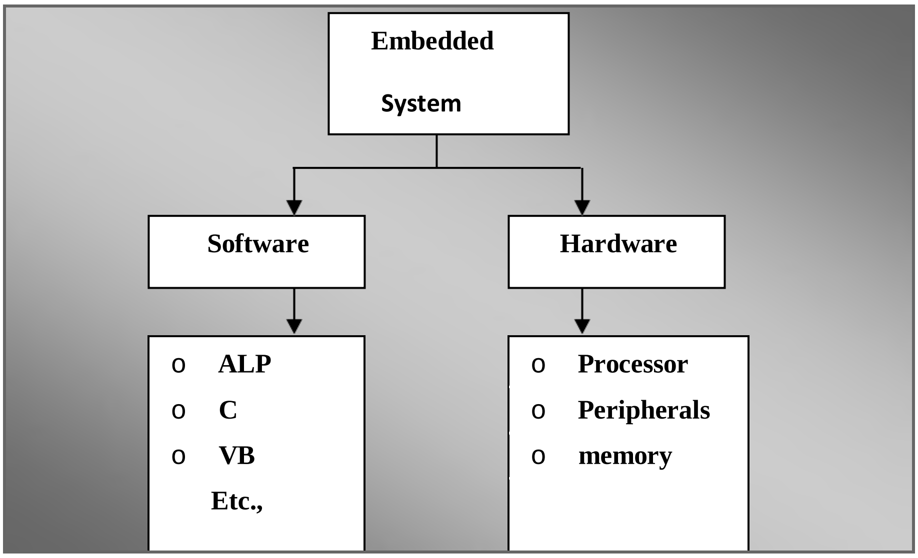


Figure 1.1: embedded system

Memory: It is used to store data or address.

Peripherals: These are the external devices connected

Processor: It is an IC which is used to perform some task

Applications of embedded systems

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

Micro Processor (µp):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor [6].

Three basic characteristics differentiate microprocessors:

- **Instruction set**: The set of instructions that the microprocessor can execute.
- **Bandwidth :** The number of bits processed in a single instruction.
- **Clock speed :** Given in megahertz (MHz), the clock speed determines how many instructions per second the processor can execute.

The ALU performs all arithmetic computations, such as addition, subtraction and logic operations (AND, OR, etc). It is controlled by the Control Unit and receives its data from the Register Array. The Register Array is a set of registers used for storing data. These registers can be accessed by the ALU very quickly. The Control Unit provides the timing and a control signal for getting data into and out of the registers and the ALU and it synchronizes the execution of instructions (we will deal with instruction execution at a later date).

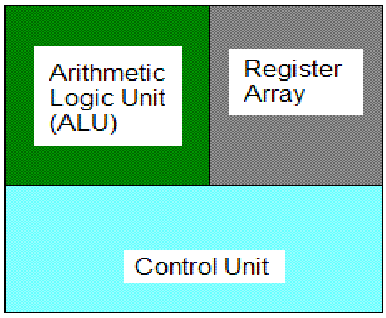


Figure 1.2: elements of microprocessor

Micro Controller (μc):

A microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.

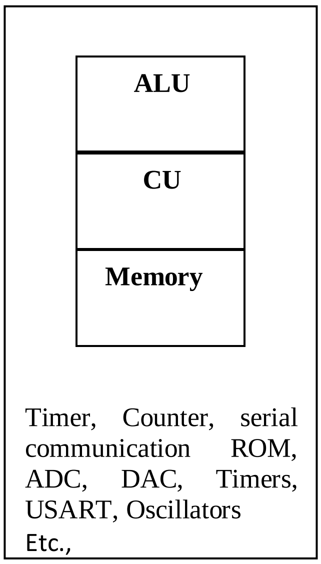


Figure 1.3: block diagram of micro controller

Application Specific Integrated Circuit (ASIC)

ASIC is a combination of digital and analog circuits packed into an IC to achieve the desired control/computation function. Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine [3].

ASIC typically contains

- CPU cores for computation and control.
- Peripherals to control timing critical functions.
- Memories to store data and program.
- Analog circuits to provide clocks and interface to the real world which is analog in nature.
- I/O's to connect to external components like LEDs, memories, monitors etc.

Computer Instruction Set

There are two different types of computer instruction set there are:

1. RISC (Reduced Instruction Set Computer) and
2. CISC (Complex Instruction Set computer)

Reduced Instruction Set Computer (RISC)

A RISC (reduced instruction set computer) is a microprocessor that is designed to perform a smaller number of types of computer instruction so that it can operate at a higher speed (perform more million instructions per second, or millions of instructions per second). Since each instruction type that a computer must perform requires additional transistors and circuitry, a larger list or set of computer instructions tends to make the microprocessor more complicated and slower in operation.

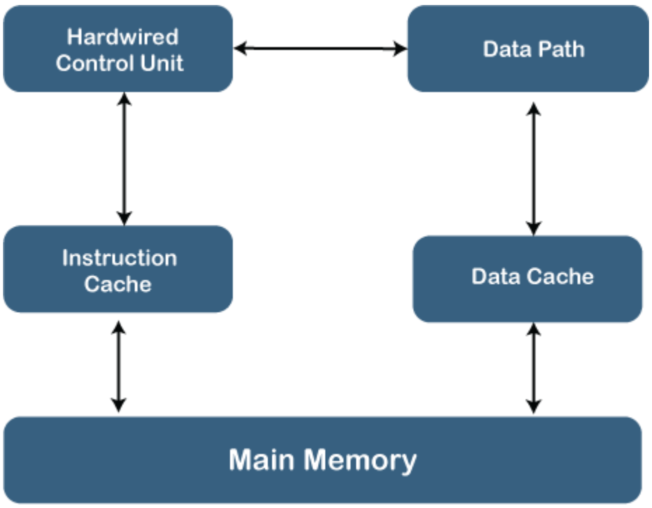


Figure 1.4(a): RISC architecture

RISC characteristics

- **Simple instruction set:** In a RISC machine, the instruction set contains simple, basic instructions, from which more complex instructions can be composed.
- **Same length instructions:** Each instruction is the same length, so that it may be fetched in a single operation.
- **1machine-cycleinstructions:** Most instructions complete in one machine cycle, which allows the processor to handle several instructions at the same time. This pipelining is a key technique used to speed up RISC machines.

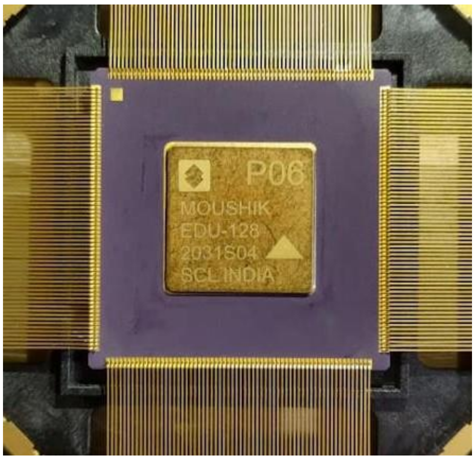


Figure 1.4(b): RISC-V of India

Complex Instruction Set Computer (CISC)

CISC, which stands for Complex Instruction Set Computer**,** is a philosophy for designing chips that are easy to program and which make efficient use of memory. Each instruction in a CISC instruction set might perform a series of operations inside the processor. This reduces the number of instructions required to implement a given program, and allows the programmer to learn a small but flexible set of instructions.

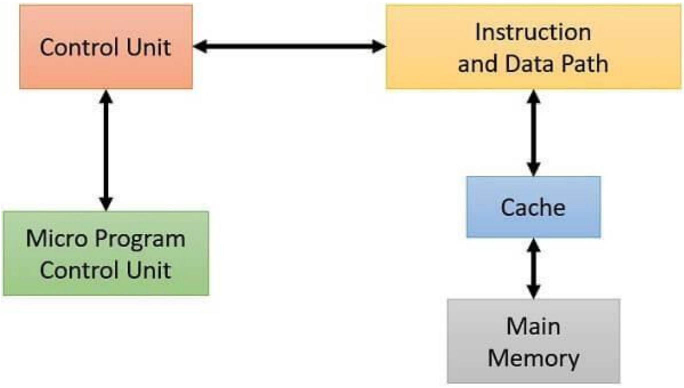


Figure 1.5(a): CISC architecture

The advantages of CISC

At the time of their initial development, CISC machines used available technologies to optimize computer performance.

- Microprogramming is as easy as assembly language to implement, and much less expensive than hardwiring a control unit.
- The ease of micro-coding new instructions allowed designers to make CISC machines upwardly compatible: a new computer could run the same programs as earlier computers because the new computer would contain a superset of the instructions of the earlier computers.
- As each instruction became more capable, fewer instructions could be used to implement a given task. This made more efficient use of the relatively slow main memory.
- Because micro program instruction sets can be written to match the constructs of high-level languages, the compiler does not have to be as complicated.

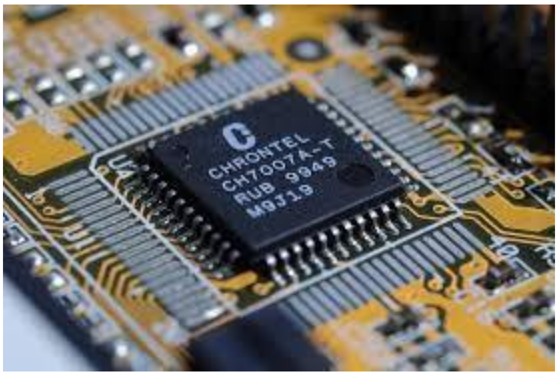


Figure 1.5(b): CISC chip

Memory Architecture

There two different type's memory architectures there are:

- Harvard Architecture
- Von-Neumann Architecture

Harvard Architecture

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

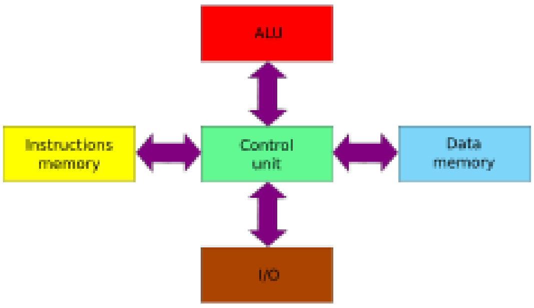


Figure 1.6: Harvard architecture

Von-Neumann Architecture

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. The von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A stored-program digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture. The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

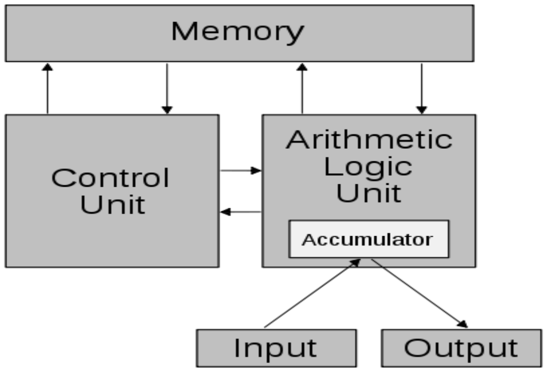


Figure 1.7: Von-Neumann Architecture

Basic Difference between Harvard and Von-Neumann Architecture

- The primary difference between Harvard architecture and the Von Neumann architecture is in the Von Neumann architecture data and programs are stored in the same memory and managed by the same information handling system.
- Whereas the Harvard architecture stores data and programs in separate memory devices and they are handled by different subsystems.
- In a computer using the Von-Neumann architecture without cache; the central processing unit (CPU) can either be reading and instruction or writing/reading data to/from the memory. Both of these operations cannot occur simultaneously as the data and instructions use the same system bus.
- In a computer using the Harvard architecture the CPU can both read an instruction and access data memory at the same time without cache. This means that a computer with Harvard architecture can potentially be faster for a given circuit complexity because data access and instruction fetches do not contend for use of a single memory pathway.
- Today, the vast majority of computers are designed and built using the Von Neumann architecture template primarily because of the dynamic capabilities and efficiencies gained in designing, implementing, operating one memory system as opposed to two. Von Neumann architecture may be somewhat slower than the contrasting Harvard Architecture for certain specific tasks, but it is much more flexible and allows for many concepts unavailable to Harvard architecture such as self-programming, word processing and so on.

Types of Embedded System

1. Real Time Systems

Real-time systems are those which give a quick response to critical situations. They are used in military, medical and industrial applications. Engineers working in these systems have high demand is current days. To develop the Realtime embedded system we require timing analysis, multitasking design, debugging, cross platform testing and architecture design. In these systems, quick response is very important.

2. Standalone Embedded System

This type of embedded system works for itself as a device without needing any interconnected computer. It can take data in the form of Analog or digital signals. This system first process data and then outputs data by displaying on the screen.

3. Networked Embedded System

Networked embedded systems are those systems which are connected to the network to give output to the attached resources. The devices in the networked embedded system are connected to the network with network interfaces. The network can be either a local area network (LAN) or a wide area network (WAN).

4. Mobile Embedded System

Mobile embedded systems are limited in resources including memory. Examples of mobile embedded systems include • Personal digital assistants (PDA) • Cellular phones • Mp3 players 6 • Digital cameras.

5. Small-Scale Embedded System

Small-scale embedded systems consist of 8–16-bit microcontroller. This system can perform tasks at a small level. They have on-chip ROM and RAM. Small scale systems can be even activated by the battery. The tools used to develop small scale embedded systems are an editor, cross assembler, assembler and integrated development environment (IDE). The purpose of this system is not computation but to control as a computer embedded inside it. It behaves as a component of a computer and its function is not to compute. The small-scale system is dedicated to some specific tasks.

6. Medium Scale Embedded System

This embedded system has 16–32-bit microprocessor or microcontroller with external RAM and ROM They can perform medium to complex level works. The integration between hardware and software is complex in these embedded systems. Programming languages used to develop medium scale embedded systems include Java, C, Visual C++, debugger, C++, RTOS, simulator, source code engineering tool and IDE.

7. Sophisticated Embedded System

The embedded system which can do large-scale works with multiple 32-64 bit chips is known as sophisticated embedded systems. They can perform distributed work on a large scale. The complexity of hardware and software is very high in these systems. In sophisticated embedded systems, hardware and software are assembled together on large scale and designing of hardware products is also included in these systems.

Power supply

Block diagram

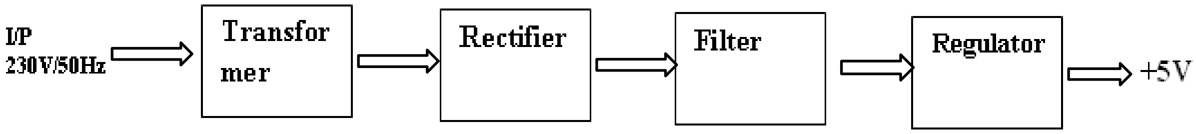


Figure 1.8: Block diagram of the power supply unit

Description

1.2.4.2(a) Transformer

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

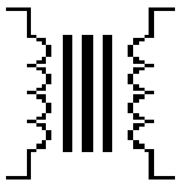


Figure 1.9: Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

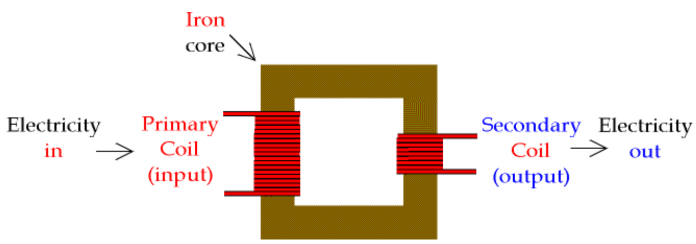


Figure 1.10: Transformer

Basic Principle:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

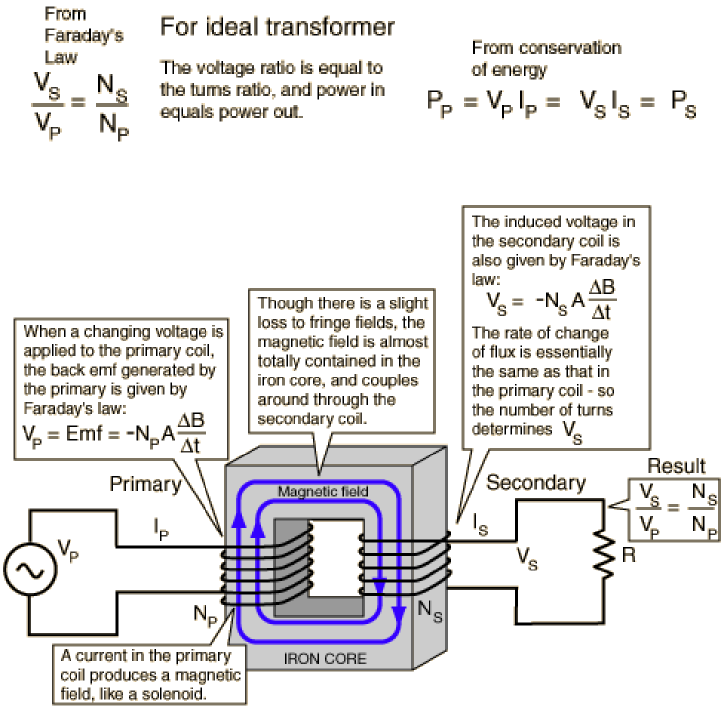


Figure 1.11: Basic Principle

Transformer Working

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

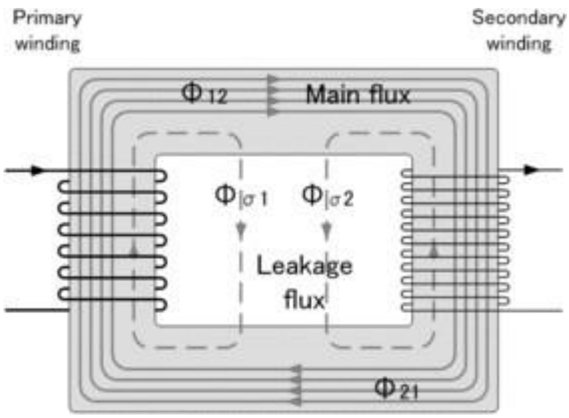


Figure 1.12: Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f.

The iron core is laminated to prevent 'eddy currents' flowing in the core. These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

Classification of Transformer:

- Step-Up Transformer
- Step-Down Transformer

Step-Down Transformer:

Step down transformers is designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step-down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers is made up of two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn's ratio of the two sets of windings determines the amount of voltage transformation.

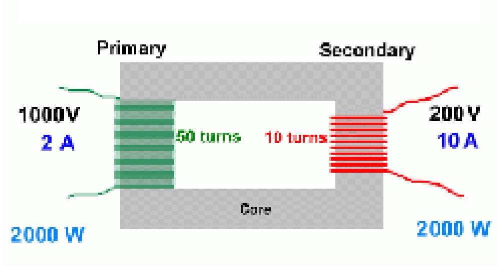


Figure 1.13: Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.

Step-Up Transformer:

A step-up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step-up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it.

A step-up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So, a step-up transformer increases the voltage and a step-down transformer decreases the voltage.

The primary components for voltage transformation are the step-up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground.

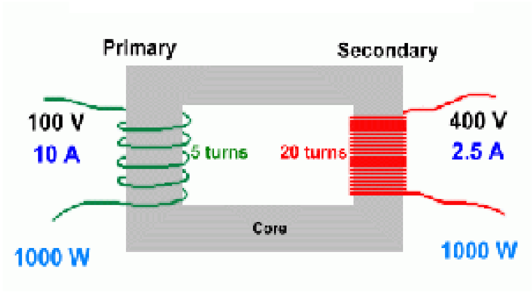


Figure 1.14: Step-Up Transformer

1.2.4.2(b) Diodes

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

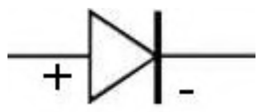


Figure 1.15: Diode Symbol

A diode is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

1.2.4.2(c) Rectifier

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as 'half-wave' and 'full-wave' rectifiers. Both use components called diodes to convert AC into DC.

THE HALF-WAVE RECTIFIER

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure.

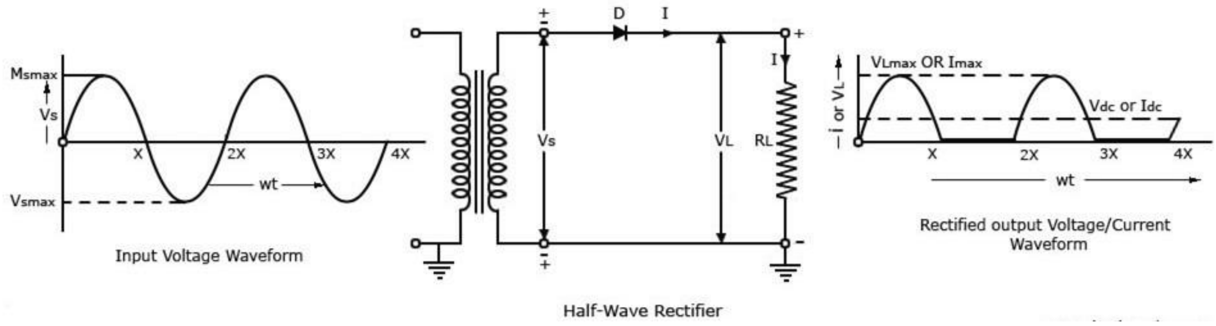


Figure 1.16: Half Wave Rectifier

Figure shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is 0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

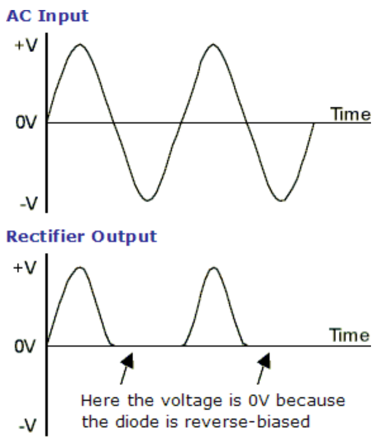


Figure 1.17: Half-Wave Rectification

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

THE FULL-WAVE RECTIFIER

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC.

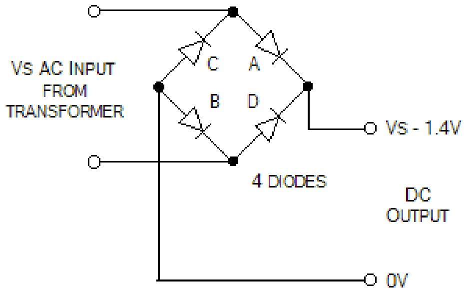


Figure 1.18: Full-Wave Rectifier

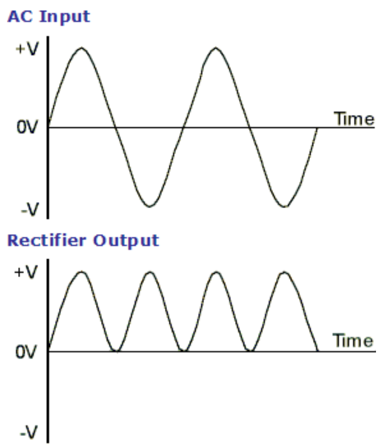


Figure 1.19: Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased.

While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

Capacitor Filter

The capacitor-input filter, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

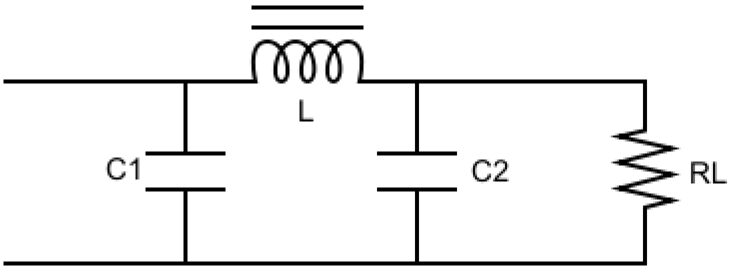


Figure 1.20: Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result only the DC component appears across the load RL.

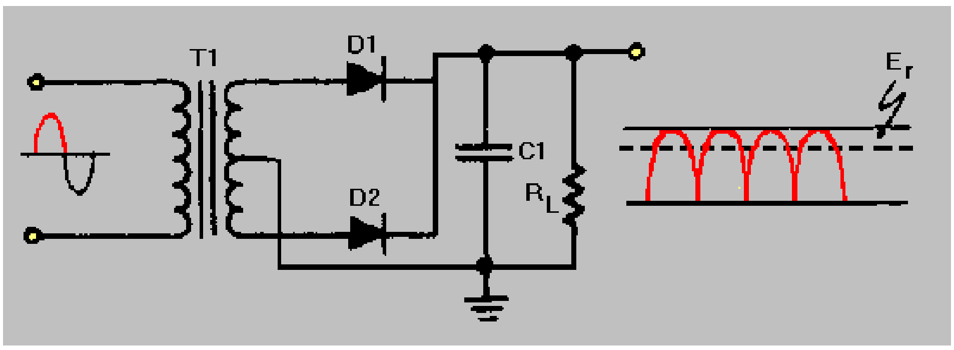


Figure 1.21: Centred Tapped Full-Wave Rectifier with a Capacitor Filter

1.2.4.2(d)Voltage Regulator:

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulators are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx'indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx'indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consist the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

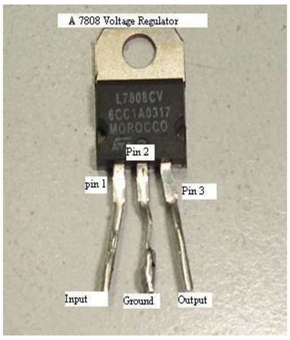


Figure 1.22: Regulator

Circuit diagram

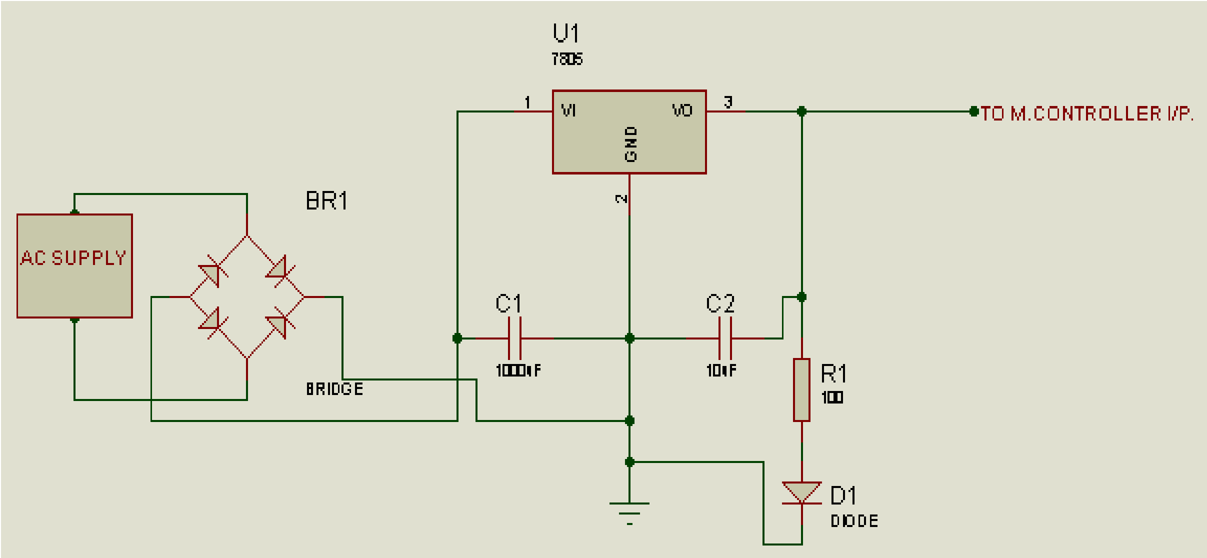


Figure 1.23: circuit diagram of power supply

CHAPTER-2

LITERATURE SURVEY

[1] IEEE paper of "Design of miniature antennas for IoT applications", by L. Lizzi, F. Ferrero, P. Monin, C. Danchesi, S. Boudaud, Published in: 2016 IEEE Sixth International Conference on Communications and Electronics (ICCE). PIC microcontroller was designed to develop a satellite dish positioning system which can be operated by using a Bluetooth control. This project helps n adjusting the position of the dish through a remote Control which acts as a transmitter whose data is received by a Bluetooth receiver which is interfaced to a Microcontroller of PIC 16F877A.

[2] "Antenna Positioning Based on IOT", by Khalid Makhdoomi, published by International Journal of Trends in Scientific Research and Development (IJTSRD), Volume 2, Issue 5, July-August 2018.This system uses sensors and mounts motors on each antenna to verify its facing direction transmitted over IoT. In case the direction of satellite or station changes with time the direction of antenna also changes accordingly. The antenna positions are visible to operator on the IoT GUI through the internet. The IoT Gecko is used to develop the GUI system for antenna monitoring. Our model allows for monitoring antenna direction as well as transmitting new coordinates to position the antenna and motor appropriately positions the antenna accordingly.

[3] "Design of Advanced Antenna Positioning System", by Rahane Suraj Dildar, Mhaske Shital Arun, Shingate Sujata Rajendra and Prof. S. B. Mandlik, published by International Journal of Research in Advent Technology, Vol.6, No.3, March 2018. Design of advanced antenna positioning system consists of various transmitters, a delay circuit & a receiver which is placed at the centre of antenna. When a signal is received at receiver, it provides adequate strength and gets transferred to an Analog to Digital Converter and the converted data is sent to a microcontroller. The microcontroller analyzes data and gives output accordingly to the motor driver which is connected to servomotor. In this way servomotor gets the adequate data or information required for correct positioning of receiving antenna.

[4] "Implementing an IOT based Antenna Positioning System", by Pooja Revane, Shradha Salaskar, KomalShelke, Priyanka Tawar, Akshata Raut, published in International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume 6 Issue IV, April 2018. In this system sensors will be mounted on the antenna to detect its direction and its direction will be changed by motors using IoT. When the direction of a transmitting station changes over time, the antenna direction must also be changed accordingly.

[5] “Automatic Dish Antenna Positioning System”, by Jadhav Seema, Shejwalkar Rakshanda, Andhale Jyoti, published by JournalNX - A Multidisciplinary Peer Reviewed Journal, ISSN No: 2581-4230, 21st - 22nd February, 2018. Dish antenna positioning is used to get broadcast signals from the satellite. The aim of this project is to control the dish automatically, which will be capable of receiving the broadcast signals from the satellite. It will rotate horizontally and vertically. The position of the dish is adjusted by an android application or Bluetooth. Power meter is used to measure the maximum signal strength. This system uses two stepper motors which helps it top move horizontally and vertically.

[6] “Microcontroller Based Wireless Automatic Antenna Positioning System” by Surya deo choudhary, Pankanj Rai, Arvind kumar, Irshad Alam used to design automatic Antenna Positioning system primarily functions to identify the source of signal. The signal may be of any type and any kind, it automatically identifies the presence of a particular signal and the antenna will remain stationary as long as the signal link is established.

[7] “Automated Antenna Positioning for Wireless Networks” by Amit Dvir, Yehuda Ben Shimol, Yoav Ben-Yehezkel, Michael Segal. This article addresses a real-life problem which is obtaining communication links between multiple base stations sites. This is done by positioning a minimal set of fixed-access relay antenna sites on a given terrain. To minimize the number of relay antenna sites is considered difficult due to substantial installation and maintenance costs. Despite the potential significant cost saving by eliminating even a single antenna site, a hardly optimal manual approach is employed due to the computation complexity of the problem.

[8] “Remote Alignment of Dish Positioning By Android Application” by Prajwal Basnet, Pranjal Grover Preeti Pannu. The project is designed to develop a dish positioning system which can be operated by using an Android application. The main application of using a dish is to receive signal from satellites and other broadcasting sources. In order to position the dish to the exact angle to receive the maximum signal of a particular frequency, it needs to be adjusted manually.

CHAPTER-3

ILLUSTRATION

Introduction to antennas

An Antenna is a transducer, which converts electrical power into electromagnetic waves and vice versa.

An Antenna can be used either as a transmitting antenna or a receiving antenna.

- A transmitting antenna is one, which converts electrical signals into electromagnetic waves and radiates them.
- A receiving antenna is one, which converts electromagnetic waves from the received beam into electrical signals.
- In two-way communication, the same antenna can be used for both transmission and reception.

Antenna can also be termed as an Aerial. Plural of it is, antennae or antennas. Now-adays, antennas have undergone many changes, in accordance with their size and shape. There are many types of antennas depending upon their wide variety of applications [2].

Figure 3.1: Satellite antenna

NEED OF ANTENNA

In the field of communication systems, whenever the need for wireless communication arises, there occurs the necessity of an antenna. Antenna has the capability of sending or receiving the electromagnetic waves for the sake of communication, where you cannot expect to lay down a wiring system. The following scenario explains this.

Radiation Mechanism

The sole functionality of an antenna is power radiation or reception. Antenna (whether it transmits or receives or does both) can be connected to the circuitry at the station through a transmission line. The functioning of an antenna depends upon the radiation mechanism of a transmission line.

A conductor, which is designed to carry current over large distances with minimum losses, is termed as a transmission line. For example, a wire, which is connected to an antenna. A transmission line conducting current with uniform velocity, and the line being a straight one with infinite extent, radiates no power.

For a transmission line, to become a waveguide or to radiate power, has to be processed as such.

- If the power has to be radiated, though the current conduction is with uniform velocity, the wire or transmission line should be bent, truncated or terminated.
- If this transmission line has current, which accelerates or decelerates with a time varying constant, then it radiates the power even though the wire is straight.
- The device or tube, if bent or terminated to radiate energy, then it is called as waveguide. These are especially used for the microwave transmission or reception.

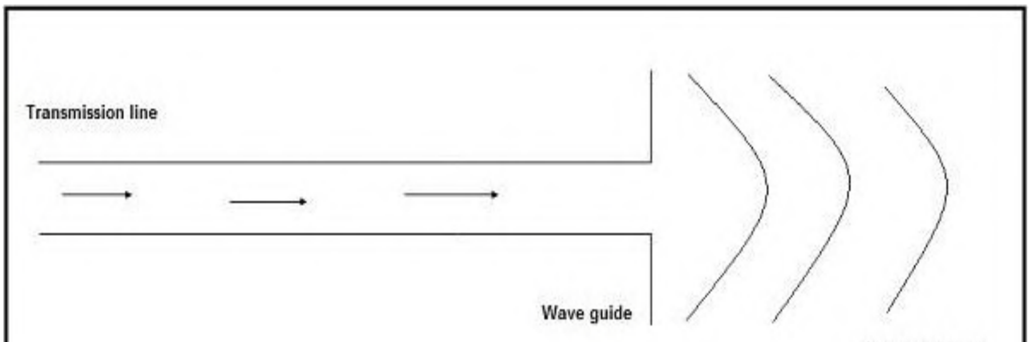


Figure 3.2: Radiation

The above diagram represents a waveguide, which acts as an antenna. The power from the transmission line travels through the waveguide which has an aperture, to radiate the energy.

TYPES OF ANTENNAS

Antennas may be divided into various types depending upon −

- The physical structure of the antenna.
- The frequency ranges of operation.
- The mode of applications etc.

Physical structure

Following are the types of antennas according to the physical structure. You will learn about these antennas in later chapters.

- Wire antennas
- Aperture antennas
- Reflector antennas
- Lens antennas
- Micro strip antennas
- Array antennas

Frequency of operation

Following are the types of antennas according to the frequency of operatio:.

- Very Low Frequency (VLF)
- Low Frequency (LF)
- Medium Frequency (MF)
- High Frequency (HF)
- Very High Frequency (VHF)
- Ultra-High Frequency (UHF)
- Super High Frequency (SHF)
- Micro wave
- Radio wave

Mode of Applications

Following are the types of antennas according to the modes of applications:

- Point-to-point communications
- Broadcasting applications

- Radar communications
- Satellite communications

Introduction to IOT

The internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. It may also refer to the billions of physical devices around the world that are now connected to the internet, all collecting and sharing data.

A ***thing*** in the internet of things can be a person with a heart monitor implant, a farm animal with a biochip transponder, an automobile that has built-in sensors to alert the driver when tire pressure is low or any other natural or man-made object that can be assigned an Internet Protocol (IP) address and is able to transfer data over a network.

Increasingly, organizations in a variety of industries are using IoT to operate more efficiently, better understand customers to deliver enhanced customer service, improve decision-making and increase the value of the business.

How IOT works?

An IoT ecosystem consists of web-enabled smart devices that use embedded systems, such as processors, sensors and communication hardware, to collect, send and act on data they acquire from their environments. IoT devices share the sensor data they collect by connecting to an IoT gateway or other edge device where data is either sent to the cloud to be analyzed or analyzed locally. Sometimes, these devices communicate with other related devices and act on the information they get from one another. The devices do most of the work without human intervention, although people can interact with the devices -- for instance, to set them up, give them instructions or access the data.

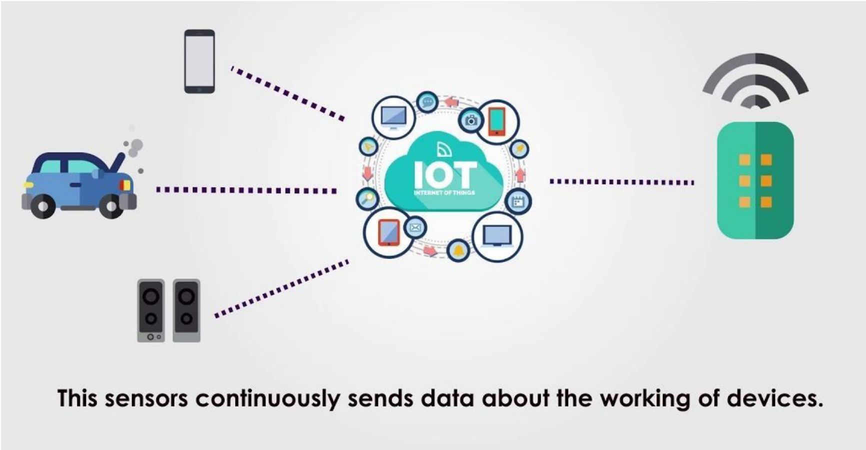


Figure 3.3: Working of IOT

Existing system

In existing system, tracking capability of a control system is critical. Antenna are controlled manually which is time consuming and accuracy was less. Place the antenna to the exact angle to obtain the full signal of a given frequency and it requires manual adjustment. Antennas are stationary and are not allowed to receive the signals which have gone the particular degree beyond the antenna is capable to receive the signal.

Disadvantages:

- Since the antennas are set manually, it is time consuming and less accurate.
- Number of antennae required is more.
- High cost and man labour is required.
- Antennas are inflexible and failed to receive the signals from all the direction.

Proposed system

Hence, we propose an antenna positioning system which enables antennas to be placed remotely over IoT. Because the position of the transmitting station changes over time even the position of the antenna should be changed accordingly. The signal received by antennas is affected by the direction of transmitting stations those changes over time so there is a need of system that Automatically changes the antenna positioning if there is any change in transmitting station over time [9]. This is used by measuring the degree of command from remote and input to drive the motor at the desired degree.

Advantages:

- Use of an IOT application to control the positioning of antenna around the globe.
- The device can receive signals from all directions.
- Increase in the range of detection.
- Decrease in man power and less time consuming.
- Increase in accuracy of the signal.
- Decrease in the number of antennas.

Block diagram of the project

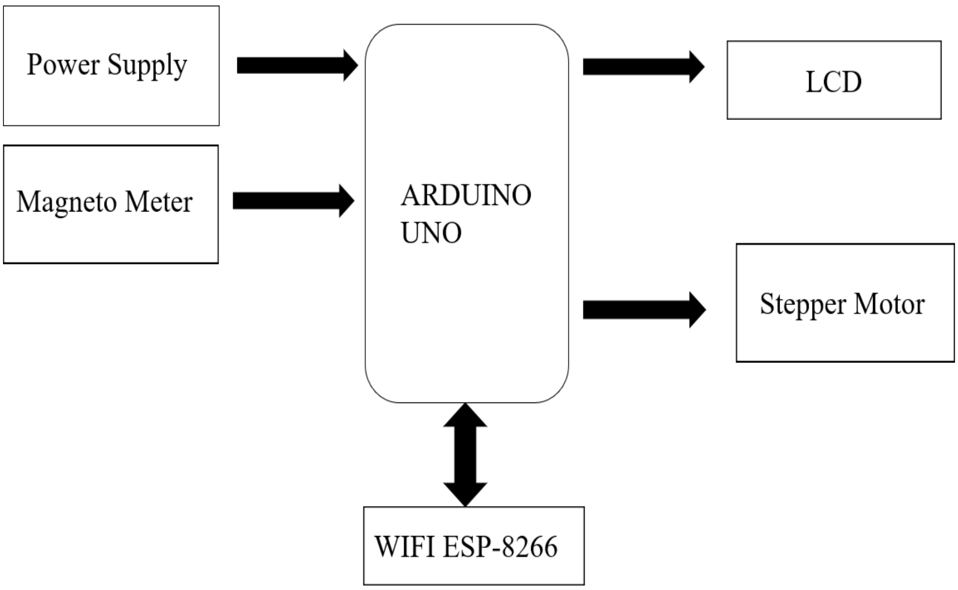


Figure 3.4: Block diagram of the project

Explanation

The power supply is given to the system, the magnetometer is considered for the direction of the antenna. When the magnetometer is moved in a certain angle the Arduino which is being programmed acts as the medium, sends the information to the stepper motor, LCD and the wi-fi module. The various acts that take place after receiving the signal from the Arduino, (1) there a rotation of the stepper motor in the desired axis with a considerable amount of change in angle (2) the LCD displays the amount to angle change and the direction of the shift (3) the wi-fi module receives the signal and notifies the user with the amount of change in angle and axis in the form of message into the mobile application.

There is another way to access the rotation of antenna with the help of mobile application. A command is given through the application for rotation. A "*" is given for the rotation of clockwise direction & "#" is given for the rotation of anti-clockwise direction.

CHAPTER-4

SOFTWARE DESCRIPTION

Introduction to proteus

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possesses 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.It is a software suite containing schematic, simulation as well as PCB designing.

ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting a new design

Step 1: Open ISIS software and select new design in File menu

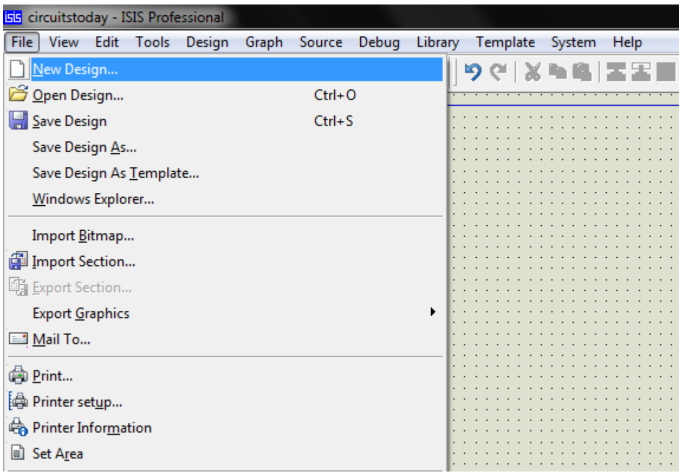


Figure 4.1: Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

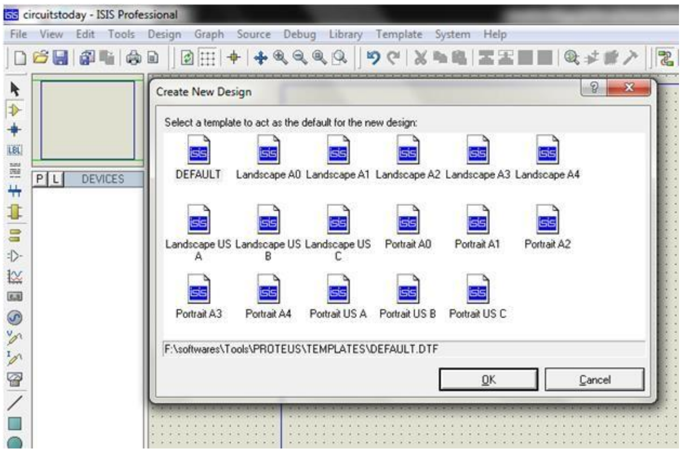


Figure 4.2: Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

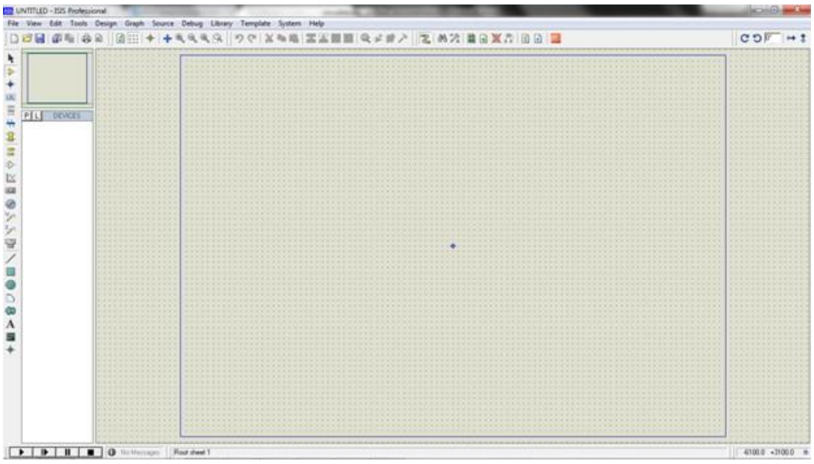


Figure 4.3: Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

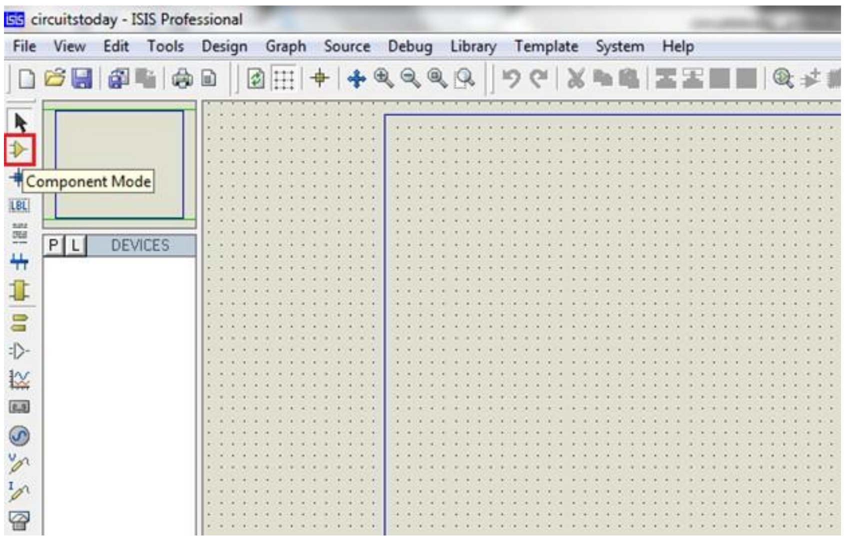


Figure 4.4: Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

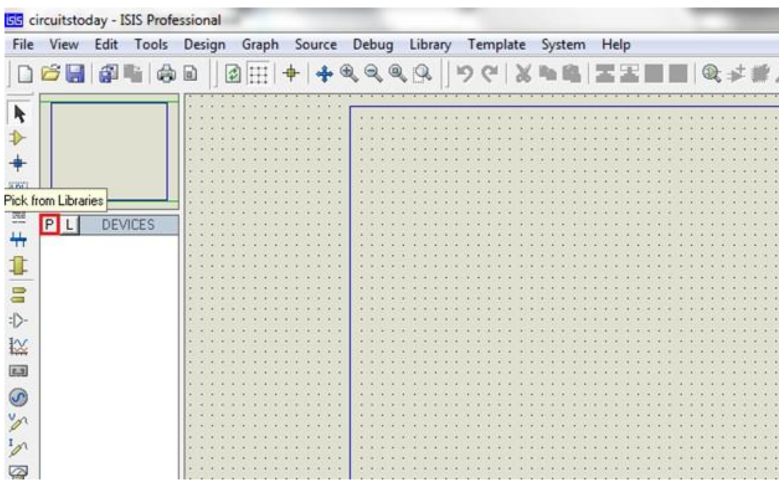


Figure 4.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keyword's text box.

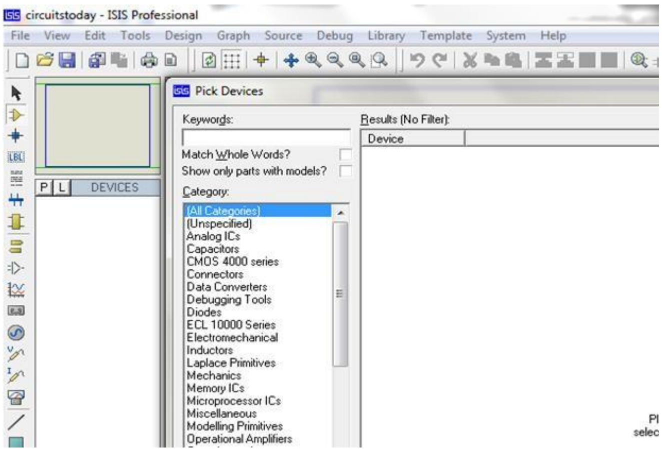


Figure 4.6: Keywords Textbox

Example shows selection of push button. Select the components accordingly.

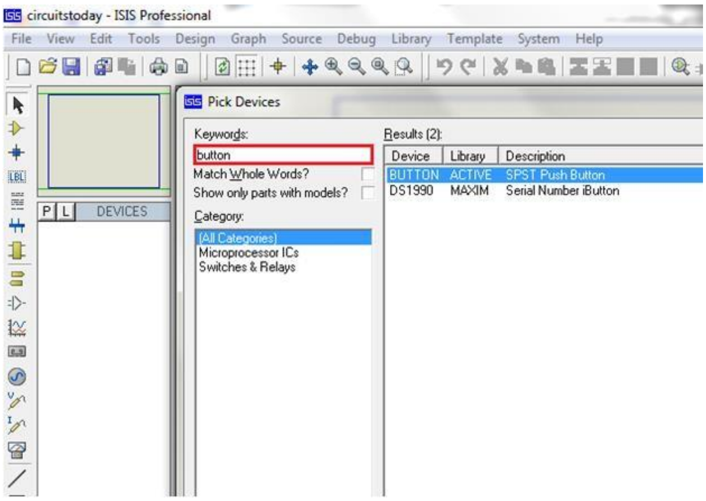


Figure 4.7: Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

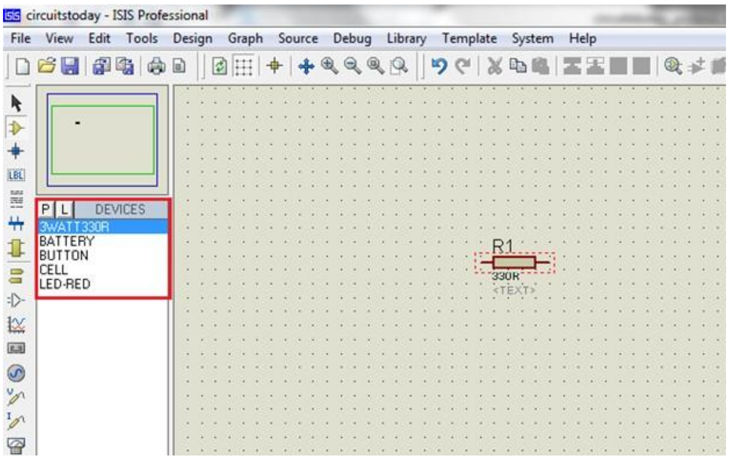


Figure 4.8: Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

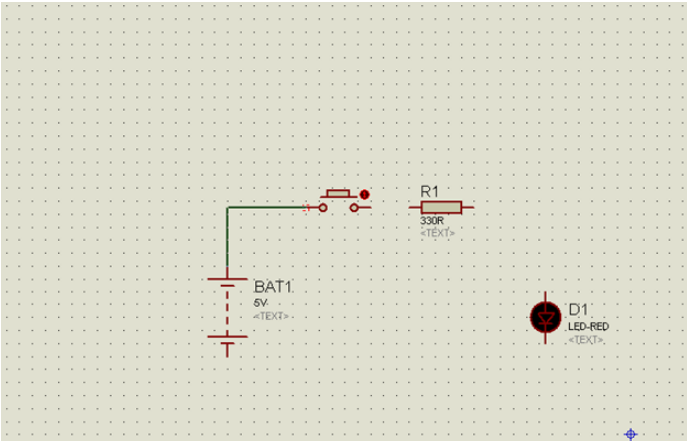


Figure 4.9: Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

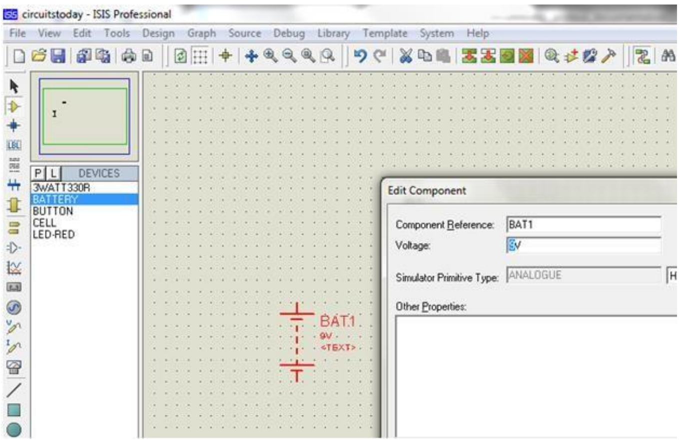


Figure 4.10: Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

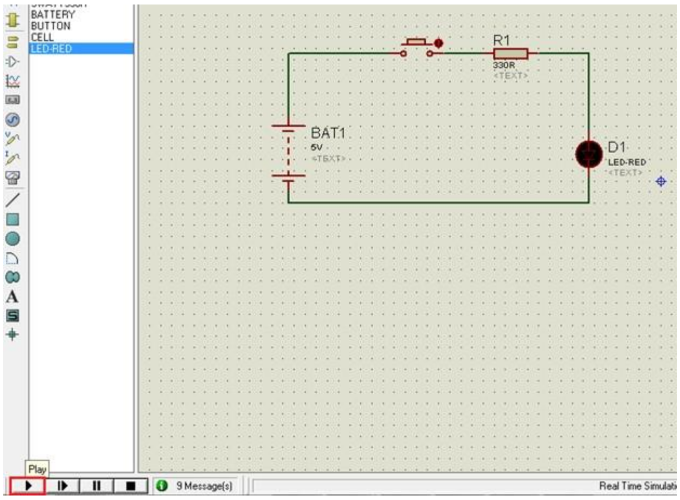


Figure 4.11: Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

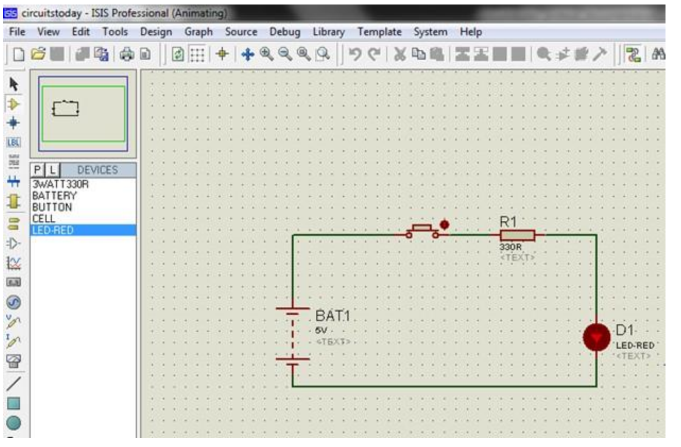


Figure 4.12: Simulation Animating

Simulation can be stepped, paused or stopped at any time.

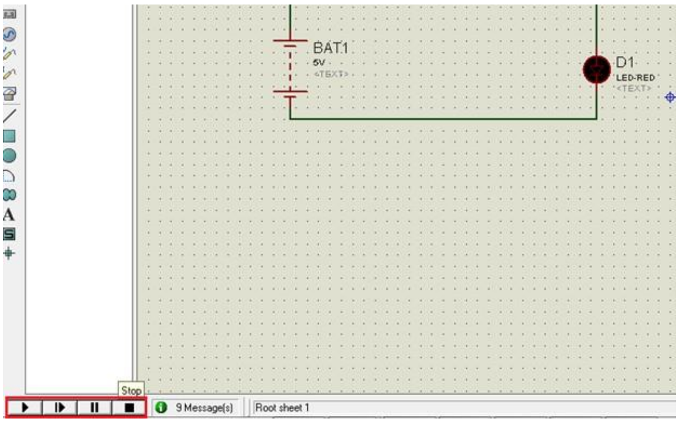


Figure 4.13: Simulation Step-Pause-Stop Buttons

Arduino IDE

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Arduino Pro IDE	
Developer(s)	Arduino Software
Preview release	v0.0.2 / 28 October 2019; 3 months ago
Repository	github.com/Arduino/Arduino
Written in	C, C++
Operating system	Windows, macOS, Linux
Platform	IA-32, x86-64, ARM
Type	Integrated development environment
License	LGPL or GPL license
Website	blog.arduino.cc/2019/10/18/arduino-pro-ide-alpha-preview-with-advanced-features/

Table 4.1: Arduino IDE description

In October 2019 the Arduino organization began providing early access to a new Arduino Pro IDE with debugging and other advanced features.

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

Installation

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

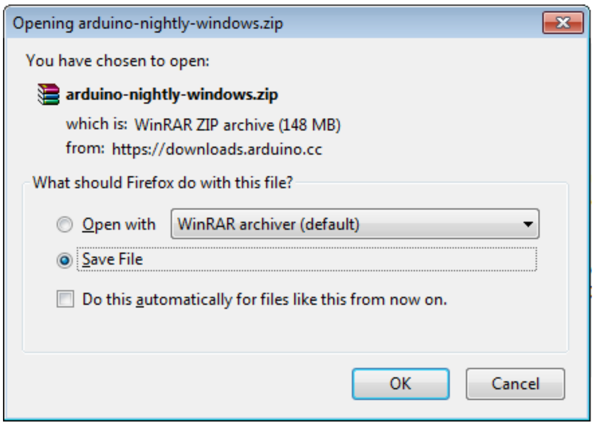
Figure 4.14: USB cable B plug

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Figure 4.15: USB cable A plug

Step 2 – Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



Opening arduino-nightly-windows.zip

You have chosen to open:

arduino-nightly-windows.zip

which is: WinRAR ZIP archive (148 MB)

from: https://downloads.arduino.cc

What should Firefox do with this file?

Open with WinRAR archiver (default)

Save File

Do this automatically for files like this from now on.

OK Cancel

Figure 4.16: Downloading program

Step 3 – Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

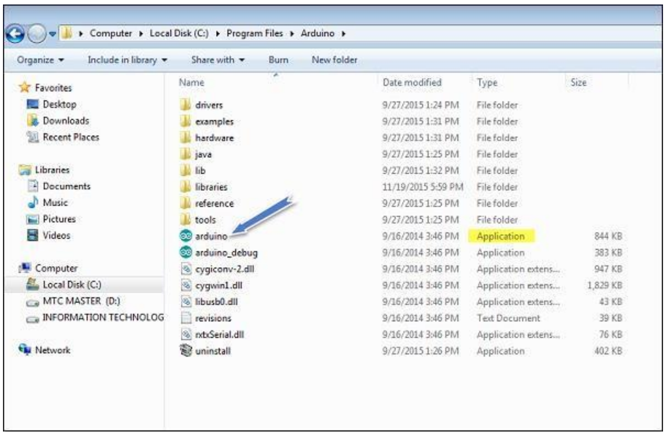


Figure 4.17: Launching the software

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.

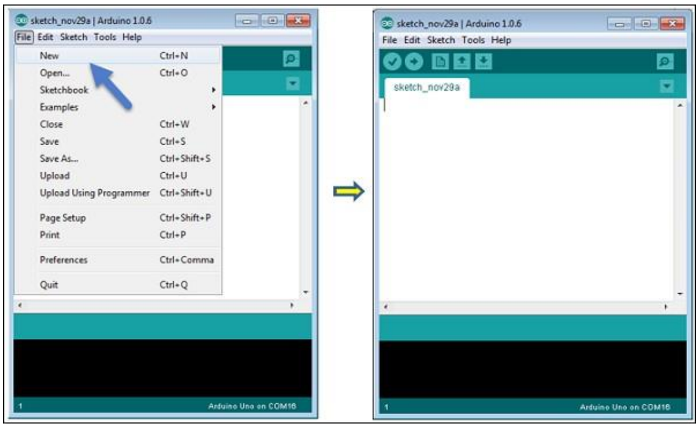


Figure 4.18: open new project

To open an existing project example, select File → Example → Basics → Blink.

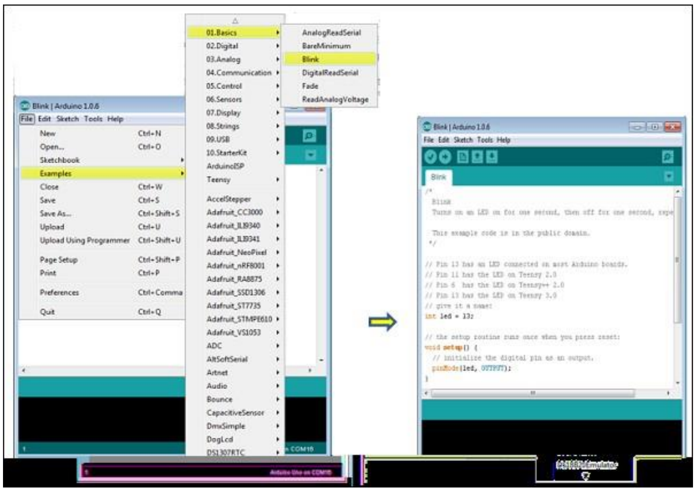


Figure 4.19: open new project

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

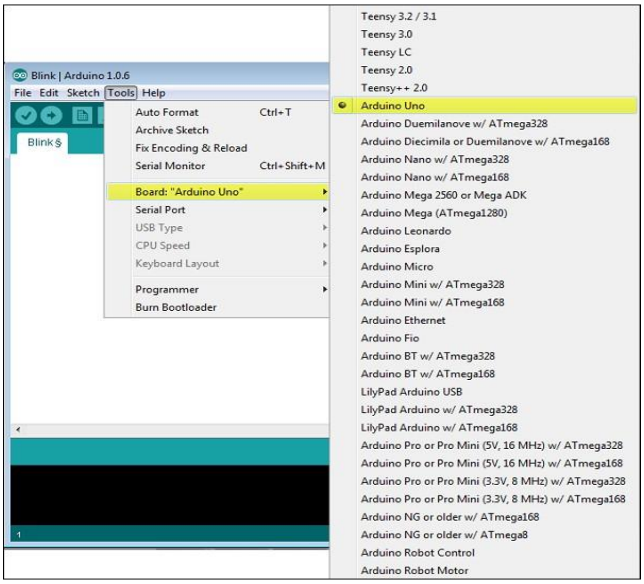


Figure 4.20: selecting board type

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

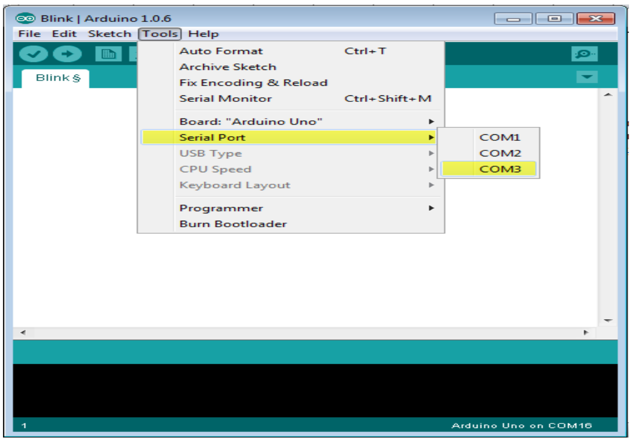


Figure 4.21: selection of port

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

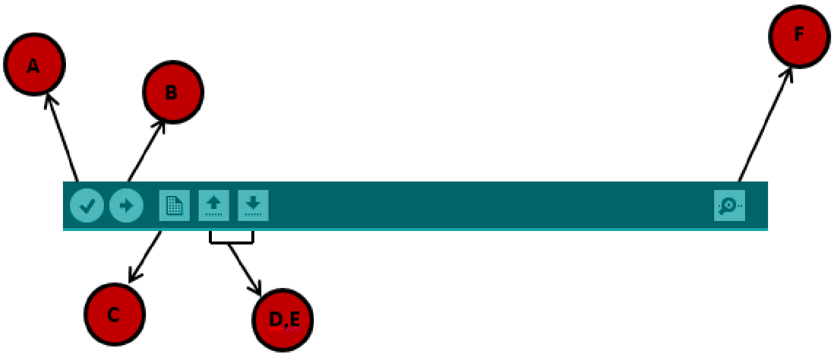


Figure 4.22: uploading the program

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketches.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

CHAPTER-5

HARDWARE DESCRIPTION

Arduino

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board. The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

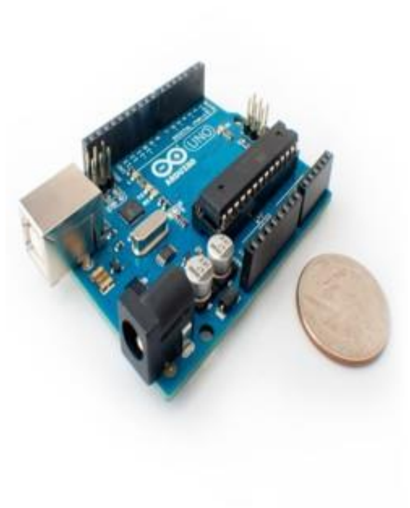
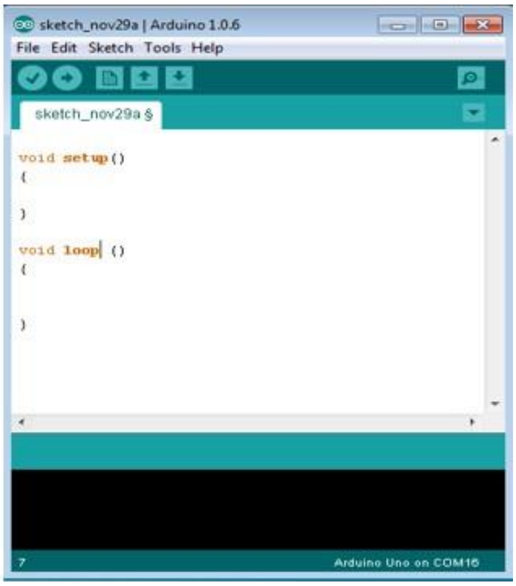



Figure 5.1: Arduino board and program page

Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE. The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately.

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header

LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 5.1: Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 5.2: Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 5.3: Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table 5.4: Arduino boards based on AT91SAM3X8E microcontroller

Board Description

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

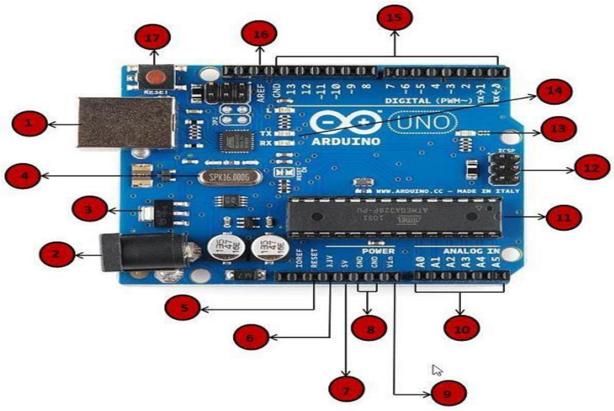


Figure 5.2: Arduino UNO

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed

	on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.

13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Table 5.5: pin configuration and description

Liquid Cristal Display

A liquid crystal display (LCD) is a thin, flat display device made up of any number of colour or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to a controller is an LCD display. Some of the most common LCDs connected to the controllers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Description Of 16x2:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all-Parallel Ports. It however doesn't show the use of the Status Port as an input. So, what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

Schematic Diagram

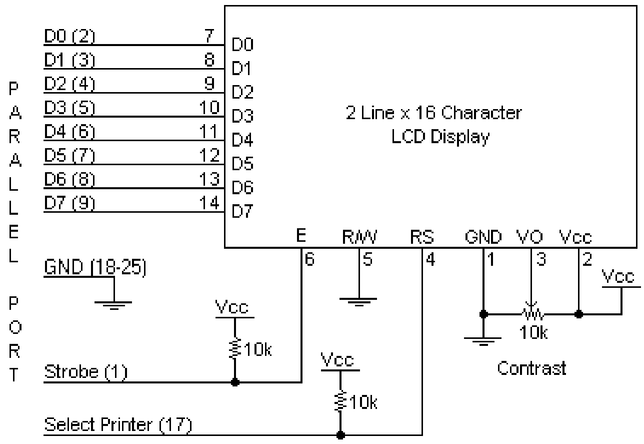


Figure 5.3: Schematic diagram of LCD

- Above is the quite simple schematic. The LCD panel's *Enable* and *Register Select* is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore, by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.
- We make no effort to place the Data bus into reverse direction. Therefore, we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result, we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.
- The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.

SPECIFICATIONS:

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Table 5.6: Specification of LCD

Pin Description:

Most LCDs with 1 controller have 14 Pins and LCDs with 2 controllers has 16 Pins (two pins are extra in both for back-light LED connections).

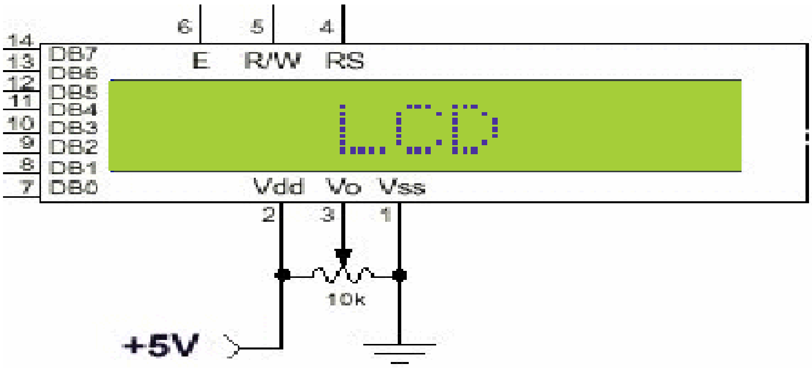


Figure 5.4: Pin diagram of 1x16 lines LCD

Control lines

EN: Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS: Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW: Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are writing commands, so RW will almost always be low. Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

LCD Commands:

There are some present commands instructions in LCD, which we need to send to LCD through some microcontroller. Some important command instructions are given below:

Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left
0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF

C1	Jump to second line, position 1
C2	Jump to second line, position 2

Table 5.7: Commands

Stepper Motor

Stepper motors are DC motors that move in discrete steps. They have multiple coils that are organized in groups called "phases". By energizing each phase in sequence, the motor will rotate, one step at a time.

With a computer controlled stepping you can achieve very precise positioning and/or speed control. For this reason, stepper motors are the motor of choice for many precision motion control applications. Stepper motors come in many different sizes and styles and electrical characteristics.

How Stepper Motor Works?

Stepper motors are very different from a regular DC motor. Instead of spinning like DC motors do, stepper motor steps at a specific resolution for each pulse. The motor that we are using needs 48 steps / pulses just to complete a single revolution! That should be enough to talk about its precision.

Another advantage of stepper motors is the fact that their speed of rotation can be achieved almost instantly even if you change the spinning direction. Stepper motor consists of a **rotor** - the permanent magnet that rotates inside, and **stator** - four coils (north, east, south, west) that are part of the case, and which **don't move**. Rotor can be moved by sequentially applying a pulsed DC voltage to one or two coils at a time.

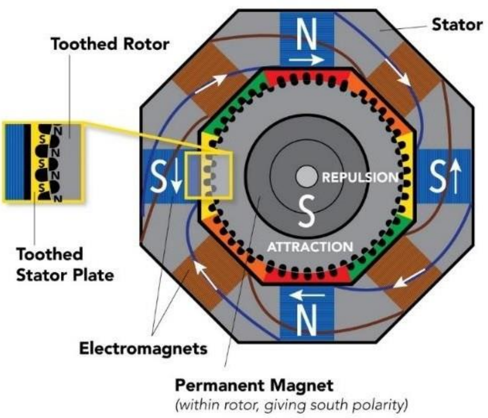


Figure 5.5: Working of stepper motor

Stepper Motor Connections

Unipolar motor should have five or six connections depending on the model. If the motor has six connections like the one pictured above, you have to join pins 1 and 2 (red) together and connect them to a (+) 12-24V voltage supply. The remaining pins; a1 (yellow), b1 (black), a2 (orange), b2 (brown) should be connected to a driver (ULN2003) as shown on the schematic.

Working

Frame 1: The top electromagnet (1) is turned on, attracting the nearest tooth of a gear-shaped iron rotor. With the teeth aligned to electromagnet 1, they will be slightly offset from electromagnet 2. **Frame 2:** The top electromagnet (1) is turned off, and the right electromagnet (2) is energized, pulling the nearest teeth slightly to the right. This results in a rotation of 3.6° in this example. **Frame 3:** The bottom electromagnet (3) is energized; another 3.6° rotation occurs. **Frame 4:** The left electromagnet (4) is enabled, rotating again by 3.6°. When the top electromagnet (1) is again enabled, the teeth in the sprocket will have rotated by one tooth position; since there are 25 teeth, it will take 100 steps to make a full rotation in this example.

A stepper motor (or step motor) is a brushless, synchronous electric motor that can divide a full rotation into a large number of steps. The motor's position can be controlled precisely without any feedback mechanism (see Open-loop controller), as long as the motor is carefully sized to the application. Stepper motors are similar to switched reluctance motors (which are very large stepping motors with a reduced pole count, and generally are closed-loop commutated.)

TYPES

There are three main types of stepper motors:

1. Permanent Magnet Stepper
2. Hybrid Synchronous Stepper
3. Variable Reluctance Stepper

Permanent magnet motors use a permanent magnet (PM) in the rotor and operate on the attraction or repulsion between the rotor PM and the stator electromagnets. Variable reluctance (VR) motors have a plain iron rotor and operate based on the principle of that minimum reluctance occurs with minimum gap, hence the rotor points are attracted toward the stator magnet poles. Hybrid stepper motors are named because they use a combination of PM and VR techniques to achieve maximum power in a small package size.

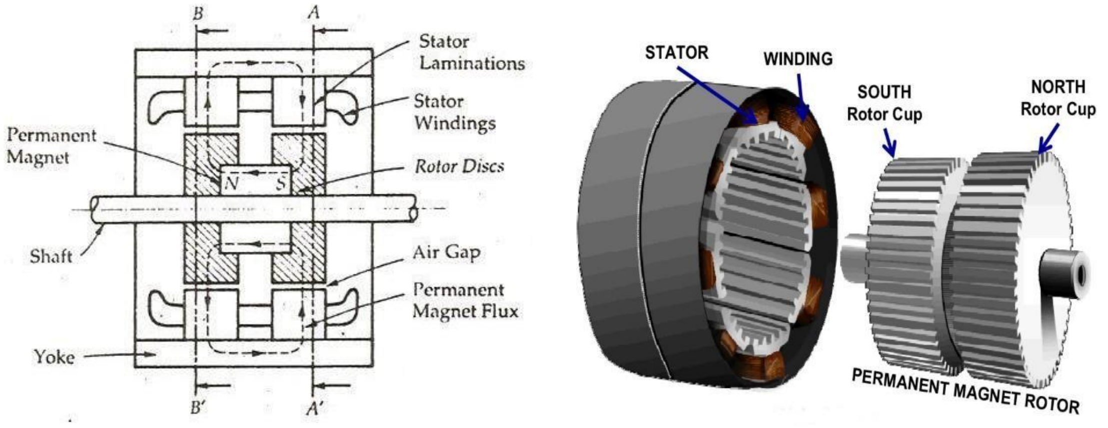


Figure 5.6: Permanent Magnet Stepper Hybrid Synchronous Stepper

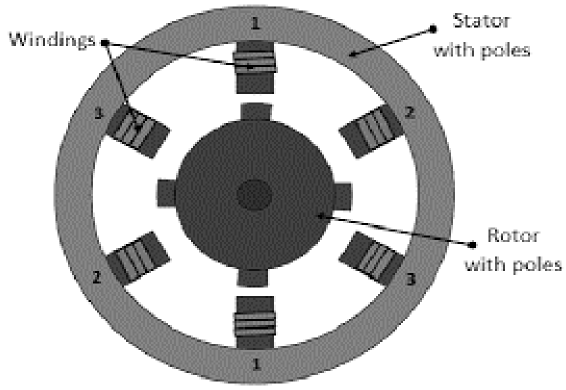


Figure 5.7: Variable reluctance stepper

►Connecting Stepper Motor

There are actually many ways you can interface a stepper motor to your controller, out of them the most used interfaces are:

1. Interface using L293D - H-Bridge Motor Driver
2. Interface using ULN2003/2004 - Darlington Arrays

We will discuss both connection techniques one by one. The above-mentioned methods need 4 controller pins for interface.

► Connecting stepper using L293D

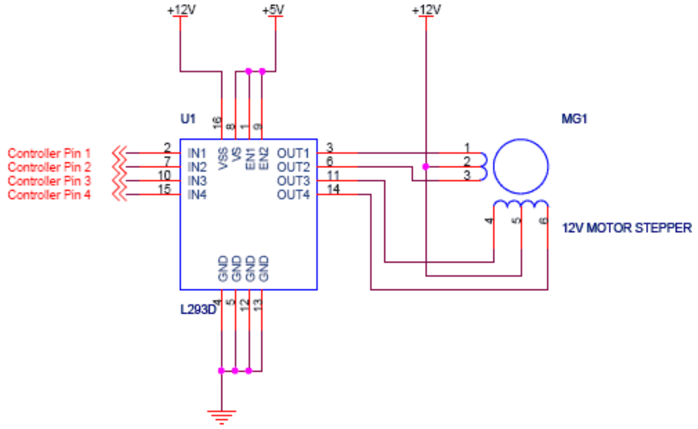


Figure 5.8: Connecting unipolar stepper using L293D

As you see in the circuit above the four pins "Controller pin 1",2,3 and 4 will control the motion and direction of the stepper motor according to the step sequence programmed in the controller.

➢ Connecting stepper using ULN2003/2004 - Darlington Arrays

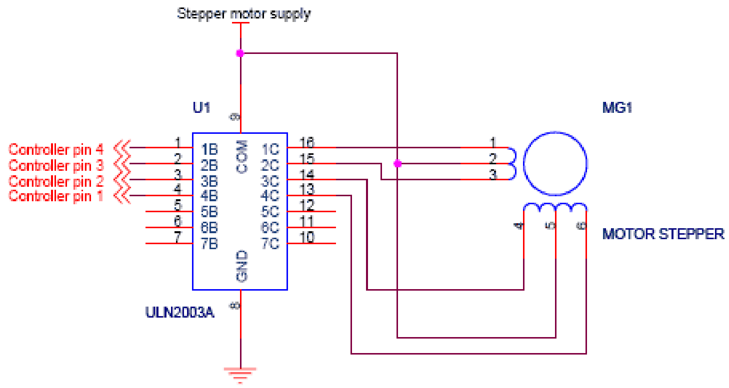


Figure 5.9: Connecting unipolar stepper using ULN2003/2004

here in this circuit too the four pins "Controller pin 1", 2, 3 and 4 will control the motion and direction of the stepper motor according to the step sequence sent by the controller. 8051 Microcontroller receives the decoded data through the input port of 8255 and process the data according to the control software to generate the control signal for the stepper motor to rotate at the desired angular position. The control signal goes to motor driver ULN 2003 to drive the stepper motor. [5]

Magneto Metre

A magnetometer is a device that measures magnetic field or magnetic dipole moment. Some magnetometers measure the direction, strength, or relative change of a magnetic field at a particular location. A compass is one such device, one that measures the direction of an ambient magnetic field, in this case, the Earth's magnetic field. Other magnetometers measure the magnetic dipole moment of a magnetic material such as a ferromagnet.

The first magnetometer capable of measuring the absolute magnetic intensity at a point in space was invented by Carl Friedrich Gauss in 1833 and notable developments in the 19th century included the Hall effect, which is still widely used.

Magnetometers are widely used for measuring the Earth's magnetic field, in geophysical surveys, to detect magnetic anomalies of various types, and to determine the dipole moment of magnetic materials. In an aircraft's attitude and heading reference system, they are commonly used as a heading reference. Magnetometers are also used by the military in magnetic mines to detect submarines.

Magnetometers can be used as metal detectors: they can detect only magnetic (ferrous) metals, but can detect such metals at a much larger depth than conventional metal detectors; they are capable of detecting large objects, such as cars, at tens of metres, while a metal detector's range is rarely more than 2 metres.

In recent years, magnetometers have been miniaturized to the extent that they can be incorporated in integrated circuits at very low cost and are finding increasing use as miniaturized compasses (MEMS magnetic field sensor).

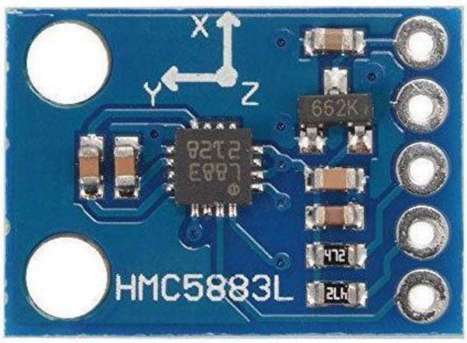


Figure 5.10: Magnetometer

Types of magnetometers

The Magnetometer experiment for the Juno orbiter for Juno can be seen here on the end of a boom. The spacecraft uses two fluxgate magnetometers. (See, also Magnetometer (Juno))

There are two basic types of magnetometer measurement. Vector magnetometers measure the vector components of a magnetic field. Total field magnetometers or scalar magnetometers measure the magnitude of the vector magnetic field. Magnetometers used to study the Earth's magnetic field may express the vector components of the field in terms of declination (the angle between the horizontal component of the field vector and magnetic north) and the inclination (the angle between the field vector and the horizontal surface).

Absolute magnetometers measure the absolute magnitude or vector magnetic field, using an internal calibration or known physical constants of the magnetic sensor. Relative magnetometers measure magnitude or vector magnetic field relative to a fixed but uncalibrated baseline. Also called variometers, relative magnetometers are used to measure variations in magnetic field.

Magnetometers may also be classified by their situation or intended use. Stationary magnetometers are installed to a fixed position and measurements are taken while the magnetometer is stationary. Portable or mobile magnetometers are meant to be used while in motion and may be manually carried or transported in a moving vehicle. Laboratory magnetometers are used to measure the magnetic field of materials placed within them and are typically stationary. Survey magnetometers are used to measure magnetic fields in geomagnetic surveys; they may be fixed base stations, as in the INTERMAGNET network, or mobile magnetometers used to scan a geographic region.

Performance and capabilities

The performance and capabilities of magnetometers are described through their technical specifications. Major specifications include

- Sample rate is the number of readings given per second. The inverse is the cycle time in seconds per reading. Sample rate is important in mobile magnetometers; the sample rate and the vehicle speed determine the distance between measurements.
- Bandwidth or bandpass characterizes how well a magnetometer tracks rapid changes in magnetic field. For magnetometers with no onboard signal processing, bandwidth is determined by the Nyquist limit set by sample rate. Modern magnetometers may perform smoothing or averaging over sequential samples, achieving a lower noise in exchange for lower bandwidth.

- Resolution is the smallest change in a magnetic field the magnetometer can resolve. A magnetometer should have a resolution a good deal smaller than the smallest change one wishes to observe.
- Quantization error is caused by recording roundoff and truncation of digital expressions of the data.
- Absolute error is the difference between the readings of a magnetometer true magnetic field.
- Drift is the change in absolute error over time.
- Thermal stability is the dependence of the measurement on temperature. It is given as a temperature coefficient in units of nT per degree Celsius.
- Noise is the random fluctuations generated by the magnetometer sensor or electronics.
- Sensitivity is the larger of the noise or the resolution.
- Heading error is the change in the measurement due to a change in orientation of the instrument in a constant magnetic field.
- The dead zone is the angular region of magnetometer orientation in which the instrument produces poor or no measurements. All optically pumped, proton-free precession, and Over Hauser magnetometers experience some dead zone effects.
- Gradient tolerance is the ability of a magnetometer to obtain a reliable measurement in the presence of a magnetic field gradient. In surveys of unexploded ordnance or landfills, gradients can be large.

WIFI ESP8266:

Description

The ESP8266 WiFi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers. The ESP8266 module is an extremely cost-effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existence interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

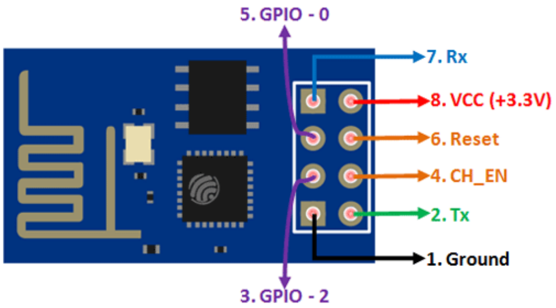


Figure 5.11: Pin configuration of the wi-fi module

Features

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- Default baud rate: 115200

CHAPTER-6

PROJECT DESCRIPTION

Schematic diagram of the project

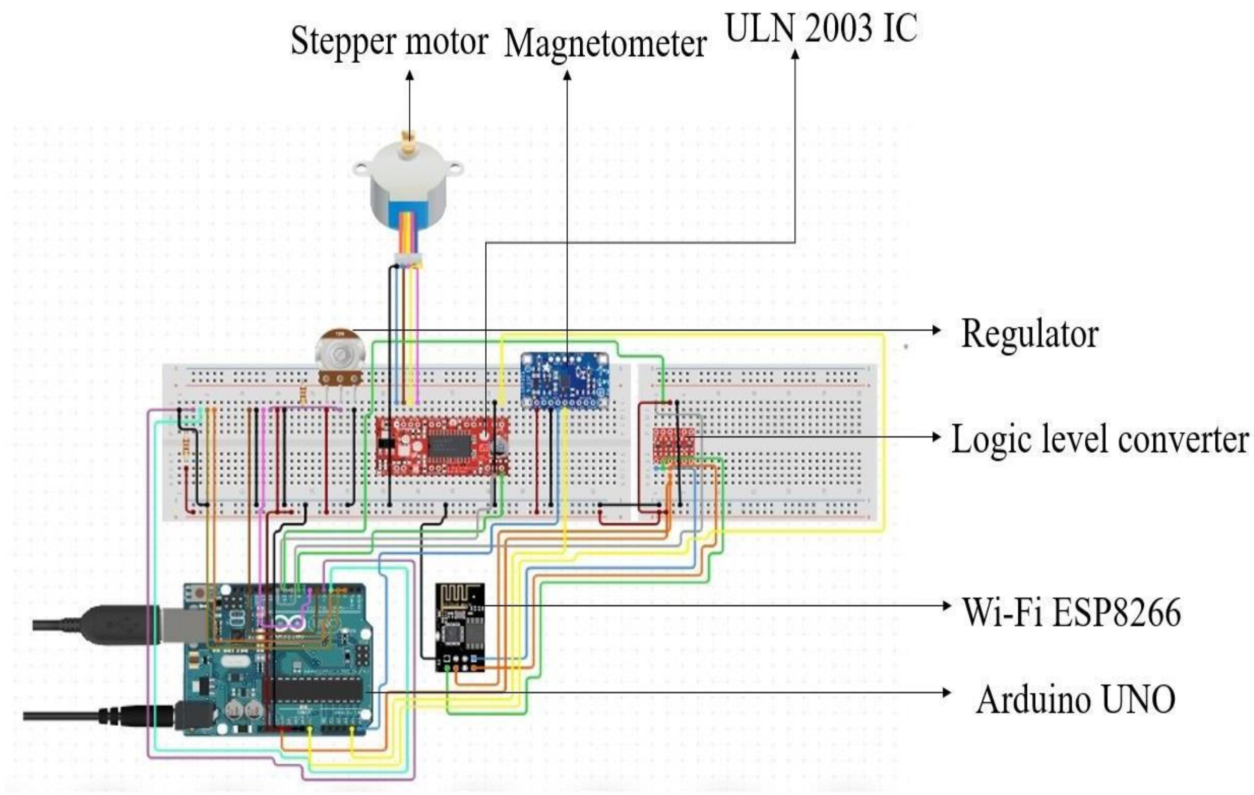


Figure 6.1: Schematic diagram of the project

The Arduino UNO is dumped with a desired code and is connected to the circuit with the help of the USB cable type B. The Arduino is now connected to the logic level converter to converter the user understandable(dumped), into the system understandable code to be executed. The A3, A4, A5 pins are considered as the connection of input and output of the X-axis, Y-axis and the Z-axis respectively. Now the logic level converter is connected to the wi-fi module and takes the input from it, which acts as a two-way wi-fi connection. The wi-fi is turned ON and the Arduino output pins are connected to the input port of the ULN2003IC, which performs the electrical form to the mechanical form of energy to be provided to the stepper motor and vice-verse from the stepper motor to the system. The regulator and capacitor are used for the stepping down of the voltage. The magnetometer is handled as the moving object to determine the direction of the antenna movement, as it moves the signal is sent to the Arduino, where it decides the angle of the change in the rotation and varies accordingly. The output pins of the ULN2003IC are connected to the stepper motor and give the appropriate angle of rotation to the stepper motor to rotate.

Flow chart and description

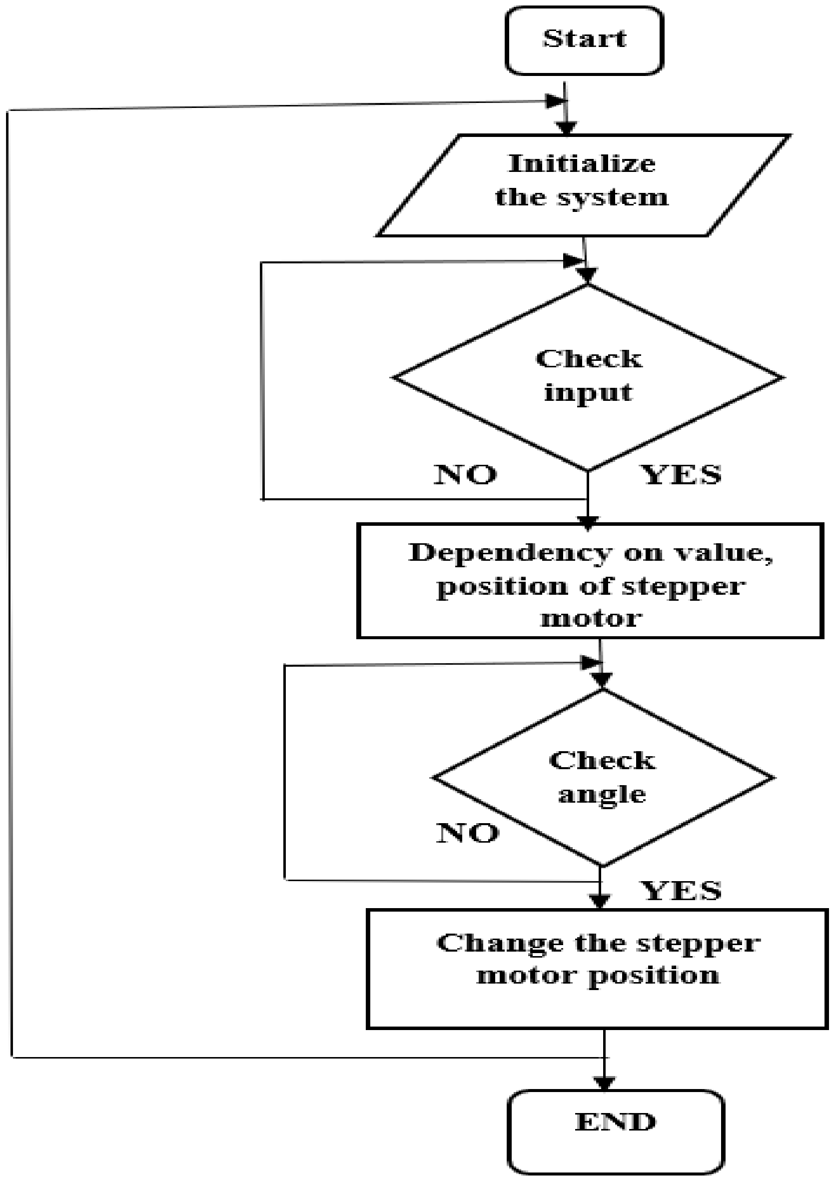


Figure 6.2: Flow chart of the code

- STEP-1: The system is turned ON and initialised.
- STEP-2: The input is checked given either from the magnetometer or the mobile application (Telnet). If No input, it returns back to the initialised stage. If YES input, then it moves to the next stage.
- STEP-3: Moving on to the next step, depending on the value of the input the system verifies the position of the stepper motor, whether to rotate or not.
- STEP-4: The angle of input is checked with the angle of stepper motor. If NO input change, the system returns to the step-3. If YES input change, it moves on to the step-5

- STEP-5: The stepper motor changes the amount of angle given as the input buy the user.
- STEP-6: The system then ends the program of execution and returns to the initialised position of the system simultaneously.

Output

There are 2 ways of varying the antenna position with an angle as the output:

1. From the magnetometer
2. From the mobile application

1. From the magnetometer

The magnetometer is kept in a constant position considering it as the Z-axis and when it is moved, there 2 cases occurring:

- When the magnetometer is moved in the forward direction (X-axis direction), the stepper motor moves in the clockwise direction.
- When the magnetometer is moved in the backward direction (Y-axis direction), the stepper motor moves in the anti- clockwise direction.

2. From the mobile application

There mobile TELNET application, the user can operate and rotate the antenna in 2 cases:

- When the user gives * <degree of angle change> then the antenna is rotated in the clockwise direction.

  Example: *16, the antenna is moved with 16 degrees in clockwise direction.
- When the user gives #<degree of angle change> then the antenna is rotate in the anti-clockwise direction.

  Example: #32, the antenna is moved with 32 degrees in anti-clockwise direction.

CHAPTER-7

RESULTS

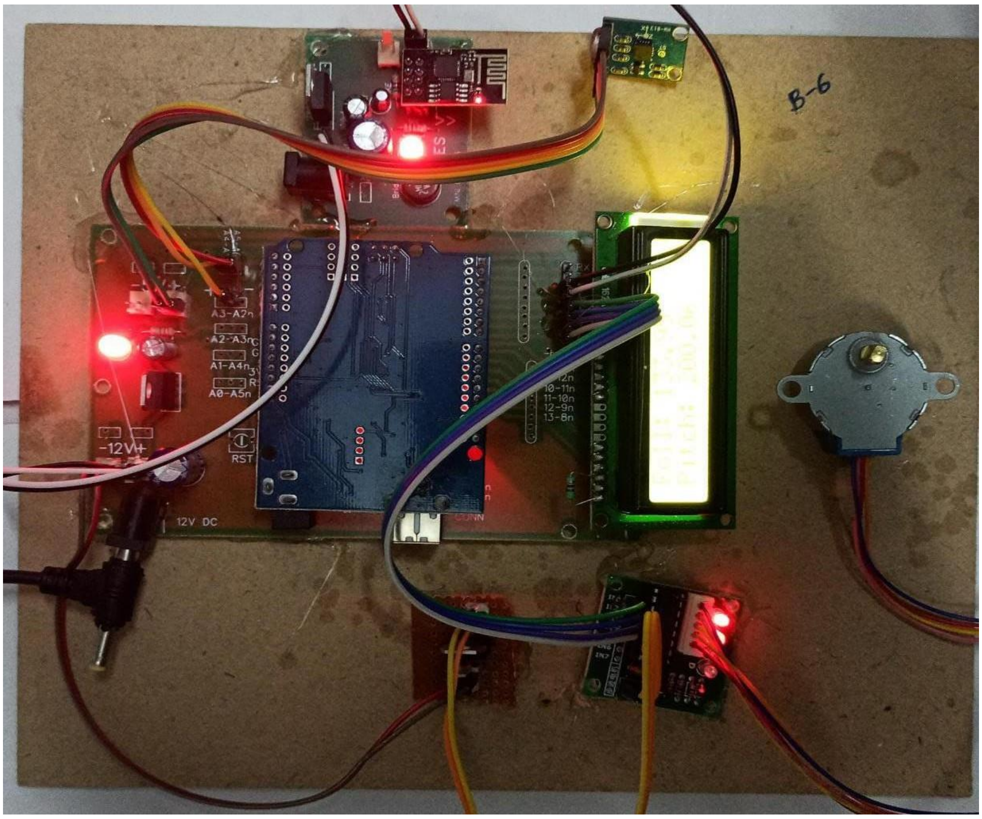


Figure 7.1: The circuit kit of the project

The Arduino UNO is connected to the system with a required code dumped in it. The LCD is displaying the variation of the angle and the direction of the movement. The stepper motor is rotated accordingly. The wi-fi module is providing the system with the wireless communication task like the mobile hotspot in our daily life.

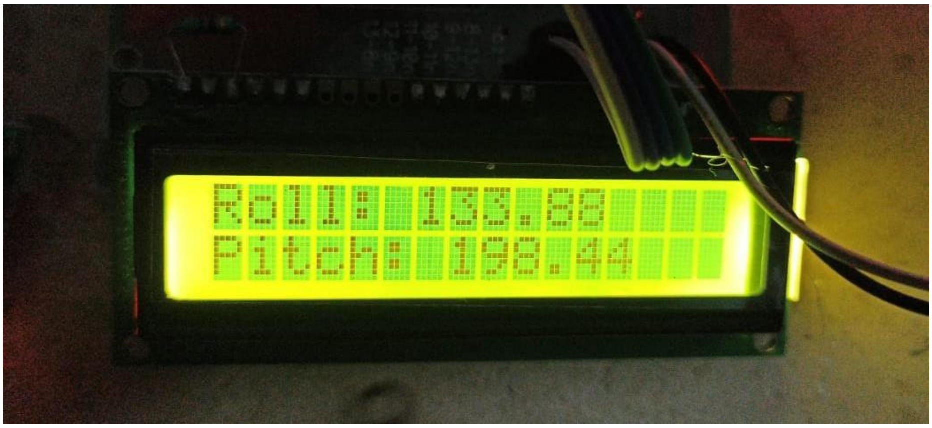


Figure 7.2: LCD displaying the angle

Here, the roll refers to the angle after the rotation in a particular direction with the reference being the Z-axis and the pitch refers to the total angle of rotation of the Z-axis when moved in other direction.

Figure 7.3: LCD displaying the direction of rotation

The position of the antenna after rotation being stable is being determined and displayed on the LCD and the direction of the antenna facing is also being shown up on the LCD for a brief report of the user understanding.

CHAPTER-8

CONCLUSION

Conclusion

The IOT is a latest and emerging technology with great significance in every domain of life. The proposed system describes the implementation of latest technologies in antenna positioning. This model allows for monitoring antenna direction as well as transmitting new coordinates to position the antenna and motor appropriately positions the antenna accordingly. IoT and microcontrollers are the most common and important technical concepts in every field of life. IoT based antenna positioning system can be beneficial in remote areas.

The Automatic Antenna Positioning System primarily functions to spot the presence of any signal. The automatic antenna positioning system is also used to position the antenna without any human interface it automatically positions itself depending upon the received signal strength. The antenna position mainly depends upon the precision of the motor. So as to beat the problem of adjusting manually, this planned system helps in adjusting the position of the antenna.

Future enhancements

- AI can be a big player here in this field. Moreover, new self-defence mechanisms will securitise IoT as well. All these connected devices will make IoT robust. Also, confirming some standard communication protocols will help in betterment of IoT security.
- The future of IoT has the potential to be limitless. Advances to the industrial internet will be accelerated through increased network agility, and the capacity to deploy, automate, orchestrate and secure diverse use cases at hyperscale.
- This may increase the need of the big data into play. Where every small detail of the IOT devices and usage is stored.
- Modification in the motors and the sensor to grasp maximum amount of the signal from every end.
- The maximum providence of the signal in a higher rate to meet the user’s need.
- The increase in the new variant of the element in the rotation of the antenna.

Code of execution

```
#include<SoftwareSerial.h>
SoftwareSerial mySerial(2,3);

#include<LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);

#include <Stepper.h>
double stepsPerRevolution = 500;
Stepper myStepper(stepsPerRevolution, 4, 6, 5, 7);  // Pin inversion to make the
library work

#include <math.h>
const int x_out = A3; /* connect x_out of module to A1 of UNO board */
const int y_out = A4; /* connect y_out of module to A2 of UNO board */
const int z_out = A5; /* connect z_out of module to A3 of UNO board */

char res[130];
char buff[130];

void sendwifi(String chr,unsigned int len)
{
  mySerial.print("AT+CIPSEND=0,");
  mySerial.println(len-1);
  delay(2000);
  mySerial.print(chr);
  delay(2000);
}

char t;

void setup()
{
  lcd.begin(16,2);
  lcd.clear();lcd.setCursor(0,0);
  lcd.print("Welcome");
  delay(1000);
  Serial.begin(9600);
  mySerial.begin(115200);
  myStepper.setSpeed(5);
  delay(200);
    mySerial.print("AT\r\n");

delay(1000);
mySerial.print("ATE0\r\n");
delay(1000);
mySerial.print("AT+CWMODE=3\r\n");
delay(1000);
mySerial.print("AT+CWSAP=\"load\",\"project1235\",5,0\r\n\r\n");
delay(1000);
mySerial.print("AT+CIPMUX=1\r\n");
delay(1000);
mySerial.print("AT+CIPSERVER=1,23\r\n");
delay(1000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("WATING FOR LINK");
while(1)
{
  if(mySerial.available())
  {
     //if(Esp.find("0,LINK"))
    if(mySerial.find("0,CONNECT"))
```

```
        {
            lcd.clear();lcd.setCursor(0, 0);lcd.print("LINK FOUND");
          break;
        }

      }
    }
    delay(1000);
    sendwifi("WELCOME \r\n",11);
    delay(500);
    delay(1000);
}
char tt;

void loop()
{
  int x_adc_value, y_adc_value, z_adc_value;
  double x_g_value, y_g_value, z_g_value;
  double roll, pitch, yaw;
  x_adc_value = analogRead(x_out); /* Digital value of voltage on x_out pin */
  y_adc_value = analogRead(y_out); /* Digital value of voltage on y_out pin */
  z_adc_value = analogRead(z_out); /* Digital value of voltage on z_out pin */

  x_g_value = ( ( ( (double)(x_adc_value * 5)/1024) - 1.65 ) / 0.330 ); /*
Acceleration in x-direction in g units */
  y_g_value = ( ( ( (double)(y_adc_value * 5)/1024) - 1.65 ) / 0.330 ); /*
Acceleration in y-direction in g units */
  z_g_value = ( ( ( (double)(z_adc_value * 5)/1024) - 1.80 ) / 0.330 ); /*
Acceleration in z-direction in g units */

  roll = ( ( (atan2(y_g_value,z_g_value) * 180) / 3.14 ) + 180 ); /* Formula for
roll */
  pitch = ( ( (atan2(z_g_value,x_g_value) * 180) / 3.14 ) + 180 ); /* Formula for
pitch */
  //yaw = ( ( (atan2(x_g_value,y_g_value) * 180) / 3.14 ) + 180 ); /* Formula for
yaw */
  /* Not possible to measure yaw using accelerometer. Gyroscope must be used if yaw
is also required */


Serial.print("Roll = ");
Serial.print(roll);
Serial.print("\t");
Serial.print("Pitch = ");
Serial.print(pitch);
Serial.print("\n\n");
delay(1000);
lcd.clear();lcd.setCursor(0,0);lcd.print("Roll: ");lcd.print(roll);
lcd.setCursor(0,1);lcd.print("Pitch: ");lcd.print(pitch);
delay(1000);

if(mySerial.available()> 0)
  {
    tt = mySerial.read();

if( tt == '#')
{
 lcd.clear();lcd.setCursor(0,0);lcd.print("Pos: Y Axis");
  lcd.setCursor(0,1);lcd.print("UP Side");
```

```
    delay(500);
    myStepper.step(-stepsPerRevolution);
    delay(3000);
    myStepper.step(0);
  }
  if( tt == '*')
  {
   lcd.clear();lcd.setCursor(0,0);lcd.print("Pos: X Axis");
    lcd.setCursor(0,1);lcd.print("Left Side");
    delay(500);
    myStepper.step(stepsPerRevolution);
    delay(3000);
    myStepper.step(0);
  }
 }
  if(roll<=300 && pitch<=20)
  {
    lcd.clear();lcd.setCursor(0,0);lcd.print("Pos: X Axis");
    lcd.setCursor(0,1);lcd.print("Left Side");
    delay(500);
    sendwifi("Axis Changed.. \r\n",18);
    delay(100);
    sendwifi("Positioning to Left Side \r\n",28);
    delay(100);
    Serial.println("Clockwise");
    myStepper.step(stepsPerRevolution);
    delay(2000);
    myStepper.step(0);
  }
  if(roll<=90 && pitch<=170)
  {
    lcd.clear();lcd.setCursor(0,0);lcd.print("Pos: Y Axis");
    lcd.setCursor(0,1);lcd.print("Up Side");
    delay(500);
    sendwifi("Axis Changed.. \r\n",18);
    delay(100);
    sendwifi("Positioning to Up Side \r\n",26);
    delay(100);
    myStepper.step(-stepsPerRevolution);
    delay(2000);
    myStepper.step(0);
  }
}
```

REFERENCES

[1] Li Chunqing, Zhang Jiancheng, “Research of ZigBee's data security and protection”, International Forum on Computer Science-Technology and Applications 2009, IEEE, 2009, pp 298 - 302

[2]https://www.tutorialspoint.com/antenna_theory/antenna_theory_fundamentals.htm

[3] Patrick Kinney, “ZigBee technology: wireless control that simply works”, Communications design conference, vol 2, 2 October 2003. retrieved 2011, from http://www.zigbee.org/en/press_kits/latest/documents/white_papers/wp_zigbeetech wireless_final.pdf

[4] Bin Yang, " Study on security of wireless sensor network based on ZigBee standard." International Conference on Computational Intelligence and Security, IEEE, 2009, pp 426 - 430.

[5] Sagarika Pal, Niladri Tripathy, “Remote Position Control System of Stepper Motor Using DTMF Technology”, International Journal of Control and Automation.

[6] Shubham Pathak, Sankalp P Taksande, Tanmay Waghmare, V. S. Nandanwar, “Satellite Dish Positioning Control by Geared Motor Using RF Module”, International Journal of Advanced Research in Science, Engineering and Technology.

[7] www.dishpointer.com

[8] www.satpointer.com

[9] Alan Bensky, “Wireless positioning and applications (Third edition)”, British library Publications.

[10] Microcontroller Based Wireless Automatic Antenna Positioning System Surya Deo Choundary, PankajRai, Arvind Kumar, Irshad Alam International Journal of Electronics and Computational System.

[11] Design and implementation of a positioning system for patch antenna IEEE International autumn meeting on NOV 20133.

[12] Wireless node localization based on RSSI using a rotating antenna on a mobile robot positioning, Navigation and communication,2009.

[13] “Microcontroller Based Wireless Automatic Antenna Positioning System” By Surya Deo Choudhary, Pankajrai, Aravind Kumar, Irshad Alam, International Journal of Electrical and Computational Systems, IJEECS ISSN 2348- 117X Volume 3, Issue 6.

[14] “Special Issues on Satellite Communications Systems and Services”, IEEE Communications Magazines, November 1991.

[15] “Remote Alignment of Dish Positioning by Android Application” by Prajwal Basnet, Pranjal Grover Preeti Pannu, International Journal of Engineering Research & Management Technology, ISSN: 2348-4039, March- 2015 Volume 2, Issue-2.

[16] Daniel Babitch, “Automatic Antenna Pointing System based on Global Positioning System (GPS) Attitude Information” U.S. Patent: 5347286, September, 13, 1994.

[17] David J. Kuether’s, “Antenna positioning system and method for simultaneous reception of signals from a plurality of satellites”, U.S. Patent: 7016643 B1, March, 21, 2006.

[18] S.Katoch, H.Jotwani, S.Pani, A.Rajawat, "A Compact Dual Band Antenna for IoT Applications", published in IEEE ,pp1594-1597, 2015.

[19] R.L Sturdivant, “Systems Engineeringof a Terabit Elliptic Orbit satellite”,published in IEEE,volume 4,pp 9941-9949 Sep 2016.

[20] Me Me Kyaw Oo, Chaw Myat New, & Hla Myo Tun “Satellite Dish Positioning Control by DC Motor Using IR Remote Control” in International Journal of Electronics and Computer Science Engineering on pages 199-207.

[21] Rahane Suraj Dildar, Mhaske Shital Arun, Shingate Sujata Rajendra, Prof.S.B.Mandalik.”Design of Advanced Antenna Positioning System.

[22] Surya Deo Choudhary, Pankaj Rai, Arvind Kumar, Irshad Alam. “Microcontroller Based Wireless Antenna Positioning System”.

[23] Prajwal Basnet, Pranjal Grover, Preeti Pannu. “Remote Alignment of Dish Positioning By Android Application.

[24] “Integrated Electronics, Analog And Digital Circuits And Systems” By Jacob Millman And Christos C. Halkias,Tata Mcgraw-Hill Edition.

[25] Ghoshal, Subrata. Embedded systems and robots: projects using the 8051 microcontroller, 1st edition, 2009, cengage learning india pvt. Ltd.

A

MAJOR PROJECT REPORT

On

RFID BASED ATTENDANCE SYSTEM WITH SMS NOTIFICATION

Submitted By

1) P. CHAITANYA CHARY (18K85A0413)

2) D. NEERAJ KUMAR (18K85A0414)

3) M. RATNAKAR REDDY (18K85A0415)

4) MD AZAM ALI (17K81A0496)

In Partial Fulfilment for The Award of The Degree

Of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS & COMMUNICATION ENGINEERING

Under the Guidance of

MS. B. SHINY SUCHARITHA

Assistant Professor

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

St. MARTIN'S ENGINEERING COLLEGE

(An UGC Autonomous Institute)

Dhulapally, Secunderabad-500 100.

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the major-project work entitled "**RFID Based Attendance System with SMS Notification**" is a bonafide work carried out by **P. Chaitanya Chary (18K85A0413), D. Neeraj Kumar (18K85A0414), M. Ratnakar Reddy (18K85A0415), Mohammed Azam Ali (17K81A0496)** in partial fulfilment of the requirements for the degree of **Bachelor of Technology in Electronics & Communication Engineering** by the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2020-21.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree.

INTERNAL GUIDE

HEAD OF THE DEPARTMENT, ECE

Ms.B.Shiny Sucharitha,
Assistant Professor,
Department of ECE.

Dr.B.Hari Krishna,
Professor.

Internal Examiner

External Examiner

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **D. NEERAJ KUMAR** WITH ROLL NO.**18K85A0414, M RATNAKAR REDDY** WITH ROLL NO.**18K85A0415, MOHAMMED AZAM ALI** WITH ROLL NO.**17K81A0496, P. CHAITANYA CHARY** WITH ROLL NO.**18K85A0413**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY**, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY**.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**RFID BASED ATTENDANCE SYSTEM WITH SMS NOTIFICATION**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. ★Kompally-500014★

ORUGANTI VENKAT
DIRECTOR
TRAININGS &
PLACEMENTS LASYA
IT SOLUTIONS PVT
LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of '**Bachelor of Technology in Department of Electronics and Communication Engineering**', session: 2018 – 2021, **St. Martin's Engineering College**, hereby declare that the project work entitled '**RFID Based Attendance System With SMS Notification**' is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

P. Chaitanya Chary	**D. Neeraj Kumar**	**M. Ratnakar Reddy**	**Md. Azam Ali**
(18K85A0413)	**(18K85A0414)**	**(18K85A0415)**	**(17K81A0496)**

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr. P. Santosh Kumar Patra**, **our Beloved Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr. B. Hari Krishna, Head &Professor**, the Dept. of Electronics and Communication Engineering for his/her advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to Our Department **Project Coordinator Mr. K. Karthik** for his/her involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

Finally, we are highly grateful and obliged to our college level **Project Guide Ms. B. Shiny Sucharitha, Assistant Professor** in ECE for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. P. Chaitanya Chary
2. D. Neeraj Kumar
3. M. Ratnakar Reddy
4. Md. Azam Ali

ABSTRACT

Now a days due to easy availability of all the information on the internet, students are less motivated to attend the classes, due to which most of the students are unable to maintain minimum attendance. This work is to simplify the attendance recording system by using Radio Frequency Identification (RFID) technology. RFID based Attendance system with SMS notification System is developed to overcome the above stated problem.

The system will be developed by using GSM (Global System for Mobile communication) technology and google firebase database and android application support. The information from RFID Database handling System will be used for taking attendance and for sending SMS alerts also.

This System interacts with parents by sending messages. Therefore, the system functionality not only records the student attendance, but also sends alert SMS to their parents when the student is absent.

Data will be automatically sent to the Google Firestore Database Server by the ESP8266-01 Wi-Fi module which is connected to IP. when a proper connection is established with server, Android Application which will allow us to monitor and control the system. The Android Application gives the information about past and current Attendance.

Keywords- Internet of Things (IoT), Smart Attendance, RFID, GSM, Arduino Software, ESP8266-01 12E, Android Studio, Android Application and Google Firebase Database.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

CHAPTER – 1

INTRODUCTION

OVERVIEW OF THE PROJECT

The goal of this project is to simplify attendance recording system by using Radio Frequency Identification technology. In the system, RFID Based Attendance System has ability to uniquely identify and take attendance of the students, in this user having RFID card which is an IC chip that has unique code contained in it and only need to place their RFID tag on the RFID reader to take attendance, and RFID reader is interfacing with Arduino Uno

Hence, it is a very time efficient system. Attendance will be taken if the RFID card ID matches the tagged ID which is predefined in memory. Otherwise, an error message will be displayed on LCD. And this attendance will send to the authorised contact which already predefined in memory via GSM module

The main aim of this paper is to design and implement an efficient Attendance system through which the required parameters are monitored remotely using internet and the data gathered ais stored in the google cloud database (Firebase) and to project the estimated trend on the Android application. A solution to simplify attendance recording system using wireless embedded computing system is proposed in this paper. This data is live updated to be viewed on the online server system.

Fig. 1.1: Overview of the project

OBJECTIVES OF THE PROJECT

The main objective of the RFID-based Attendance System project is to take the attendance of students or employees. The microcontroller does the task of storing the attendance of the respective person in the microcontroller memory. To demolish the flaws associated with the manual attendance system.

The immense potential of RFID based systems has been identified and explained in literature earlier. It will open the door to applications which were unthought of a few years ago. Also, the evolution of the technology and role of individual components has been explained in detail

The use of Wi-Fi module in RFID based systems has also been shown. Such a system significantly improves the current manual process of taking attendance in the University Environment. It also promotes a fully automated approach in monitoring the students in the Campus

SCOPE OF THE PROJECT

Main objective of RFID based Attendance System project is to take the attendance of students or employees. Microcontroller does the task of storing the attendance of the respective person in the Microcontroller memory. The existing attendance system is manual, and it is taken on paper, and it consumes lot of time.

Overcome the old attendance system which the attendance has to be taken manually. Design a database that supports such a system that has been mentioned earlier. Develop a smart attendance system that be implemented in classes, laboratory etc. by combining the software with the proposed hardware.

MATERIAL REQUIREMENT

HARDWARE REQUIREMENT

1. Arduino UNO R3
2. RC522 RFID Reader Module
3. SIM800L GSM Module
4. ESP8266-01 Wi-Fi Module
5. 16X2 I2C LCD Module
6. Power Adapter

SOFTWARE REQUIREMENT

1. Arduino IDE
2. Android Studio
3. Google Firebase Database CLI

INTRODUCTION TO EMBEDDED SYSTEMS

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor, or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.

The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes. The applications software on such processors is sometimes referred to as firmware

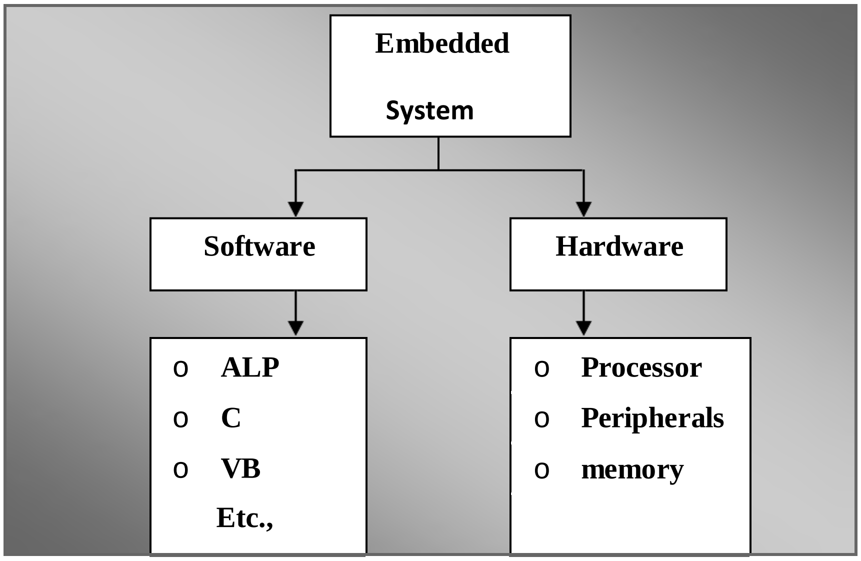


Fig 1.5 Block diagram of Embedded System

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

Memory: It is used to store data or address.

Peripherals: These are the external devices connected

Processor: It is an IC which is used to perform some task

Applications of embedded systems

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

Processors are classified into four types like:

- Micro Processor (μp)
- Micro controller (μc)
- Digital Signal Processor (DSP)
- Application Specific Integrated Circuits (ASIC)

Micro Processor (μp):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Three basic characteristics differentiate microprocessors:

- **Instruction set**: The set of instructions that the microprocessor can execute.
- **Bandwidth :** The number of bits processed in a single instruction.
- **Clock speed :** Given in megahertz (MHz), the clock speed determines how many instructions per second the processor can execute.

In both cases, the higher the value, the more powerful the CPU. For example, a 32-bit microprocessor that runs at 50MHz is more powerful than a 16-bit microprocessor that runs at 25MHz. In addition to bandwidth and clock speed, microprocessors are classified as being either RISC (reduced instruction set computer) or CISC (complex instruction set computer).

A microprocessor has three basic elements, as shown above. The ALU performs all arithmetic computations, such as addition, subtraction and logic operations (AND, OR, etc). It is controlled by the Control Unit and receives its data from the Register Array. The Register Array is a set of registers used for storing data. These registers can be accessed by the ALU very quickly. Some registers have specific functions - we will deal with these later. The Control Unit controls the entire process. It provides the timing and a control signal for getting data into and out of the registers and the ALU and it synchronizes the execution of instructions (we will deal with instruction execution at a later date).

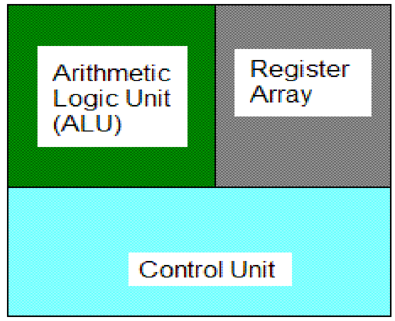


Figure 1.5.1 Three Basic Elements of a Microprocessor

Micro Controller (µc):

A microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the Microprocessors used in Personal computers or other general purpose applications.

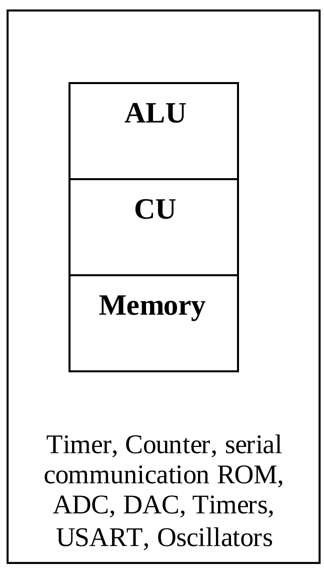


Figure 1.5.2 Block Diagram of Micro Controller (µc)

Digital Signal Processors (DSPs):

Digital Signal Processors is one which performs scientific and mathematical operation. Digital Signal Processor chips - specialized microprocessors with architectures designed specifically for the types of operations required in digital signal processing. Like a general-purpose microprocessor, a DSP is a programmable device, with its own native instruction code. DSP chips are capable of carrying out millions of floating point operations per second, and like their better-known general-purpose cousins, faster and more powerful versions are continually being introduced. DSPs can also be embedded within complex "system-on-chip" devices, often containing both analog and digital circuitry.

Application Specific Integrated Circuit (ASIC)

ASIC is a combination of digital and analog circuits packed into an IC to achieve the desired control/computation function

ASIC typically contains

- CPU cores for computation and control
- Peripherals to control timing critical functions
- Memories to store data and program
- Analog circuits to provide clocks and interface to the real world
- I/Os to connect to external components like LEDs, memories, etc.

Computer Instruction Set

There are two different types of computer instruction set there are:

1. RISC (Reduced Instruction Set Computer) and
2. CISC (Complex Instruction Set computer)

Reduced Instruction Set Computer (RISC)

A RISC (reduced instruction set computer) is a microprocessor that is designed to perform a smaller number of types of computer instruction so that it can operate at a higher speed (perform more million instructions per second, or millions of instructions per second). Since each instruction type that a computer must perform requires additional transistors and circuitry, a larger list or set of computer instructions tends to make the microprocessor more complicated and slower in operation.

Besides performance improvement, some advantages of RISC and related design improvements are:

- A new microprocessor can be developed and tested more quickly if one of its aims is to be less complicated.
- Operating system and application programmers who use the microprocessor's instructions will find it easier to develop code with a smaller instruction set.
- The simplicity of RISC allows more freedom to choose how to use the space on a microprocessor.

Higher-level language compilers produce more efficient code than formerly because they have always tended to use the smaller set of instructions to be found in a RISC computer.

RISC characteristics

- **Simple instruction set:**

In a RISC machine, the instruction set contains simple, basic instructions, from which more complex instructions can be composed.

- **Same length instructions.**

  Each instruction is the same length, so that it may be fetched in a single operation.

- **1machine-cycle instructions.**

  Most instructions complete in one machine cycle, which allows the processor to handle several instructions at the same time. This pipelining is a key technique used to speed up RISC machines.

Complex Instruction Set Computer (CISC)

CISC, which stands for **Complex Instruction Set Computer,** is a philosophy for designing chips that are easy to program and which make efficient use of memory. Each instruction in a CISC instruction set might perform a series of operations inside the processor. This reduces the number of instructions required to implement a given program, and allows the programmer to learn a small but flexible set of instructions.

The Advantages of CISC

At the time of their initial development, CISC machines used available technologies to optimize computer performance.

- Microprogramming is as easy as assembly language to implement, and much less expensive than hardwiring a control unit.
- The ease of micro-coding new instructions allowed designers to make CISC machines upwardly compatible: a new computer could run the same programs as earlier computers because the new computer would contain a superset of the instructions of the earlier computers.
- As each instruction became more capable, fewer instructions could be used to implement a given task. This made more efficient use of the relatively slow main memory.
- Because micro program instruction sets can be written to match the constructs of high-level languages, the compiler does not have to be as complicated.

The Disadvantages of CISC

Still, designers soon realized that the CISC philosophy had its own problems, including:

- Earlier generations of a processor family generally were contained as a subset in every new version --- so instruction set & chip hardware become more complex with each generation of computers.
- So that as many instructions as possible could be stored in memory with the least possible wasted space, individual instructions could be of almost any length---this means that different instructions will take different amounts of clock time to execute, slowing down the overall performance of the machine.
- Many specialized instructions aren't used frequently enough to justify their existence --- approximately 20% of the available instructions are used in a typical program.
- CISC instructions typically set the condition codes as a side effect of the instruction. Not only does setting the condition codes take time, but programmers have to remember to examine the condition code bits before a subsequent instruction changes them.

Memory Architecture

There two different type's memory architectures there are:

- Harvard Architecture
- Von-Neumann Architecture

Harvard Architecture

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

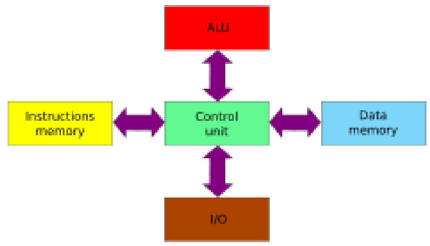


Figure 1.5.8 Harvard Architecture

Modern uses of the Harvard architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where trade-offs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

- Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behavior must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.
- Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by

Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Von-Neumann Architecture

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. The **von Neumann architecture** is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A **stored-program** digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

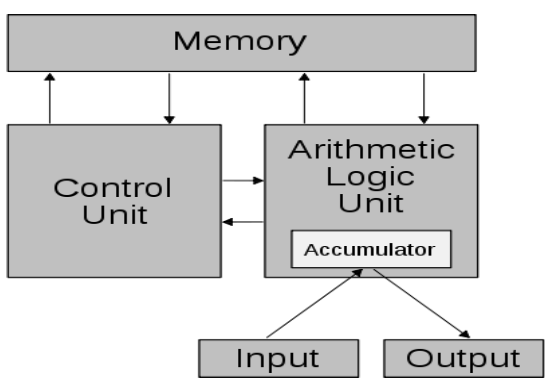


Figure 1.5.9 Schematic of the Von-Neumann Architecture

Basic Difference between Harvard and Von-Neumann Architecture

- The primary difference between Harvard architecture and the Von Neumann architecture is in the Von Neumann architecture data and programs are stored in the same memory and managed by the same information handling system.
- Whereas the Harvard architecture stores data and programs in separate memory devices and they are handled by different subsystems.
- In a computer using the Von-Neumann architecture without cache; the central processing unit (CPU) can either be reading and instruction or writing/reading data to/from the memory. Both of these operations cannot occur simultaneously as the data and instructions use the same system bus.
- In a computer using the Harvard architecture the CPU can both read an instruction and access data memory at the same time without cache. This means that a computer with Harvard architecture can potentially be faster for a given circuit complexity because data access and instruction fetches do not contend for use of a single memory pathway.
- Today, the vast majority of computers are designed and built using the Von Neumann architecture template primarily because of the dynamic capabilities and efficiencies gained in designing, implementing, operating one memory system as opposed to two. Von Neumann architecture may be somewhat slower than the contrasting Harvard Architecture for certain specific tasks, but it is much more flexible and allows for many concepts unavailable to Harvard architecture such as self-programming, word processing and so on.

CHAPTER - 2

LITERATURE SURVEY

LITERATURE REVIEW ON RESEARCH AREA

Title : 1 RFID Attendance Monitoring System

RFID Attendance Monitoring System (2013), this system is an electronic device which is use in attendance management system by faculty members. It provides robust, secure and automatic attendance administration in faculties.

Radio-Frequency Identification Attendance Monitoring has programmed RFID software specially for optimizing attendance tracking. Thus, combined RFID hardware with this software, it automates the whole system.

Title :2 RFID Technology Based Attendance Management System

According to Sumita Nainan and et.al (2013), the primary aim of the research is to uniquely identify individual students based on their unique tag identifiers. The research should shower light on how scalable and efficient the system is. A systematic and serialised approach is required to solve this conundrum. The key characteristics of the application include; Perform automated attendance, Generate report of attendees for a particular course, Error free tag identifier detection, Easy scalability to incorporate more records, Integrity and security in data storage.

This paper concentrates on the principal purpose to overcome the human errors while recording student attendance and the creation of a data centric student attendance database system with an improved overall efficiency.

Title : 3 School of Information Technology Faculty Monitoring using Radio Frequency Identification (RFID)

According to Justin Lee and et.al (2013), this study was to improve the faculty monitoring of University of Baguio in School of Information Technology Department. This helps the Student Assistant to easily locate the Faculty Members if needed and can generate a summary report. This may also lessen the tardiness of Faculty Members. RFID systems also provide good personal security access to confidential data's.

REVIEW OF RELATED WORKS

A number of related works exist in literature, application of RFID Technology to different areas and specifically to the area of academic attendance monitoring problem. In [6], authors designed and implemented a model of a secured and portable embedded reader system to read the biometric data from the electronic passport. The authors attempted to solve problems of reliability, security and privacy in E-passports by authenticating holder online using Global System of Mobile Communications (GSM) network.

The GSM network is the main interface between identification centre and the e- passport reader. The communication data is protected between server and e-passport reader by using AES to encrypt data for protection while transferring through GSM network. Author in [5] reviewed the current research application of RFID to different areas with emphasis on application for supply chain management and developed a taxonomic framework to classify literature which enables swift and easy content analysis to help identify areas for future research. Authors in [9] reviewed the use of RFID in an integrated- circuit(IC) packaging house to resolve inventory transaction issues. His study suggests that RFID contributes significant improvements to the water receiving process and the inventory transaction process that reduce labour cost and man-made errors. In [10], an automated attendance management system was implemented both in electronic and mobile platform using stationary matrix AR 400 RFID reader with four circulatory polarized antennae and Symbol MC9000-G handheld RFID reader respectively. In the electronic platform, the attendance management system

DISADVANTAGES OF EXISTING SYSTEM

- Need to enter attendance data manually.
- Replication of attendance data may be a tedious work.
- There is always a risk of human error.
- Need spend a lot of time correcting these errors
- Can be easily manipulated.

PROPOSED SYSTEM

The proposed system uses the Radio Frequency Identification Technology (RFID) and interacts with the objects through wireless communication. The proposed model is further adaptable and distributive in nature to simplify attendance system. The implemented system consists of an Arduino UNO R3 as a main processing unit for the entire system and all the Sensing Modules and communication devices are connected with the microcontroller. The Sensing modules can be operated by the microcontroller to retrieve the data from them and analyse the data with data in memory.

RFID card which is an IC chip that has unique code contained in it and only need to place their RFID tag on the RFID reader. After sensing the data from RFID module devices, Microcontroller compares the data and identifies the user with the Associated user data such as Name, ID, Phone Number and Class Details. An alert will be sent to the authorized contact which already predefined in memory via GSM module.

The data will be automatically sent to the Google Firestore Database Server by the ESP8266-01 Wi-Fi module which is connected to IP. when a proper connection is established with server, Android Application which will allow us to monitor and control the system. The Android Application gives the information about past and current Attendance.

ADVANTAGES OF THE PROPOSED SYSTEM

- Fully Automated.
- No Human Error Occurs.
- Error Free Data and can be used for different purposes without duplication.
- Fast and Reliable.
- Low Power Consumption, Even works with Battery Supply.

CHAPTER – 3

PROJECT DESCRIPTION

BLOCK DIAGRAM

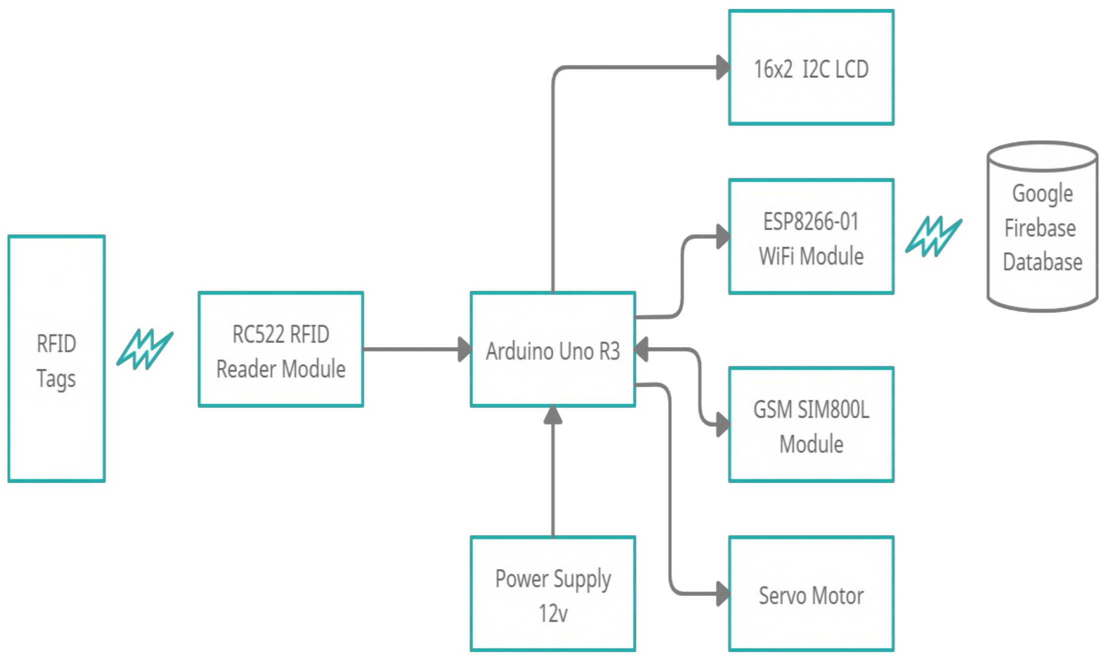


Fig. 3.1 Block Diagram of RFID Based Attendance System

SCHEMATIC DIAGRAM

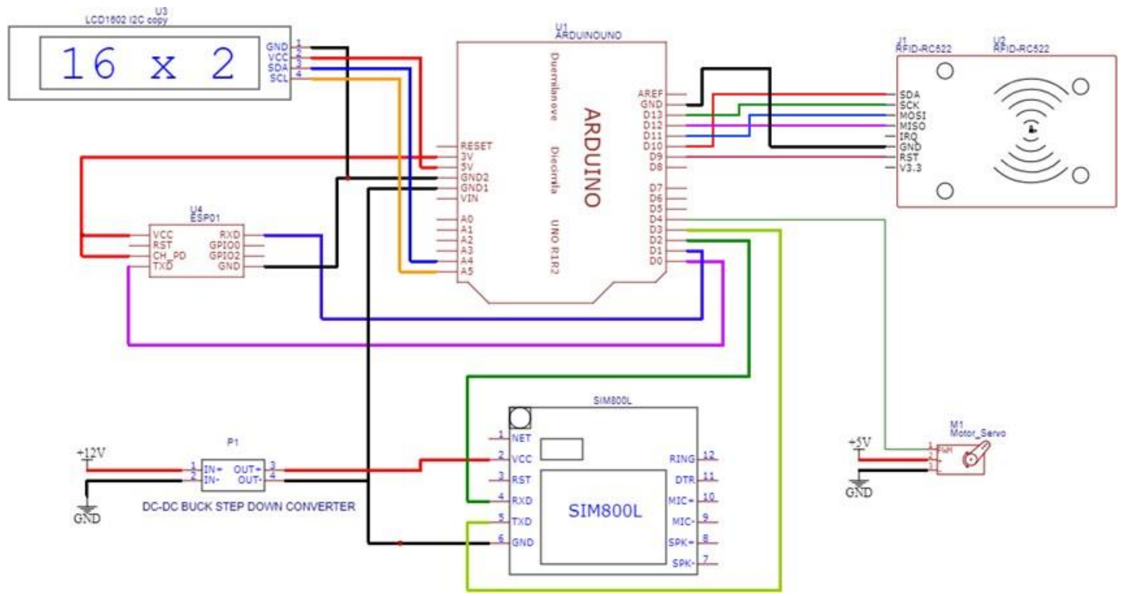


Fig. 3.2 Schematic Diagram of RFID Based Attendance System

SCHEMATIC EXPLANATION

The proposed system uses the Radio Frequency Identification Technology (RFID) and interacts with the objects through wireless communication. The proposed model is further adaptable and distributive in nature to simplify attendance system. The implemented system consists of an Arduino UNO R3 as a main processing unit for the entire system and all the Sensing Modules and communication devices are connected with the microcontroller. The Sensing modules can be operated by the microcontroller to retrieve the data from them and analyse the data with data in memory.

RFID card which is an IC chip that has unique code contained in it and only need to place their RFID tag on the RFID reader. After sensing the data from RFID module devices, Microcontroller compares the data and identifies the user with the Associated user data such as Name, ID, Phone Number and Class Details. An alert will be sent to the authorized contact which already predefined in memory via GSM module.

The data will be automatically sent to the Google Firestore Database Server by the ESP8266-01 Wi-Fi module which is connected to IP. when a proper connection is established with server, Android Application which will allow us to monitor and control the system. The Android Application gives the information about past and current Attendance.

Fig. 3.2.1 Working of RFID Based Attendance System

FLOWCHART OF THE RFID BASED ATTENDANCE SYSTEM

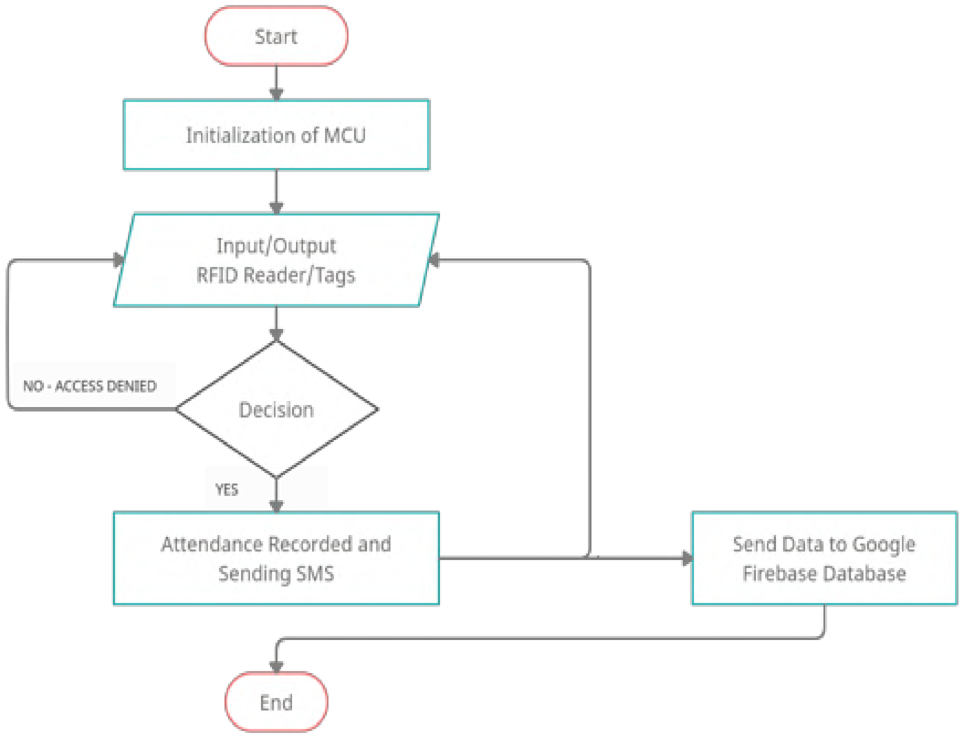


Fig. 3.3 Flowchart of RFID Based Attendance System

CHAPTER – 4

HARDWARE DESCRIPTION

ARDUINO UNO R3

Fig 4.1 Arduino UNO R3

The Arduino Uno is a microcontroller board based on the ATmega328 . It has 14 digital input/output pins, 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

Technical Specifications

- Microcontroller ATmega328
- Operating Voltage 5V
- Input Voltage 7-12V
- Input Voltage 6-20V
- Digital I/O Pins 14 (of which 6 provide PWM output)
- Analog Input Pins 6
- DC Current per I/O Pin 40 mA

- DC Current for 3.3V Pin 50 mA
- Flash Memory 32 KB of which 0.5 KB used by bootloader
- SRAM 2 KB EEPROM 1 KB Clock Speed 16 MH

Features

- High Performance, Low Power AVR® 8-Bit Microcontroller Family
- Advanced RISC Architecture
  - 131 Powerful Instructions
  - Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation─Up to 20 MIPS Throughput at 20MHz
  - On-chip 2-cycle Multiplier

- High Endurance Non-volatile Memory Segments
  - 4/8/16/32KBytes of In-System Self-Programmable Flash program memory
  - 256/512/512/1KBytes EEPROM─512/1K/1K/2KBytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C(1)
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read While-Write Operation

o Programming Lock for Software Security

- QTouch® library support
  - Capacitive touch buttons, sliders and wheels
  - QTouch and QMatrix™ acquisition
  - Up to 64 sense channels

- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode

- Real Time Counter with Separate Oscillator
- Six PWM Channels

- 8-channel 10-bit ADC in TQFP and QFN/MLF package λ
  - Temperature Measurement
- 6-channel 10-bit ADC in PDIP Package
  - Temperature Measurement
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Byte-oriented 2-wire Serial Interface (Philips I2C compatible)
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator
- Interrupt and Wake-up on Pin Change
- I/O and Packages
- Operating Voltage: 1.8 - 5.5V
- Temperature Range: -40°C to 85°C
- Speed Grade:
  - 0 - 4MHz @ 1.8 - 5.5V
  - 0 - 10MHz @ 2.7 - 5.5.V
  - 0 - 20MHz @ 4.5 - 5.5V
- Power Consumption at 1MHz, 1.8V, 25°C
- Active Mode: 0.2mA
- Power-down Mode: 0.1µA
- Power-save Mode: 0.75µA (Including 32kHz RTC)

PIN DESCRIPTION

For pin description of Arduino UNO, let us assume some basic numbering. Let the numbering begin with the RX Pin (D0). So, RX is Pin 1, TX is Pin 2, D2 is Pin 3 and so on. On the other side, NC is Pin 19, IOREF is Pin 20 etc. Overall, there are 32 pins on the Arduino UNO Board.

With this information, let us now see the pin description of Arduino UNO.

Pin Number	Pin Name	Description	Alternative Functions
1	RX / D0	Digital IO Pin 0 Serial RX Pin	Generally used as RX
2	TX / D1	Digital IO Pin 1 Serial TX Pin	Generally used as TX
3	D2	Digital IO Pin 2	
4	D3	Digital IO Pin 3	Timer (OC2B)
5	D4	Digital IO Pin 4	Timer (T0/XCK)
6	D5	Digital IO Pin 5	Timer (OC0B/T1)
7	D6	Digital IO Pin 6	
8	D7	Digital IO Pin 7	
9	D8	Digital IO Pin 8	Timer (CLK0/ICP1)
10	D9	Digital IO Pin 9	Timer (OC1A)
11	D10	Digital IO Pin 10	Timer (OC1B)

12	D11	Digital IO Pin 11	SPI (MOSI) Timer (OC2A)
13	D12	Digital IO Pin 12	SPI (MISO)
14	D13	Digital IO Pin 13	SPI (SCK)
15	GND	Ground	
16	AREF	Analog Reference	
17	SDA / D18	Digital IO Pin 18	I2C Data Pin
18	SCL / D19	Digital IO Pin 19	I2C Clock Pin
19	NC	Not Connected	
20	IOREF	Voltage Reference	
21	RESET	Reset (Active LOW)	
22	3V3	Power	
23	5V	+5V Output from regulator or +5V regulated Input	
24	GND	Ground	
25	GND	Ground	
26	VIN	Unregulated Supply	
27	A0	Analog Input 0	Digital IO Pin 14

28	A1	Analog Input 1	Digital IO Pin 15
29	A2	Analog Input 2	Digital IO Pin 16
30	A3	Analog Input 3	Digital IO Pin 17
31	A4	Analog Input 4	Digital IO Pin 18 I2C (SDA)
32	A5	Analog Input 5	Digital IO Pin 19 I2C (SCL)

Table 4.1.3 Arduino Uno Pinout

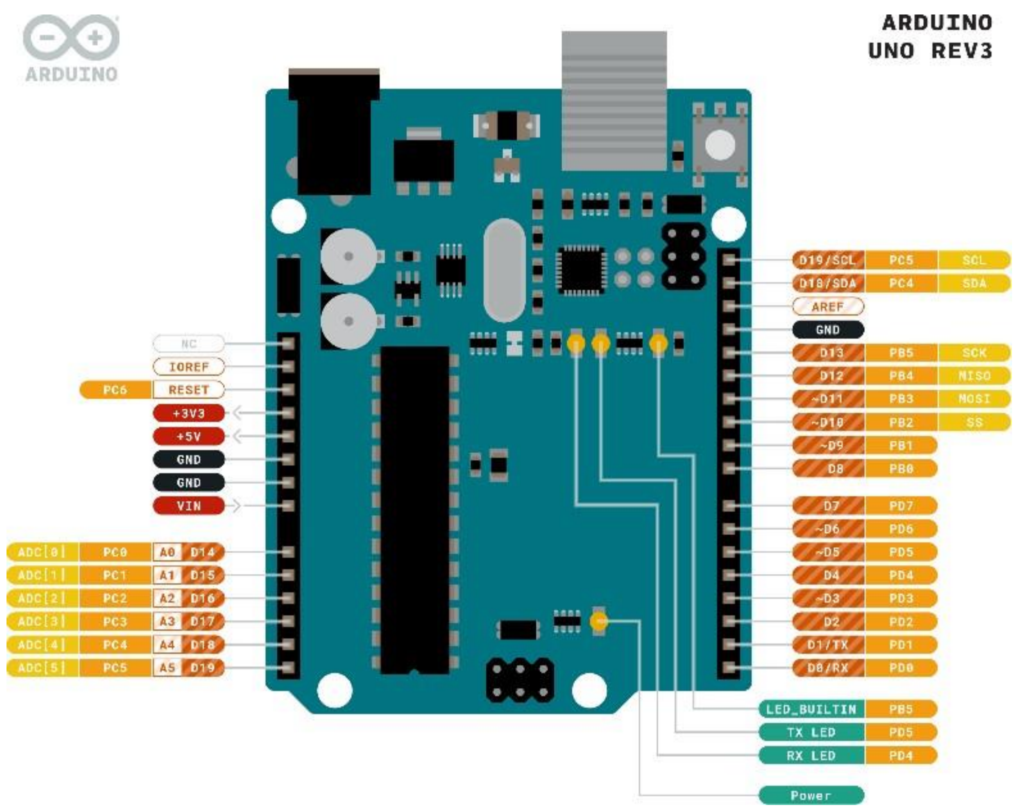


Fig 4.1.3 Arduino Uno PinOut Diagram

RC522 RFID Reader Module

The RC522 is a 13.56MHz RFID module that is based on the MFRC522 controller from NXP semiconductors. The module can supports I2C, SPI and UART and normally is shipped with a RFID card and key fob. It is commonly used in attendance systems and other person/object identification applications.

Fig 4.2: RC522 RFID Reader Module

The reader can communicate with a microcontroller over a 4-pin Serial Peripheral Interface (SPI) with a maximum data rate of 10Mbps. It also supports communication over I2C and UART protocols.The module comes with an interrupt pin. It is handy because instead of constantly asking the RFID module “is there a card in view yet? “, the module will alert us when a tag comes into its vicinity.

The operating voltage of the module is from 2.5 to 3.3V, but the good news is that the logic pins are 5-volt tolerant, so we can easily connect it to an Arduino or any 5V logic microcontroller without using any logic level converter.

RC522 Pinout Configuration

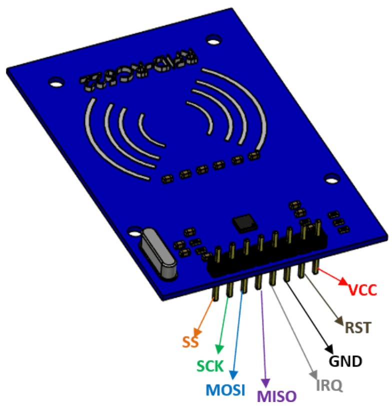


Fig 4.2.1: RC522 Pinout Configuration

Pin Number	**Pin Name**	**Description**
1	Vcc	Used to Power the module, typically 3.3V is used
2	RST	Reset pin – used to reset or power down the module
3	Ground	Connected to Ground of system
4	IRQ	Interrupt pin – used to wake up the module when a device comes into range
5	MISO/SCL/Tx	MISO pin when used for SPI communication, acts as SCL for I2c and Tx for UART.
6	MOSI	Master out slave in pin for SPI communication
7	SCK	Serial Clock pin – used to provide clock source
8	SS/SDA/Rx	Acts as Serial input (SS) for SPI communication, SDA for IIC and Rx during UART

RC522 Features

- Highly integrated analog circuitry to demodulate and decode responses
- Buffered output drivers to connect an antenna with minimum number of external components
- Supports ISO/IEC 14443A / MIFARE®
- Typical operating distance in Reader/Writer mode for communication to a ISO/IEC 14443A MIFARE® up to 50 mm depending on the antenna size and tuning
- Supports MIFARE® Classic encryption in Reader/Writer mode
- Supports ISO/IEC 14443A higher transfer speed communication up to 848 kbit/s
- Support of the MFIN / MFOUT
- Additional power supply to directly supply the smart card IC connected via MFIN / MFOUT
- Supported host interfaces
- SPI interface up to 10 Mbit/s
- I2C interface up to 400 kbit/s in Fast mode, up to 3400 kbit/s in High-speed mode
- serial UART in different transfer speeds up to 1228.8 kbit/s, framing according to the RS232 interface with voltage levels according pad voltage supply
- Comfortable 64 byte send and receive FIFO-buffer
- Flexible interrupt modes
- Hard reset with low power function
- Power-down mode per software
- Programmable timer
- Internal oscillator to connect 27.12 MHz quartz
- 2.5 - 3.3 V power supply
- CRC Co-processor
- Free programmable I/O pins
- Internal self-test

RC522 Specifications

- 13.56MHz RFID module
- Operating voltage: 2.5V to 3.3V
- Communication : SPI, I2C protocol, UART
- Maximum Data Rate: 10Mbps
- Read Range: 5cm
- Current Consumption: 13-26mA
- Power down mode consumption: 10uA (min)

Applications of RC522 Module

- Automatic billing systems
- Attendance systems
- Verification/Identification system
- Access control systems

SIM800L GSM Module

SIM800L is a miniature cellular module which allows for GPRS transmission, sending and receiving SMS and making and receiving voice calls. Low cost and small footprint and quad band frequency support make this module perfect solution for any project that require long range connectivity. After connecting power module boots up, searches for cellular network and login automatically. On board LED displays connection state (no network coverage - fast blinking, logged in - slow blinking).

Fig. 4.3 SIM800L GSM Module

SIM800 is a quad-band GSM/GPRS module that works on frequencies 850MHz GSM, 900MHz EGSM, 1800MHz DCS, and 1900MHz PCS.It also features GPRS multi-slot class 12/class 10 (optional), and supports CS-1, CS-2, CS-3, and CS-4 GPRS coding schemes.

It has one UART port. It also has one USB port that can be used for updating firmware and for debugging. Audio channels are also there, which include a microphone input and a receiver output.SIM800 has one SIM card interface.It integrates TCP/IP protocol.

SIM800 can be controlled/configured using simple AT commands. A host microcontroller can send AT commands over the UART interface and control the SIM800. It is operated on a supply in the range of 3.4 to 4.4V.

It can be used for sending/receiving messages, making calls, sending/receiving data over the internet, etc. This makes it useful for applications such as home automation, agriculture automation, etc.

An antenna is required to use the module for any kind of voice or data communications as well as some SIM commands. So, selecting an antenna could be a crucial thing. There are two ways you can add an antenna to your SIM800L module.

Fig. 4.3a SIM800L GSM Module with Helical Antenna

The first one is a Helical GSM antenna which usually comes with the module and solders directly to NET pin on PCB. This antenna is very useful for projects that need to save space but struggles in getting connectivity especially if your project is indoors.

Fig. 4.3b SIM800L GSM Module with 3dBi Antenna

The second one is any 3dBi GSM antenna along with a U.FL to SMA adapter which can be obtained online for less than $3. You can snap-fit this antenna to small u.fl connector located on the top-left corner of the module. This type of antenna has a better performance and allows putting your module inside a metal case – as long the antenna is outside

One of the most important parts of getting the SIM800L module working is supplying it with enough power.

Depending on which state it is in, the SIM800L can be a relatively power-hungry device. The maximum current draw of the module is around 2A during transmission burst. It usually will not pull that much, but may require around 216mA during phone calls or 80mA during network transmissions

SIM800L Specifications

- Quad-band 850/900/1800/1900MHz
- GPRS multi-slot class12 connectivity: max. 85.6kbps(down-load/up-load)
- GPRS mobile station class B
- Controlled by AT Command (3GPP TS 27.007, 27.005 and SIMCOM enhanced AT Commands)
- Supports Real Time Clock
- Supply voltage range 3.4V ~ 4.4V
- Supports A-GPS
- Supports 3.0V to 5.0V logic level
- Low power consumption, 1mA in sleep mode
- Compact size 23mm x 35mm x 5.6mm
- Micro SIM Card

SIM800L Pinout Configuration

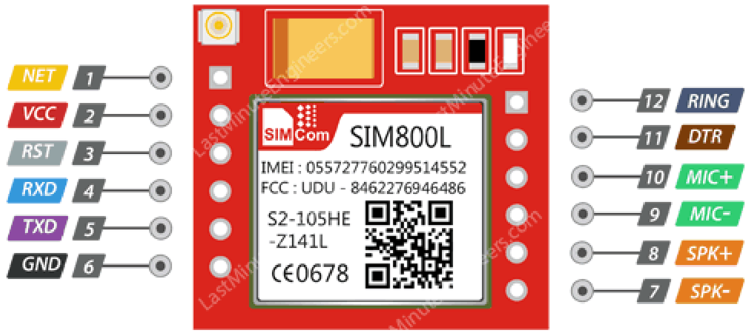


Fig. 4.3.2 SIM800L Pinout

- **NET** is a pin where you can solder Helical Antenna provided along with the module.
- **VCC** supplies power for the module. This can be anywhere from 3.4V to 4.4 volts. An external power source like Li-Po battery or DC-DC buck converters rated 3.7V 2A would work.
- **RST** (Reset) is a hard reset pin. If you absolutely got the module in a bad space, pull this pin low for 100ms to perform a hard reset.
- **RxD** (Receiver) pin is used for serial communication.
- **TxD** (Transmitter) pin is used for serial communication.
- **GND** is the Ground Pin and needs to be connected to GND pin on the Arduino.
- **RING** pin acts as a Ring Indicator. It is basically the 'interrupt' out pin from the module. It is by default high and will pulse low for 120ms when a call is received. It can also be configured to pulse when an SMS is received.
- **DTR** pin activates/deactivates sleep mode. Pulling it HIGH will put module in sleep mode, disabling serial communication. Pulling it LOW will wake the module up.
- **MIC±** is a differential microphone input. The two microphone pins can be connected directly to these pins.
- **SPK±** is a differential speaker interface. The two pins of a speaker can be tied directly to these two pins.

SIM800L Features

- Network support: Quad-Band 850/900/1800/1900 MHz – works on GSM networks in all countries across the world.
- TTL serial interface compatible with 3.3V and 5V MCU
- The small size is suitable for all kinds of sleeve belt type and embedded.
- Playing card slot, convenient change mobile phone card. Can side reserve replacement cell phone bayonet into the box.
- Using the IPX interface can arbitrarily change the antenna. The default PCB antenna greatly reduces & use the space.
- The board with gold plated board, more stable performance.
- When wiring neat guarantee board running more stable.
- This module adds a protection device, which can realize the prevention, high pressure, reduce the loss of the board, suitable for the project and learn

AT Commands of SIM800L Module

To send or read SMS from the module we need to set the module to SMS mode and then use the AT Commands for SMS.

- To set the GSM Module in SMS mode, use the following code.

  AT+CMGF=1

- Once it returns OK, then use the second command to read the SMS.

  AT+CMGR=1

- To Send an SMS, set the GSM Modem to SMS mode using 'AT+CMGF=1' and then use the following command.

  AT+CMGS="+91XXXXXXXX"

- The modem will return an arrow to enter the message (>). Type the message and hit enter.
- To exit from SMS Mode, we need to send 'CTRL' in ASCII Form.
  The '26' is the ASCII Character for 'CTRL'.

  println((char)26)

ESP8266-01 WI-FI MODULE

The ESP-01 ESP8266 Serial WIFI Wireless Transceiver Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a Wi-Fi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost-effective board with a huge, and ever growing, community.

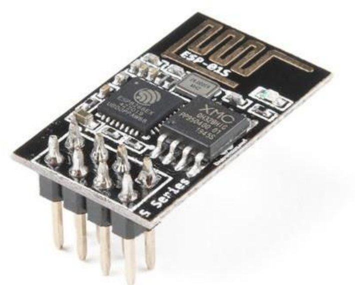


Fig. 4.4 ESP8266-01 Wi-Fi Mofule

This module has a powerful enough onboard processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existing interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions and requires no external RF parts.

There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support. In the Documents section below you will find many resources to aid you in using the ESP8266, even instructions on how to transform this module into an IoT (Internet of Things) solution!

Note: The ESP8266 Module is not capable of 5-3V logic shifting and will require an external Logic Level Converter. Please do not power it directly from your 5V dev board.

ESP-01 Wi-Fi module is developed by Ai-thinker Team. core processor ESP8266 in smaller sizes of the module encapsulates Tensilica L106 integrates industry-leading ultra-low power 32-bit MCU micro, with the 16-bit short mode, Clock speed support 80 MHz, 160 MHz, supports the RTOS, integrated Wi-Fi MAC/BB/RF/PA/LNA, on-board antenna.

The module supports standard IEEE802.11 b/g/n agreement, complete TCP/IP protocol stack. Users can use the add modules to an existing device networking or building a separate network controller.

ESP8266 is high integration wireless SOCs, designed for space and power constrained mobile platform designers.

It provides unsurpassed ability to embed Wi-Fi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minimal space requirement.

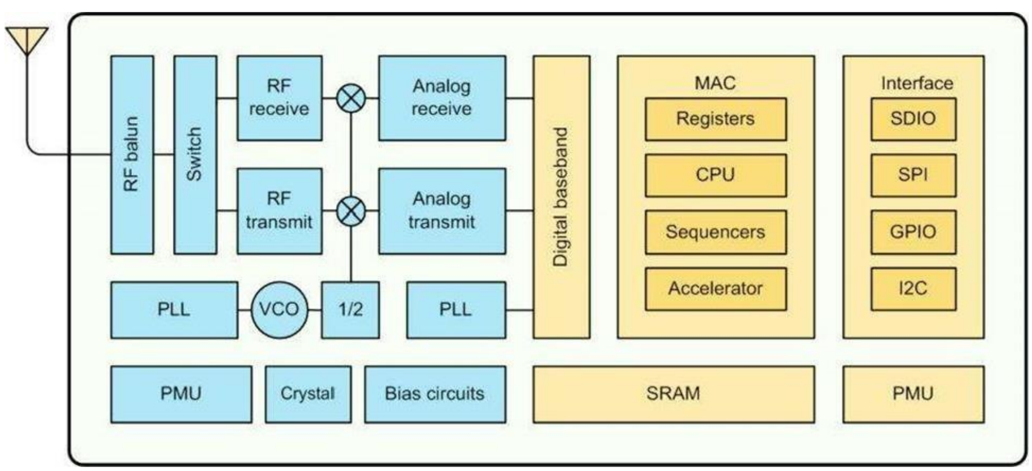


Fig. 4.4a ESP8266 Architecture

ESP8266EX offers a complete and self-contained Wi-Fi networking solution; it can be used to host the application or to offload Wi-Fi networking functions from another application processor.

When ESP8266EX hosts the application, it boots up directly from an external flash. In has integrated cache to improve the performance of the system in such applications.

Alternately, serving as a Wi-Fi adapter, wireless internet access can be added to any micro controller based design with simple connectivity (SPI/SDIO or 12C/UART interface).

ESP8266-01 Specifications

- 802.11 b/g/n
- Integrated low power 32-bit MCLI
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, INA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- Wi-Fi 2.4 GHz, support WPA,WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and iOS devices
- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, 12C, 12S, IRDA, PWM, GPIO
- STBC, IXI MIMO, 2x1 MIMO
- A-MPDU & A-MSDIJ aggregation and 0.4s guard interval
- Deep sleep power <10uA, Power down leakage current < 5uA
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- +20dBm output power in 802.11b mode
- Operating temperature range -40C 125C

ESP8266-01 Parameters

Categories	Items	Values
Wi-Fi Parameters	Wi-Fi Protocols	802.11 b/g/n
	Frequency Range	2.4GHz-2SGHz (2400M-2483SM)
Hardware Parameters	Peripheral Bus	UART/HSPI/12C/12S/ir Remote Control
		GPIO/PWM
	Operating Voltage	
	Operating Current	Average value.' 80mA
	Operating Temperature Range	-aom2s•
	Ambient Temperature Range	Normal temperature
	Package Size	14.3mrn*24.8mm *3mm
	External Interface	N/A
Software Parameters	Wi-Fi mode	station/softAP/SoftAP+station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	IJART Download / OTA (via network) / download and write firmware via host
	Software Development	Supports Cloud Server Development / SDK for custom firmware development
	Network Protocols	IPv4» TCP/IJDP/HTTP/FTP
	User Configuration	AT Instruction Set* Cloud Server* Android/iOS App

Table 4.4.2 ESP8266-01 Parameters

ESP8266-01 Module Pin Description

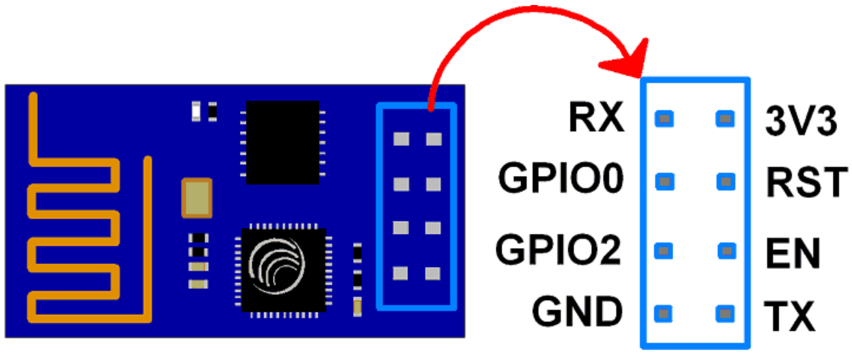


Fig. 4.4.3 ESP8266-01 Pinout

- **3V3**: - 3.3 V Power Pin.
- **GND**: - Ground Pin.
- **RST**: - Active Low Reset Pin.
- **EN**: - Active High Enable Pin.
- **TX**: - Serial Transmit Pin of UART.
- **RX**: - Serial Receive Pin of UART.
- **GPIO0 & GPIO2**: - General Purpose I/O Pins. These pins decide what mode (boot or normal) the module starts up in. It also decides whether the TX/RX pins are used for Programming the module or for serial I/O purpose.
- To program the module using UART, Connect GPIO0 to ground and GPIO2 to VCC or leave it open. To use UART for normal Serial I/O leave both the pins open (neither VCC nor Ground).

ESP8266-01 Functional Description

ESP8266EX is embedded with Tensilica L106 32-bit micro controller (MCU), which features extra low power consumption and 16-bit RSIC. The CPU clock speed is 80MHz. It can also reach a maximum value of 160MHz. ESP8266EX is often integrated with external sensors and other specific devices through its GPIOs; codes for such applications are provided in examples in the SDK.

Internal SRAM and ROM

ESP8266EX Wi-Fi SOC is embedded with memory controller, including SRAM and ROM. MCU can visit the memory units through iBus, dBus, and ABB interfaces. All memory units can be Visited upon request, while a memory arbiter Will decide the running sequence according to the time When these requests are received by the processor.

According to our current version of SDK provided, SRAM space that is available to users is assigned as below:

- RAM size < 36kB when ESP8266EX is working under the station mode and is connected to the router, programmable space accessible to user in heap and data section is around 36kB.)
- There is no programmable ROM in the SOC, therefore, user program must be stored in an external SPI flash.

External SPI Flash

This module is mounted with an 1 MB external SPI flash to store user programs. If larger definable storage space is required, a SPI flash with larger memory size is preferred. Theoretically speaking, up to 16 MB memory capacity can be supported.

Suggested SPI Flash memory capacity:

OTA is disabled: the minimum flash memory that can be supported is 512 kB; OTA is enabled: the minimum flash memory that can be supported is 1 MB.

Several SPI modes can be supported, including Standard SPI, Dual SPI, and Quad SPI.

Therefore, please choose the correct SPI mode When you are downloading into the flash, otherwise firmware/programs that you downloaded may not work in the right way.

Crystal

Currently, the frequency of crystal oscillators supported include 40MHz, 26MHz and 24MHz. The accuracy of crystal oscillators applied should be ±IOPPM, and the operating temperature range should be between -200C and 85 0C.

When using the downloading tools, please remember to select the right crystal oscillator type. In circuit design, capacitors CI and C2, which are connected to the earth, are added to the input and output terminals of the crystal oscillator, respectively. The values of the two capacitors can be flexible, ranging from 6pF to 22pF, however, the specific capacitive values of Cl and C2 depend on further testing and adjustment on the overall performance of the whole circuit. Normally, the capacitive values of Cl and C2 are within 10pF if the crystal oscillator frequency is 26MHz, while the values of Cl and C2 are 10pF<C1, C2<22pF if the crystal oscillator frequency is 40MHz.

Interfaces

Interface	Pin Name	Description
HSPI	MISO MOSI CLK CS	SPI Flash 2, display Screen, and MCIJ can be connected using HSPI interface.
PWM	R G B	Currently the PWM interface has four channels. PWM interface can be used to control LED lights, buzzers & so on.
IR Remote Control	IR-T IR-R	The functionality of Infrared remote-control interface can be implemented via software programming.

ADC	TOUT	ESP8266EX integrates a 1 0-bit analog ADC. It can be used to test the power supply voltage of VDD3P3 (Pin3 and Pin4) and the input power voltage of TOUT (Pin 6).
12C	SCL SDA	12C interface can be used to connect external sensor products and display screens, etc.

Interface	**Pin Name**	**Description**
UART	TXD RXD RTS UART TXD	Devices with UART interfaces can be connected with the module.
		UART will output some printed information when the device is powered On and is booting up, if this issue exerts influence on some specific applications, users can exchange the inner pins of UART when initializing exchange UTXD, URXD with URTS, UCTS.
I2S	12S Input: (12Sl_DATA) ; (12Sl_BCK) (12Sl_WS); 12S Output: (12SO_BCK) (12SO_DATA); (12SO_WS)	I2S interface is mainly used for collecting, processing, and transmission of audio data.

ESP8266-01 Schematic Circuit

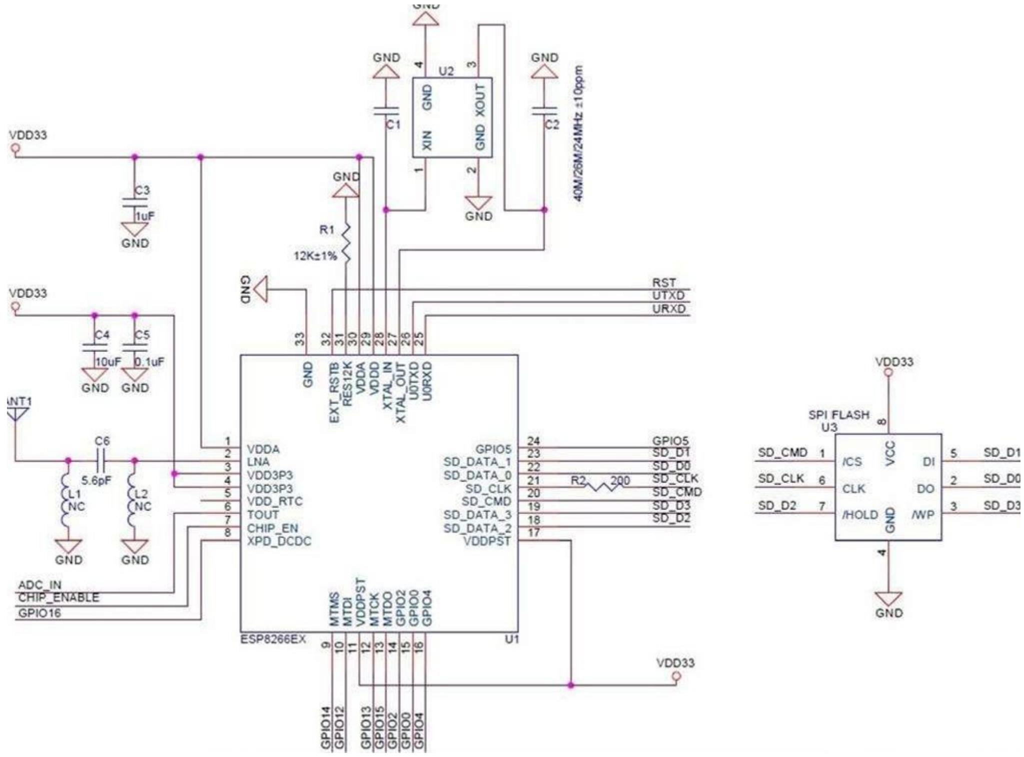


Fig. 4.4.6 ESP8266-01 Schematic Circuit

Advantages of ESP8266-01 Module

- Low cost
- Integrated support for WIFI network
- Reduced size of the board
- Low energy consumption

Disadvantages

- Need to learn a new language and IDE
- Reduced pinout
- Scarce documentation

Applications

- Home appliances
- Home automation
- Smart plugs and lights
- Industrial wireless control
- Baby monitors
- IP cameras
- Sensor networks
- Wearable electronics
- Wi-Fi location-aware devices
- Security ID tags
- Wi-Fi position system beacons

16X2 LIQUID CRYSTAL DISPLAY

An LCD is an electronic display module which uses liquid crystal to produce a visible image. The 16×2 LCD display is a very basic module commonly used in DIYs and circuits. The 16×2 translates to a display 16 characters per line in 2 such lines. In this LCD each character is displayed in a 5×7-pixel matrix.

Fig. 4.5a: JHD 16x2 LCD

To connect an LCD display with an Arduino, you may have noticed that it consumes a lot of pins on the Arduino. Even in 4-bit mode, the Arduino still requires a total of seven connections – which is half of Arduino's available digital I/O pins.

The solution is to use an I2C LCD Display. It consumes only two I/O pins that are not even part of a set of digital I/O pins and can also be shared with other I2C devices.
A typical I2C LCD display consists of a HD44780 based character LCD display and an I2C LCD adapter.

Fig. 4.5b: JHD 16x2 LCD with I2C Module

16x2 LCD with I2C Module Pin Configuration

Fig 4.5.1 16x2 LCD with I2C Module Pinout

- **GND**
  - GND is a ground pin and should be connected to the ground of MCU.
- **VCC**
  - VCC supplies power to the module and the LCD. Connect it to the 5V output of the MCU or a separate power supply.
- **SDA**
  - Serial Data pin. This line is used for both transmit and receive. Connect to the SDA pin on the MCU.
- **SCL**
  - Serial Clock pin. This is a timing signal supplied by the Bus Master device. Connect to the SCL pin on the MCU.

Power Supply Adapter

12V power supplies are one of the most common power supplies in use today. In general, a 12VDC output is obtained from a 50VAC or 240VAC input using a combination of transformers, diodes and transistors. 12V power supplies can be of two types: 5V regulated power supplies, and 12V unregulated power supplies. 12V regulated power supplies come in three styles: Switching regulated AC to DC, Linear regulated AC to DC, and Switching regulated DC to DC.

Fig. 4.6: 12V Power Supply Adapter

Regulated 12VDC power supplies regulate the 12VDC output voltage using a complex high frequency switching technique that employs pulse width modulation and feedback. regulated power supplies also employ extensive EMI filtering and shielding to attenuate both common and differential mode noise conducted to the line and load. Galvanic isolation is standard in our 12 VDC switchers, affording our users input to output and output to ground isolation for maximum versatility. Regulated power supplies are highly efficient, small and lightweight, and are available in both AC-DC single and wide-adjust output and DC-DC configurations. Our Low Profile wide adjust output switchers can be voltage or current regulated and are externally programmable.

CHAPTER - 5

SOFTWARE DESCRIPTION

ARDUINO IDE 1.8.13

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards.

Fig. 5.1: Arduino IDE 1.8.13

The source code for the IDE is released under the GNU General Public License, version The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures.

User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution.

The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

By default, avrdude is used as the uploading tool to flash the user code onto official Arduino boards.

With the rising popularity of Arduino as a software platform, other vendors started to implement custom open source compilers and tools (cores) that can build and upload sketches to other microcontrollers that are not supported by Arduino's official line of microcontrollers.

Arduino IDE Features

- **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50

- **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

- **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well.

- For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

- **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers.

  - The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based.

  - Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

- **Open source and extensible hardware** - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers c an make their own version

- Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money

Installing Arduino IDE

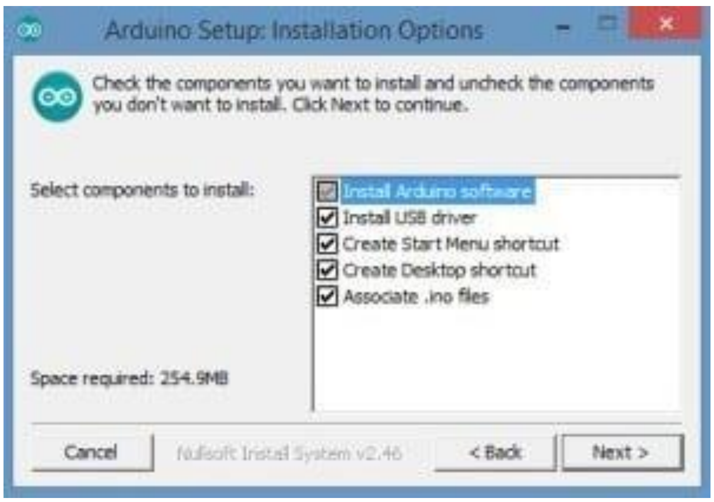
1. Visit http://www.arduino.cc/en/main/software to download the latest Arduino IDE version for your computer's operating system. There are versions for Windows, Mac, and Linux systems. At the download page, click on the "Windows Installer" option for the easiest installation.

2. Save the .exe file to your hard drive.

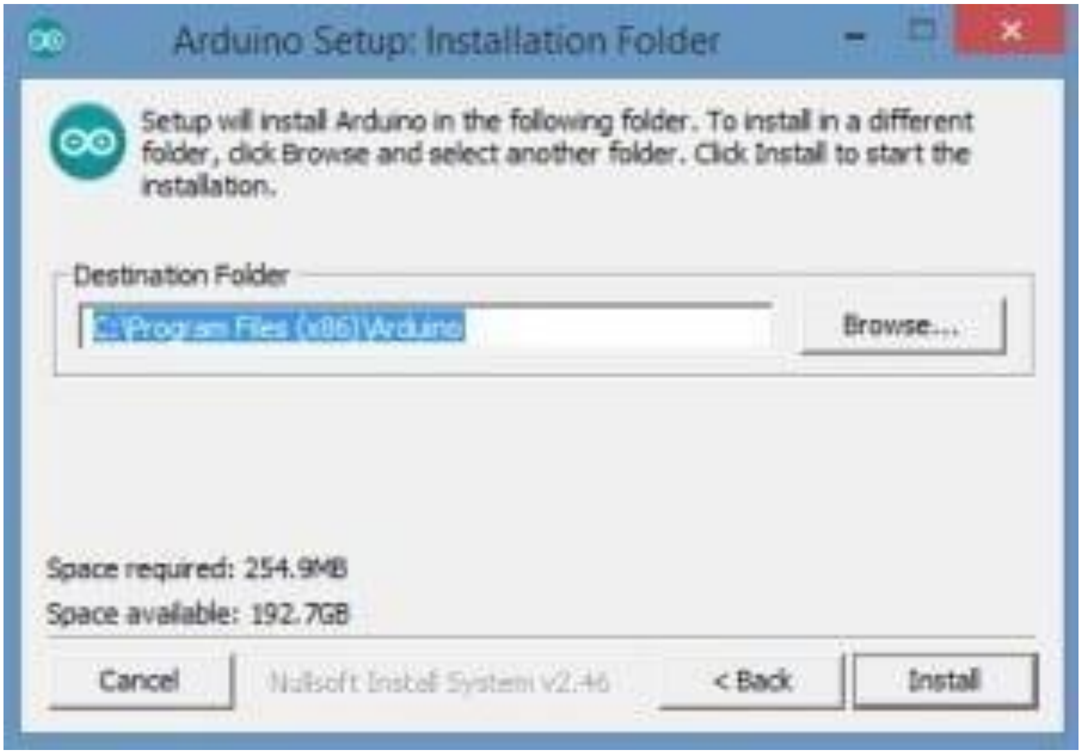
3. Open the .exe file.

4. Click the button to agree to the licensing agreement:

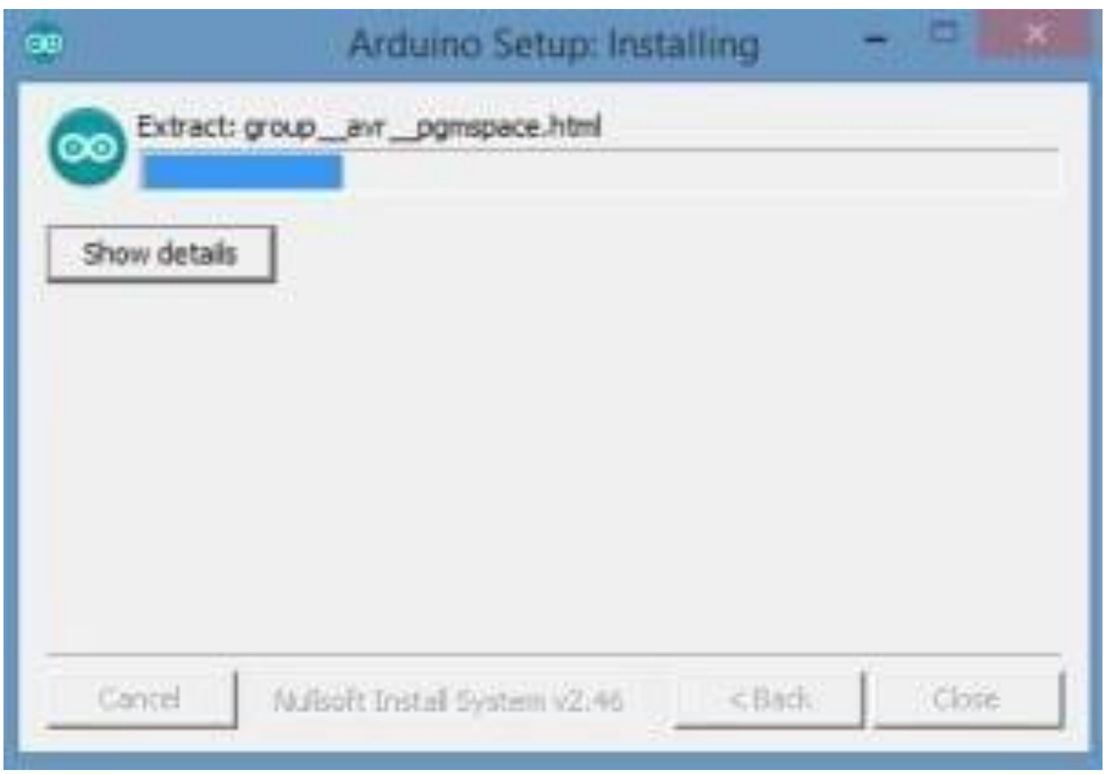
5. Decide which components to install, then click "Next":



6. Select which folder to install the program to, then click “Install”:



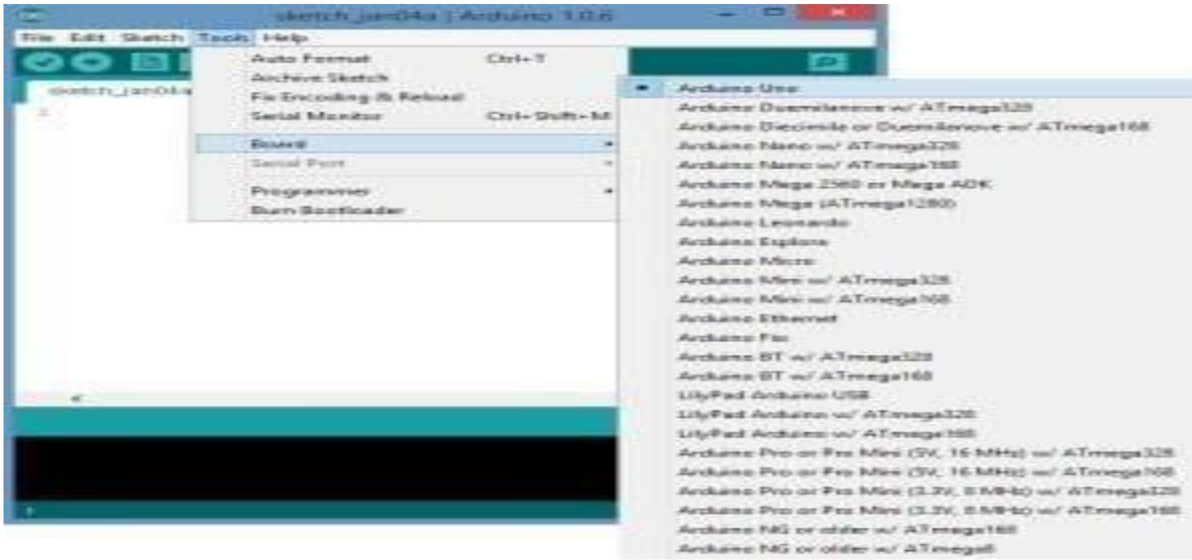
7. Wait for the program to finish installing, then click “Close”:



8. Now find the Arduino shortcut on your Desktop and click on it. The IDE will open up and you’ll see the code editor:

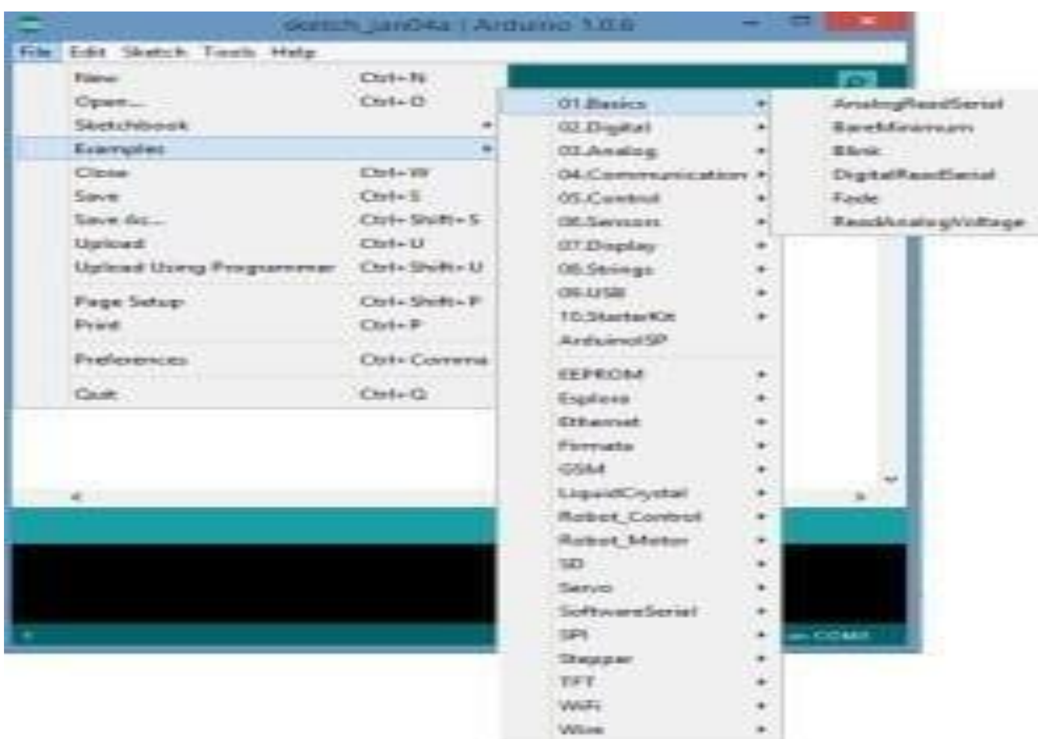
- **CONFIGURING THE ARDUINO IDE**

The next thing to do is to make sure the software is set up for your particular Arduino board. Go to the “Tools” drop-down menu, and find “Board”. Another menu will appear, where you can select from a list of Arduino models. I have the Arduino Uno R3, so I chose “Arduino Uno”.



- **EXPLORING THE ARDUINO IDE**

If you want, take a minute to browse through the different menus in the IDE. There is a good variety of example programs that come with the IDE in the “Examples” menu. These will help you get started with your Arduino right away without having to do lots of research:



Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020. It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.

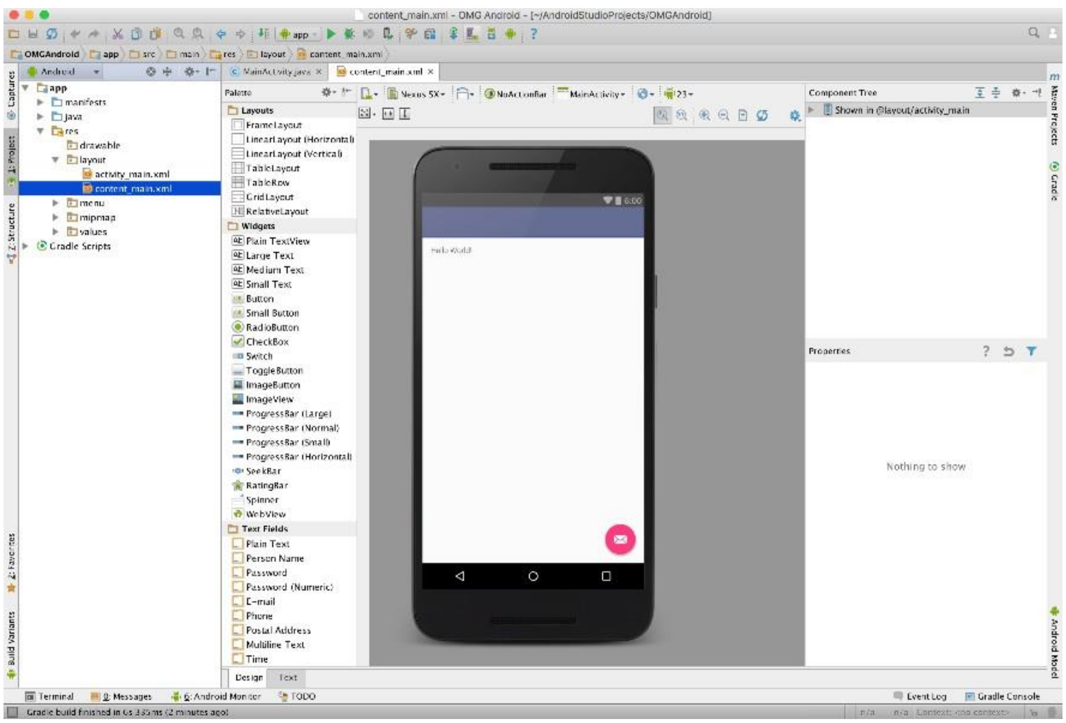


Fig. 5.2 Android Studio IDE

Android Studio Features

- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations.
- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine.

Android Virtual Device (Emulator) to run and debug apps in the Android studio. Android Studio supports all the same programming languages of IntelliJ e.g. Java, C++, and more with extensions, such as Go and Android Studio 3.0 or later supports Kotlin and all Java 7 language features and a subset of Java 8 language features that vary by platform version. External projects backport some Java 9 features. While IntelliJ states that Android Studio supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12. At least some new language features up to Java 12 are usable in Android.

Once an app has been compiled with Android Studio, it can be published on the Google Play Store. The application has to be in line with the Google Play Store developer content policy

System Requirements

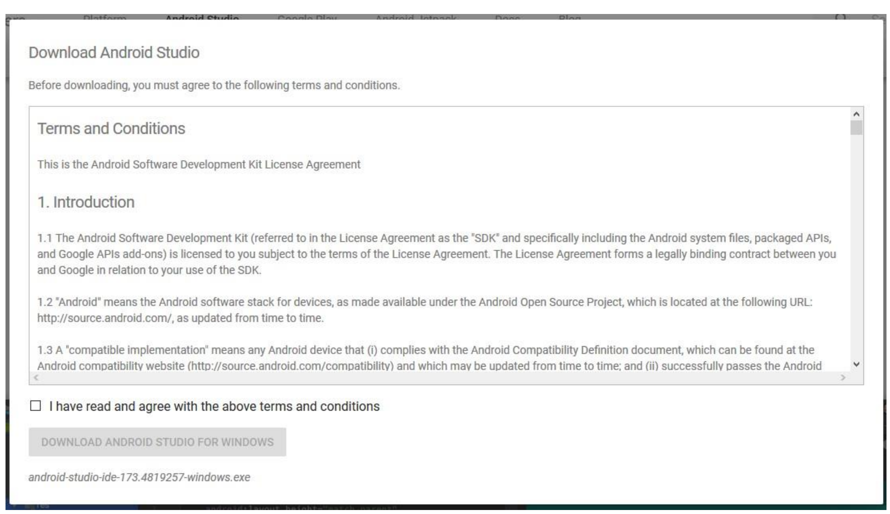
Operating System	Microsoft Windows	Mac	Linux
Operating System Version	Microsoft® Windows® 7/8/10 (32- or 64-bit) *The Android Emulator only supports 64-bit Windows.*	Mac® OS X® 10.10 (Yosemite) or higher, up to 10.14 (macOS Mojave)	GNOME or KDE desktop *Tested on Linux based on Debian (4.19.67-2rodete2).*
Random Access Memory (RAM)	4 GB RAM minimum; 8 GB RAM recommended.		
Free disk space	2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image).		
Minimum required JDK version	Java Development Kit 8		
Minimum screen resolution	1280 x 800		

Installation of Android Studio

To install Android Studio on Windows, proceed as follows:

- If you downloaded an .exe file (recommended), double-click to launch it.
- If you downloaded a .zip file, unpack the ZIP, copy the android-studio folder into your Program Files folder, and then open the android-studio > bin folder and launch studio64.exe (for 64-bit machines) or studio.exe (for 32-bit machines).
- Follow the setup wizard in Android Studio and install any SDK packages that it recommends.
- Head over to this link to get the Android Studio executable or zip file .
- Click on the download android studio button .

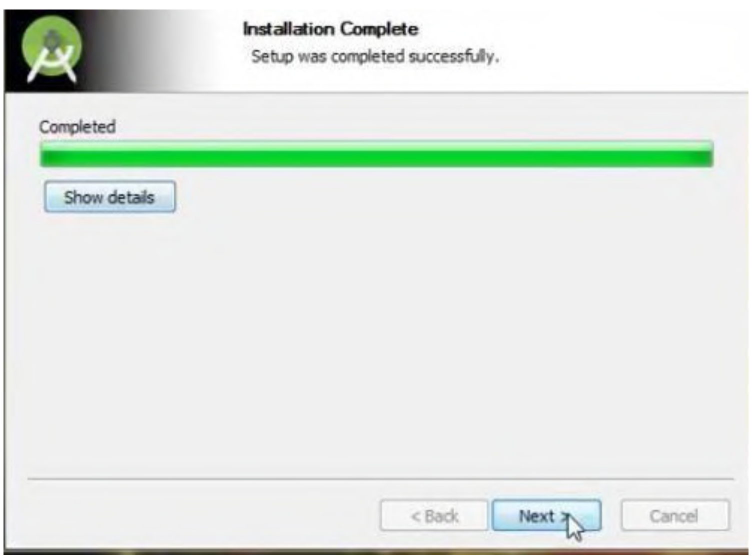
- Click on the “I have read and agree with the above terms and conditions” checkbox followed by the download button.



- Click on Save file button in the appeared prompt box and the file will start downloading .

- After the downloading has finished, open the file from downloads and run it . It will prompt the following dialogue box .

- Click on next .
- In the next prompt it'll ask for a path for installation. Choose a path and hit next.
- Note :The installation path should have the required minimum space.
- It will start the installation, and once it is completed, it will be like the image shown below .



- Click on next .

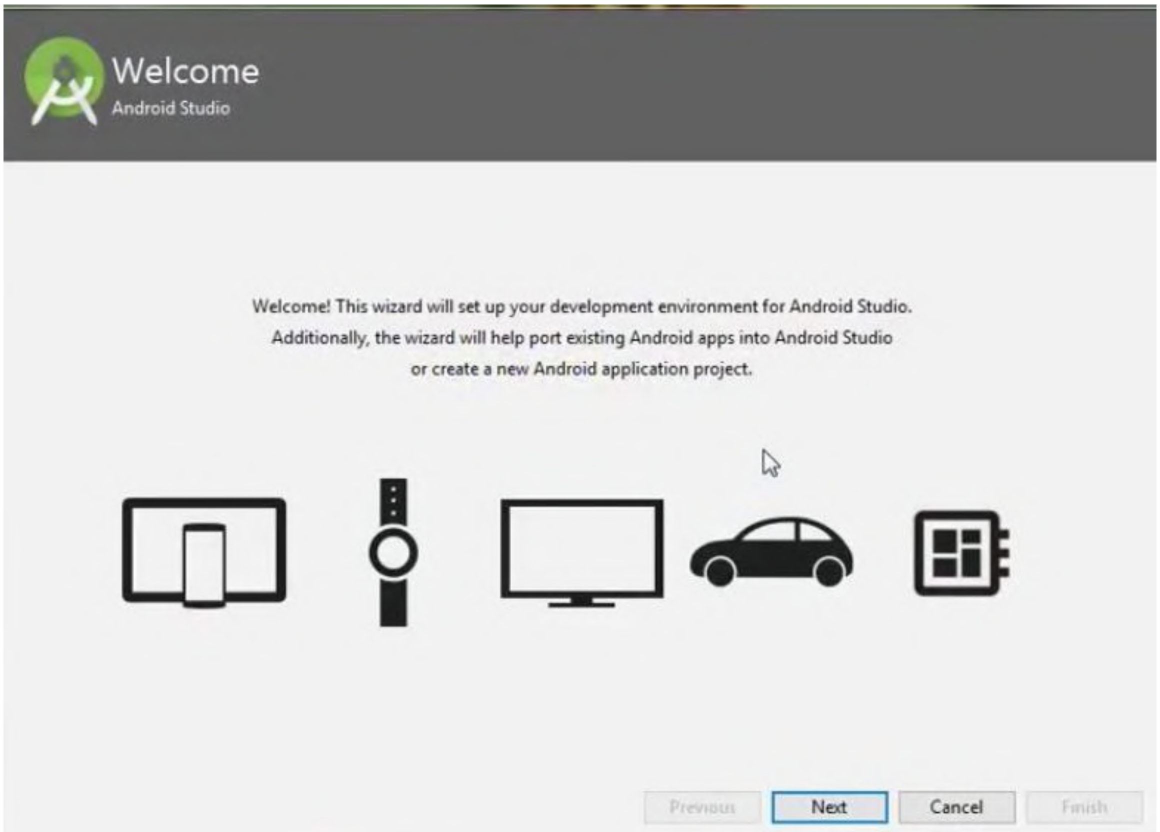
- Once “Finish” is clicked, it will ask whether the previous settings needs to be imported [if android studio had been installed earlier], or not.

  It is better to choose the ‘Don’t import Settings option’ .

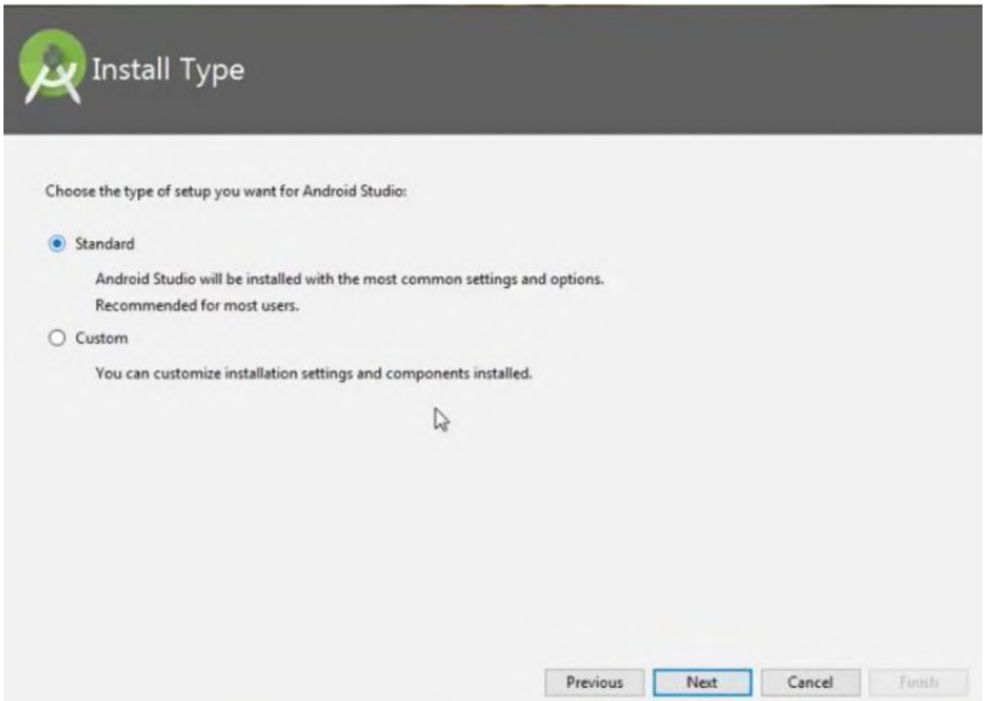
- Click the OK button.
- This will start the Android Studio.

- Meanwhile it will be finding the available SDK components .

- After it has found the SDK components, it will redirect to the Welcome dialog box .



- Click on next .

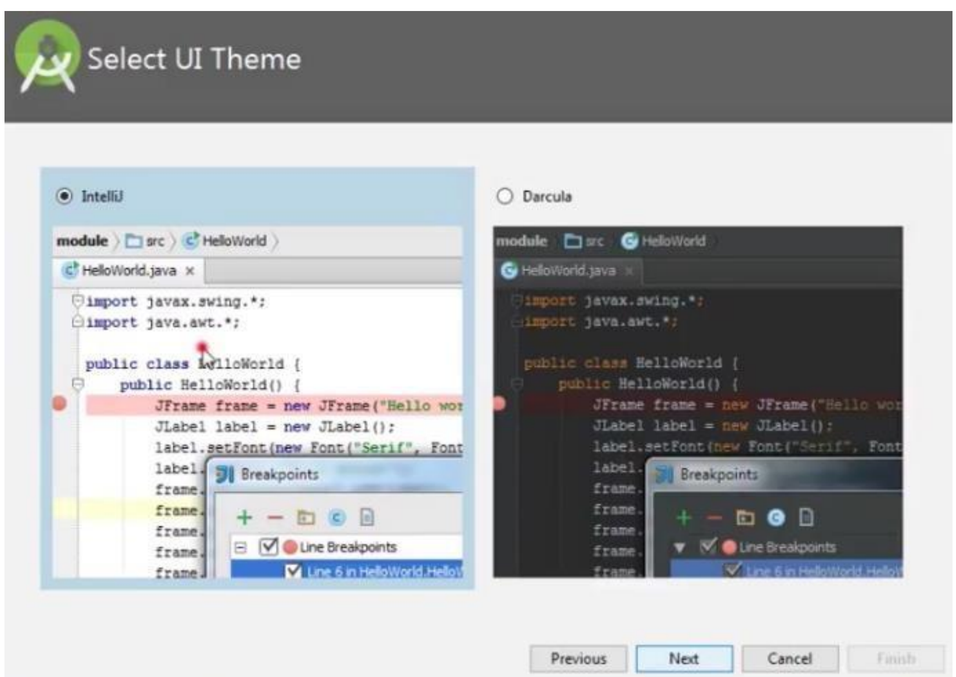


- Choose Standard and click on Next.

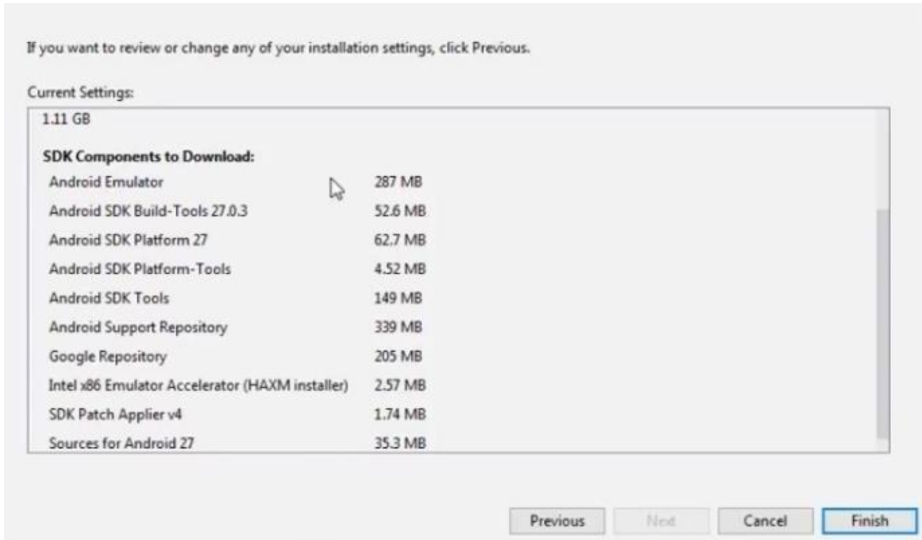
  Now choose the theme, whether Light theme or the Dark one .

  The light one is called the IntelliJ theme whereas the dark theme is called Darcula

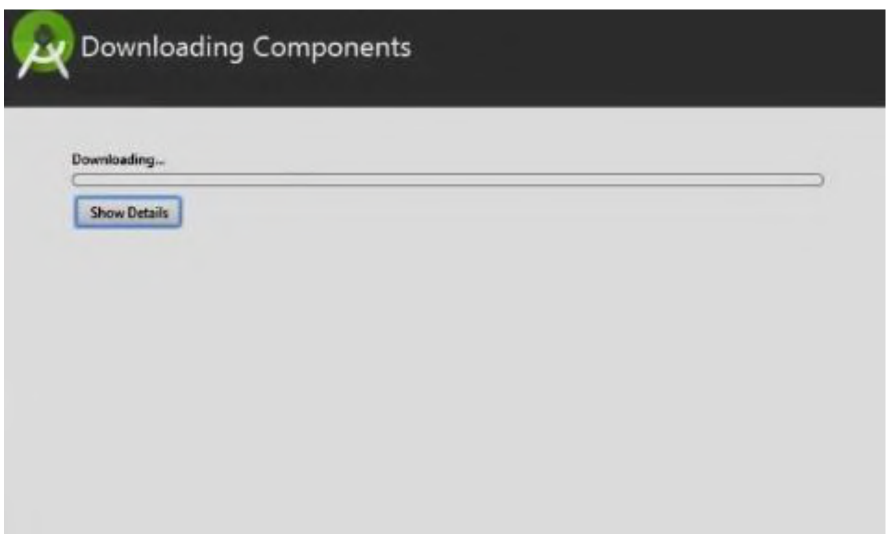
- Choose as required.



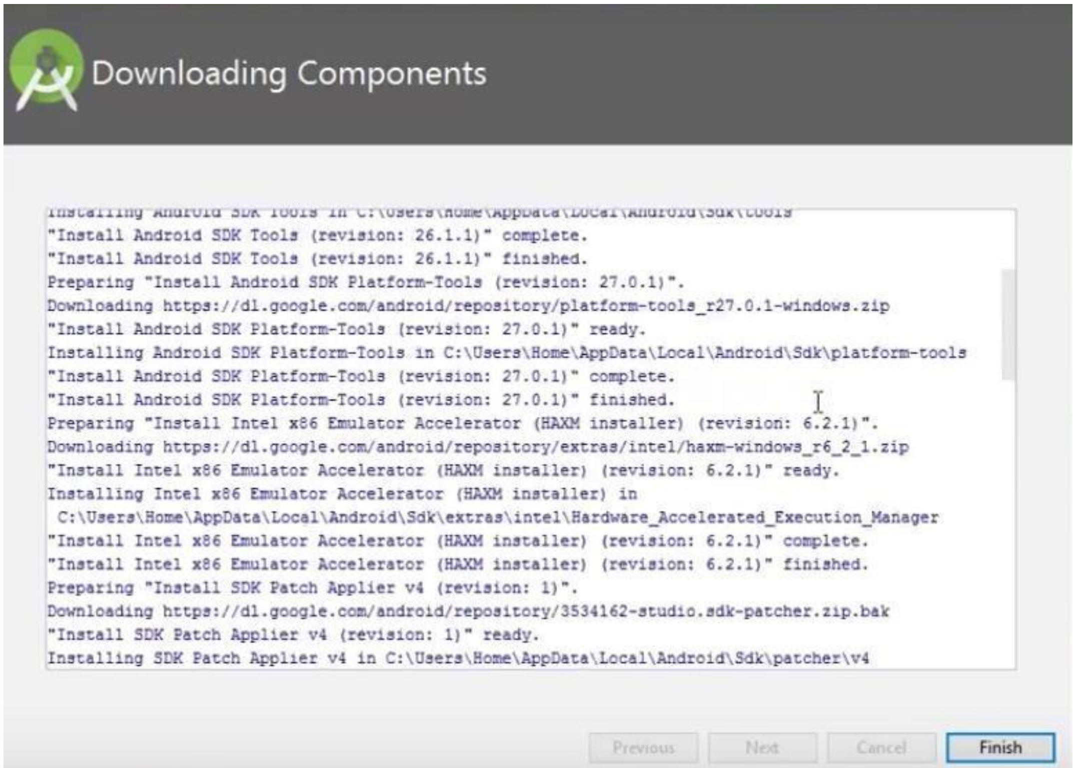
- Click on the Next button
- Now it is time to download the SDK components .



- Click on Finish .

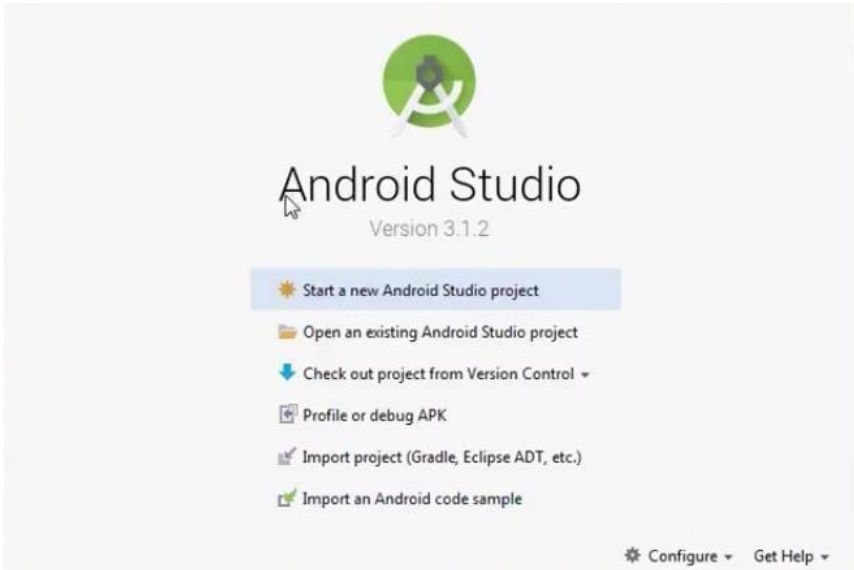


- It has started downloading the components



- The Android Studio has been successfully configured.
  Now it’s time to launch and build apps.
- Click on the Finish button to launch it.

- Click on ‘Start new android project’ to build a new app.



CHAPTER - 6

PROJECT IMPLEMENTATION

RESULTS

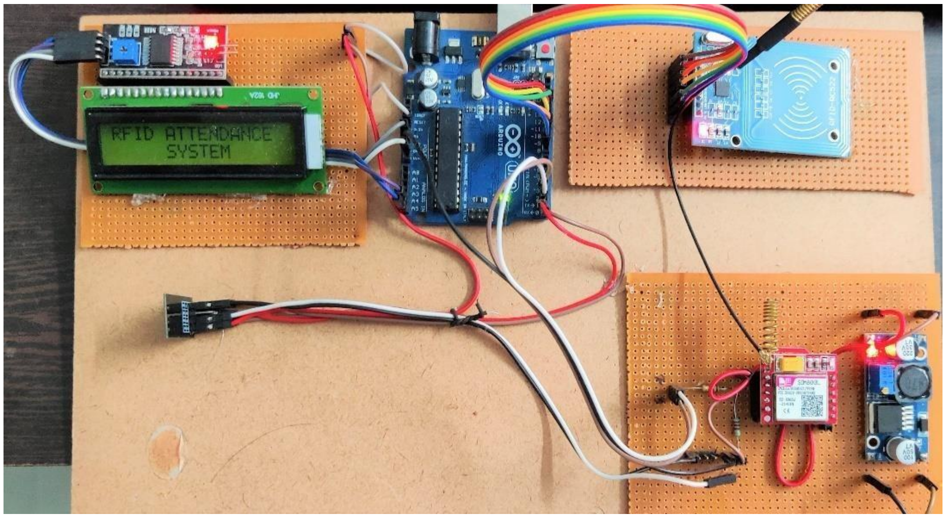


Fig 6.1a Welcome Screen

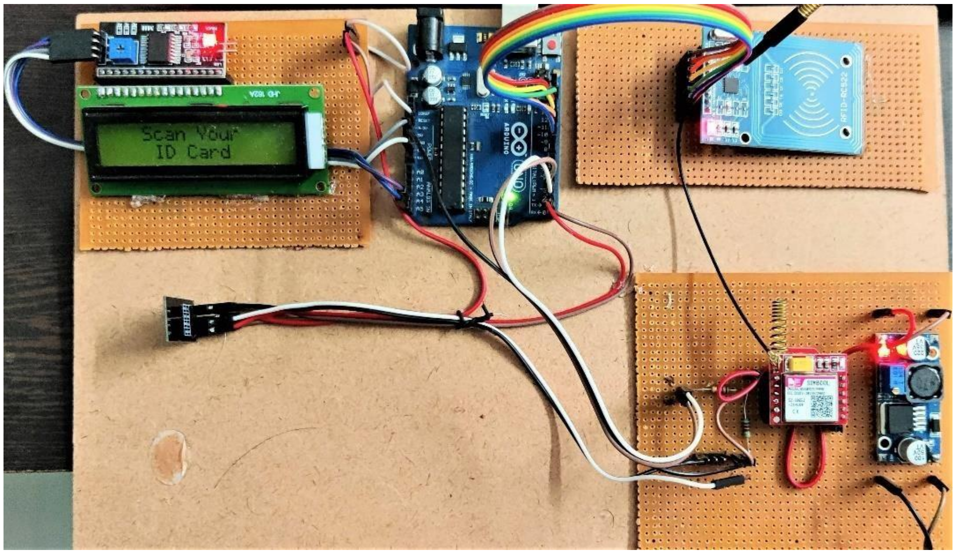


Fig 6.1b Scan your card Screen

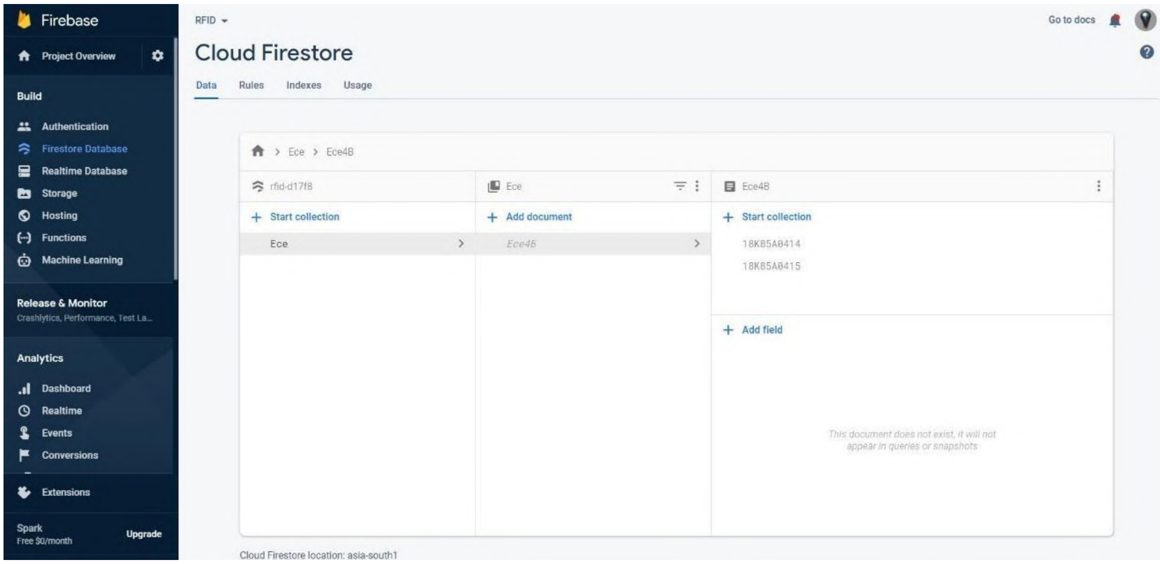
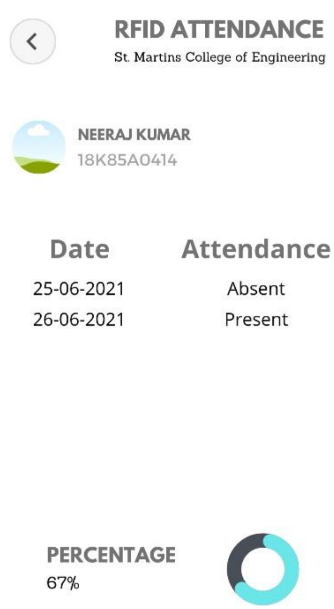


Fig 6.1c Google Firebase Database



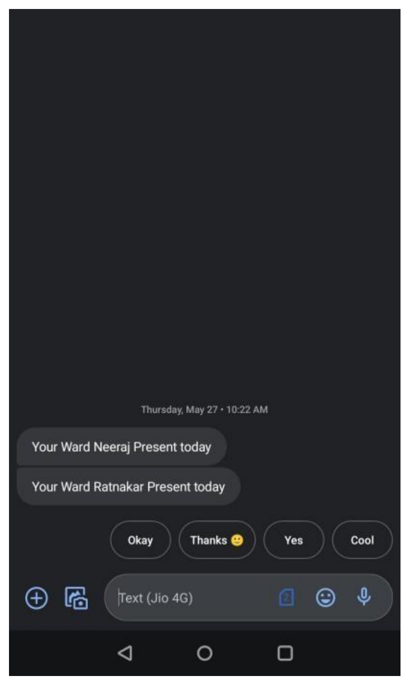


Fig 6.1d Android Application

Fig 6.1e SMS Notification

CODE IMPLEMENTATION

```
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <MFRC522.h>
SoftwareSerial sim800l(3,2);
LiquidCrystal_I2C lcd(0x27,16,2);
MFRC522 mfrc522(10, 9);
const char *ids[] = {"F1 98 59 02","45 D9 E7 30","34 AB C0 30","34 AB C0 40"};
const char *phn[] =
{"+918897999600","+918897999600","+918897999600","+918897999600"};
const char *roll[] = {"18K85A0415","18K85A0414","18K85A0413","18K85A0416"};
const char *names[] = {"Ratnakar","Neeraj","Chaitanya","Shankar"};
int allow=0;
int index=0;
String dt="20210527";
int len = (sizeof (ids) / sizeof (*ids));
void setup()
{
  Serial.begin(9600);
  sim800l.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();
  lcd.begin(16,2);lcd.init();lcd.backlight();
  lcd.setCursor(0,0);lcd.print("RFID ATTENDANCE");
  lcd.setCursor(0,1);lcd.print("     SYSTEM     ");
  delay(3000);
```

```
  lcd.clear();
}
void loop()
{
  allow=0;
  index=0;
  lcd.setCursor(0,0);lcd.print("  Scan  Your  ");
  lcd.setCursor(0,1);lcd.print("    ID Card    ");
  if ( ! mfrc522.PICC_IsNewCardPresent())
  {
    return;
  }
  if ( ! mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
rfidcheck();
}
void rfidcheck(){
  String content= "";
  byte letter;
  for (byte i = 0; i < mfrc522.uid.size; i++) {
     content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
     content.concat(String(mfrc522.uid.uidByte[i], HEX));
  }
  content.toUpperCase();
  for (int x = 0; x < len; x++) {
```

```
        if (strcmp (content.substring(1).c_str(), ids[x]) == 0) {
          allow=1;
          index=x;
        }
    }
  if (allow==1)
    {
     lcd.clear();
     lcd.setCursor(0,0); lcd.print("Authorized access");
     lcd.setCursor(0,1); lcd.print(ids[index]);
     delay(2000);lcd.clear();
    uploadData();
     SendSms();
    }
  else
    {
     lcd.clear();
     lcd.setCursor(0,0); lcd.print("Access denied");
     lcd.setCursor(0,1); lcd.print(content.substring(1));
     delay(2000);
     lcd.clear();
    }
}
void SendSms()
{
  lcd.setCursor(0,0);
  lcd.print("Sending Data...");
```

```
  delay(1000);

  sim800l.println("AT");

  delay(500);

  sim800l.println("AT+CMGF=1");

  delay(500);

 sim800l.print("AT+CMGS=\"");sim800l.print(phn[index]);sim800l.println("\"");

  delay(500);

  sim800l.print("Your Ward ");sim800l.print(names[index]);sim800l.print(" Present 
today");

  delay(500);

  sim800l.write(26);

  lcd.clear();

  lcd.setCursor(0,1); lcd.print("SMS Sent");

  delay(1000);

}

void uploadData(){

      String values = "--"+(String)roll[index]+""+dt;

      Serial.flush();

      delay(1000);

      Serial.print(values);

      delay(2000);

  }
```

CHAPTER - 7

PROJECT TESTING

TESTING THE RFID TAG, RFID READER AND LCD DISPLAY

This has to do with powering and testing the RFID Card of each student, the RFID reader and the LCD segment. When the system is powered, it displays “Swipe your RFID CARD”, meaning that the system is ready to scan card as shown below in Fig 6.1.

Fig 7.1a: LCD powered ON

When the card is swiped across the RFID reader, the reader will sense it, decode the embedded code on it and sends the decoded signal to the central microcontroller to process, if found valid, details will be display on the LCD screen as shown below in Fig 2

Fig 7.1b: Access Granted

TESTING THE SIM800L GSM MODULE.

As the central microcontroller is sending the decoded information from RFID reader to LCD and computer screen for display, it also sends SMS notification through the Sim800LGSM module to a particular programmed student phone number as back up means. The SMS notification will send “Attendance, Student name, Registration number” as shown below in Fig 6.2 and Fig 6.3

Fig 7.2: Sending SMS to Parent

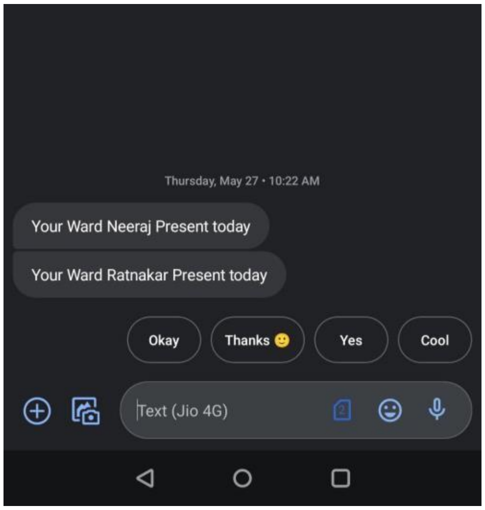


Fig 7.3: SMS Received by Parent

CHAPTER - 8

ADVANTAGES AND APPLICATIONS

Advantages of The Project

- Fully Automated.
- No Human Error Occurs.
- Error Free Data and can be used for different purposes without duplication.
- Fast and Reliable.
- Low Power Consumption, Even works with Battery Supply.

Applications of The Project

- Can be used to take attendance for student in school, college, and university.
- It also can be used to take attendance for workers in working places
- Used for Library Access and Books Management token

CHAPTER - 9

CONCLUSION AND FUTURE SCOPE

CONCLUSION

- We have developed an RFID based student attendance system with SMS notification and Database integration which is helpful in saving valuable time of both students and lecturers and helps to generate accurate reports when required.

- The SMS feature serves as a backup if data or connection loss occurs on the PC.

- Every Information is updated to College Portals and Android Applications in Real-time

FUTURE SCOPE

- We can extend this project by implement UHF RFID Reader module. So that attendance can be taken without placing the card on the reader.
- We can use Raspberry pi for fully Automating the Circuit with AI and ML Concepts.

PUBLICATIONS

CONFERENCE: Online Mega International Conference on **"Smart Modernistic in Electronics and Communication"**

PAPER ID: ICSMEC21-0042.

BIBLIOGRAPHY

[1] Barwuah, Adjei Further Education Development Agency, London (England). http://files.eric.ed.gov

[2] Philip VillaminMojaresLipa City Colleges, iNotified: An SMS and RFID-Based Notification System of Lipa City No. 10 G.A. Solis St., Lipa City Colleges, Lipa City, Batangas, Philippines 4217 phijuv@yahoo.com

[3] Rhowel M. Dellos ,ElinorHemedes Developing a User Monitoring System for the LPU-Laguna Library College of Engineering, Lyceum of the Philippines-Laguna 01-09 International Organization of Scientific Research IOSR Journal of Engineering (IOSRJEN) 2250-3021, ISSN (p): 2278-8719 Vol. 04, Issue 06 (June. 2014), ||V2|| PP www.iosrjen.org

[4] SumitaNainan , Romin Parekh , Tanvi Shah RFID Technology Based Attendance Management System IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 1, January 2013 ISSN (Print): 1694-0784 | ISSN (Online): 1694-0814 www.IJCSI.org

[5] Walkley, Phill Monitoring Student Attendance.FEDA Paper. International Journal of Computer Applications (0975 – 8887) 57 Volume 39– No.3, February 2012 http://research.ijcaonline.org

REFERENCES

[1]. C, Bardaki, P, Kourouthanassis and K, Pramatari. Deploying RFID-Enabled Services in the Retail Supply Chain, Lessons Learned toward the Internet of Things. Information Systems Management, 29(3), 233-245, 2012.

[2]. S, Chitresh and K, Amit. An efficient Automatic Attendance Using Fingerprint Verification Technique, International Journal on Computer Science and Engineering (IJCSE), 2(2), 264-269, 2010.

[3]. A. T, Dawes. Is RFID Right for Your Library, Journal of Access Services, 2(4), 7-13, 2004. Radio Frequency Identification (RFID) Based Attendance System with Short Message Servi DOI: 10.9790/0661-2102010108 www.iosrjournals.org 8 | Page

[4]. C. Geoffrey. Automatic Access Control System using Student's Identification Card based on RFID Technology, Thesis, Faculty of Electrical Engineering, University of Technology Malaysia, 2012

[5]. S, Lee, K, Ha and K , Lee. A pyroelectric infrared sensor-based indoor location-aware system for the smart home, IEEE Trans. Consumer Electron. 52(4), 311 -1317, 2006.

[6]. T, Lim, S , Sim and M , Mansor. RFID Based Attendance System. IEEE Symposium on Industrial Electronics and Applications. 2009

[7]. C, Liu and L, Chen. Applications of RFID technology for improving production efficiency in an Integrated-circuit packaging house, International Journal of Production Research, 47(8), 2203-2216, 2009.

[8] . O, Longe. Implementation of Student Attendance System using RFID Technology, B.Tech Project Report, Ladoke Akintola University of Technology, Ogbomoso, Nigeria, 2009.

[9]. K, Mahajan, P, Pandey and B, Pandher. Application of RFID Technology in Libraries and Role of Librarian. Pp 109-118, 2010

[10]. D, Maltoni, D, Maio, A, Jain and S, Prabhaker. Handbook of Fingerprint Recognition, Springer, New York, Pp 13-20, 2003.

[11]. A, Mohamed, A, Abdel-Hamid and K, Mohammed. Implementation of an Improved secure system detection for E passport by using EPC RFID tags, World Academy of Science, Engineering and Technology Journal, 6(1), 1-5, 2009.

[12]. A, Nambiar. A supply chain perspective of RFID Systems, World Academy of Science, Engineering and Technology Journal, Volume 6(2), 1-5, 2009

[13]. A, Olanipekun and K, Boyinbode. A RFID Based Automatic Attendance System in Educational Institutions of Nigeria, B.Tech Project Report. Department of Computer Science, Federal University of Technology, Akure, Nigeria. 2015

[14]. G, Ononiwu and O, Nwaji. Radio Frequency Identification Based Attendance System with Automatic Door Unit, Academic Research International, 2(2), 2223-9553, 2009.

[15]. A, Parvathy, V , Raj and M , Reddy M. RFID based exam hall maintenance system, IJCA Special Issue on Artificial Intelligence Techniques-Novel Approaches & Practical Applications (AIT). 2011

[16]. A, Qaiser and A, Khan. Automation of Time and Attendance using RFID Systems, 2nd International Conference on Emerging Technologies IEEE-ICET. pp. 60-63, 2006.

[17]. RFID Journal, Gillette Confirms RFID Purchase. http://www.rfidjournal.com/article/articleview/258/1/1/. [Accessed August 2018]

[18]. RFID SensNet Lab. Automatic Attendance System. Texas A & M University, Texas, USA, 2005

A

MAJOR PROJECT REPORT

on

FINGER PRINT BASED DEVICE SWITCHER

Submitted by

1) Mr. A. Shankar (18K85A0416)

2) Ms. M. Ramya (18K85A0417)

3) Ms. D. Shivaraj (17K81A0456)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Mrs. V. Sreeja

Associate Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500100

NBA &NAAC A+ Accredited

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE
UGC AUTONOMOUS
Dhulapally, Secunderabad-500 100
NBA &NAAC A+ Accredited

BONAFIDE CERTIFICATE

This is to certify that the project entitled "Recursive Block Based Key Point Matching Copy Move Image Forgery Detection" is being submitted by **1.Mr. A. Shankar 18K85A0416, 2.Ms. M. Ramya 18K85A0417 3.Mr. D. Shivaraj 17K81A0456** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OFTECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Internal Guide	**Head of the Department**
Mrs. V. sreeja Department of Electronics and Communication Engineering	Dr. Hari Krishna Department of Electronics and Communication Engineering

Internal Examiner	**External Examiner**

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT A.SHANKAR WITH ROLL NO.18K85A0416, D.SHIVA RAJ WITH ROLL NO.18K85A0421, M.RAMYA WITH ROLL NO.18K85A0417, OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "FINGERPRINT AUTHENTICATED DEVICE SWITCHER" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of '**Bachelor of Technology in Department of Electronics and Communication Engineering**', session: 2017-2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project work entitled '**Finger Print Based Device Switcher**' is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

A. Shankar	M. Ramya	D. Shivaraj
17K81A0401	17K81A0402	17K81A0456

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success**.**

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr. P. Santosh Kumar Patra**, our beloved **Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B. Hari Krishna**, **Head & Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our department **Project Coordinator, Mrs. V. Sreeja, Associate Professor** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our **Class Project Coordinator, Mr. K. Karthik,** Department of ECE and our project **Internal guide, Mrs. V. Sreeja, Associate Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level **Project Coordinators Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. Student Name
2. Student Name
3. Student Name

ABSTRACT

User Based Device control using fingerprint sensing. We come across areas where we need to switch no device for a particular user as soon as he enters. The device could be his Pc, fan, lights etc. For this purpose we suggest a fingerprint based device switcher project. The system switches on device for a particular user automatically on sensing him through his fingerprint scan. For this we use a fingerprint scanner interfaced to a microcontroller based circuit. As soon as a person enters he/she is allowed to scan his/her finger on the scanner. On scanning the data is sent to the microcontroller, based on this data the scanned copy is now verified for authentication. If the user is verified the microcontroller switches on a device (light) to indicate as the users verification.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

OBJECTIVE OF THE PROJECT

The system switches on device for a particular user automatically on sensing him through his fingerprint scan. For this we use a fingerprint scanner interfaced to a microcontroller based circuit. As soon as a person enters he/she is allowed to scan his/her finger on the scanner.

On scanning the data is sent to the microcontroller, based on this data the scanned copy is now verified for authentication. If the user is verified the microcontroller switches on a device (light) to indicate as the users verification.

Introduction of Embedded System

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.

The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes. The applications software on such processors is sometimes referred to as firmware

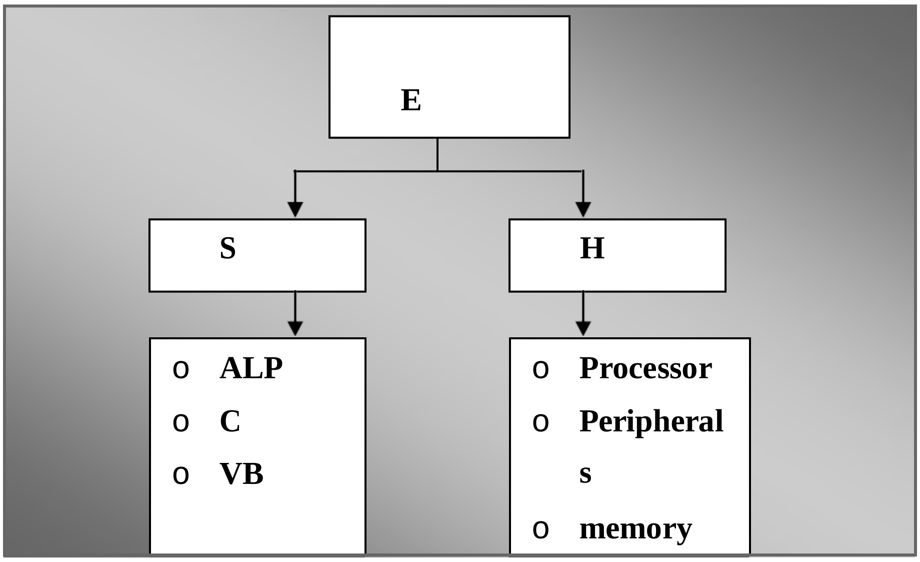


Figure 1.1 Block diagram of Embedded System

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

Memory: It is used to store data or address.

Peripherals: These are the external devices connected

Processor: It is an IC which is used to perform some task

Applications of embedded systems

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

Processors types

- Micro Processor (µp)
- Micro controller (µc)
- Digital Signal Processor (DSP)

Application Specific Integrated Circuits (ASIC)

- Micro Processor (μp):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Three basic characteristics differentiate microprocessors

- **Instruction set**: The set of instructions that the microprocessor can execute.
- **Bandwidth :** The number of bits processed in a single instruction.
- **Clock speed :** Given in megahertz (MHz), the clock speed determines how many instructions per second the processor can execute.

In both cases, the higher the value, the more powerful the CPU. For example, a 32-bit microprocessor that runs at 50MHz is more powerful than a 16-bit microprocessor that runs at 25MHz. In addition to bandwidth and clock speed, microprocessors are classified as being either RISC (reduced instruction set computer) or CISC (complex instruction set computer).

A microprocessor has three basic elements, as shown above. The ALU performs all arithmetic computations, such as addition, subtraction and logic operations (AND, OR, etc). It is controlled by the Control Unit and receives its data from the Register Array. The Register Array is a set of registers used for storing data. These registers can be accessed by the ALU very quickly. Some registers have specific functions - we will deal with these later. The Control Unit controls the entire process. It provides the timing and a control signal for getting data into and out of the registers and the ALU and it synchronizes the execution of instructions (we will deal with instruction execution at a later date).

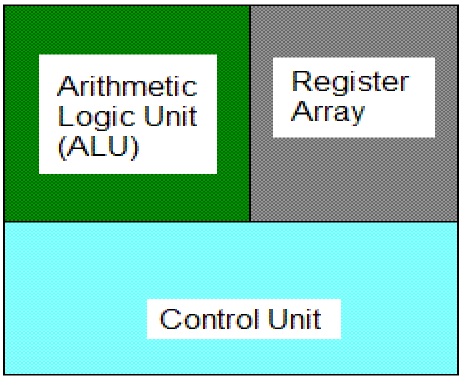


Fig1.2 basic elements of microprocessor

Three Basic Elements of a Microprocessor

Micro Controller (µc):

A microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.

Digital Signal Processors (DSPs)

Digital Signal Processors is one which performs scientific and mathematical operation. Digital Signal Processor chips - specialized microprocessors with architectures designed specifically for the types of operations required in digital signal processing. Like a general-purpose microprocessor, a DSP is a programmable device, with its own native instruction code. DSP chips are capable of carrying out millions of floating point operations per second, and like their better-known general-purpose cousins, faster and more powerful versions are continually being introduced. DSPs can also be embedded within complex "system-on-chip" devices, often containing both analog and digital circuitry.

Application Specific Integrated Circuit (ASIC)

ASIC is a combination of digital and analog circuits packed into an IC to achieve the desired control/computation function

ASIC typically contains

CPU cores for computation and control

Peripherals to control timing critical functions

Memories to store data and program

Analog circuits to provide clocks and interface to the real world which is analog in nature

Os to connect to external components like LEDs, memories, monitors etc.

Instruction Set

There are two different types of computer instruction set there are:

1. RISC (Reduced Instruction Set Computer) and

2. CISC (Complex Instruction Set computer)

Reduced Instruction Set Computer (RISC)

A RISC (reduced instruction set computer) is a microprocessor that is designed to perform a smaller number of types of computer instruction so that it can operate at a higher speed (perform more million instructions per second, or millions of instructions per second). Since each instruction type that a computer must perform requires additional transistors and circuitry, a larger list or set of computer instructions tends to make the microprocessor more complicated and slower in operation.

Besides performance improvement, some advantages of RISC and related design improvements are:

A new microprocessor can be developed and tested more quickly if one of its aims is to be less complicated.

Operating system and application programmers who use the microprocessor's instructions will find it easier to develop code with a smaller instruction set.

The simplicity of RISC allows more freedom to choose how to use the space on a microprocessor.

Higher-level language compilers produce more efficient code than formerly because they have always tended to use the smaller set of instructions to be found in a RISC computer.

RISC characteristics

Simple instruction set:

In a RISC machine, the instruction set contains simple, basic instructions, from which more complex instructions can be composed.

Same length instructions.

Each instruction is the same length, so that it may be fetched in a single operation.

1machine-cycleinstructions.

Most instructions complete in one machine cycle, which allows the processor to handle several instructions at the same time. This pipelining is a key technique used to speed up RISC machines.

Complex Instruction Set Computer (CISC)

CISC, which stands for Complex Instruction Set Computer**,** is a philosophy for designing chips that are easy to program and which make efficient use of memory. Each instruction in a CISC instruction set might perform a series of operations inside the processor. This reduces the number of instructions required to implement a given program, and allows the programmer to learn a small but flexible set of instructions.

The advantages of CISC:

At the time of their initial development, CISC machines used available technologies to optimize computer performance.

Microprogramming is as easy as assembly language to implement, and much less expensive than hardwiring a control unit.

The ease of micro-coding new instructions allowed designers to make CISC machines upwardly compatible: a new computer could run the same programs as earlier computers because the new computer would contain a superset of the instructions of the earlier computers.

As each instruction became more capable, fewer instructions could be used to implement a given task. This made more efficient use of the relatively slow main memory.

Because micro program instruction sets can be written to match the constructs of high-level languages, the compiler does not have to be as complicated.

The disadvantages of CISC

Still, designers soon realized that the CISC philosophy had its own problems, including:

Earlier generations of a processor family generally were contained as a subset in every new version --- so instruction set & chip hardware become more complex with each generation of computers.

So that as many instructions as possible could be stored in memory with the least possible wasted space, individual instructions could be of almost any length---this means that different instructions will take different amounts of clock time to execute, slowing down the overall performance of the machine.

Many specialized instructions aren't used frequently enough to justify their existence --- approximately 20% of the available instructions are used in a typical program.

CISC instructions typically set the condition codes as a side effect of the instruction. Not only does setting the condition codes take time, but programmers have to remember to examine the condition code bits before a subsequent instruction changes them.

Memory Architecture

There two different type's memory architectures there are:

Harvard Architecture

Von-Neumann Architecture

Harvard Architecture

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The **Harvard architecture** is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

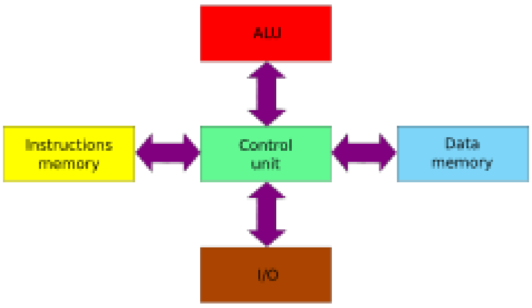


Figure 1.3 Harvard Architecture

Modern uses of the Harvard architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behaviour must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.

Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

Von-Neumann Architecture

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. The **von Neumann architecture** is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A **stored-program** digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

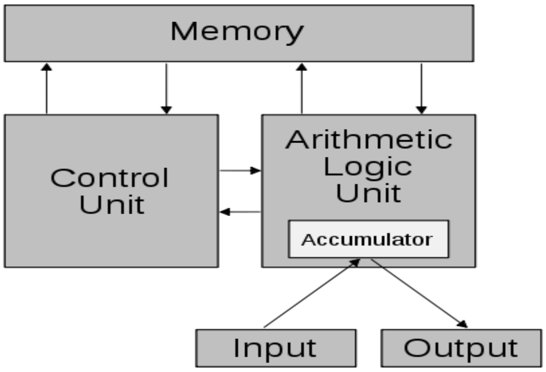


Figure 1.4 Schematic of the Von-Neumann Architecture.

Difference between Harvard and Von-Neumann Architecture

The primary difference between Harvard architecture and the Von Neumann architecture is in the Von Neumann architecture data and programs are stored in the same memory and managed by the same information handling system.

Whereas the Harvard architecture stores data and programs in separate memory devices and they are handled by different subsystems.

In a computer using the Von-Neumann architecture without cache; the central processing unit (CPU) can either be reading and instruction or writing/reading data to/from the memory. Both of these operations cannot occur simultaneously as the data and instructions use the same system bus.

In a computer using the Harvard architecture the CPU can both read an instruction and access data memory at the same time without cache. This means that a computer with Harvard architecture can potentially be faster for a given circuit complexity because data access and instruction fetches do not contend for use of a single memory pathway.

Today, the vast majority of computers are designed and built using the Von Neumann architecture template primarily because of the dynamic capabilities and efficiencies gained in designing, implementing, operating one memory system as opposed to two. Von Neumann architecture may be somewhat slower than the contrasting Harvard Architecture for certain specific tasks, but it is much more flexible and allows for many concepts unavailable to Harvard architecture such as self-programming, word processing and so on.

CHAPTER 2

LITERATURE SURVEY

Fingerprint matching has been successfully used by law enforcement for more than a century. The technology is now finding lot of other applications such as identity management and access control. In this context, an automated fingerprint recognition system and identification of key challenges are described along with the switching opportunities. The description is like a product design in this report implementing RTOS (Real time operating system) under the domain of embedded system. Fingerprint Recognition is a widely popular but complex pattern recognition Problem. In this paper, we have come with a novel approach to solve the existing problems with a proper Embedded System Design. In this project as well the fingerprint detection makes it coherent with the application involved within. The solutions viable are never ending but the switching mechanisms are to be sought after

From earlier times, security was and also till now is an issue of concern in our households and also in office, shops, etc. Everyone has a fear of unauthorized person entering to their home or office without their knowledge. The normal door can be fitted with locks which are capable of breaking with the use of an alternate key. Alternatives to this system can be found like the password or pattern system in the locks which again has the possibility of getting exposed and opening the lock. So, a solution to such problems can be by combining door lock with biometrics. Biometric verification is any means by which a person can be uniquely identified by evaluating one or more distinguishing biological traits. Unique identifiers include fingerprints, hand geometry, earlobe geometry, retina and iris patterns, voice waves, DNA, and signatures. Here we will use fingerprint for biometric verification as it is one such thing which is unique to every individual and the use of fingerprint as the key to door locks can overcome the security problem of unauthorized people trespassing to our homes, shops, offices, etc to a great extent as duplicacy in such key is not possible. Also, this system will not lead to problems like losing keys because we do not require carrying keys if this system is used instead of traditional locks. So, using arduino we will try to implement the system with features which will increase the security level.

Current biometric recognition systems for website authentication are mostly web server based, needing server support and infrastructure, and sometimes dedicated external hardware, for online authentication. Not many web servers support this kind of infrastructure for authentication. On the other hand, pure device based authentication systems are of the `all or none' type, serving to authenticate users for every action when using the device, or having authentication once and then keeping the device free to use. In this paper we propose a hybrid model, where fingerprint authentication is used in combination with the auto complete function on the browser for logging in to certain types of websites, or accessing only certain kind of information on the browser. In this method, the fingerprint module, inbuilt on certain mobile devices like Samsung Galaxy, is triggered automatically when certain pre-configured rules are met, such as the user visiting certain categories of websites. Also, based on the identity of the person swiping their fingerprint, the device has the ability to switch between one of a number of preconfigured security modes. Such a model can enable enhanced authentication for logging in to websites on mobile devices. We present the results of tests on a browser enabled with this system to study average response times, accuracy and effect on browser performance.

Network security has always had an issue with secure authentication and identification. In the current mixed device network of today, the number of nodes on a network has expanded but these nodes are often unmanaged from a network security perspective. The solution proposed requires a paradigm shift, recognition of what has already happened, identity is for sale across the internet. That identity is the users' network ID, their behaviour, and even their behaviour in using the networks. Secondly a majority of the devices on the Internet have been fingerprinted. Use of device fingerprinting can help secure a network if properly understood and properly executed. The research into this area suggests a solution. Which is the use of device fingerprints including clock skews to identify the devices and a dual- authentication process targeted at authenticating the device and the user? Not only authenticating the identity presented but also combining them into a unified entity so failure to authenticate part of the entity means the whole is denied access to the network and its resources.

CHAPTER 3

BLOCK DIAGRAM

Block Diagram Of Project:

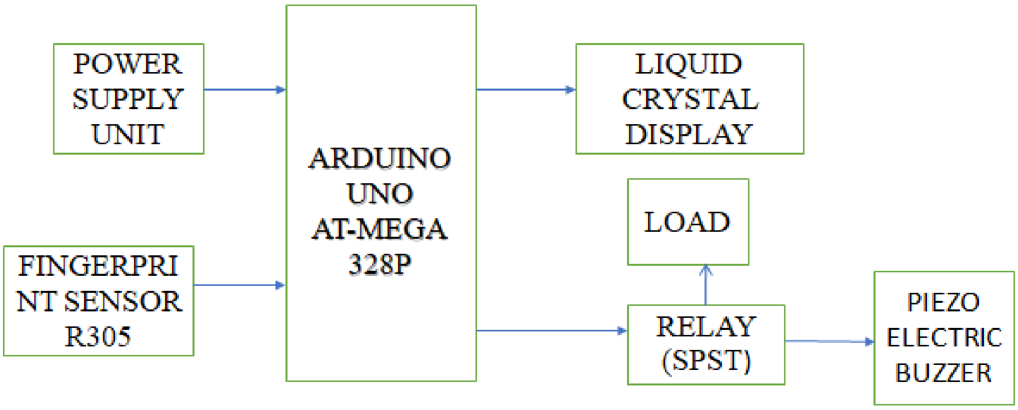


Fig 3.1 Block diagram of finger print based device switcher

The block diagram of finger print based device switcher is

The Proposed system finger print device switcher used for the authentication of any switching device. it means when user will be easily switch on his device and the wrong person will be switched on then the buzzer will be ringed otherwise correct person will accessed then the device will be on. Here we will connected the Lcd display used for it will display the information about to the project we will see in the display.

COMPONENT DESCRIPTION

ARDUINO UNO

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

Arduino is used as a controlling device .It comes comes with an open supply hardware feature that permits users to develop their own kit.

3.2.1LIQUID CRISTAL DISPLAY

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

3.2.3 Fingerprint Recognition Sensor

The sensor is a solid-state fingerprint sensor that reliably captures fingerprint information. It is designed to integrate into devices for improved security and convenience. The sensor provides a reliable, quick and user-friendly alternative to passwords, PIN's and other forms of user authentication.

This fingerprint scanner is capable of gathering and storing unique finger prints. Simply hold your finger on the optical scanner, query the device over serial, and you will be issued a unique ID. Use that ID within your embedded system to determine access levels, time clocks, door locks, etc. Unit includes 4 pin connector cable to connect and read to controller. The outputs is TTL level serial data.

CHAPTER 4

HARDWARE AND SOFTWARE COMPONENTS

POWER SUPPLY

Block Diagram

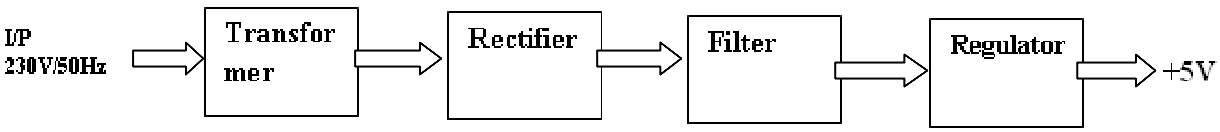
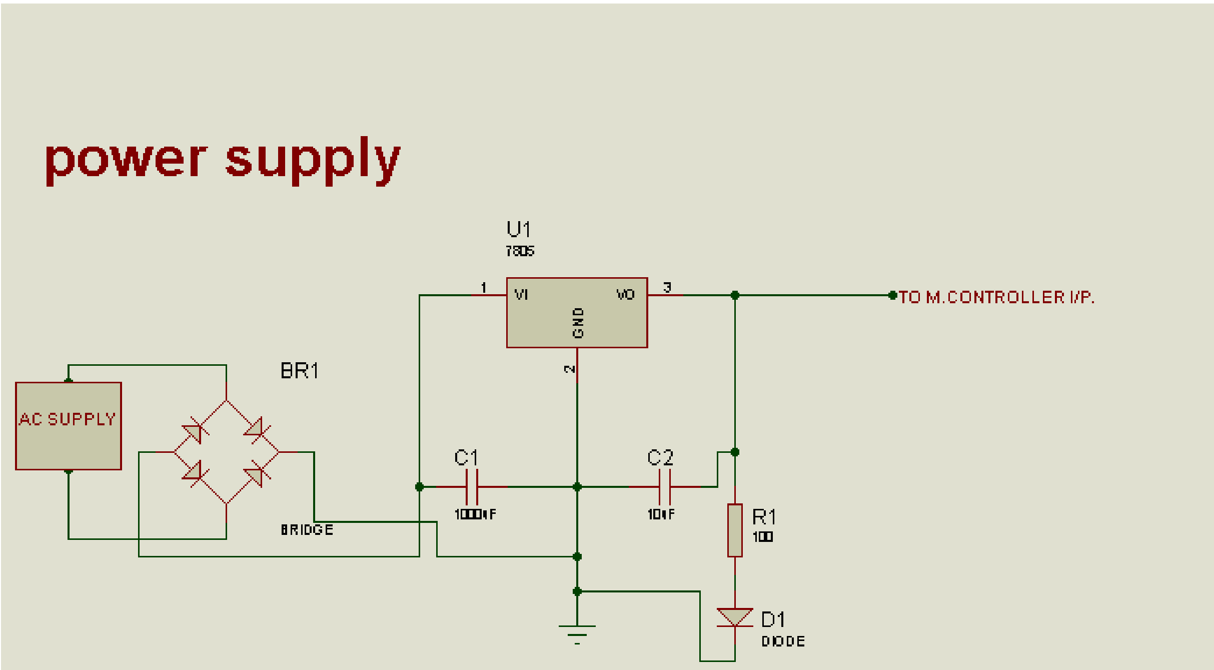


Figure 4.1 Block diagram Power Supply

Circuit Diagram



circuit diagram of power supply

Transformer

A **transformer** is a device that transfers electrical energy from one circuit to another through inductively coupled conductors the transformer's coils. A varying current in the first or primary winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

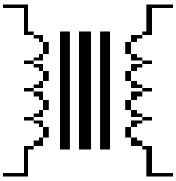


Figure: 4.3 Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

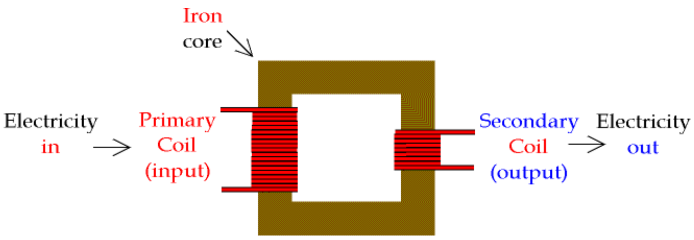


Figure 4.4 Transformer

Principle of trans former:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

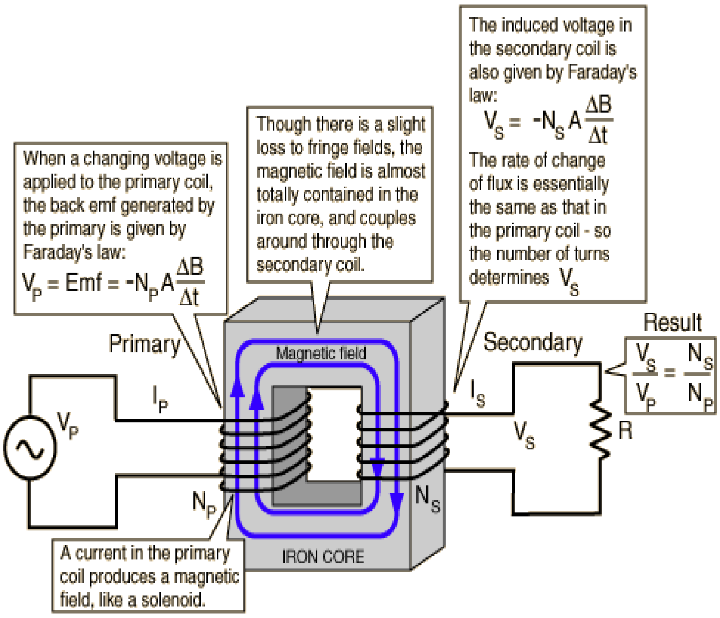


Figure: 4.5Basic Principle

Transformer Working

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

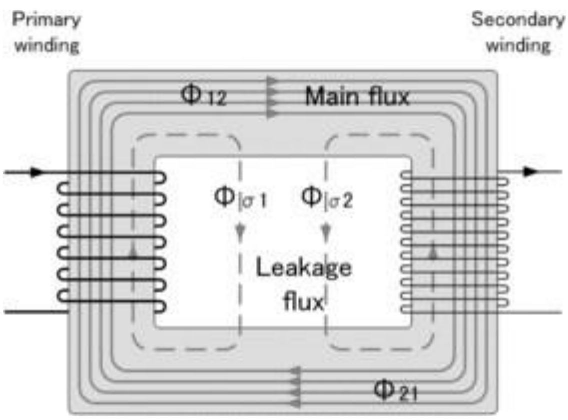


Figure: 4.6Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f.

The iron core is laminated to prevent 'eddy currents' flowing in the core. These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

Classification of Transformer:

- Step-Up Transformer
- Step-Down Transformer

Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn's ratio of the two sets of windings determines the amount of voltage transformation.

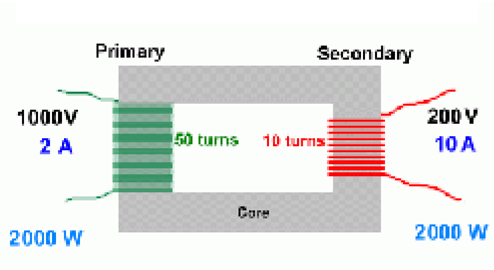


Figure: 4.7 Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single-phase step-down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase step up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

Step-Up Transformer

A step-up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply.

A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

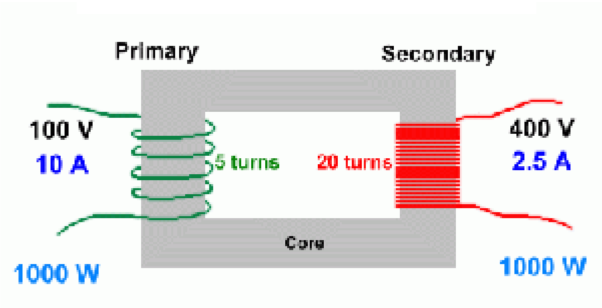


Figure: 4.8Step-Up Transformer

Applications

Generally these **Step-Up Transformers** are used in industries applications only.

Turns Ratio and Voltage

The ratio of the number of turns on the primary and secondary coils determines the ratio of the voltages...

$$\frac{V_s}{V_p} = \frac{N_s}{N_p}$$

...where V_p is the primary (input) voltage, V_s is the secondary (output) voltage, N_p is the number of turns on the primary coil, and N_s is the number of turns on the secondary coil.

Diodes

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

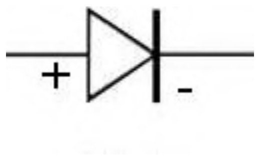


Figure 4.9 Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

Rectifier

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

The Half-wave Rectifier

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure .

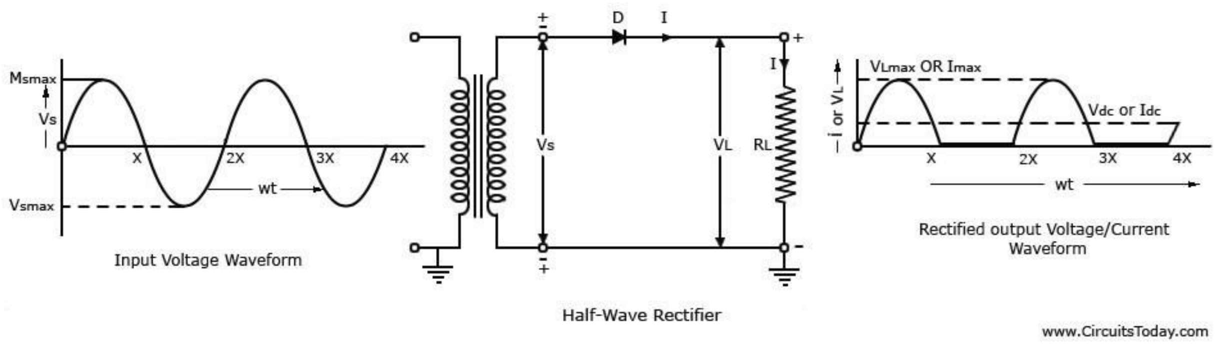


Figure 4.10 Half Wave Rectifier

Figure 2 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is 0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

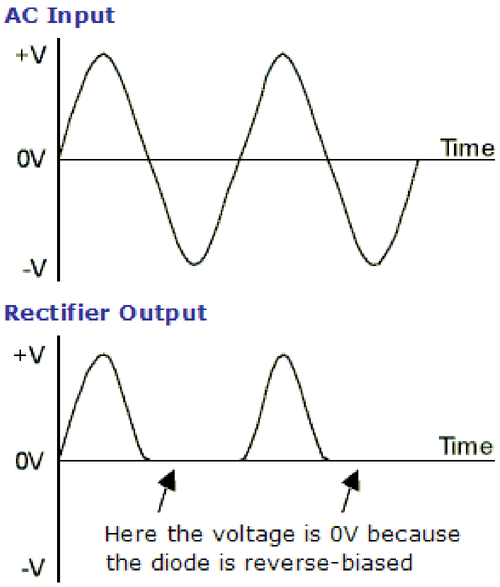


Figure 4.11 Half-Wave Rectification

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

The Full-wave Rectifier

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC. The resulting waveform is shown in figure 4.

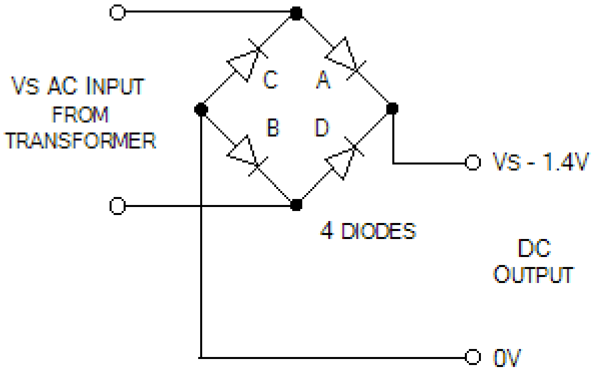


Figure:4.12 Full-Wave Rectifier

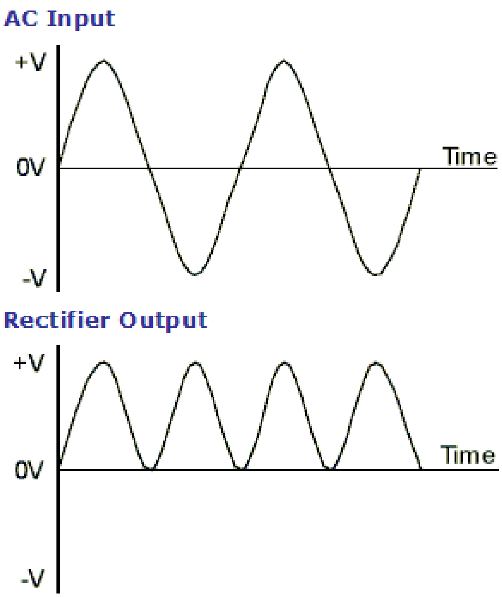


Figure :4.13Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased.

While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

Capacitor Filter

The **capacitor-input filter**, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

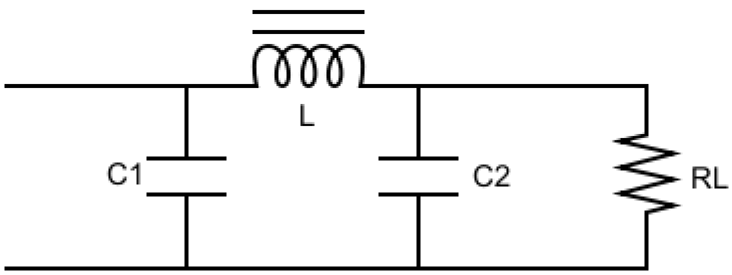


Figure: 4.14Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result only the DC component appears across the load RL.

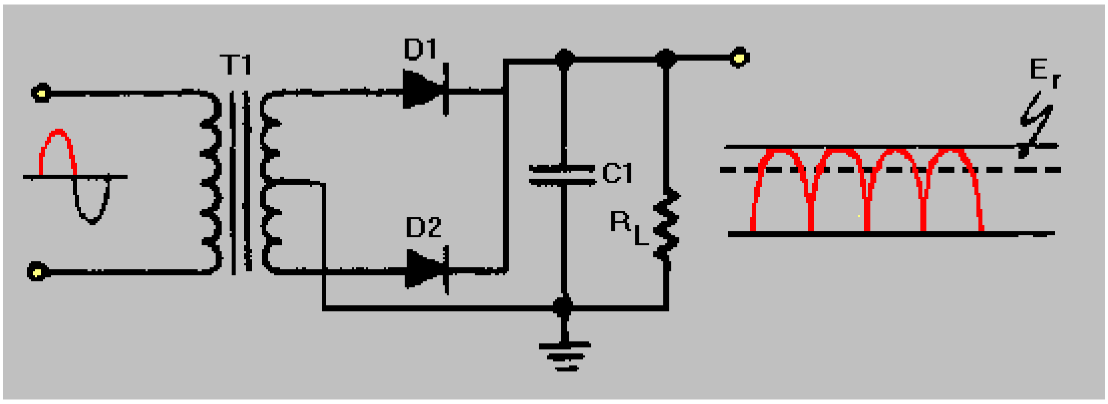


Figure: 4.15Centered Tapped Full-Wave Rectifier with a Capacitor Filter

Voltage Regulator

A **voltage regulator** is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx'indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx'indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consists the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

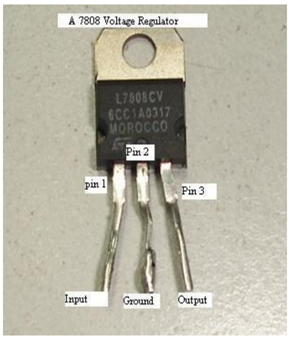


Figure 4.16 Regulator

Arduino

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

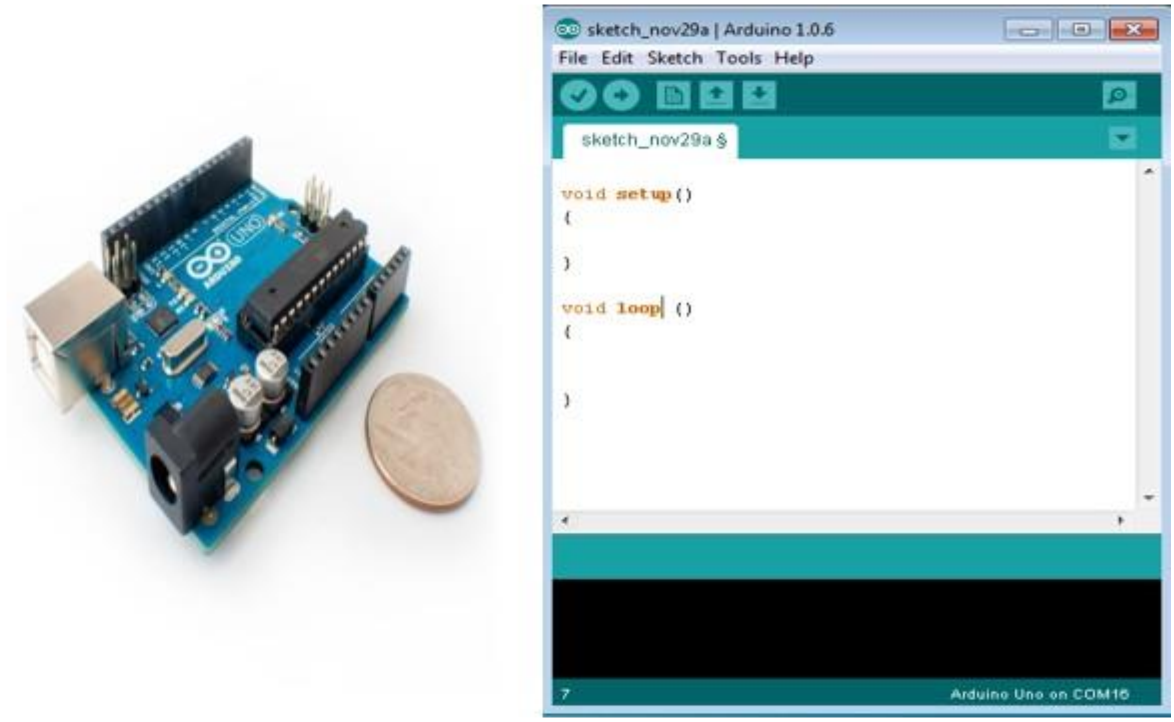


FIG:4.17 ARDUINO PLATFORM

Ardunio Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Table 4.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Voltage	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U 2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mh z	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header

Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 4.2Arduino boards based on ATMEGA32u4 microcontroller

Board Name	**Operating Voltage**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Progra mming Interface**
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MH z	5V	16MHz	14	6	6	1	Nativ e USB
Pro micro 3.3V/8MH z	5V	16MHz	14	6	6	1	Nativ e USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Voltage	Clock Speed	DIGITAL INPUTS	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Arduino boards based on AT91SAM3X8E microcontroller

Board Name	Operating Voltage	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

4.2.2Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

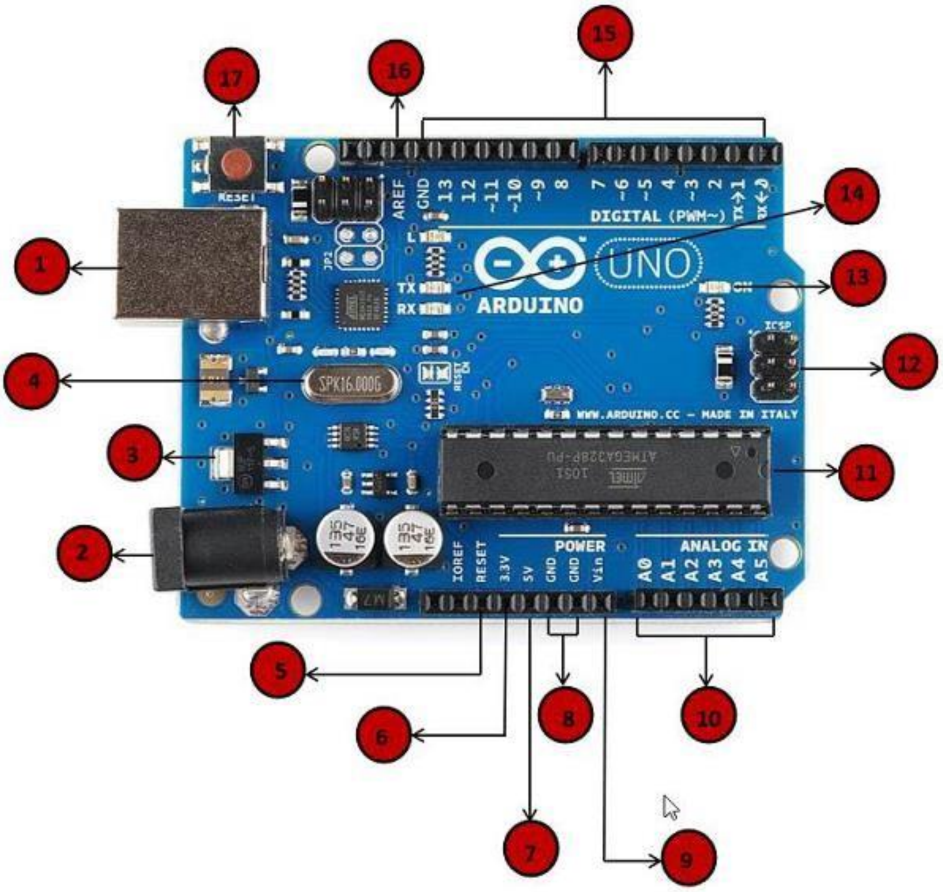
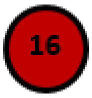


FIGURE:4.18 PIN DISCRIPTION OF ARDUINO UNO

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital

	value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1)

	or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Liquid Cristal Display

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCDs connected to the contollers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines(RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

Description Of 16x2

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all-Parallel Ports. It however doesn't show the use of the Status Port as an input. So what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

Schematic diagram

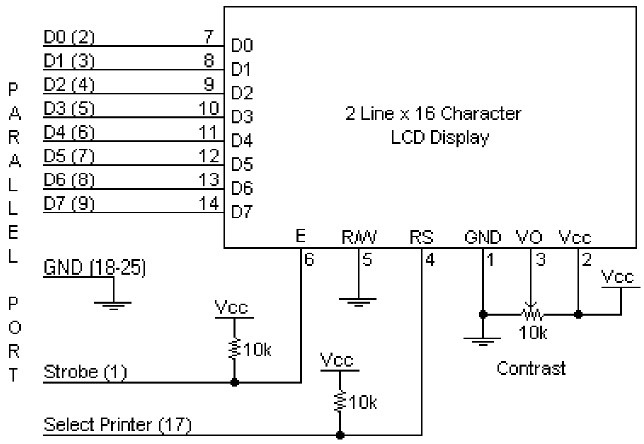


FIGURE:4.19 Schamatic diagram of lcd

- Above is the quite simple schematic. The LCD panel's *Enable* and *Register Select* is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of w
- hich may have no internal pull up resistors.
- We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if

the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

- The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.

16 x 2 Alphanumeric LCD Module Features:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

TABLE:4.5 16 x 2 Alphanumeric LCD Module Specifications

Pin	Symbol	Level	Function
1	V_{SS}	-	Power, GND
2	V_{DD}	-	Power, 5V
3	Vo	-	Power, for LCD Drive
4	RS	H/L	Register Select Signal H: Data Input L: Instruction Input

5	R/W	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)
6	E	H,H->L	Enable
7-14	DB0-DB7	H/L	Data Bus; Software selectable 4- or 8-bit mode
15	NC	-	**NOT CONNECTED**
16	NC	-	**NOT CONNECTED**

FEATURES

- 5 x 8 dots with cursor
- Built-in controller (KS 0066 or Equivalent)
- + 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)
- N.V. optional for + 3V power supply

Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

Figure :4.20 Address locations for a 1x16 line LCD

Even limited to character based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8,16,20,24,32 and 40 characters are all standard, in one, two and four line versions.

Several different LC technologies exists. “supertwist” types, for example, offer Improved contrast and viewing angle over the older “twisted nematic” types. Some modules are available with back lighting, so that they can be viewed in dimly-lit conditions. The back lighting may be either “electro-luminescent”, requiring a high voltage inverter circuit, or simple LED illumination.

PIN DESCRIPTION

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections).

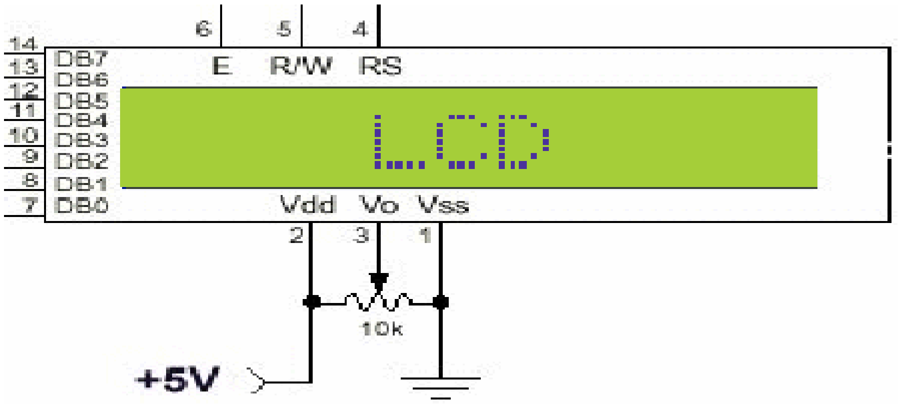


Figure 4.21 Pin diagram of 1x16 lines LCD

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Figure 4.22 Pin specifications

CONTROL LINE

EN: Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS: Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW: Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands, so RW will almost always be low. Finally, the data bus consists of 4 or

8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Logic status on control lines

- E - 0 Access to LCD disabled
- 1 Access to LCD enabled
- R/W - 0 Writing data to LCD
- 1 Reading data from LCD
- RS - 0 Instructions
- 1 Character

Writing data to the LCD

1) Set R/W bit to low
2) Set RS bit to logic 0 or 1 (instruction or character)
3) Set data to data lines (if it is writing)
4) Set E line to high
5) Set E line to low

Read data from data lines (if it is reading) on LCD

1) Set R/W bit to high
2) Set RS bit to logic 0 or 1 (instruction or character)
3) Set data to data lines (if it is writing)
4) Set E line to high
5) Set E line to low

Entering Text

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the d.i.l. switch pack with a of re-wiring is necessary.

LCD Commands

There are some present commands instructions in LCD, which we need to send to LCD through some microcontroller. Some important command instructions are given below:

Table 4.6 Commands

Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left
0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

Fingerprint Sensor

The fingerprint sensor can read different fingerprints and store in its own flash memory. The sensor can perform three functions namely Add(Enroll) , Empty Database or search database and return the ID of stored fingerprint. Any of three functions can be called simply by making the pin low of the sensor or pressing onboard three switches. The response is either error or ok which is indicated by onboard LED. The response is also returned as single serial data byte.

The return byte is a valid ID or error code. The response byte is a single byte at 9600 bps thus making whole sensor very easy to use. We have provided indicating LEDs and function switch already so it's ready to use when you receive it. Just give power and start using the sensor using onboard switches. Then you can move on making external application using these functions.

Inputs and Outputs of Sensor

Input: Two ways to trigger the function of fingerprint sensor

1) Onboard switch: Add, Empty or Search.

2) Make pin low from external microcontroller for 5ms as per function required to be executed.

Outputs(Response): Two ways to monitor output response after a function is executed

1) Onboard LEDs: ERROR or OK

2) Read byte after executing function

Types of function

There are namely three functions you can call for the fingerprint sensor. We will see each in brief.

Add(Enroll) Function: Adds a fingerprint to database and return a byte of newly added ID. Return values are from 0x00 to 0xFE. In case of error like no finger placed, return code is 0xFF. Here 0xFF means error executing function

Search Function: When a finger is put and search funtion is called, it returns a matching ID if found in its existing memory. Return values are from 0x00 to 0xFE. In case of error like no finger placed, return code is 0xFF. Here 0xFF means error executing function.

Empty Function: When you wish to empty all fingerprint data stored on sensor you can use thisfunction. After executing this function, you will get 0xCC as OK or 0xFF in case of error.

About Fingerprint Recognition Sensor Module

The sensor is a solid-state fingerprint sensor that reliably captures fingerprint information. It is designed to integrate into devices for improved security and convenience. The sensor provides a reliable, quick and user-friendly alternative to passwords, PIN's and other forms of user authentication.

This fingerprint scanner is capable of gathering and storing unique finger prints. Simply hold your finger on the optical scanner, query the device over serial, and you will be issued a unique ID. Use that ID within your embedded system to determine access levels, time clocks, door locks, etc. Unit includes 4 pin connector cable to connect and read to controller. The outputs is TTL level serial data.

A biometric sensor, fingerprint sensor to be specific, also known as the fingerprint reader, is a fingerprint image capture device, the very front end of the biometric fingerprint identification/verification module. The fingerprint sensor captures the fingerprint images, matches the uniqueness of each print read by the sensor and compares it to the one stored in its module or local system database.

It consists of optical fingerprint sensor, high performance DSP processor and Flash. It boasts of functions such as fingerprint enrollment, fingerprint deletion, fingerprint verification, fingerprint upload, fingerprint download, etc.

3.3.4.2 Applications

- Computer peripherals – improves security and convenience
- Transportation systems – validation of operators, drivers and inspectors
- Medical equipment – authorization of operator or technician
- Physical access systems – approval for entry
- Kiosks and vending machines – confirmation of person receiving the selection
- Point of Sale terminals – authentication of tellers and cashiers

Specification

- Fingerprint enrollment time <250ms
- Fingerprint search time <1s（100 fingerprint, average value in test）
- Resolutions 500 DPI
- Security level 5, Adjustable
- Capacity 768 Templates
- FAR <0.0001%
- FRR <0.01%

- Power Supply 4.3V to 6V
- Working Current <80mA
- Peak Current <90mA
- Communication Interface TTL
- Communication Baud Rate 57600 bps
- Working Temperature -10 deg C to +40 deg C
- Working Humidity 40% RH to 85% TH (no dew)
- Module Dimensions 60x21x25 mm (LxWxH)

Electrical Interface

Module is connected to HOST via 4 PIN cable(provided with module).

Pin Number Pin Name Details

1 +Ve Positive power pin(+), Apply proper filtered DC regulated power (Typical 5V)

2 TXD Module Serial Transmit Output, Open-drain output, need to use pull-up resistance to supply in application

(Typical value: 10KΩ)

3 RXD Module Serial Receive Input, Wide voltage input up to 7V (TTL level)

4 GND Power supply ground

Note: The PIN close to the edge of circuit board is PIN4: Ground

Relays

A relay is an electrically operated switch. These are remote control electrical switches that are controlled by another switch, such as a horn switch or a computer as in a power train control module, devices in industries, home based applications. Relays allow a small current pin, 4-pin, 5-pin, and 6-pin, single switch or dual switches. Relays are used throughout the automobile. Relays which come in assorted sizes, ratings, and applications, are used as remote control switches. A typical vehicle can have 20 relays or more.

basics on relay handling

- To maintain initial performance, care should be taken to avoid dropping or hitting the relay.
- Under normal use, the relay is designed so that the case will not detach. To maintain initial performance, the case should not be removed. Relay characteristics cannot be guaranteed if the case is removed.
- Use of the relay in an atmosphere at standard temperature and humidity with minimal amounts of dust, SO 2, H 2 S, or organic gases is recommended.
- Please avoid the use of silicon-based resins near the relay, because doing so may result in contact failure. (This applies to plastic sealed type relays, too.)
- Care should be taken to observe correct coil polarity (+, –) for polarized relays.
- Proper usage requires that the rated voltage be impressed on the coil. Use rectangular waves for DC coils and sine waves for AC coils.
- Be sure the coil impressed voltage does not continuously exceed the maximum allowable voltage.
- Absolutely avoid using switching voltages and currents that exceed the designated values.
- The rated switching power and life are given only as guides. The physical phenomena at the contacts and contact life greatly vary depending on the type of load and the operating conditions. Therefore, be sure to carefully check the
- type of load and operating conditions before use.
- Do not exceed the usable ambient temperature values listed in the catalog.
- Use the flux-resistant type or sealed type if automatic soldering is to be used.
- Use alcohol based cleaning solvents when cleaning is to be performed using a sealed relay
- Avoid ultrasonic cleaning of all types of relays.
- Avoid bending terminals, because it may cause malfunction.
- As a guide, use a Faston mounting pressure of 40 to 70N {4 to 7kgf}for relays with tab terminals.

A relay is used to isolate one electrical circuit from another. It allows a low current control circuit to make or break an electrically isolated high current circuit path. The basic relay consists of a coil and a set of contacts. The most common relay coil is a length of magnet wire wrapped around a metal core. When voltage is applied to the coil, current passes through the wire and creates a magnetic field. This magnetic field pulls the contacts together and holds them there until the current flow in the coil has stopped. The diagram below shows the parts of a simple relay.

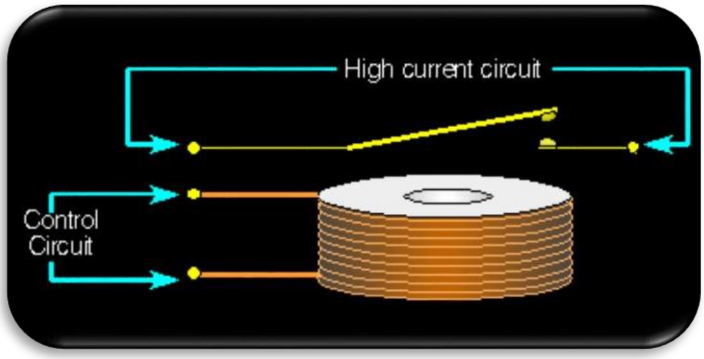


Figure:4.23 Relay

Operation

When a current flows through the coil, the resulting magnetic field attracts an armature that is mechanically linked to a moving contact. The movement either makes or breaks a connection with a fixed contact. When the current is switched off, the armature is usually returned by a spring to its resting position shown in figure 6.6(b). Latching relays exist that require operation of a second coil to reset the contact position.

By analogy with the functions of the original electromagnetic device, a solid-state relay operates a thyristor or other solid-state switching device with a transformer or light-emitting diode to trigger it.

Pole and throw

Since relays are switches the terminology applied to switches is also applied to relays. A relay will switch one or more *poles*, each of whose contacts can be thrown by energizing the coil in one of three ways

- Normally-open (**NO**) contacts connect the circuit when the relay is activated; the circuit is disconnected when the relay is inactive. It is also called a **Form A** contact or "make" contact.

- Normally-closed (**NC**) contacts disconnect the circuit when the relay is activated; the circuit is connected when the relay is inactive. It is also called a **Form B** contact or "break" contact.
- Change-over (**CO**), or double-throw (**DT**), contacts control two circuits: one normally-open contact and one normally-closed contact with a common terminal. It is also called a **Form C** contact or "transfer" contact ("break before make"). If this type of contact utilizes a "make before break" functionality, then it is called a **Form D** contact.

SPST RELAY

SPST relay stands for Single Pole Single Throw relay. Current will only flow through the contacts when the relay coil is energized.

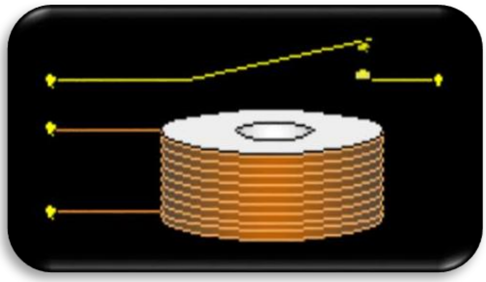


Figure:4.24 SPST Relay

SPDT Relay

SPDT Relay stands for Single Pole Double Throw relay. Current will flow between the movable contact and one fixed contact when the coil is De-energized and between the movable contact and the alternate fixed contact when the relay coil is energized. The most commonly used relay in car audio, the Bosch relay, is a SPDT relay.

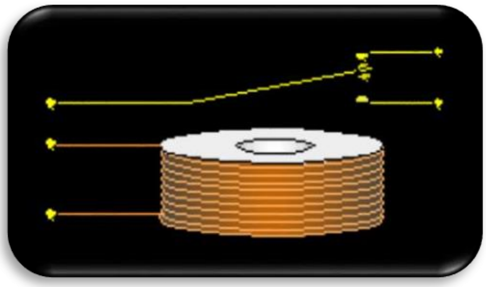


Figure:4.25 SPDT Relay

DPST Relay

DPST relay stands for Double Pole Single Throw relay. When the relay coil is energized, two separate and electrically isolated sets of contacts are pulled down to make contact with their stationary counterparts. There is no complete circuit path when the relay is De-energized.

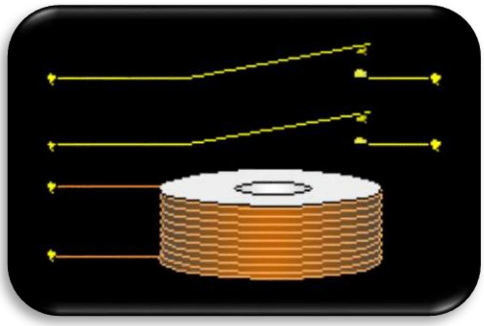


Figure:4.26 DPST Relay

DPDT Relay

DPDT relay stands for Double Pole Double Throw relay. It operates like the SPDT relay but has twice as many contacts. There are two completely isolated sets of contacts.

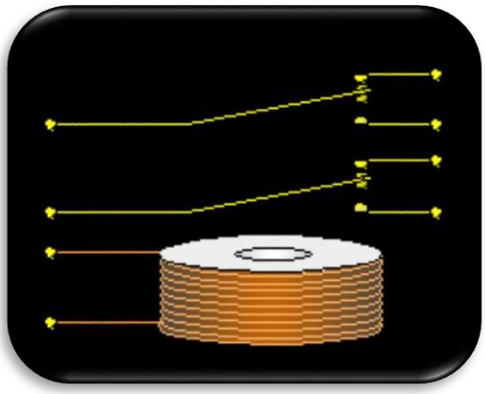


Figure:4.27 DPDT Relay

This is a 4 Pole Double Throw relay. It operates like the SPDT relay but it has 4 sets of isolated contacts.

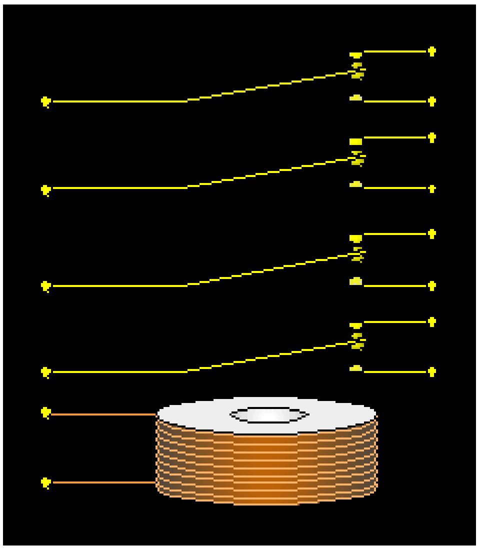


Figure: 4.28 4 Pole Double Throw relay

Types of relay:

1. Latching Relay
2. Reed Relay
3. Mercury Wetted Relay
4. Machine Tool Relay
5. Solid State Relay (SSR)

Latching relay

Latching relay, dust cover removed, showing pawl and ratchet mechanism. The ratchet operates a cam, which raises and lowers the moving contact arm, seen edge-on just below it. The moving and fixed contacts are visible at the left side of the image.

A **latching relay** has two relaxed states (bi-stable). These are also called "impulse", "keep", or "stay" relays. When the current is switched off, the relay remains in its last state. This is achieved with a solenoid operating a ratchet and cam mechanism, or by having two opposing coils with an over-center spring or permanent magnet to hold the armature and contacts in position while the coil is relaxed, or with a remanent core. In the ratchet and cam example, the first pulse to the coil turns the relay on and the second pulse turns it off. In the two coil example, a pulse to one coil turns the relay on and a pulse to the opposite coil turns the relay off. This type of relay has the advantage that it consumes power only for an instant, while it is being switched, and it retains its last setting across a power outage. A remanent core latching relay requires a current pulse of opposite polarity to make it change state.

Figure:4.29 Latching relay

Reed relay

A reed relay has a set of contacts inside a vacuum or inert gas filled glass tube, which protects the contacts against atmospheric corrosion. The contacts are closed by a magnetic field generated when current passes through a coil around the glass tube. Reed relays are capable of faster switching speeds than larger types of relays, but have low switch current and voltage ratings.

Mercury-wetted relay

A **mercury-wetted reed relay** is a form of reed relay in which the contacts are wetted with mercury. Such relays are used to switch low-voltage signals (one volt or less) because of their low contact resistance, or for high-speed counting and timing applications where the mercury eliminates contact bounce. Mercury wetted relays are position-sensitive and must be mounted vertically to work properly. Because of the toxicity and expense of liquid mercury, these relays are rarely specified for new equipment. See also mercury switch.

Machine tool relay

A **machine tool relay** is a type standardized for industrial control of machine tools, transfer machines, and other sequential control. They are characterized by a large number of contacts (sometimes extendable in the field) which are easily converted from normally-open to normally-closed status, easily replaceable coils, and a form factor that allows compactly installing many relays in a control panel. Although such relays once were the backbone of automation in such industries as automobile assembly, the programmable logic controller (PLC) mostly displaced the machine tool relay from sequential control applications.

Solid-state relay

A solid state relay (SSR) is a solid state electronic component that provides a similar function to an electromechanical relay but does not have any moving components, increasing long-term reliability. With early SSR's, the tradeoff came from the fact that every transistor has a small voltage drop across it. This voltage drop limited the amount of current a given SSR could handle. As transistors improved, higher current SSR's, able to handle 100 to 1,200 Amperes, have become commercially available. Compared to electromagnetic relays, they may be falsely triggered by transients.

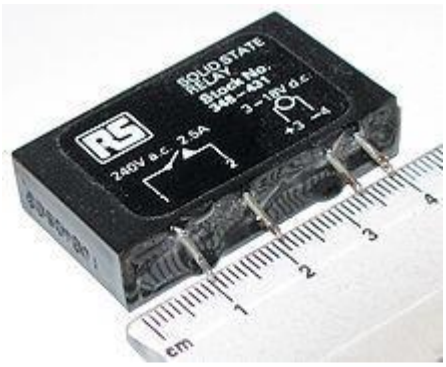


Figure: 4.30 Solid relay, which has no moving parts

Specification

- Number and type of contacts – normally open, normally closed, (double-throw)
- Contact sequence – "Make before Break" or "Break before Make". For example, the old style telephone exchanges required Make-before-break so that the connection didn't get dropped while dialing the number.
- Rating of contacts – small relays switch a few amperes, large contactors are rated for up to 3000 amperes, alternating or direct current

- Voltage rating of contacts – typical control relays rated 300 VAC or 600 VAC, automotive types to 50 VDC, special high-voltage relays to about 15 000 V
- Coil voltage – machine-tool relays usually 24 VAC, 120 or 250 VAC, relays for switchgear may have 125 V or 250 VDC coils, "sensitive" relays operate on a few milli-amperes

Applications:

Relays are used:

- To control a high-voltage circuit with a low-voltage signal, as in some types of modems,
- To control a high-current circuit with a low-current signal, as in the starter solenoid of an automobile,
- To detect and isolate faults on transmission and distribution lines by opening and closing circuit breakers (protection relays),
- To isolate the controlling circuit from the controlled circuit when the two are at different potentials, for example when controlling a mains-powered device from a low-voltage switch. The latter is often applied to control office lighting as the low voltage wires are easily installed in partitions, which may be often moved as needs change. They may also be controlled by room occupancy detectors in an effort to conserve energy,
- To perform logic functions. For example, the boolean AND function is realized by connecting relay contacts in series, the OR function by connecting contacts in parallel. Due to the failure modes of a relay compared with a semiconductor, they are widely used in safety critical logic, such as the control panels of radioactive waste handling machinery.
- As oscillators, also called vibrators. The coil is wired in series with the normally closed contacts. When a current is passed through the relay coil, the relay operates and opens the contacts that carry the supply current. This stops the current and causes the contacts to close again. The cycle repeats continuously, causing the relay to open and close rapidly. Vibrators are used to generate pulsed current.
- To generate sound. A vibrator, described above, creates a buzzing sound because of the rapid oscillation of the armature. This is the basis of the electric bell, which consists of a vibrator with a hammer attached to the armature so it can repeatedly strike a bell.

➢ To perform time delay functions. Relays can be used to act as an mechanical time delay device by controlling the release time by using the effect of residual magnetism by means of a inserting copper disk between the armature and moving blade assembly.

Switches

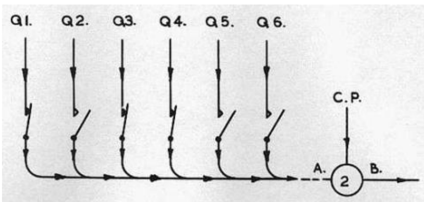
Introduction

A switch is an electrical component that can break an electrical circuit, interrupting the current or diverting it from one conductor to another. The most familiar form of switch is a manually operated electromechanical device with one or more sets of electrical contacts. Each set of contacts can be in one of two states: either 'closed' meaning the contacts are touching and electricity can flow between them, or 'open', meaning the contacts are separated and non-conducting.

Figure:4.31 switches

CONSTRUCTION

Keys are placed for giving instructions to the particular part. For example if need to operate a robot .keys are placed and each key is giving an instruction through which the robot moves. If we need to operate the



Switches with larger numbers of poles or throws can be described by replacing the "S" or "D" with a number or in some cases the letter "T" (for "triple"). In the rest of this article the terms SPST, SPDT and intermediate will be used to avoid the ambiguity in the use of the word "way".

WORKING

A pair of contacts is said to be "closed" when current can flow from one to the other. When the contacts are separated by an insulating air gap, an air space, they are said to be "open", and no current can flow at typical voltages.

Switches are classified according to the arrangement of their contacts in electronics. Electricians installing building wiring use different nomenclature, such as "one-way", "two-way", "three-way" and "four-way" switches

In a push-button type switch, in which the contacts remain in one state unless actuated, the contacts can either be normally open (abbreviated "n.o." or "no") until closed by operation of the switch, or normally closed ("n.c. or "nc") and opened by the switch action. A switch with both types of contact is called a changeover switch. These may be "make-before-break" which momentarily connect both circuits, or may be "break-before-make" which interrupts one circuit before closing the other

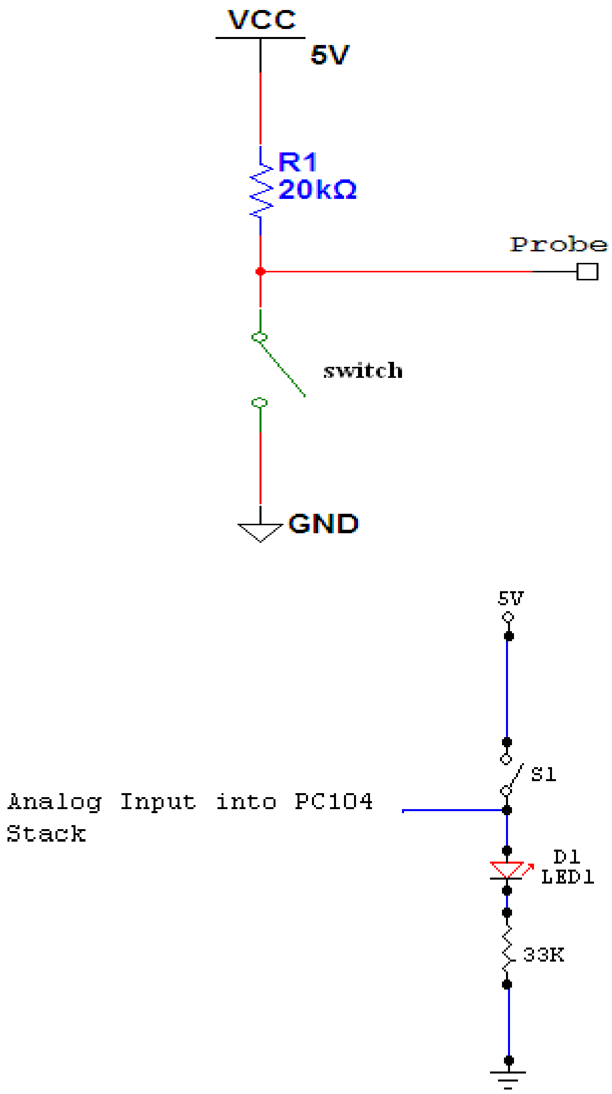


Figure:4.32 circuits of the switch

From the above circuit it explains the working of the switch when supply is given to the circuit the resistor take the current and oppose to the certain extend until the switch is pressed. These switches can be connected parallel or serial in required manner.

We can have resistor, capacitor, light resistor diode etc as the combination of the switch. For example w the reset circuit in which we the resistor and capacitor as the combination.

APPLICATIONS

- They are used for the given instructions to the required device like robot, driving the motor, etc.
- Switches can arrange in the matrix from or in the required from according to the application.
- Reset circuit of the controller, motor also has the switches for its particular functioning.

SOFTWARE REQUIERMENTS

Proteus

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing.

ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.

The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design

Step 1: Open ISIS software and select New design in File menu

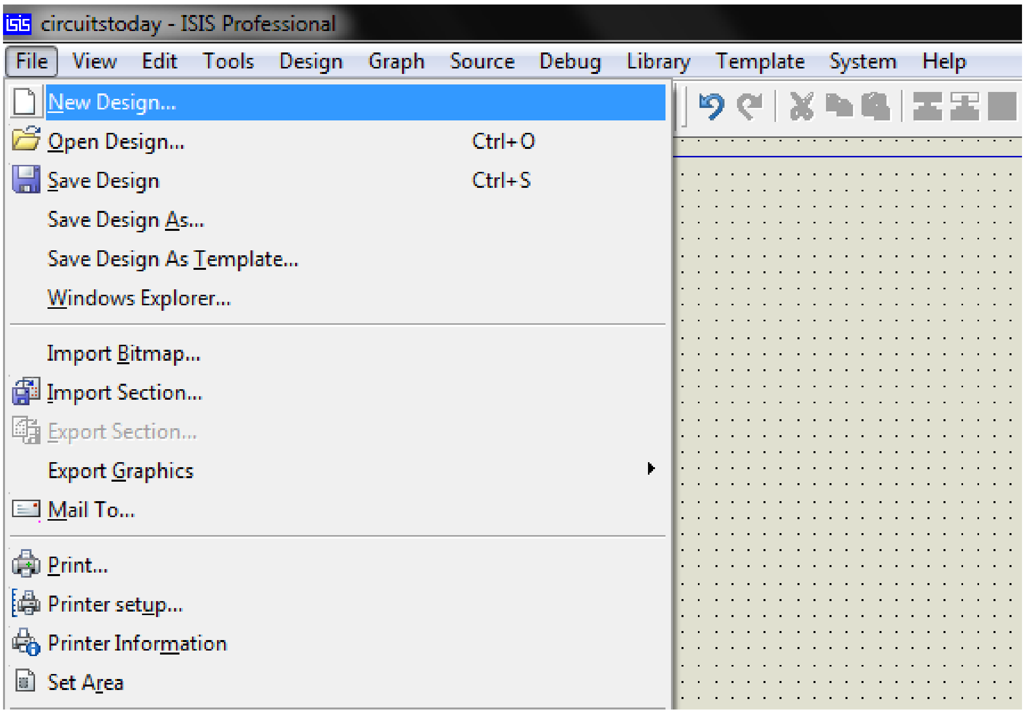


Figure 4.33 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

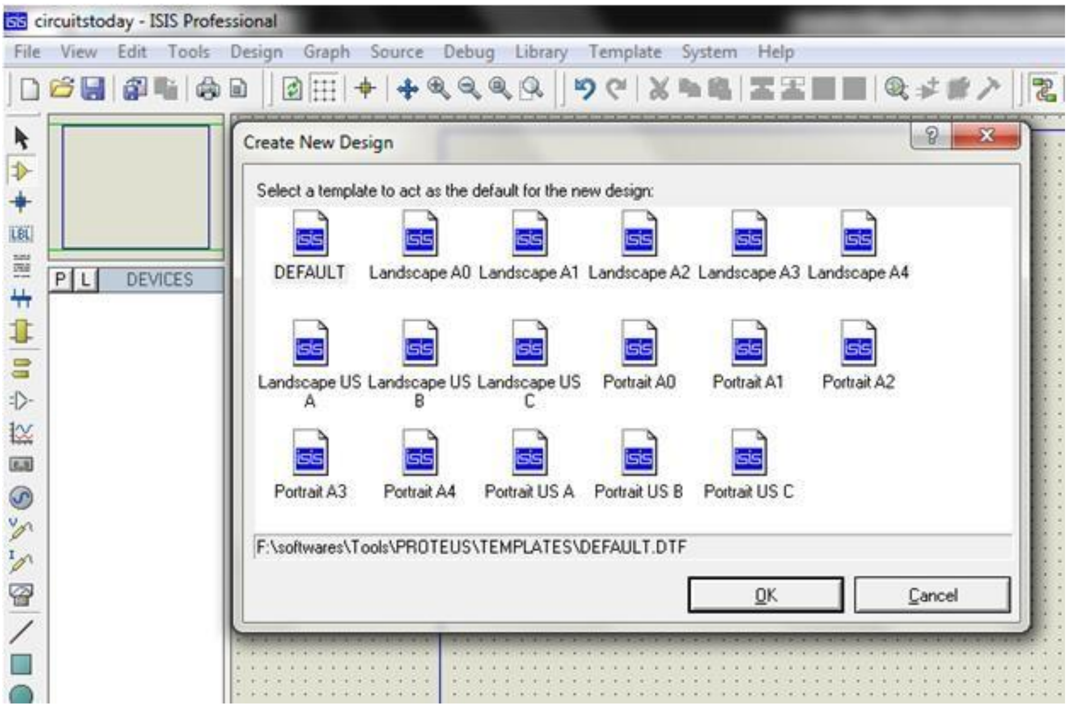


Figure:4.34 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

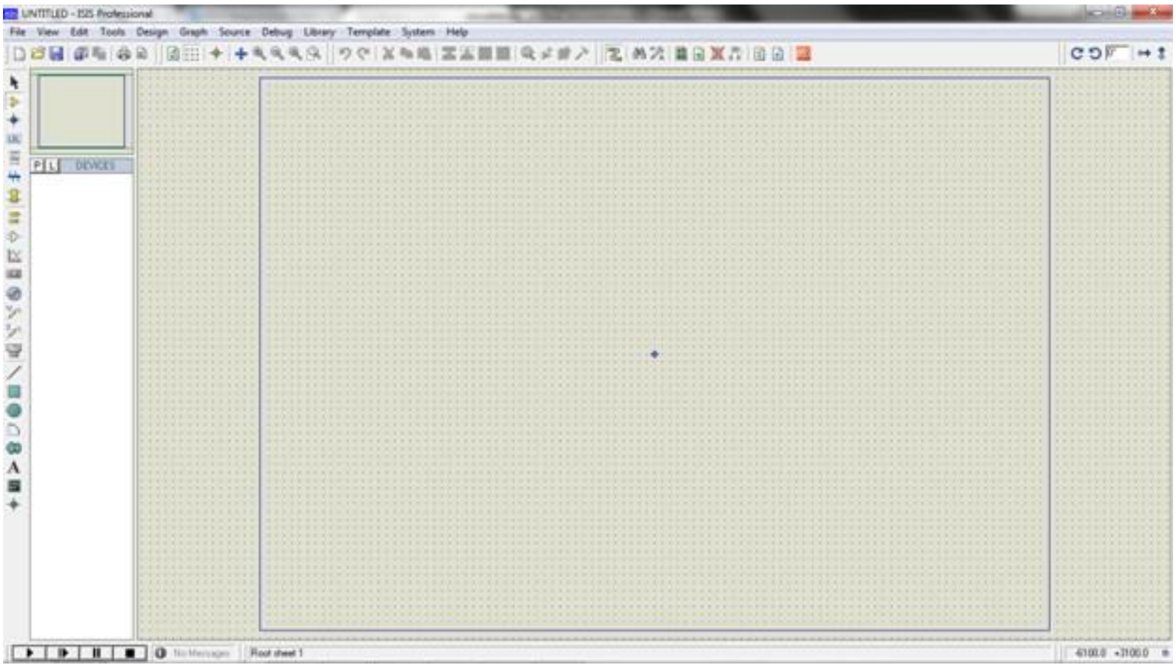


Figure: 4.35 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

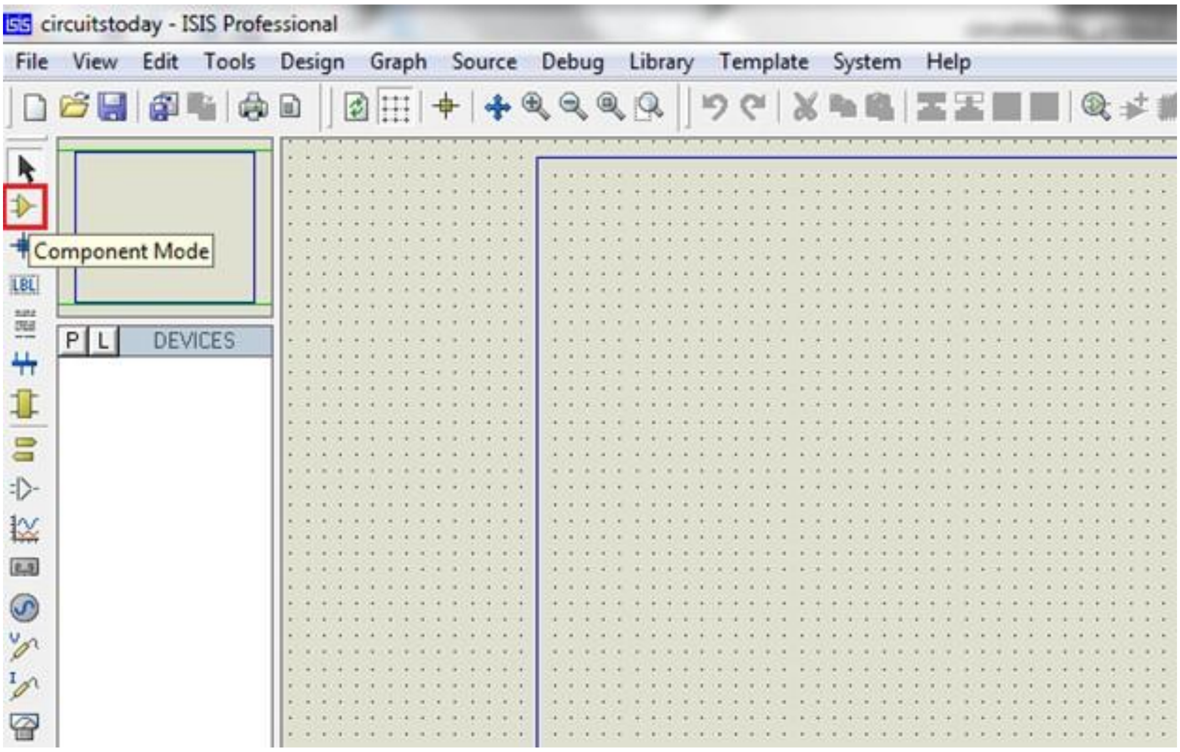


Figure: 4.36 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

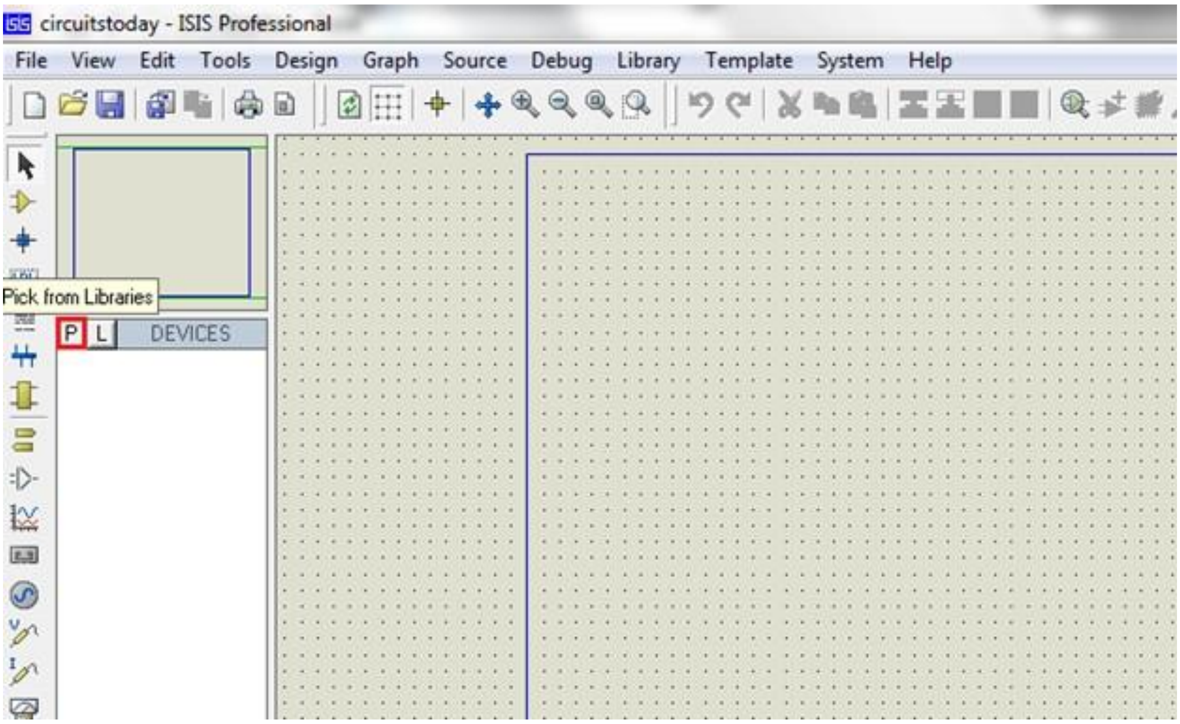


Figure: 4.37 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

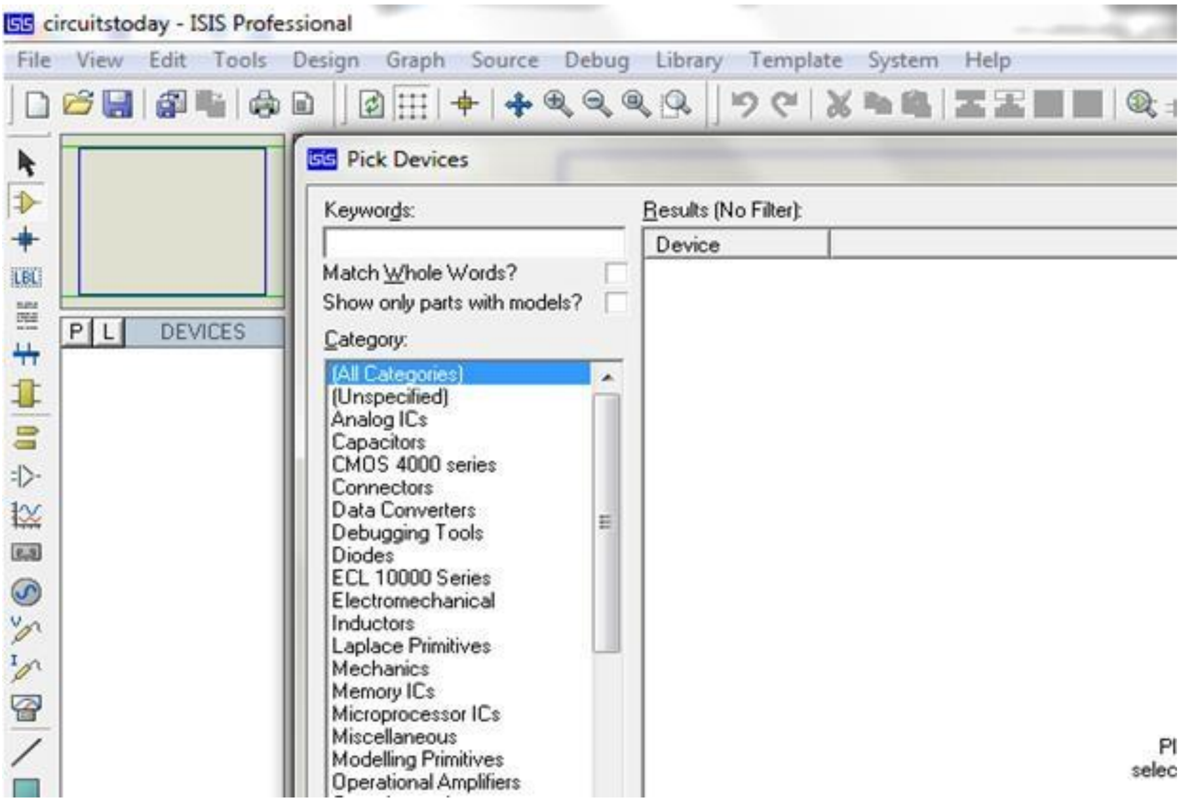


Figure:4.38 Keywords Textbox

Example shows below selection of push button. Select the components accordingly.

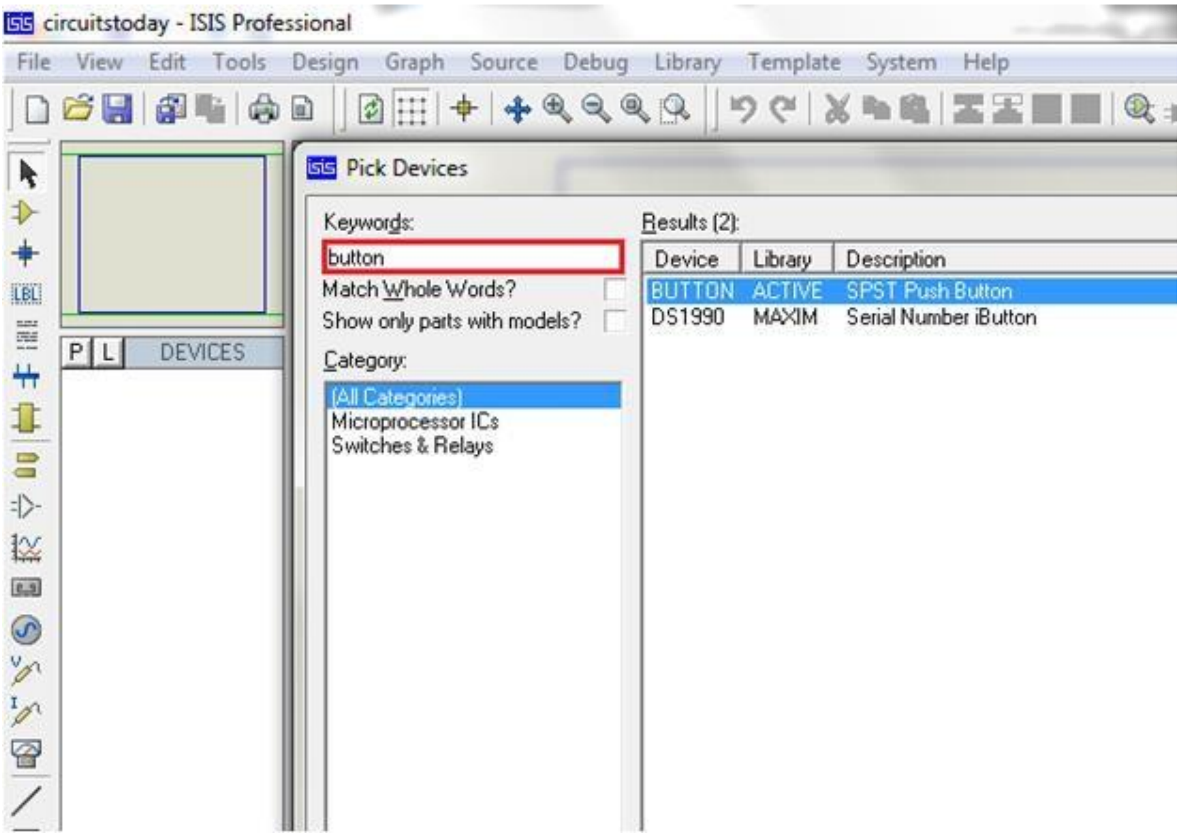


Figure:3.34 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

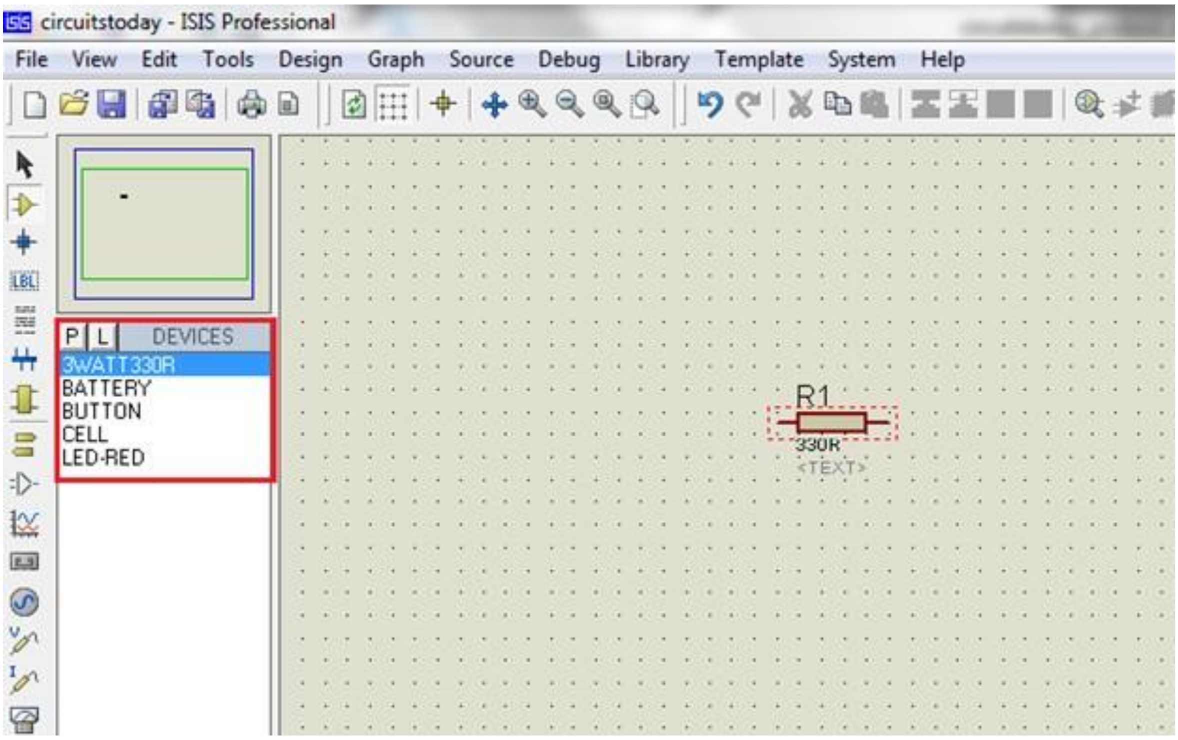


Figure:4.40 Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

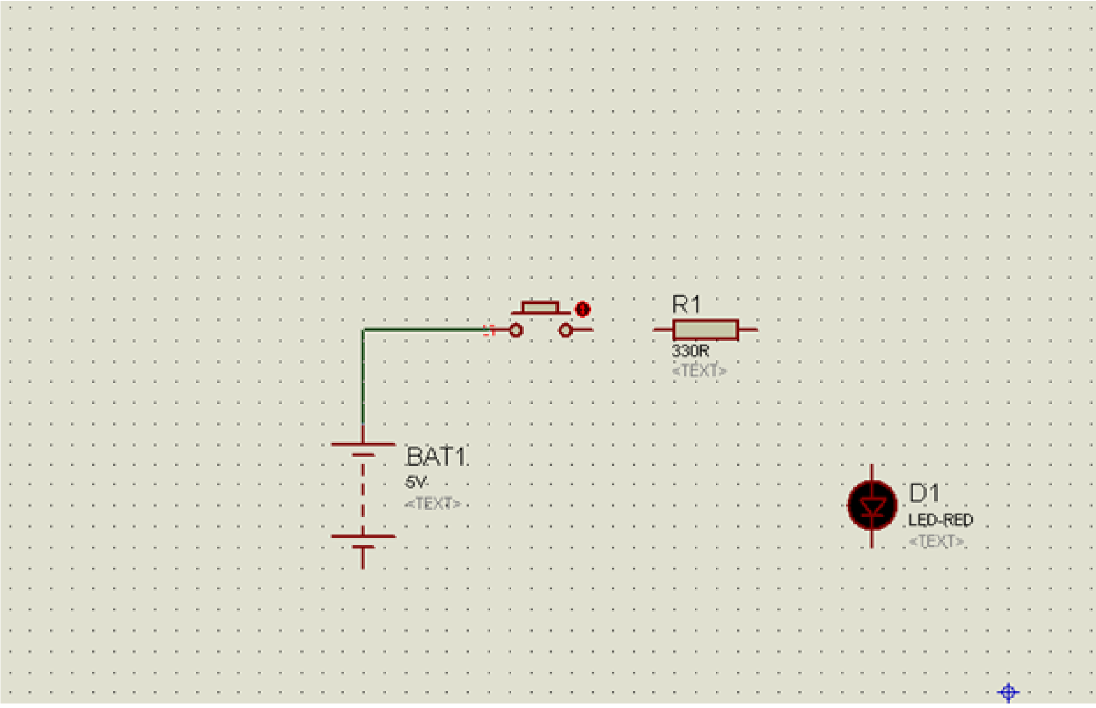


Figure:4.41 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

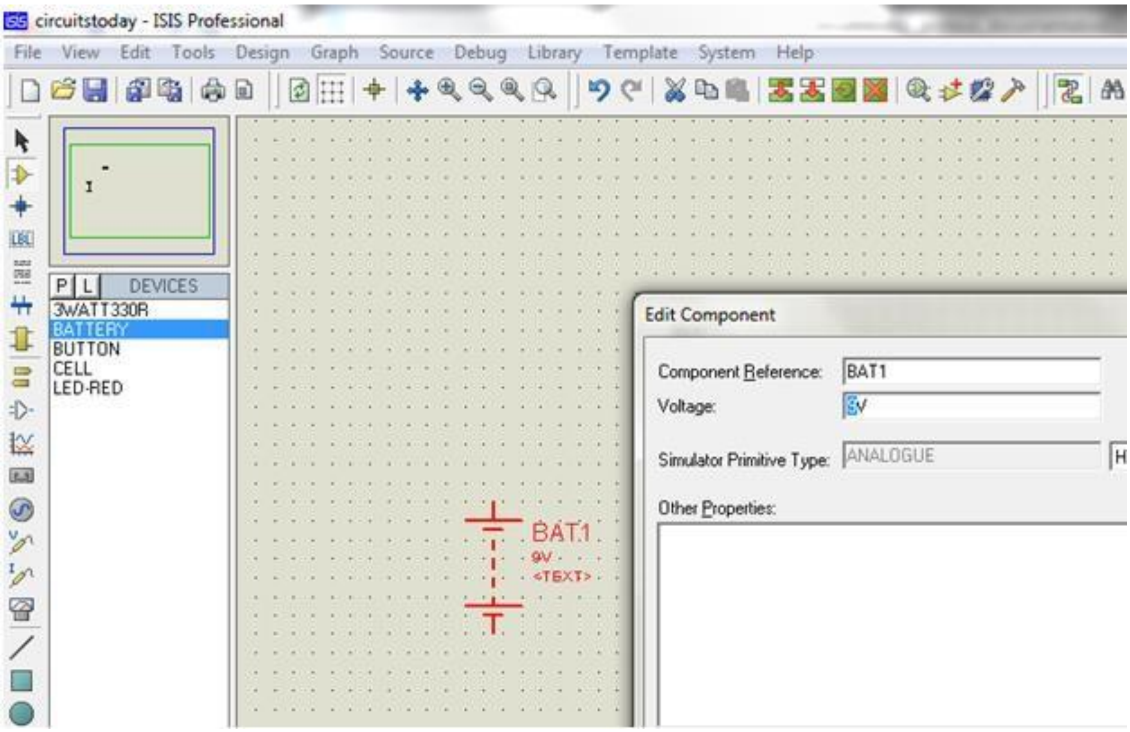


Figure: 4.42 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

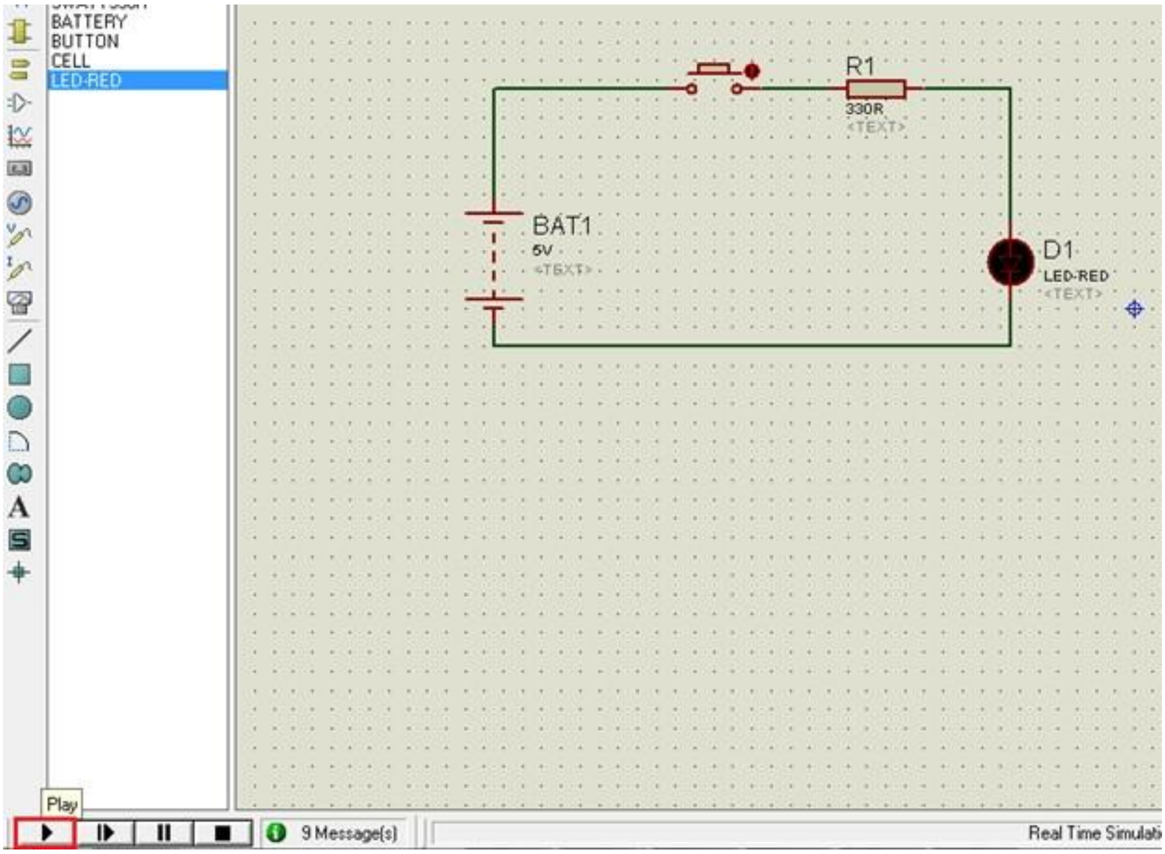


Figure:4.43 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

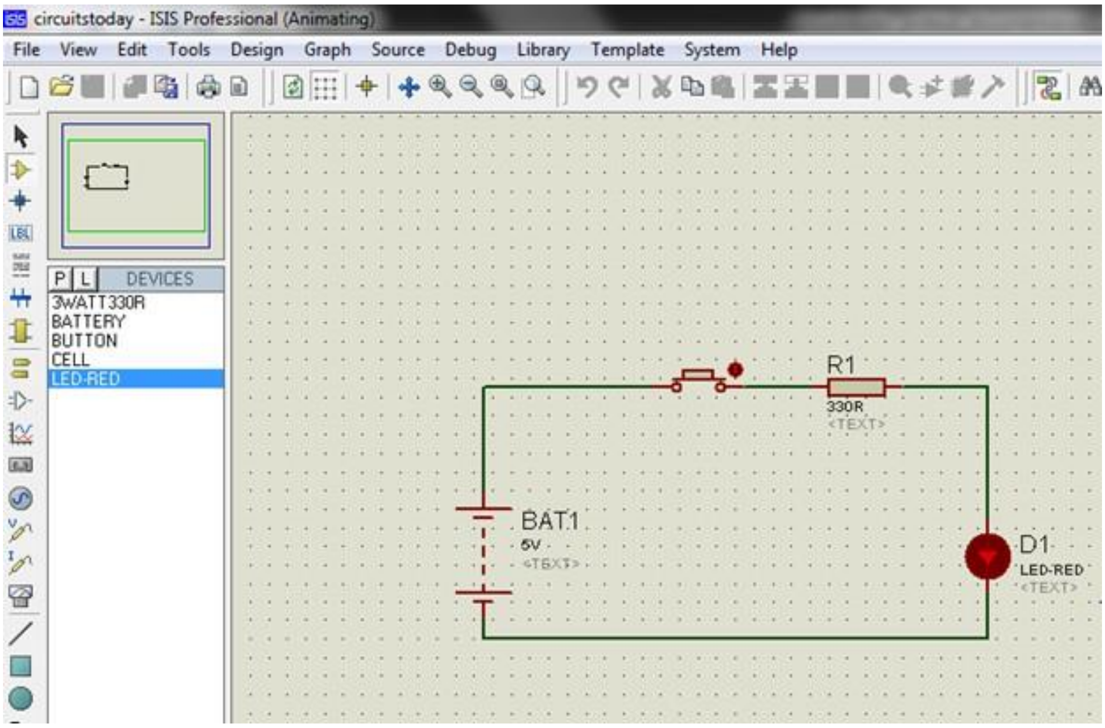


Figure:4.44 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

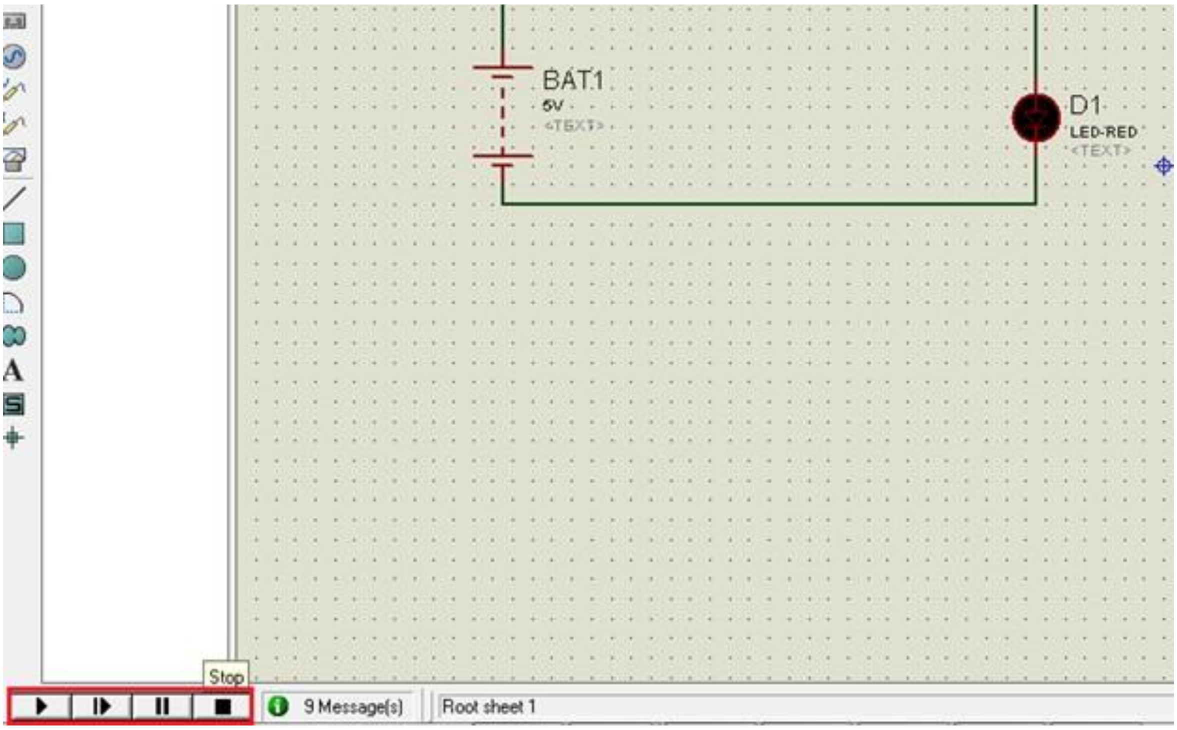


Figure:4.45 Simulation Step-Pause-Stop Buttons

Arduino IDE

The **Arduino Integrated Development Environment (IDE)** is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Arduino Pro IDE	
Developer(s)	Arduino Software
Preview release	v0.0.2 / 28 October 2019; 3 months ago[8]
Repository	github.com/arduino/Arduino
Written in	C, C++
Operating system	Windows, macOS, Linux
Platform	IA-32, x86-64, ARM
Type	Integrated development environment
License	LGPL or GPL license
Website	blog.arduino.cc/2019/10/18/arduino-pro-ide-alpha-preview-with-advanced-features/

In October 2019 the Arduino organization began providing early access to a new Arduino Pro IDE with debugging[9] and other advanced features.

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

Installation

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig:4.46 usb cable

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

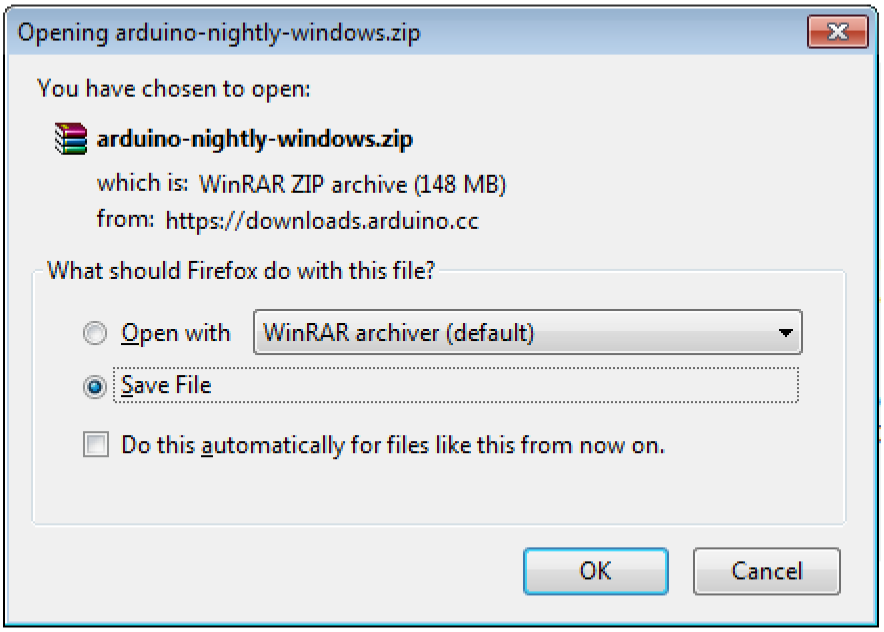


Fig:4.47 down load aurdino ide

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

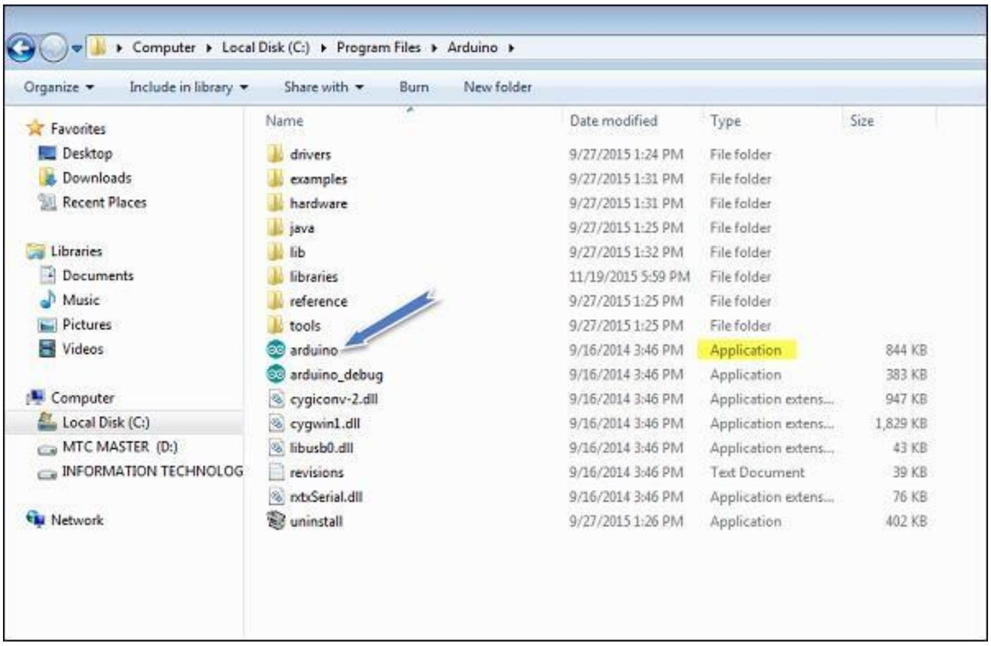


Fig:4.48launching aurdino ide

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.

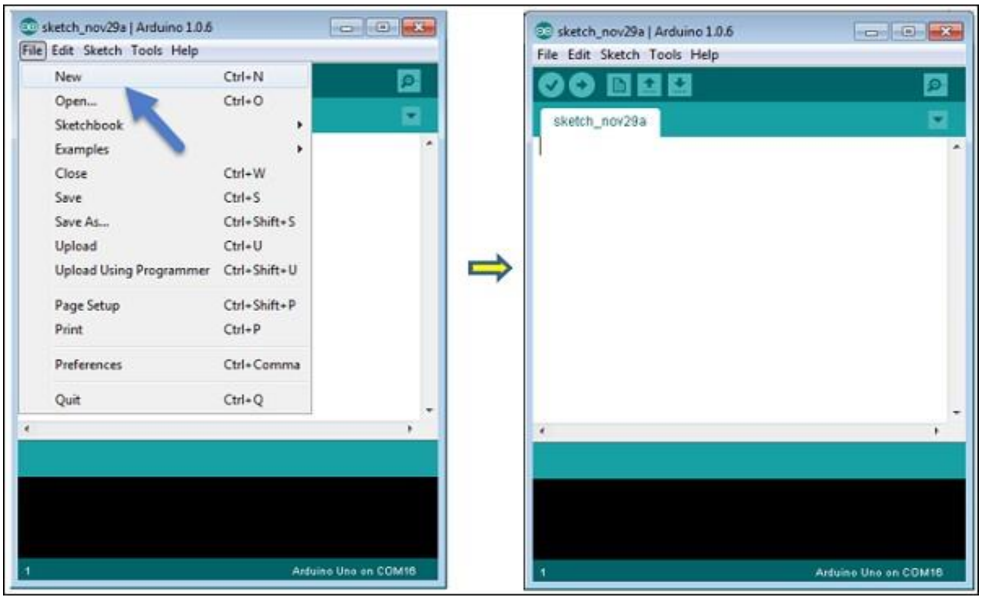


Fig:4.49 opeaning first project

To open an existing project example, select File → Example → Basics → Blink.

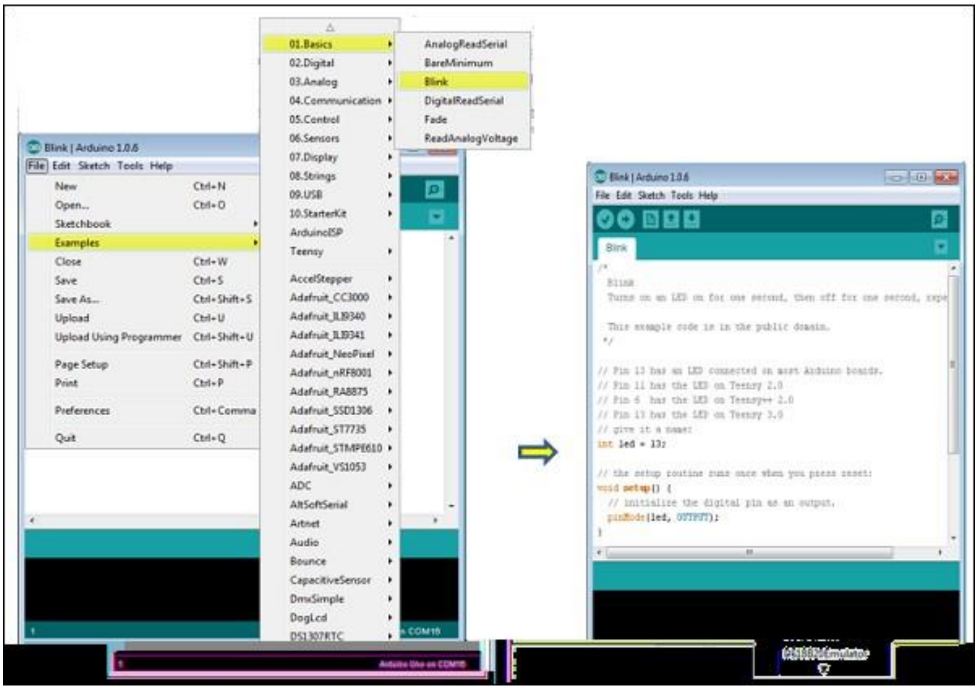


Fig:4.50selecting aurdino board

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

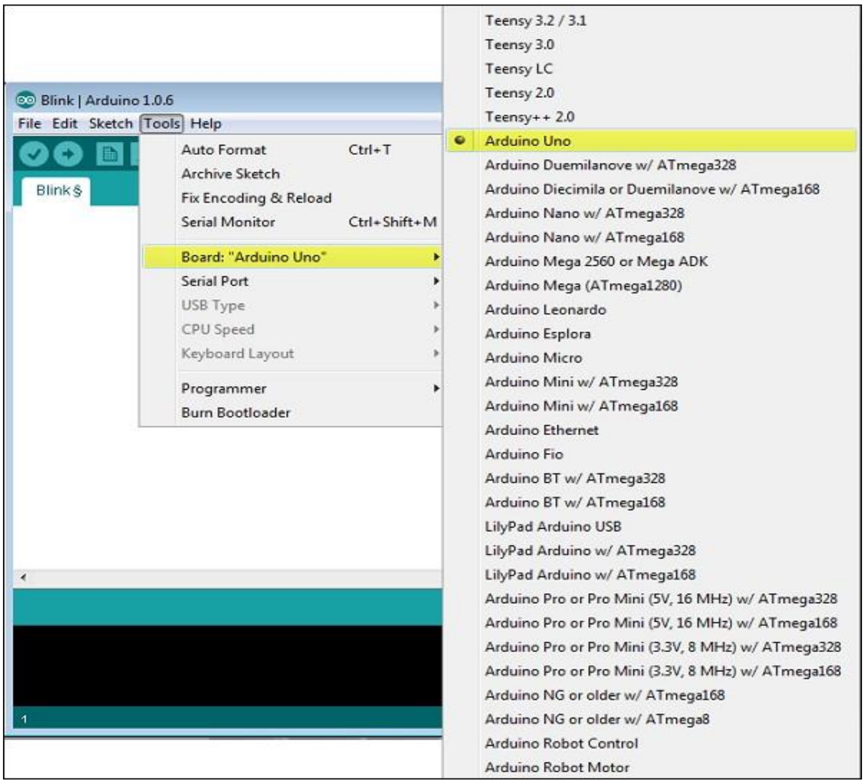


Fig:4.50 selecting aurdino board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

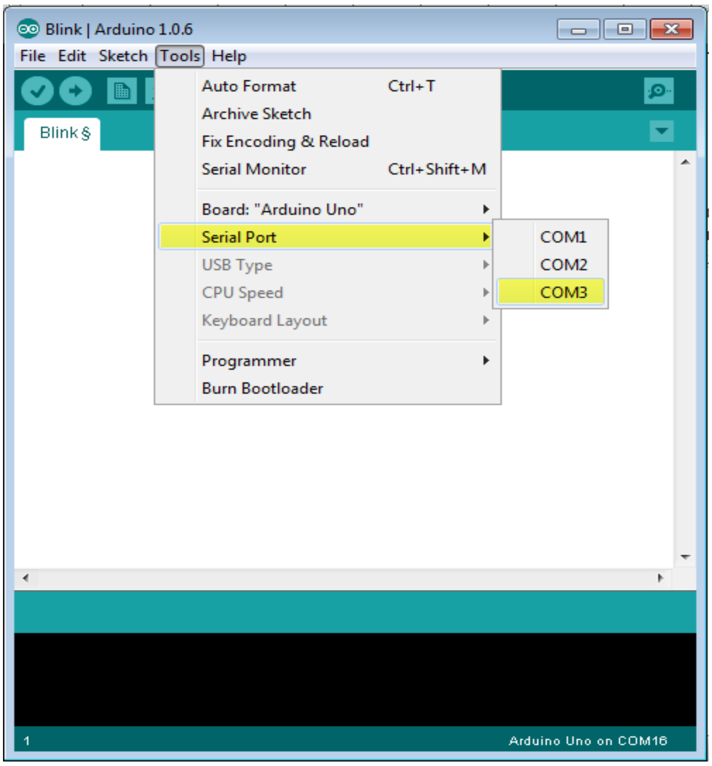


Fig:4.51 selecting serial poart

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

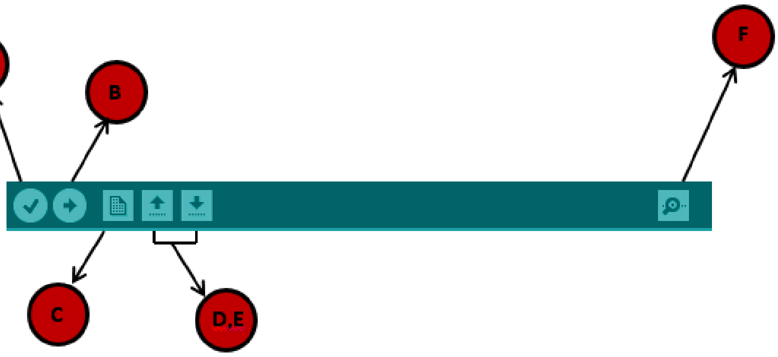


Fig:4.52 upload the program on board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

WORKING

The first we will give the 5v dc supply to give the kit .then switch on the device, then the finger print module will ask to enrolled the finger print . After enrolled the finger print is enrolled will display in LCD display board. Then automatically it will store the finger print. Then we switch on the device then the LCD display will asks the put your finger. After that if the correct person will authenticate the device will be switched The door lock will be opened. After that the unauthorized person will try to switch on the kit the device will not opened .the door will be locked .the buzzer will ringing

CHAPTER 5

RESULTS

RESULTS

The Finger print based device switcher project prototype as shown in below figure below

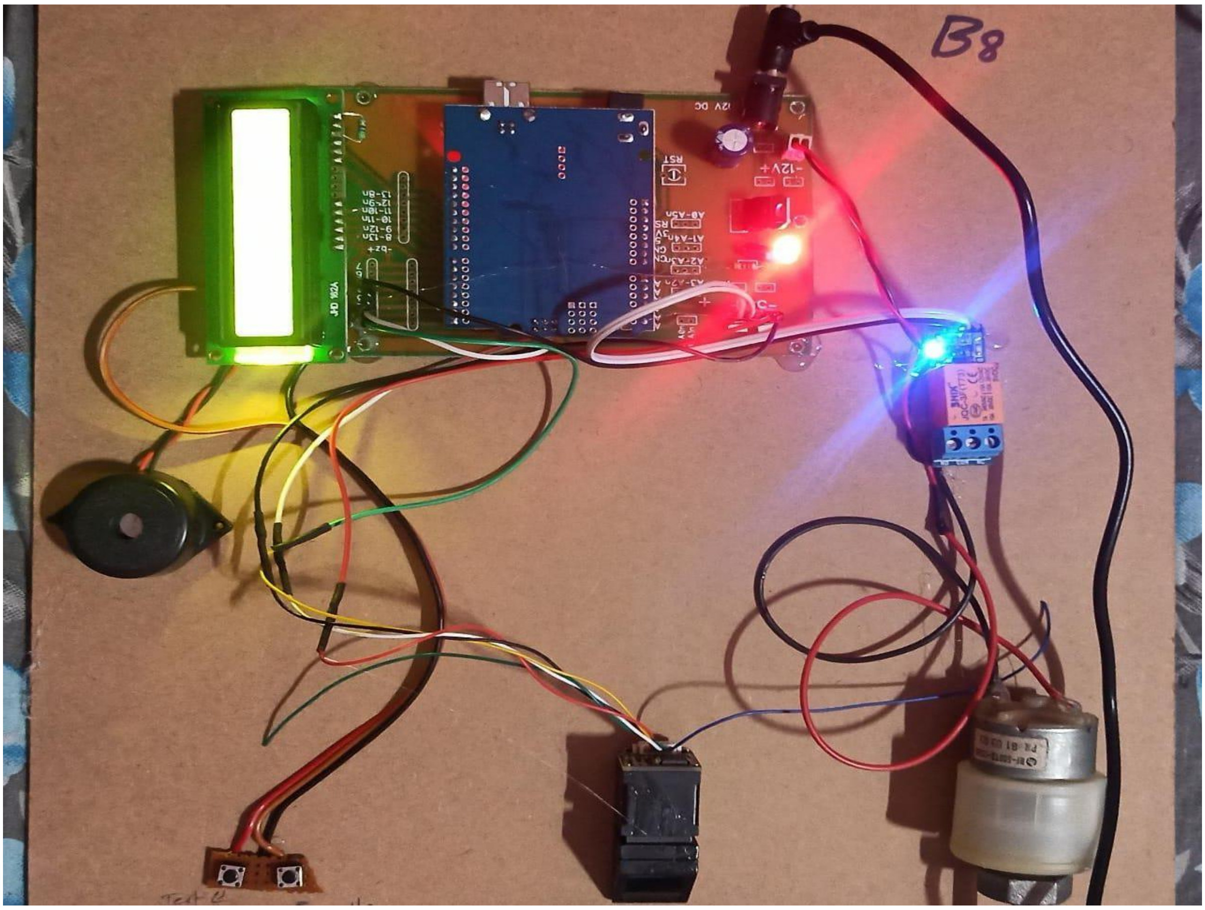


Fig:5.1 Finger print base device switcher proto type

RESULTS

RESULT 1

When the person to store the finger print first he was enrolled the finger print the stored the finger print

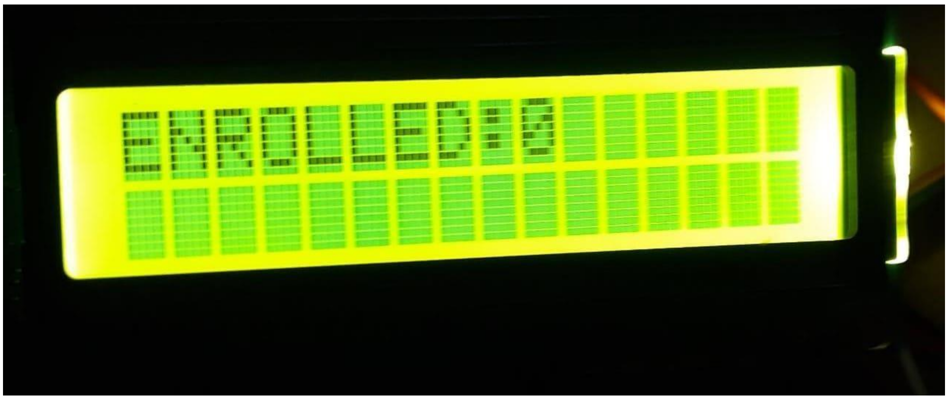


Fig:5.2 Enrolled the finger print

RESULT 2

Here the correct person eill authenticated the finger print then the device will be switched on

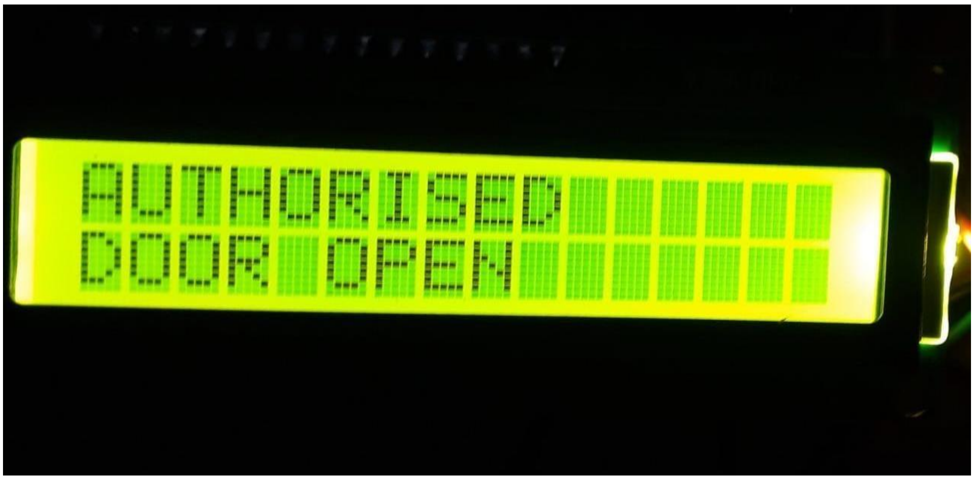


Fig:5.3 the door will opeaned

RESULT 3

When the un authorized person will try to switch on the device then it will automatically turned off and and the door will be locked

Fig:5.4 Door will locked

CHAPTER 6

ADVANTAGES AND DIS ADVANTAGES

Advantages:

- easy to construct
- less cost
- easy to transport
- specified persons to unlock the device
- 3 rd party will not allowed

Disadvantages:

- It will use to unlock fixed places
- It has a low bit rate

Applications

- It is used industrial applications
- Used in home and personal locker systems
- Used to switch any missionary and devices
- Used in automobiles

CHAPTER 7

CONCLUSION AND FUTURE ENHANCE MENT

Future scope:

New fingerprint sensing technologies are constantly being explored and developed. For example, Multispectral Fingerprint Imaging and this finger print of Msi technology more accurate to the conventional optical sensors

Conclusion

This project uses very popular finger print sensing technology for the purpose of electrical switching for security and various other switching purposes. The technology is mainly used for the identification and access control of various switches for identifying individuals that need surveillance or automation.

REFERENCES

[1] Balaji, Speech of Shri V S Sampath, CEC for Defence Estates Day Lecture 2014

[2] R. Haenni, E. Dubuis, and U. Ultes-Nitsche, “Research on e-voting technologies." Bern University of Applied Sciences, Technical Report 5, 2008.

[3] G.V.L. N. Rao. Democracy at Risk! Citizens for Veri_ability, Transparency & Accountability in Elections, New Delhi,

[4] Election Commission of India.

[5] A. K. Agarwala, D. T. Shahani, and P. V. Indiresan. Report of the expert committee for evaluation of the upgraded electronic voting machine (EVM). Sept. 2006.

[6] Wikipedia. Results of the 2009 Indian general election by parliamentary constituency Sanjay Kumar, Manpreet Singh,”DESIGN OF A SECURE ELECTRONIC VOTING SYSTEM USING FINGERPRINT TECHNIQUE”,IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 4, No 1, July 2013

[7] Press Trust of India.

```
/*
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2,3);*/
#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);
char buff[200],k=0;
char res[130];
unsigned char
enroll[12]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X03,0X01,0X00,0X05};  //
ok
unsigned char
generate_ch[13]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X04,0x02,0X01,0X00,
0X08};  //ok
unsigned char
generate_ch1[13]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X04,0x02,0X02,0X0
0,0X09};  //ok
unsigned char un_cmd[12]={0xef,0x01,0xff,0xff,0xff,0xff, 0x01,0x00,0x03,0x05,0x00,0x09
};
unsigned char
store[12]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X06,0X06,0X02,0x00};
//ok
```

```
unsigned char
identify[17]={0xef,0x01,0xff,0xff,0xff,0xff,0x01,0x00,0x08,0x1b,0x01,0x00,0x00,0x01,0x0
1,0x00,0x27};
void serialFlush(){
  while(Serial.available() > 0) {
    char t = Serial.read();
  }
}
int fpenroll(char);
int fpsearch();
int s1=5,s2=6,m1=4,buz=7;
void setup() {
  char ret;
  pinMode(s1, INPUT_PULLUP);
  pinMode(s2, INPUT_PULLUP);
  pinMode(m1, OUTPUT);
 // pinMode(m2, OUTPUT);
  pinMode(buz, OUTPUT);
     digitalWrite(buz,HIGH);
     digitalWrite(m1,LOW);
    // digitalWrite(m2,LOW);
     Serial.begin(9600);
 // mySerial.begin(57600);//57600
  lcd.begin(16,2);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("FINGERPRINT AUTHENTICATED DEVICE
SWITCHERFINGERPRINT AUTHENTICATED DEVICE SWITCHER");
```

```
}
 int err =0;
 int idk = 0,eid=0;
void loop()
{
  //digitalWrite(buz,LOW);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("PUT UR FINGER"); delay(1000);
if(digitalRead(s1) == 0)
{
  //Serial.println("Enrolling");
  lcd.clear();lcd.setCursor(0, 0);lcd.print("ENROLLING..");
  if(fpenroll(eid) == -1)
  {
   // Serial.print("Enroll failed:");Serial.print(err);Serial.println("");
    err=0;
    lcd.clear();lcd.setCursor(0, 0);lcd.print("ENROLL FAILED");
    digitalWrite(buz,LOW);delay(1000);digitalWrite(buz,HIGH);
  }
  else
  {
    lcd.clear();lcd.setCursor(0, 0);lcd.print("ENROLLED:");lcd.print((int)eid);
    //Serial.print("Enroll Success to id:");Serial.print((int)eid);Serial.println("");
    //Serial.print("*E");Serial.print((int)eid);Serial.println("#");
    eid++;
  }
```

```
  delay(2000);
 // lcd.clear();lcd.setCursor(0, 0);lcd.print("SELECT OPTION");
}
if(digitalRead(s2) == 0)//identify
{
  lcd.clear();lcd.setCursor(0, 0);lcd.print("SEARCHING..");
    idk = fpsearch();
    if(idk == -1)
    {
      err=0;lcd.clear();lcd.setCursor(0, 0);lcd.print("UNAUTHOISED ");
      lcd.clear();lcd.setCursor(0, 0);lcd.print("DOOR LOCKED");

        digitalWrite(buz,LOW);
        delay(400);
        digitalWrite(buz,HIGH);
    }
    else
    {
      lcd.clear();lcd.setCursor(0, 0);lcd.print("IDENTIFIED:");lcd.print((int)idk);

      if(idk == 0)
      {
        lcd.clear();lcd.setCursor(0, 0);lcd.print("AUTHORISED ");
        //lcd.clear();
        lcd.setCursor(0, 1);lcd.print("DOOR OPEN");
```

```
  digitalWrite(m1,HIGH);
  delay(2000);
  digitalWrite(m1,LOW);
}
if(idk == 1)
{
   lcd.clear();lcd.setCursor(0, 0);lcd.print("AUTHORISED ");
  //lcd.clear();
  lcd.setCursor(0, 1);lcd.print("DOOR OPEN");
  digitalWrite(m1,HIGH);
  delay(2000);
  digitalWrite(m1,LOW);
}
if(idk == 2)
{
  lcd.clear();lcd.setCursor(0, 0);lcd.print("AUTHORISED ");
  //lcd.clear();
  lcd.setCursor(0, 1);lcd.print("DOOR OPEN");
  digitalWrite(m1,HIGH);
  delay(2000);
  digitalWrite(m1,LOW);
}
if(idk == 3)
{
  lcd.clear();lcd.setCursor(0, 0);lcd.print("AUTHORISED ");
```

```
        //lcd.clear();
        lcd.setCursor(0, 1);lcd.print("DOOR OPEN");
        digitalWrite(m1,HIGH);
        delay(2000);
        digitalWrite(m1,LOW);
      }
     }
    delay(2000);
    //lcd.clear();lcd.setCursor(0, 0);lcd.print("SELECT OPTION");
}

}

int ct=0;
char dummy=0x0f;
int fpenroll(char id)
{

  serialFlush();
  //enroll buffer send 12 bytes
  for(int i =0;i<12;i++)
     Serial.write(enroll[i]);
  res[9] = 1;//
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
```

```
if(res[9] == 0){}
else{err=1;return -1;}

//generate ch buffer
for(int i =0;i<13;i++)
   Serial.write(generate_ch[i]);
res[9] = 1;
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=2;return -1;}

//enroll buffer send 12 bytes
for(int i =0;i<12;i++)
   Serial.write(enroll[i]);
res[9] = 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=3;return -1;}

//generate ch1 buffer
for(int i =0;i<13;i++)
   Serial.write(generate_ch1[i]);
res[9] = 1;
```

```
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=4;return -1;}

//uncmd buffer send 12 bytes
for(int i =0;i<12;i++)
   Serial.write(un_cmd[i]);
res[9] = 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=5;return -1;}

//store buffer send 12 bytes
for(int i =0;i<12;i++)
   Serial.write(store[i]);
dummy = 0x0f+id;
Serial.write((uint8_t)id);
Serial.write((uint8_t)0x00);
Serial.write((uint8_t)dummy);
res[9] = 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){return id;}
```

```
  else{err=6;return -1;}
}
int fpsearch()
{
  ct=0;
   serialFlush();
  //enroll buffer send 12 bytes
  for(int i =0;i<12;i++)
      Serial.write(enroll[i]);
  res[9] = 1;//
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){}
  else{err=1;return -1;}


   //generate ch buffer
  for(int i =0;i<13;i++)
      Serial.write(generate_ch[i]);
  res[9] = 1;
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){}
  else{err=2;return -1;}
  //enroll buffer send 12 bytes
```

```
  for(int i =0;i<17;i++)
     Serial.write(identify[i]);
  res[9] = 1;//
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){return (int)res[11];}
  else{err=1;return -1;}
}
```

A

PROJECT REPORT

On

ZIGBEE BASED SECURE WIRELESS COMMUNICATION USING AES

Submitted by

Mr. Angirekula Subash (17K81A0464)

Ms. Bodhanapu Archeshma (17K81A0467)

Ms. Vallamshetty Shravya Guptha (17K81A04B7)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATIONS ENGINEERING

Under The Guidance of

Mr. Venkanna Mood

Associate Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

St. MARTIN’S ENGINEERING COLLEGE

AN AUTONOMOUS INSTITUTE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

Dhulapally, Secunderabad-500 100
NBA & NAAC A+ Accredited

2020-2021

Department Of Electronics & Communication Engineering

BONAFIDE CERTIFICATE

This is to certify that the project entitled “Zigbee based secure wireless communication using AES’’, is being submitted by **Mr. Angirekula Subash (17K81A0464), Ms. Bodhanapu Archeshma (17K81A0467), Ms. Vallamshetty Shravya Guptha (17K81A04B7)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

< Signature>	Head of the Department
Mr. Venkanna Mood	Dr. B. Hari Krishna
Department of ECE	Department of ECE

Internal Examiner	External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **A.SUBASH** WITH ROLL NO.**17K81A0464, B.ARCHESHMA** WITH ROLL NO.**17K81A0467**, **V.SHRAVYA GUPTHA** WITH ROLL NO.**17K81A04B7**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**ZIGBEE BASED SECURE WIRELESS COMMUNICATION USING AES**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. ★ Kompally-500014 ★

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

DECLARATION

We, the student of **Bachelor of Technology** in Department of Electronics and Communication Engineering, session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled Zigbee Based Secure Wireless Communication using AES is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

Angirekula Subash	Bodhanapu Archeshma	Vallamshetty Shravya Guptha
(17K81A0464)	(17K81A0467)	(17K81A04B7)

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr. P. Santosh Kumar Patra**, **our beloved Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr. B. Hari Krishna**, **Head &Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept. Project coordinator Mr. K. Karthik** for his involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time to come.

Finally, we are highly grateful and obliged to our college level Project Guide **Mr. Venkanna Mood**, **Associate Professor in ECE** advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

1. Angirekula Subash
2. Bodhanapu Archeshma
3. Vallamshetty Shravya Guptha

ABSTRACT

The Zigbee based secured wireless communication using AES encryption allows us to communicate wirelessly with security feature. The data transfer during communication between two system is encrypted using 128-bit AES encryption which is highly secure. The data can be decrypted with correct key only, otherwise it returns some garbage value. This is two-way communication system where we can transmit as well as receive at both ends. We used ATMega microcontroller interfaced with xbee along with LCD display to send message and key, also have USB keyboards connected to each system and powered by 12V supply. After starting system, we will able to enter message on system. The maximum limit of message is 32 characters. After that system asks for key, the key limit is 16character it can be number or alphabet. Entering key will send the encrypted message to other system. Then the other system ask key to view the message. If the users enter correct key the message gets decrypted otherwise it will show garbage value thus securing the wireless communication.

INDEX

LISTS OF CONTENTS

CHAPTER-2

LITERATURE SURVEY

CHAPTER-3

PROJECT DESCRIPTION

CHAPTER-4

HARDWARE AND SOFTWARE DESCRIPTION

CHAPTER-5

SOFTWARE TESTING AND CODE IMPLEMENTATION

CHPATER-6

RESULTS ANALYSIS

CHAPTER-7

LIMITATIONS AND ADVANTAGES

CHAPTER-8

CONCLUSION AND FUTURE WORK

LIST OF FIGURES

LISTS OF TABLES

CHAPTER-1

INTRODUCTION

Overview of the Project

Nowadays communication plays a vital role in almost all industries. Be it sports, banks, armed forces, shopping malls etc., our project Zigbee based secure wireless communication using AES helps us to communicate i.e., transfer or receive data and since were using aes our data will be encrypted that means when an external person tries to hack or get access to our information, they'll need to enter a predefined key which is password in order to access the information. And since were using Zigbee we don't need to connect the transmitter and receiver on a daily basis this connection lasts for a longer duration to be approx. for a couple of months. The technology defined by the Zigbee specification is intended to be simpler and less expensive than other wireless personal area networks such as Bluetooth or more general wireless networking such as WIFI. Applications include wireless light switches, home energy monitors, traffic management systems, and other consumer and industrial equipment that requires short-range low-rate wireless data transfer. Its low power consumption limits transmission distances to 10-100 meters line of sight, depending on power output and environmental characteristics. ZigBee devices can transmit data over long distances by passing data through a mesh network of intermediate devices to reach more distant ones. As one of its defining features, ZigBee provides facilities for carrying out secure communications protecting establishment and transport of cryptographic keys.

Objective of the Project

Wireless technology is being developed rapidly nowadays advancement in microelectromechanical systems brings integration of sensing signal processing and RF capability on very small devices all kind of portable applications tend to be able to communicate without the use of any wires. Aim of wireless communication is to gather information or perform certain task in the environment a typical sensor node contains 3 C's are collection, competition and communication units. Based on the request of the sink, gathered information will be transmitted wirelessly. The collection unit has series of sensors, completion unit contains microcontroller and memory. Finally, the communication unit contains transceiver to transmit and receive. Zigbee is one of the most widely utilised wireless sensor network standards with low power, low data rate, low cost and short time delay characteristics, simple to develop and deploy and provides robust security and high data reliability. Name of the Zigbee came from zigzagging patterns of honey bees between flowers, represents the communication between nodes in a mesh network.

Scope of the Project

Zigbee has a very promising future in front of it. By rapid rise in home networking, Zigbee would provide revolutionizing statistics in the upcoming years which would entirely change to the wireless world. Various types of areas such as defence, national security, monitoring and control etc., can be facilitated by devices based on Zigbee standards. It leads to the cheap wireless technology, so that it can be widely used for low-rate data transfer. Zigbee aims to achieve greater efficiency. Range of the system can be increased by building a mesh wireless network. Using flexible PCB or SCB will further reduce size and weight of the sensor. Different wireless protocol such as ANT or Wibree will be investigated to further reduce power consumption since the amount of data transmission is very small. To simplify the device, the wireless link can be replaced with 1 or 2 GB mini/SD card.

Material Requirement

Hardware Requirements

- Arduino UNO
- Liquid Crystal Display.
- Zigbee
- USB-TTL
- Power supply.

Software Requirements

- Ardunio IDE
- Proteus software.

- X-CTU Software

Procurement Of Equipment’s

We brought all the required hardware components in a electronic manufacturing shop and before assembling and interfacing hardware components to Arduino UNO micro-controller, we connected individual sensors to Arduino in Proteus Software and understood working of every individual sensor in Proteus Software. Now, we assembled all sensors and interfaced to Arduino UNO micro-controller using connecting wires through bread board and we wrote program to work all the hardware sensors in Arduino IDE, we understood the working of all sensors in a practical way. We calculated and got all health vital information of coma patient’s and displayed on LCD and also updated on Server. So, doctor can check and track the patient’s health status from any- where in the world.

Introduction to Embedded System

An embedded system is a combination of computer hardware and software, either fixed in capability or programmable, designed for a specific function or functions within a larger system. Industrial machines, agricultural and process industry devices, automobiles, medical equipment, cameras, household appliances, airplanes, vending machines and toys, as well as mobile devices, are possible locations for an embedded system. An embedded system is a dedicated computer system designed for one or two specific functions. This system is embedded as a part of a complete device system that includes, such as electrical and mechanical components [1]. The embedded system is unlike the general-purpose computer, which is engineered to manage a wide range of processing tasks. Because an embedded system is engineered to perform certain tasks only, design engineers may optimize size, cost, power consumption, reliability and performance. Embedded systems are typically produced on broad scales and share across a variety of environments and applications. An embedded system is one that has computer hardware with software embedded in it as one of its most important components. It is a dedicated computer-based system for an application or product. An embedded system has three main components.

- It has hardware
- It has main application software.
- It has a real time operating system (RTOS)

An embedded system has software designed to keep in view three constraints:

(i) Available system memory.

(ii) Available processor speed.

(iii) The need to limit power dissipation when running the system.

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious. Some of these computers are however very simple systems as compared with a personal computer. The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes. The applications software on such processors is sometimes referred to as firmware. The simplest devices consist of a single microprocessor (often called a chip) which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine [3]. As the embedded system is the combination of both software and hardware.

History of Embedded Systems

Embedded systems date back to the 1960s. Charles Stark Draper developed an integrated circuit (IC) in 1961 to reduce the size and weight of the Apollo Guidance Computer, the digital system installed on the Apollo Command Module and Lunar Module. The first computer to use ICs, it helped astronauts collect real-time flight data. In 1965, Auto now a part of Boeing, developed the D-17B, the computer used in the Minuteman I missile guidance system.

Block Diagram of Embedded System

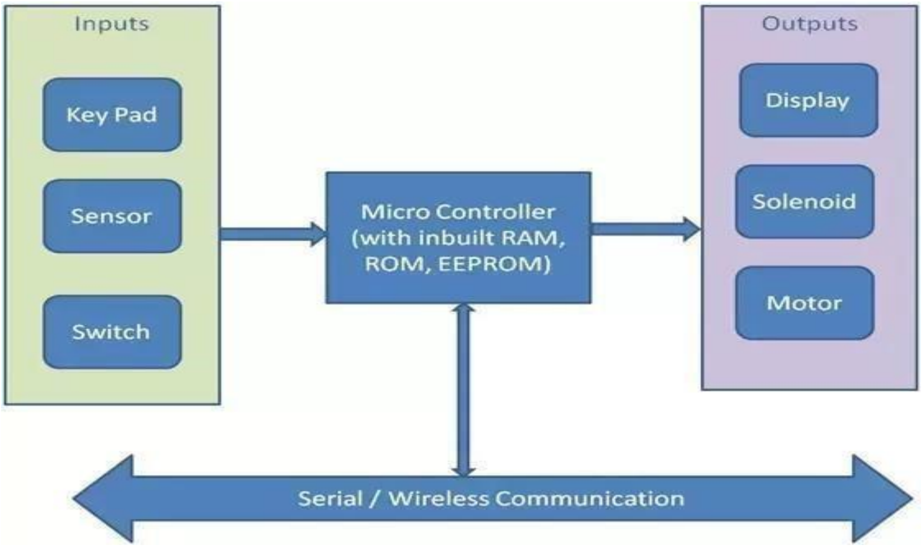


Figure.1.6.1: Block diagram of embedded system

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

Embedded System Hardware

Embedded system hardware can be microprocessor- or microcontroller-based. In either case, an integrated circuit is at the heart of the product that is generally designed to carry out computation for real-time operations[4]. Microprocessors are visually indistinguishable from microcontrollers, but while the microprocessor only implements a central processing unit (CPU) and, thus, requires the addition of other components such as memory chips, microcontrollers are designed as self-contained systems.

Embedded System Software

A typical industrial microcontroller is unsophisticated compared to the typical enterprise desktop computer and generally depends on a simpler, less-memory intensive program environment. The simplest devices run on bare metal and are programmed directly using the chip CPU's machine code language. Often, embedded systems use operating systems or language platforms tailored to embedded use, particularly where real-time operating environments must be served. At higher levels of chip capability, such as those found in SoCs designers have increasingly decided the systems are generally fast enough and the tasks tolerant of slight variations in reaction time that near-real-time approaches are suitable.

Examples of Embedded System

Some examples of embedded systems are below

- ATM
- Digital Cameras
- Microwave ovens
- Factory controllers
- Washing machine
- Calculator
- TV remote
- Traffic lights
- Digital watches
- Mp3 player
- Video games consoles

• Printers

• GPS receivers

• Dishwashers

• Thermostats

• Anti-lock banking system

Types of Embedded System

1. Real Time Systems

Real-time systems are those which give a quick response to critical situations. They are used in military, medical and industrial applications. Engineers working in these systems have high demand is current days. To develop the Realtime embedded system we require timing analysis, multitasking design, debugging, cross platform testing and architecture design. In these systems, quick response is very important.

2. Standalone Embedded System

This type of embedded system works for itself as a device without needing any interconnected computer. It can take data in the form of Analog or digital signals. This system first process data and then outputs data by displaying on the screen.

3. Networked Embedded System

Networked embedded systems are those systems which are connected to the network to give output to the attached resources. The devices in the networked embedded system are connected to the network with network interfaces. The network can be either a local area network (LAN) or a wide area network (WAN).

4. Mobile Embedded System

Mobile embedded systems are limited in resources including memory. Examples of mobile embedded systems include

• Personal digital assistants (PDA)

• Cellular phones

• Mp3 players

• Digital cameras

5. Small-Scale Embedded System

Small-scale embedded systems consist of 8–16-bit microcontroller. This system can perform tasks at a small level. They have on-chip ROM and RAM. Small scale systems can be even activated by the battery. The tools used to develop small scale embedded systems are an editor, cross assembler, assembler and integrated development environment (IDE). The purpose of this system is not computation but to control as a computer embedded inside it. It behaves as a component of a computer and its function is not to compute. The small-scale system is dedicated to some specific tasks.

6. Medium Scale Embedded System

This embedded system has 16–32-bit microprocessor or microcontroller with external RAM and ROM They can perform medium to complex level works. The integration between hardware and software is complex in these embedded systems. Programming languages used to develop medium scale embedded systems include Java, C, Visual C++, debugger, C++, RTOS, simulator, source code engineering tool and IDE.

7. Sophisticated Embedded System

The embedded system which can do large-scale works with multiple 32-64 bit chips is known as sophisticated embedded systems. They can perform distributed work on a large scale. The complexity of hardware and software is very high in these systems. In sophisticated embedded systems, hardware and software are assembled together on large scale and designing of hardware products is also included in these systems.

Applications of Embedded Systems

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment

- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

Microcontroller

A microcontroller is a small computer on a single metal oxide semiconductor (MOS) integrated circuit (IC) chip. A microcontroller contains one or more CPUs along with memory and programmable input/output peripherals. Program memory in the form of ferroelectric RAM, NOR flash or OTP ROM is also often included on chip, as well as a small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications consisting of various discrete chips[7]. In modern terminology, a microcontroller is similar to, but less sophisticated than, a system on a chip (SoC). SoC may include a microcontroller as one of its components, but usually integrates it with advanced peripherals like graphics processing unit (GPU), Wi-Fi module, or one or more coprocessors.

Figure.1.7.1: Microcontroller

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control nondigital electronic systems. In the context of the internet of things, microcontrollers are an economical and popular means of data collection, sensing and actuating the physical world as edge devices.

Some microcontrollers may use four-bit words and operate at frequencies as low as 4 kHz for low power consumption (single-digit milliwatts or microwatts). They generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (CPU clock and most peripherals off) may be just nanowatts, making many of them well suited for long lasting battery applications[8]. Other microcontrollers may serve performance-critical roles, where they may need to act more like a digital signal processor (DSP), with higher clock speeds and power consumption.

Microprocessor	Microcontroller
CPU is stand alone, RAM,ROM, I/O & timer are separate.	CPU, RAM,ROM, I/O & timer all are on single chip.
Designer can decide amount of RAM,ROM, & I/O ports.	Fixed amount of on-chip RAM,ROM, & I/O ports.
High processing power	Low processing power
High power consumption	Low power consumption
Typically 32/64 bit	8/16 bit
General purpose	Single purpose(control oriented)
Less reliable	Highly reliable
Eg.- 8086,8085	8051

Table 1.7.1: Difference between microprocessor and microcontroller

History

The origins of both the microprocessor and the microcontroller can be traced back to the invention of the MOSFET (metal-oxide-semiconductor field-effect transistor), also known as the MOS transistor. It was invented by Mohamed M. Atalla and Dawon Kahng at Bell Labs in 1959, and first demonstrated in 1960.

The same year, Atalla proposed the concept of the MOS integrated circuit, which was an integrated circuit chip fabricated from MOSFETs. By 1964, MOS chips had reached higher transistor density and lower manufacturing costs than bipolar chips[2]. MOS chips further increased in complexity at a rate predicted by Moore's law, leading to large-scale integration (LSI) with hundreds of transistors on a single MOS chip by the late 1960s. The application of MOS LSI chips to computing was the basis for the first microprocessors, as engineers began recognizing that a complete computer processor could be contained on a single MOS LSI chip.

Figure.1.7.2: 8051 Microcontroller Board

Most microcontrollers at this time had concurrent variants. One had EPROM program memory, with a transparent quartz window in the lid of the package to allow it to be erased by exposure to ultraviolet light. These erasable chips were often used for prototyping. The other variant was either a mask programmed ROM or a PROM variant which was only programmable once. For the latter, sometimes the designation OTP was used, standing for "one-time programmable. For the erasable variants, quartz was required, instead of less expensive glass, for its transparency to ultraviolet light— to which glass is largely opaque—but the main cost differentiator was the ceramic package itself. In 1993, the introduction of EEPROM memory allowed microcontrollers (beginning with the Microchip PIC16C84) to be electrically erased quickly without an expensive package as required for EPROM, allowing both rapid prototyping, and in-system programming. (EEPROM technology had been available prior to this time, but the earlier EEPROM was more expensive and less durable, making it unsuitable for low-cost mass-produced microcontrollers.)

Zigbee Technology

So, where does Zigbee Technology fit into the existing pool of Wireless Technologies? The Zigbee Wireless Technology fits into a market that is not completely filled by other technologies. For example, other wireless technologies like WiFi and Bluetooth are striving to go faster, add more features, provide streaming of high-definition content and run for hours. In contrast, the Zigbee Technology is designed for low data rates, fit into a tiny stack, controlling simple devices like LEDs or Thermostats and run for years. Although Zigbee Technology has many applications like Home Automation and Military, the main category that the Zigbee Technology has been aimed for is called as Wireless Sensor Network or WSN.

Wireless Sensor Network (WSN) is a collection (network) of wireless sensors that collect, store and process environmental information and also communicate with neighboring networks. More about WSN later. WSN has few unique requirements like low power and low cost and Zigbee Technology achieves these requirements with the help of a constraint: Low Data Rate.

Wireless Sensor Network (WSN) is a technology that being warmly used both for research and to help human activities. System security is one of the important things that must be considered both in wireless and wire line network. Wireless sensor networks are expanding which are vulnerable and therefore require an effective security mechanism. Wireless sensor networks have several constraints such as limited memory, energy and computational capabilities that create constraints when coupled with security at sensor nodes. To solve the problem, this final project will implement and analyze security system in wireless sensor network refers to ZigBee standard. ZigBee schemes can be synchronized with new protocols targeted at low rates, small power devices, and sensor nodes. ZigBee requires cryptography that is expected to save power, computing power, and storage resources. To that end, the chosen security system is using AES encryption algorithm (Advanced Encryption Standard) which is implemented directly on ZigBee. In this research successfully apply AES128 decryption and encryption algorithm on wireless sensor network. The test used is a passive attack that can only see and capture the package only. In a security performance analysis, the confidentiality parameter is not fulfilled if it does not use security. Integrity parameters are fulfilled using or without security. The largest throughput value is 1122 bytes / s at a distance of 21 meters. And the last delay at a distance of 49.5 meters with a value of 4.1483 s.

Wireless Sensor Network (WSN) is a technology used for research as well and to facilitate daily life. WSN is a network infrastructure that can detect, calculate and have communication elements that can transmit data to the administrator to measure, observe, and give orders if there are certain conditions. The security system is one of the important things that must be considered both wireless and wired networks. WSN users should believe that the information they receive from the network is reliable, accurate, and not damaged by irresponsible parties. Wireless sensor networks have several constraints such as memory limitations, energy and computational capabilities that create constraints when added to the security of the node sensor. To send information that has been read by the node ZigBee needed to send the information to the user WSN. To solve the problem, this research applies and analyzes system security on wireless sensor network refers to ZigBee standard. ZigBee is an emerging standard for low power, low level wireless communications with the goal of both systems being able to work together and can include all devices within the scope of power despite low WSN node sensors. ZigBee requires cryptography which is expected to save power, computing power, and storage resources. In the end, the security system is chosen using AES encryption algorithm (Advanced Encryption Standard) which is implemented directly on ZigBee.

What is ZigBee Technology

Zigbee Technology is a Wireless Communication Standard that defines a set of protocols for use in low data rate, short to medium range wireless networking devices like sensors and control networks. The target of Zigbee Technology is low cost, low power, battery operated wireless sensors that do not need to constantly update its status and also allows sleep mode or low power mode for its electronics and radio.

Zigbee Technology is based on IEEE 802.15.4 Standard and the Zigbee devices operate in the unlicensed 2.4 GHz ISM Band (ISM – Industrial, Scientific and Medical). Although 2.4 GHz band is commonly used worldwide for commercial Zigbee devices, there are devices that use a different set of frequency bands like 784 MHz, 868 MHz and 915 MHz in China, Europe and USA (and Australia) respectively.

Since we are talking about low data rates of Zigbee Technology from the beginning, let us see the data rates supported by Zigbee. The data rates in Zigbee Technology are dependent on the frequency band. For example, the 868 MHz band supports a data transfer rate of 20 kbps while the more common 2.4 GHz band supports data rates up to 250 kbps. Before digging deep in to Zigbee Technology, let us take a look at two applications, where Zigbee can be used. One application is the in-home patient monitoring, where a patient wears a Zigbee device, which periodically collects the information like blood pressure and heart rate.

This data is then wirelessly transmitted to a local server in the patient's home (local server – like a PC). The local server performs basic analysis and the vital information is sent to the Doctor over the internet. Another application of Zigbee is the building's structural health monitoring. This application is very useful in earthquake prone areas. Several Zigbee based wireless sensors like accelerometers are installed throughout the build. These sensors, which form a wireless sensor network, collect information that can be used to detect signs of damage and evaluate whether the building is safe for public or not.

Why Low Data Rates in Zigbee?

When other wireless technologies like WiFi and Bluetooth are aiming to provide high speed data transfers, why the data rates are low in Zigbee standard? The answer is very simple. The intention behind the development of Zigbee Technology is to use it in wireless monitor and control. The amount of information and the frequency of communication in such applications is very less.

Even though it is possible for an IEEE 802.15.4 network to achieve higher data rates, the Zigbee Technology, which is based on IEEE 802.15.4, is not designed to do so.

Zigbee Alliance

The Zigbee Alliance works to simplify wireless product integration to help product manufacturers introduce energy-efficient wireless control into their products faster and more cost-effectively. Alliance members create standards that offer reliable, secure, low-power and easy-to-use wireless communication, using an open standards development process to guide their work. The alliance is organized by committees, work groups, study groups, task forces and special interest groups.

Figure.1.8.1: Zigbee Alliance

There are three types of membership with different rights and benefits:

- An **adopter** offers access to final, approved specifications, participation in interoperability events and access to standard work/task group documents and development activities.
- A **participant** offers voting rights in work groups and has early access to all Zigbee Alliance standards and specifications in development.
- A **promoter** offers automatic voting rights in all work groups, final approval rights on all standards and a seat on the alliance's board of directors.

Zigbee Network Topologies

The Zigbee Network Layer in the Zigbee Stack is responsible for formation of the network. There are three Zigbee Network Topologies: Star, Cluster Tree and

Mesh. These three Zigbee Topologies come under one of the two network topologies mentioned in the IEEE 802.15.4 i.e., Star and Peer-to-Peer.

Star Topology

In a Star Network Configuration, there is one Coordinator and any number of End Devices. All the End Devices are connected to the coordinator and the individual End Devices are isolated, both physically and electrically i.e. no direct communication between end devices.

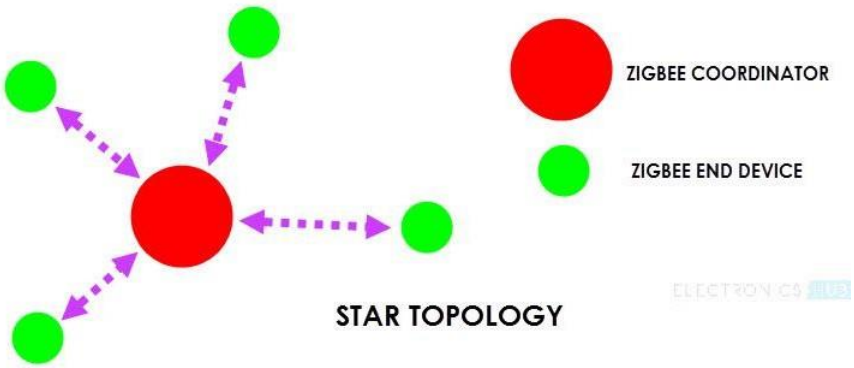


Figure.1.8.2: Star Topology

All the information must pass through the coordinator, even the information from one End Device to another.

Cluster Tree Topology

The Cluster Tree Topology is a type of Peer-to-Peer Topology. In Zigbee Cluster Tree Topology, the End Devices join the network via the Coordinator or the Router. As the Zigbee Router extends the range of the Zigbee Network, the End Device need not be in the range of the coordinator.

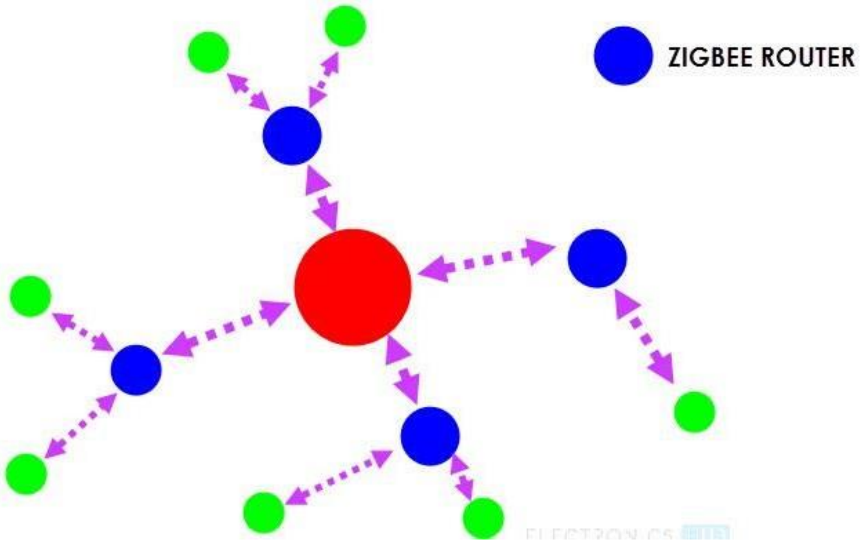


Figure.1.8.3: Cluster Tree Topology

Even in Tree Topology, the End Devices cannot communicate with each other directly but the Router can communicate with other Routers and the Coordinator.

Mesh Topology

The Zigbee Mesh Topology is also a Peer-to-Peer Topology and is an extension to the Cluster Tree Topology. The End Devices that are configured as FFD can directly communicate with other FFD Devices (either Routers or End Devices). But End Devices configured as RFD, still need to communicate through Routers or Coordinators.

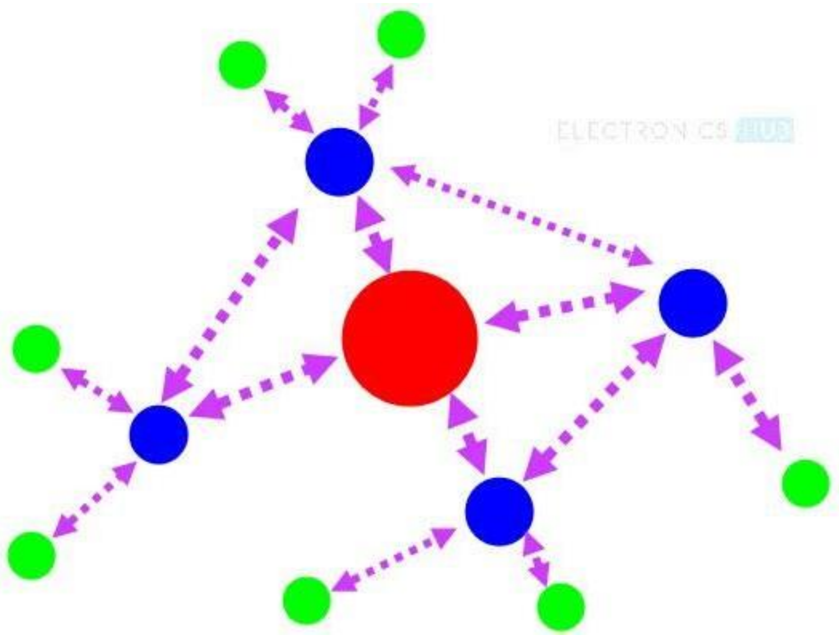


Figure.1.8.4: Mesh Topology

Applications of Zigbee Technology

Zigbee Networking and Zigbee Technology has a wide range of application like Home Automation, Healthcare and Material Tracking. Let us see few Applications of Zigbee Technology, where Zigbee Devices can increase efficiency and reduce cost.

- Home Automation
    - Security Systems
    - Meter Reading Systems
    - Light Control Systems
    - HVAC Systems
- Consumer Electronics
    - Gaming Consoles
    - Wireless Mouse
    - Wireless Remote Controls
- Industrial Automation
    - Asset Management
    - Personnel Tracking
    - Livestock Tracking
- Healthcare
- Hotel Room Access
- Fire Extinguishers

Motivation of the Project

The main reason for preferring this project is to make sure that the data communication is done securely from transmitter to receiver. The system, primarily intended for the benefit of the locations where the data communication is low, can be used and implemented by various other types of individuals as well. For example, a hospital, library, working office, hotel, restaurant etc use data transfer protocols on a day-to-day basis and this can be securely done using secure ZigBee communication which is encrypted using advanced encryption standards.

CHAPTER-2

LITERATURE SURVEY

Literature Survey

As per the exploration paper of Amar Abana [1] explains that Named Data Networking (NDN) is a new architecture which allows communications using data's natural names rather than hosts' logical addresses. In recent years, several research projects have demonstrated the ability of NDN to support emerging IoT applications like home automation, smart cities and smart farming applications. This work aims to integrate NDN with ZigBee to give NDN a better support for INTERNATIONAL JOURNAL OF CURRENT ENGINEERING AND SCIENTIFIC RESEARCH applications that are known to require wireless sensing/actuating abilities, mobility support and low power consumption. For this purpose, we present our NDN-overZigBee design and we show through experiments conducted with three different scenarios the suitability and the ease of use of NDN in IoT context. The choice of ZigBee is motivated by the fact that it is a network specification for low power Wireless Personal Area Networks (WPANs) and supports a large number of nodes.

Jiasong Mu Liang Han [2] said that The ZigBee network is widely studied and deployed recently because of its low cost and simplicity features. However, the power consumption issue needs a further improvement since the application requirements are not fully satisfied. The emerging 5G communication technology is characterized by the smarter devices and the native support for the M2M communication. On that basis, the 5G terminals are capable of joining the existing ZigBee networks and have the potential to improve the data transmission. In this paper, we investigate the performance of the ZigBee networks in the 5G environment for different scenarios. Then a Nearest Access Routing (NAR) algorithm based on the physical depth is proposed for different communication types. To reduce the loads in ZigBee networks, the data flow in the neighbourhood of 5G terminals is gathered and transmitted via the IP networks. The simulation results showed that NAR effectively share the communication in ZigBee networks. It leads to better performances with higher packet delivery ratio, less hop counts from ZigBee devices, lower latency, fewer packets sent by ZigBee nodes and zero routing overheads.

Jiangsu [3] proposes that a modified AODV Junior routing protocol is adopted as part of the combined routing strategy in the ZigBee networks. However, its routing overhead caused by the routing discovery and the routing table maintenance needs to be further reduced. Considering the deploying environment, the emerging 5G technology tends to be ubiquitous in the near future. Due to its native support to the smart devices and the M2M communication, the 5G terminals may be used as gateways for the ZigBee networks to improve the existing routing method. The ZigBee coordinator and the 5G terminals, which both have the accessibility to the internet, are defined as Associated Gateways (AG). Aiming to the higher efficiency and the shorter path, an improved ZigBee AODV (ZAODV) routing protocol using associated gateways for the heterogeneous networks in the 5G environment is proposed in this paper. Besides the regular ZigBee function, by sharing the neighbour and routing information via IP network, the AGs are also responsible for collaboratively finding the optimum path and transmitting the packets to reduce the consumption for ZigBee devices. Moreover, an additional routing information collecting method is developed to further improve the routing performance. The proposed algorithm is evaluated based on simulation results. It is shown that our routing method outperforms the existing ones by higher picketer deliver ratio, shorter path length, lower latency, fewer packets sent per ZigBee node and lower routing overhead.

Chia-HsinCheng [4] proposes a multichannel ZigBee Wireless Sensor Network (WSN) for robust data transmission. This study sought to increase the Packet Delivery Ratio (PDR) through the application of multi-channel technology. Based on a cluster-tree construction protocol, we implemented a multi-channel time division multiple-access scheme using hardware devices. Furthermore, ZigBee and Wireless LAN (WLAN) both operate on the 2.4 GHz band, referred to as the industrial, scientific, and medical unlicensed band. As a result, ZigBee faces severe interference problems in the presence of WLAN. In this study, we developed an interference avoidance method to improve conditions for the coexistence of ZigBee and WLAN and then conducted experiments using a practical network implementation to evaluate the resulting performance. Our experiment results demonstrate considerable improvements in the PDR of ZigBee transmission in an environment with interference from WLAN.

Jacek Stępień [5] conversed that an implementation of mobile nodes tracking system based on ZigBee and Wi-Fi wireless networks is presented. On the base of known algorithmic as well as circuit solutions a simple yet universal system, applied in prototype application dedicated for person's localization in museum premises has been developed. Since system INTERNATIONAL JOURNAL OF CURRENT ENGINEERING AND SCIENTIFIC RESEARCH utilizes entirely wireless communication, it can be applied in any closed objects. The system has been preliminarily verified in real in-situ environment.

Rob Toulson [6] e states introduce concepts and applications relating to wireless data communication. It starts with a review of the principles, including the electromagnetic spectrum, radio transceivers, protocols and antennae. The background to short distance, personal area network protocols is introduced, through reference to IEEE Working Groups. Bluetooth is introduced, and trialled using the RN-41 Bluetooth module. The RN-41 is used to develop key Bluetooth concepts. Simple PC-tombed and mbed-to-mbed Bluetooth links are demonstrated. The Zigbee protocol, its typical networks and applications are introduced. XBee radio modules are used to illustrate simple Zigbee links, in transparent mode. Use of the XBee in its more flexible API mode is then introduced, which opens the door to more advanced Zigbee applications.

Jia Jia [7] said that the physical layer of ZigBee communication systems was defined by IEEE 802.15.4 and has good external white noise resistance due to its spread spectrum characteristic and error correction of the baseband coding process. However, previous research has shown the performance of ZigBee to degrade in the presence of impulsive noise. In this regard, an improvement of the ZigBee receiver is warranted in order to improve the decoding process. A novel Error-Balanced Wavelet filtering approach utilizing the multiresolution property is proposed to suppress the impact of impulsive noise prior to symbol detection and thus improve the Bit Error Rate (BER) performance of the ZigBee demodulation process. This assessment is based on computer simulations and verifies that the overall transmission performance is improved by our proposed approach. The results obtained are also compared with existing impulsive noise suppression approaches and it is shown that our waveletbased method outperforms other methods in improving the system.

BER. L.K. Wadhwa [8] et.al. proposes that Energy efficiency and network lifetime are main concerns in WSN. In order to improve these factors ZigBee plays an important role. Low cost, low data rate features of ZigBee results in low power consumption and makes it useful in wireless sensor networks, increasing life of small batteries of nodes in the network. Since tree routing in ZigBee does not require any routing tables to send the packet to the destination, it can be used in ZigBee end devices that have limited resources. Routing protocols such as AODV (Ad-hoc On Demand distance Vector routing), ZTR (ZigBee Tree Routing), and STR (Shortcut Tree Routing) are compared on the basis of different performance metrics (End to end delay, routing overload, throughput, packet delivery ratio). The performance evaluation shows that STR achieves better performance as compared to other two routing protocols. But there are some limitations of the STR method. Performance of packet delivery ratio of STR is less as compared to AODV. Performance of end-toend delay of STR is poor as compared to AODV. Hence ESTR is proposed. The main aim of proposed ESTR [Extended STR] is to present new ZigBee network routing protocol with goal of improving the performance of ESTR in terms of PDR and delay against STR and AODV.

Shashwat Pathak. [9] explains an energy efficient wireless telemonitoring scenario of cardiac patients through ZigBee, based on variable duty cycle being rendered to sensors. In an intra-hospital telemedicine scenario, Electro Cardio Gram (ECG) signals of patients are acquired through ECG sensor nodes having transmission capability and these ECG signals are received by Personal Digital Assistant (PDA) kept at nursing station through ZigBee network. ECG signals are further transmitted to Doctor's PDA. If the duty cycle is varied as per the load or number of active sensors, total energy consumed in idle mode can be avoided and total energy consumed by sensors is reduced hence increasing total network lifetime. This paper, comparatively analyses the energy efficiency of ZigBee sensors with different percentage of duty cycle on the basis of energy consumption parameter under variable load conditions. The matrices used in performance evaluation are energy consumption in transmit mode, energy consumption in received mode and energy consumption in idle mode.

Existing Methods

The previously existing method of this project was using keyboards i.e., the data transfer would be possible with AES through a lcd using a keyboard. Which was not efficient, and would consume more power comparatively to our proposed system and in order to receive the transferred information the receiver had to enter the predefined password every time they wanted to access the transferred information. And this wasn't as secure as compared to our proposed system cause of the risk of someone else getting access to our information.

Proposed Methods

In our proposed method we use ZigBee as a transfer protocol to transfer the data and that too from a system to system. In this we will have a better efficiency and since the transfer of data is from system to system, we don't need to enter the predefined password at the receiver every time we want to access the received information. Since the ZigBee is designed in pairs at the time of manufacturing.

CHAPTER-3

PROJECT DESCRIPTION

BLOCK DIAGRAM OF THE PROJECT:

The Zigbee based secured wireless communication using AES encryption allows us to communicate wirelessly with security feature. The data transfer during communication between two system is encrypted using 128-bit AES encryption which is highly secure. The data can be decrypted with correct key only, otherwise it returns some garbage value. This is two-way communication system where we can transmit as well as receive at both ends. We used ATMega microcontroller interfaced with x-bee along with LCD display to send message and key, also have USB keyboards connected to each system and powered by 12V supply. After starting system, we will able to enter message on system. The maximum limit of message is 32 characters. After that system asks for key, the key limit is 16character it can be number or alphabet. Entering key will send the encrypted message to other system. Then the other system ask key to view the message. If the users enter correct key the message gets decrypted otherwise it will show garbage value thus securing the wireless communication.

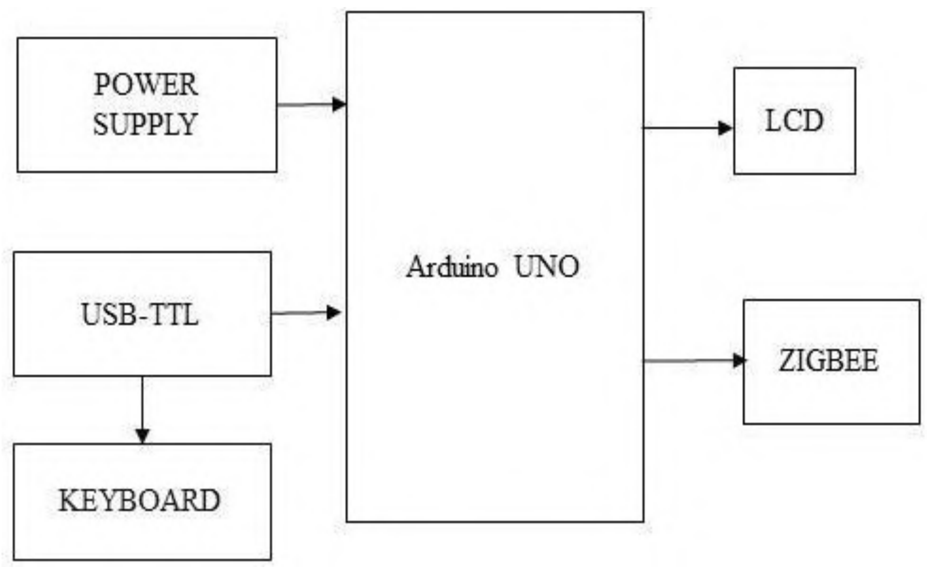


Figure 3.1.1: Block Diagram of Transmitter

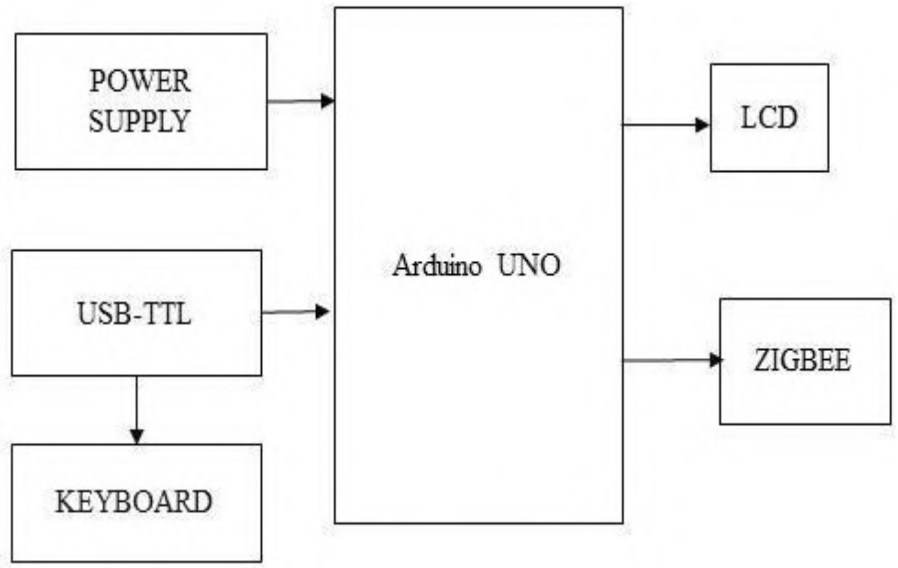


Figure 3.1.2: Block Diagram of Receiver

The Block of Circuits are

1) Arduino UNO
2) Power Supply Unit
3) Zigbee
4) USB-TTL
5) LCD
6) Keyboard

1. Arduino UNO

Arduino is open-source software. Arduino Uno is a microcontroller, ATmega328Pcontain in it. It has 14 digital i/o pins and 6 analog input pin, a 16 MHz quartz crystal, a USB connection, a power jack and a reset button. Simply connect to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. Embedded C program is used for coding.

2. Power Supply

Every electrical and electronic device that we use in our day-to-day life will require a power supply. In general, we use an AC supply of 230V 50Hz, but this power has to be changed into the required form with required values or voltage range for providing power supply to different types of devices.

3. Zigbee

Zigbee is a standards-based wireless technology developed to enable low-cost, low-power wireless machine-to-machine (M2M) . Zigbee is for low-data rate, low-power applications and is an open standard. This, theoretically, enables the mixing of implementations from different manufacturers, but in practice, Zigbee products have been extended and customized by vendors and, thus, plagued by interoperability issues.

4. USB-TTL

This is an USB2.0 to TTL UART converter module which is based on CP2102 bridge by silabs. This module can be used with laptop's which don't have standard serial port. This module creates a virtual com port using usb on your computer which can support various standard baud rates for serial communication.

5. LCD

The LCD here is used to display us the information. Here we use two LCD’s one at Transmitter, one at Receiver. When we send the information from transmitter to receiver. The information which we have sent is displayed on the LCD. The information is display on both the LCD’s i.e., on transmitter side and receiver side.

6. Keyboard

In this project we use two keyboard’s one at transmitter side and the other one at receiver side. To type the information which we want to send. we use keyboards to type the information.

SCHEMATIC DESCRIPTION:

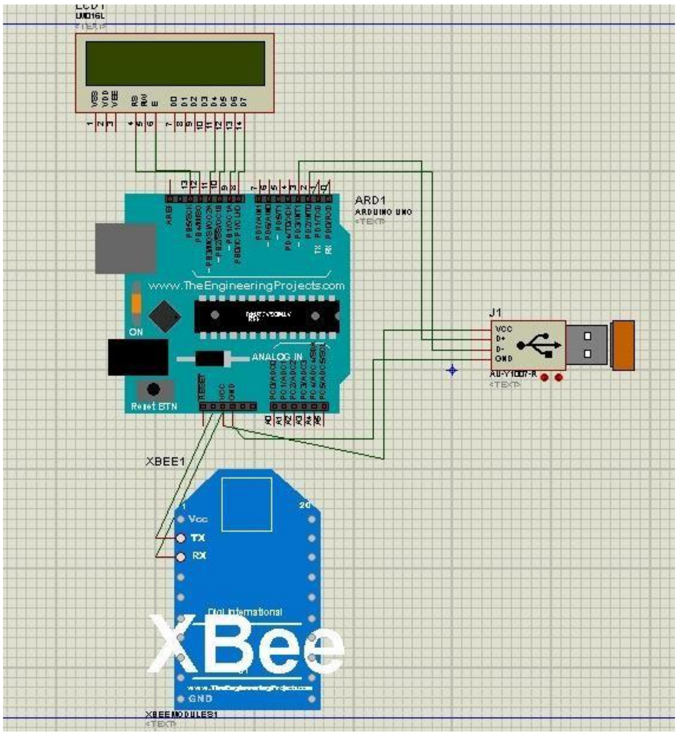


Figure 3.2.1: Schematic Diagram of Transmitter

- Here we can see the schematic diagram of the transmitter side where 2 Zigbee's are connected to ports 2 and 3 of Arduino and the ports 13,12,11,10,9 and 8 are connected to Arduino.
- The lcd we use is a 16,2 lcd i.e., 16 characters per line and 2 such lines.
- We will be having 2 possibilities of sending the information either through a Zigbee or from serial ports.
- We give a delay to make things stable and we transmit the information.
- Then we will read the information received through the X-CTU software.

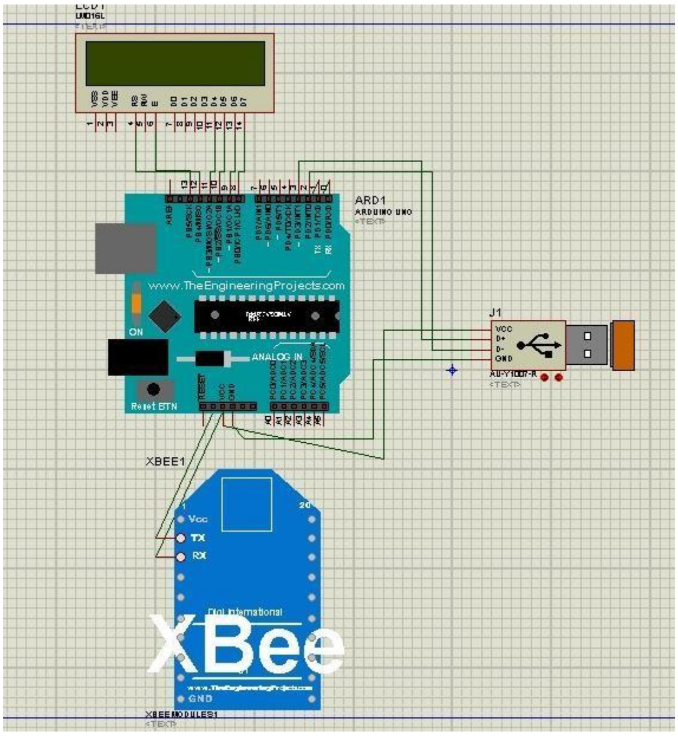


Figure 3.2.2: Schematic Diagram of Receiver

- Here we can see the schematic diagram of the transmitter side where 2 Zigbee's are connected to ports 2 and 3 of Arduino and the ports 13,12,11,10,9 and 8 are connected to Arduino.
- The lcd we use is a 16,2 lcd i.e., 16 characters per line and 2 such lines.
- We will be having 2 possibilities of sending the information either through a Zigbee or from serial ports.
- We give a delay to make things stable and we transmit the information.
- Then we will read the information received through the X-CTU software here we can see the schematic diagram of the transmitter side where 2 Zigbee's are connected to ports 2 and 3 of Arduino and the ports 13,12,11,10,9 and 8 are connected to Arduino.
- The lcd we use is a 16,2 lcd i.e., 16 characters per line and 2 such lines.
- We will be having 2 possibilities of sending the information either through a Zigbee or from serial ports.
- We give a delay to make things stable and we transmit the information.

Then we will read the information received through the X-CTU software.

CHAPTER-4

HARDWARE AND SOFTWARE DESCRIPTION

Hardware Components

The hardware components used for the implementation of “Zigbee Based Secure Wireless Communications using AES” are

1. Arduino UNO
2. Power Supply Unit
3. Zigbee
4. LCD’S
5. USB TTLS-2
6. Keyboard

Arduino UNO

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message and turn it into an output activating a motor, turning on an led, publishing as it reached a wider community, the Arduino board started something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board.

To do so you use the Arduino programming language (based Wiring), the Arduino Software (IDE), based on Processing. Arduino was born at the area Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming wearable, 3dprinting, and embedded environments. All Arduino boards are completely open-source, empowering from everyday objects to complex scientific instruments.

Professional gathered around open-source plat form, the software too is open-source, it is growing through the contributions of users worldwide, the Arduino board is shown in below figure.

Figure.4.1.1: Arduino uno board

Why Arduino?

Arduino Uno is inexpensive. Arduino is cross platform, its IDE can run on multiple Operating Systems such as Windows, Macintosh, Linux and many more. It has simple and clear programming environment. Its software and hardware are open-source and extensible, that is, Arduino programming can be expanded through C++ libraries and can also leap to AVR ‘C’ language programming.

It is simple and accessible to user experience; Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux.

There are many other microcontrollers platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handy board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems.

- **Inexpensive**
  Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50
- **Cross-platform**
  The Arduino Software (IDE) runs on Windows, Macintosh OSX, and

Linux operating systems. Most microcontroller systems are limited to Windows.

- **Simple, clear programming environment**
  The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- **Open source and extensible software**
  The Arduino software is published as opensource tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- **Open source and extensible hardware**
  The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money and the parts of Arduino board is shown below.

Figure.4.1.2: Arduino board description

Features of Arduino

Micro controller	Atmega328
Operating voltage	5V
Input Voltage(recommended)	7-12V
Input Voltage	6-20V
Digital I/O	14(of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 Ma
DC Current for 3.3V Pin	50 Ma
Flash Memory	32 KB (Atmega328 of which 0.5 KB used by bootloader)
SRAM	2 KB (ATmega32)
EEPROM	1 KB(ATmeha328)
Clock Speed	16 MHz

Table 4.1.1: Features of Arduino uno

- It is an easy USB interface. This allows interface with USB as this is like a serial device.
- The chip on the board plugs straight into your USB port and supports on your computer as a virtual serial port. The benefit of this setup is that serial communication is an extremely easy protocol which is time-tested and USB makes connection with modern computers and makes it comfortable.
- It is easy-to-find the microcontroller brain which is the ATmega328 chip. It has more number of hardware features like timers, external and internal interrupts, PWM pins and multiple sleep modes.
- It is an open-source design and there is an advantage of being open source is that it has a large community of people using and troubleshooting it. This makes easy to help in debugging projects.
- It is 16MHz clock which is fast enough for most applications and does not speeds up the microcontroller.
- It is very convenient to manage power inside it and it had a feature of built- in voltage regulation. This also be powered directly off a USB port without any

external power. You can connect an external power source of up to 12v and this regulates it to both 5v and 3.3v.

- 14 digital pins and 6 analog pins. This sort of pins allows you to connect hardware to your Arduino Uno board externally. These pins are used as a key for extending the computing capability of the Arduino Uno into the real world. This has an ICSP connector for bypassing the USB port and interfacing the Arduino directly as a serial device. This port is necessary to re-boot load your chip if it corrupts and can no longer used to your computer.
- It has a 32 KB of flash memory for storing your code.
- An on-board led is attached to digital pin 13 to make fast the debugging of code and to make the debug process easy finally it has the button to reset.

Pin Configuration on Arduino UNO

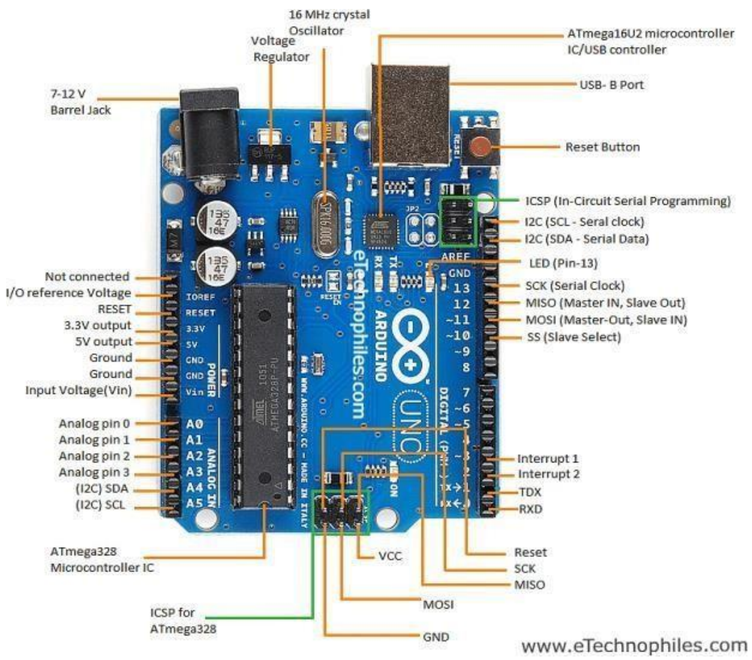


Figure.4.1.3: Pin configuration

- **Reset Pin:** resets the microcontroller.
- **GND :** used to connect the circuit to Ground
- **A0-A5:** used to provide analog input in the range of 0-5V.
- **Digital pins(0-13) :** can be used as input or output pins.
- **0(RX),1(TX) :** used to receive and transmit TTL serial data.
- **AREF :** To provide reference voltage for input voltage.
- **SPI :** used for SPI communication.

Working of Arduino

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The 14 digital input/output pins can be used as input or output pins by using pinMode(), digitalRead() and digitalWrite() functions in arduino
programming. Each pin operate at 5V and can provide or receive a maximum of 40mA current, and has an internal pull-up resistor of 20-50 KOhms which are disconnected by default. Out of these 14 pins, some pins have specific functions as listed below:

- **Serial Pins 0 (Rx) and 1 (Tx):** Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.
- **External Interrupt Pins 2 and 3:** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM Pins 3, 5, 6, 9 and 11:** These pins provide an 8-bit PWM output by using analogWrite() function.
- **SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK):** These pins are used for SPI communication.
- **In-built LED Pin 13:** This pin is connected with an built-in LED, when pin 13 is HIGH – LED is on and when pin 13 is LOW, its off.
- **Analog pins (A0-A5) :** used to provide analog inputs in the range of 0-5V.

- Analog pin 4 (SDA) and pin 5 (SCA) also used for TWI communication using Wire library.

Arduino Uno has a couple of other pins as explained below:

- **AREF:** Used to provide reference voltage for analog inputs with analogReference () function.
- **Reset Pin:** resets the microcontroller.

Advantages

Inexpensive -Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50.

- Cross-platform

The Arduino software runs on Windows, Macintosh OSX, and the Linux operating systems. Most microcontroller systems are limited to Windows.

- Simple programming environment

Here the Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino.

- Open source and extensible software

The Arduino software is published as open-source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

- Open source and extensible hardware

The Arduino is based on Atmel's ATmega8 and ATmega168 microcontrollers. The plans for the modules are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even 19 relatively inexperienced users that can build the breadboard version of the module in order to understand how it works and save money.

Power Supply

The Arduino Uno can be powered via through USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wallwart) or battery. The adapter can be connected by plugging a 2.1mm centre positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. the power pins are as follows.

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3.3V.** A 3.3volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.

Programming

The Arduino Uno can be programmed with the Arduino Software (IDE). Select Arduino Uno from the Tools Board After your code is compiled using Arduino IDE, it should be uploaded to the main microcontroller of the Arduino UNO using a USB connection. Because the main microcontroller doesn’t have a USB transceiver, you need a bridge to convert signals between the serial interface (UART interface) of the microcontroller and the host USB signals. The ATmega328 on the Arduino Uno comes pre-programmed with a boot loader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino are similar. The ATmega16U2 (or 8U2 in the rev1&rev2 boards) firmware source code is available in the Arduino repository. The ATmega16U2 or 8U2 is loaded with a DFU bootloader, which can be activated by On Rev1 boards connecting the solder jumper on the back of the board and then resetting the 8U2.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). Arduino is an open-source hardware platform that is being used by people around the globe for building electronics projects. It is an integrated platform which contains
both the physical and programmable circuit otherwise known as microcontroller and a software (or IDE) that you can run on your computer to write and upload the code onto the physical board.

Arduino Board is quite popular among many people who want to get started with electronics, and unlike other embedded system boards Arduino does not require any additional hardware to upload the code (generally known as programmer).
Here we will try to understand about the Arduino Architecture and its functionalities. The processor of the Arduino Board uses Harvard Architecture for which the program code and program data have separate memory. The memory of it is divided into two namely program memory and data memory. The data will be stored in the data memory whereas the program code will be stored in the flash program memory. For ex: The Atmega328 microcontroller has 32kb of flash memory, 2kb of SRAM, 1kb of EPROM and operates at 16MHz clock speed. Some of the other basic functions of Arduino are:

- **Digital write pin** is used to write the digital value of the given pin.
- **Pin mode pin** is used to set the pin to I/O mode.
- **Analog read pin** Reads the value from the specified Analog pin.
- **Analog write pin** writes the value from the specified Analog pin
- **Serial. Begins pin** sets the beginning of serial communication by setting the rate of bit
- **Digital read pin** reads the digital value of given pin.

Introduction to ATMega328

ATmega328 is an eight (8) bit Microcontroller. It can handle the data sized of up to eight (8) bits. It is an AVR based micro-controller. Its built-in internal memory is around 32KB. It operates ranging from 3.3V to 5V. It has an ability to store the data even when the electrical supply is removed from its biasing terminals. it is the excellent features include the cost efficiency and low power dissipation, programming lock for security purposes, real timer counter with the separate oscillator and It is generally or normally used in Embedded Systems applications. ATmega-328 is shown in the figure given below.

Figure.4.1.4: ATMega328 Microcontroller

Arduino UNO Atmega328

The Arduino UNO is a micro controller board based on the ATmega. UNP means one in Italian and is named to mark the upcoming release of the given Arduino 1.0 it has 14 digital input/output pins, 6 analog inputs, 16MHz ceramic resonator. A USB connection. A power jack, An ICSP header and a reset button.it contains information everything needed to support the microcontroller. simply connect it to a computer with a USB cable or power it with an ac-to-dc adapter or battery to get started.

The UNO differs from all preceding boards in that it does not use the FTDI USB-to serial driver chip. Instead, it features the ATMega 16U2ATMega 8U2 up to version instead R2) programmed as a USB to-serial converter.

Architecture and Working of Arduino Uno

The Arduino UNO can be also powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter or battery. The adapter can connect by plugging a 2.1mm centre positive plug into the board’s power jack. Leads from a battery can be inserted in the Gnd and vin pin headers of the power connector. The board can operate on an external supply of 6to 20 volts. If supplied with less than 7v however, the 5v pin may supply less than five volts and the board may be unstable. If using more than 12v.the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The processor of the Arduino board uses the Harvard architecture where the program code and program data have separate memory. It consists of two memories such as program memory and data memory. Where in the data is stored in data memory and the code is stored in the flash program memory. The Atmega328 microcontroller has 32kb of flash memory, 2kb of SRAM 1kb of EPROM and operates with a 16MHz clock speed. Architecture of Atmega328 is shown below.

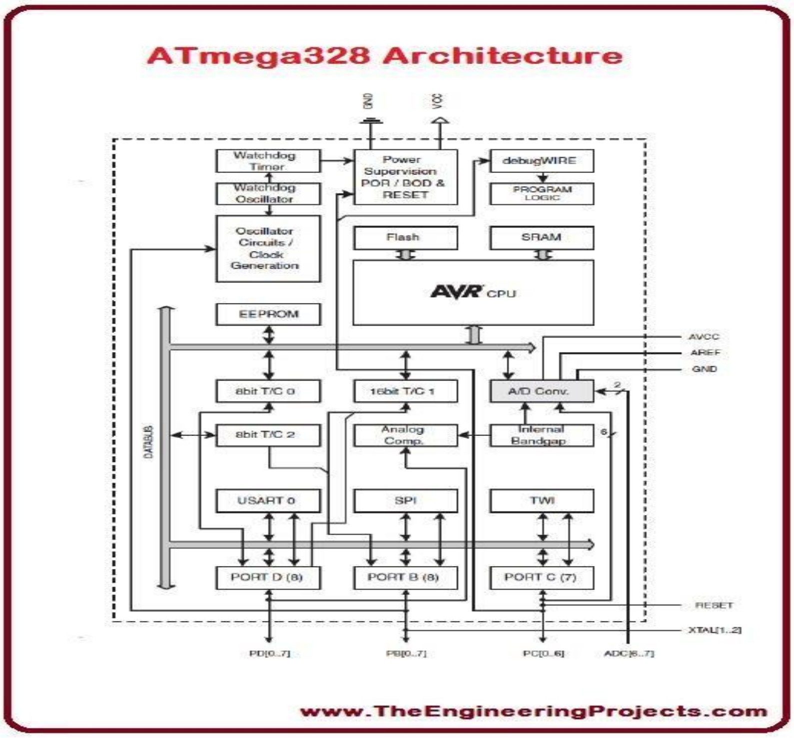


Figure.4.1.5: ATmega328P Architecture

Atmega328 Pins Description

Functions associated with the pins must be known in order to use the device appropriately. ATmega-328 pins are divided into different ports which are given in detail below.

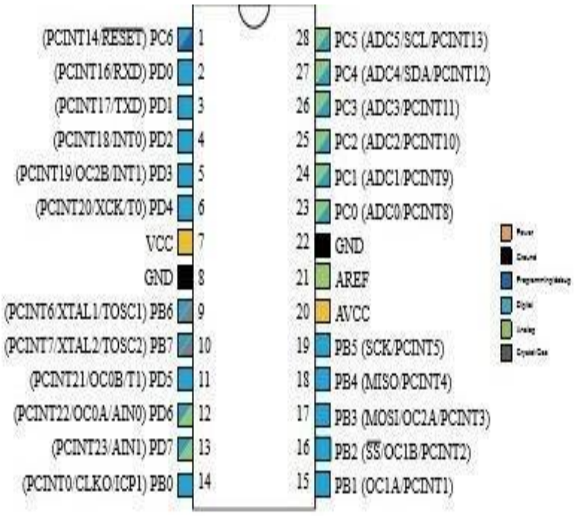


Figure.4.1.6: ATmega328P Ports

AVCC is a supply voltage pin for analog to digital converter.

VCC is a digital voltage supply

GND denotes Ground and it has a 0V.

Port A consists of the pins from PA0 to PA7. These pins serve as analog input to analog to digital converters. If analog to digital converter is not used, port A acts as an eight (8) bit bidirectional input/output port.

Port B consists of the pins from PB0 to PB7. This port is an 8 bit bidirectional port having an internal pull-up resistor.

Port C consists of the pins from PC0 to PC7. The output buffers of port C **Port D** consists of the pins from PD0 to PD7. It is also an 8bit input/output port having an internal pull-up resistor. All of the AVR ports are shown in figure given below.

Power Supply

Every electrical and electronic device that we use in our day-to-day life will require a power supply. In general, we use an AC supply of 230V 50Hz, but this power has to be changed into the required form with required values or voltage range for providing power supply to different types of devices.

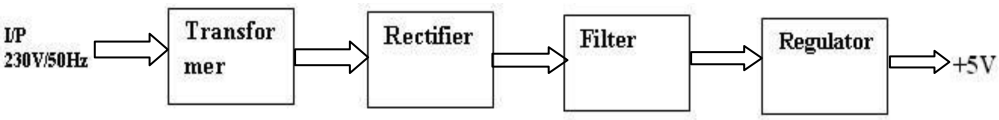


Figure.4.1.7: Block Diagram of Power supply

There are various types of power electronic converters such as step-down converter, step-up converter, voltage stabilizer, AC to DC converter, DC 25 to DC converter, DC to AC converter, and so on. For example, consider the microcontrollers that are used frequently for developing many embedded systems-based projects and kits used in real-time applications. These microcontrollers require a 5V DC supply.

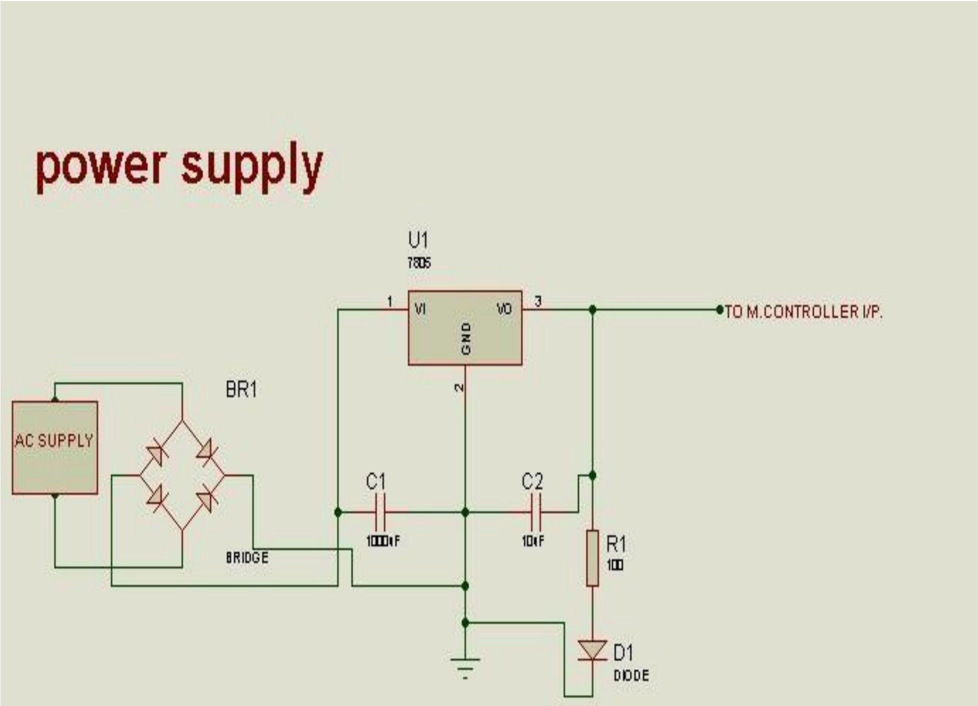


Figure. 4.1.8: Circuit diagram of Power supply

ZIGBEE

Xbee is the module using Zigbee protocol. Zigbee is a wireless communication protocol like Wi-Fi and Bluetooth. ZigBee defines a network layer above the 802.15.4 layers to support advanced mesh routing capabilities. Xbee can be used for wireless communication with low power consumption. A 3.6V 600mA Lithium battery may last 6 - 12 months for powering up an Xbee while the wireless range can up to 1 mile. It talks with well-known UART interface and makes it easy to use. It is simple and straight forward if you only use 2 Xbee for communication. People use this for their own electronics projects for wireless communication.

The Xbee Wireless Communication Starter Kit provides everything you need to harness the powerful and versatile technology of Zigbee 802.15.4 wireless communication. Do you want to free your robot of wires? Send sensory data back to your PC? Control your robot from up to 300 feet away without having to worry about interference issues inherit in most 2.4ghz technology? Zigbee technology is the answer. From simply 'cutting' your serial cable, to creating a wireless mesh network perfect for swarm robotics, you can take your robotics projects to the next level with Zigbee!

Figure.4.1.9: X-bee Module

This is the very popular 2.4GHz XBee module from Digi (formally Maxstream). These modules take the 802.15.4 stack (the basis for Zigbee) and wrap it into a simple to use serial command set. These modules allow a very reliable and simple communication between microcontrollers, computers, systems, really anything with a serial port! Point to point and multi-point networks supported.

ZigBee is a specification for a suite of high-level communication protocols using small, low-power digital radios based on the IEEE 802.15.4-2003 standard for wireless personal area networks (WPANs), such as wireless headphones connecting with cell phones via short-range radio. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other WPANs, such as Bluetooth. ZigBee is targeted at radio-frequency (RF) applications that require a low data rate, long battery life, and secure networking.

History of Zigbee Technology

In the year 1990, the digital radio networks with self-organizing ad hoc were implemented. The Zigbee specification like IEEE 802.15.4-2003 was approved in the year 2004, on December 14. The Specification 1.0 was announced by Zigbee Alliance in the year 2005, on June 13, called the Specification of ZigBee 2004.

Overview

ZigBee is a low-cost, low-power, wireless mesh networking proprietary standard. The low cost allows the technology to be widely deployed in wireless control and monitoring applications, the low power-usage allows longer life with smaller batteries, and the mesh networking provides high reliability and larger range. The ZigBee Alliance, the standards body that defines ZigBee, also publishes application profiles that allow multiple OEM vendors to create interoperable products.

ZigBee operates in the industrial, scientific and medical (ISM) radio bands; 868 MHz in Europe, 915 MHz in the USA and Australia, and 2.4 GHz in most jurisdictions worldwide. Because ZigBee can activate (go from sleep to active mode) in 15 msec or less, the latency can be very low and devices can be very responsive — particularly compared to Bluetooth wake-up delays, which are typically around three seconds. Because ZigBee's can sleep most of the time, average power consumption can be very low, resulting in long battery life.

Protocol

The protocols build on recent algorithmic research (Ad-hoc On-demand Distance Vector, neuRFon) to automatically construct a low-speed ad-hoc network of nodes. In most large network instances, the network will be a cluster of clusters. It can also form a mesh or a single cluster. The current profiles derived from the ZigBee protocols support beacon and non-beacon enabled networks.

Zigbee specification

ZigBee is the specification of a low-cost, low-power wireless communications solution, meant to be integrated as the main building block of ubiquitous networks.

It is maintained by the ZigBee Alliance, which develops the specification and certifies its proper implementation. As of 2007, the latest publicly available revision is the 2006 version.

Zigbee Architecture

Zigbee system structure consists of three different types of devices as Zigbee Coordinator, Router, and End device. Every Zigbee network must consist of at least one coordinator which acts as a root and bridge of the network. The coordinator is responsible for handling and storing the information while performing receiving and transmitting data operations.

Zigbee routers act as intermediary devices that permit data to pass to and for through them to other devices. End devices have limited functionality to communicate with the parent nodes such that the battery power. The number of routers, coordinators, and end devices depends on the type of networks such as star, tree, and mesh networks. Zigbee protocol architecture consists of a stack of various layers where IEEE 802.15.4 is defined by physical and MAC layers while this protocol is completed by accumulating Zigbee's own network and application layers.

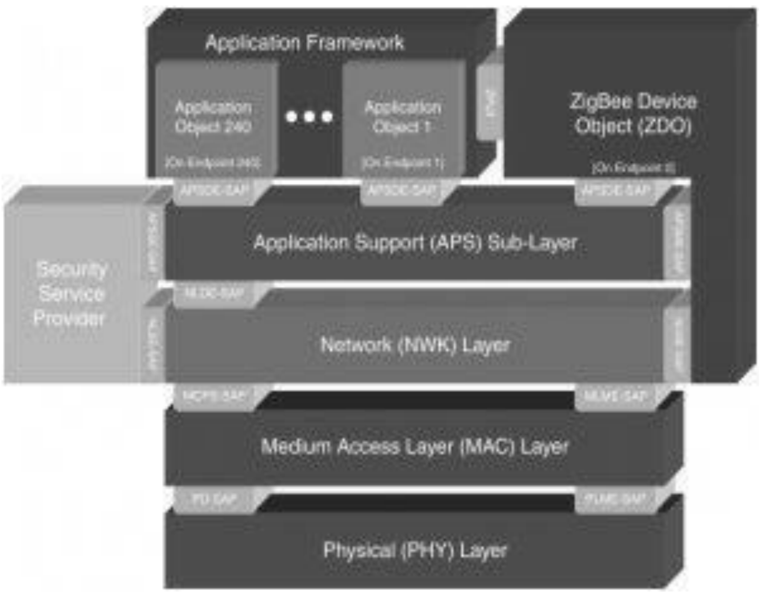


Figure.4.1.10: Zigbee Architecture

- **Physical Layer**: This layer does modulation and demodulation operations upon transmitting and receiving signals respectively. This layer's frequency, data rate, and a number of channels are given below.
- **MAC Layer**: This layer is responsible for reliable transmission of data by accessing different networks with the carrier sense multiple access collision avoidances (CSMA). This also transmits the beacon frames for synchronizing communication.
- **Network Layer**: This layer takes care of all network-related operations such as network setup, end device connection, and disconnection to network, routing, device configurations, etc.
- **Application Support Sub-Layer**: This layer enables the services necessary for Zigbee device objects and application objects to interface with the network layers for data managing services. This layer is responsible for matching two devices according to their services and needs.
- **Application Framework**: It provides two types of data services as key-value pair and generic message services. The generic message is a developer-defined structure, whereas the key-value pair is used for getting attributes within the application objects. ZDO provides an interface between application objects and the APS layer in Zigbee devices. It is responsible for detecting, initiating, and binding other devices to the network.

Zigbee Operating Modes

Zigbee two-way data is transferred in two modes: Non-beacon mode and Beacon mode. In a beacon mode, the coordinators and routers continuously monitor the active state of incoming data hence more power is consumed. In this mode, the routers and coordinators do not sleep because at any time any node can wake up and communicate.

These beacon networks are work for time slots which means, they operate when the communication needed results in lower duty cycles and longer battery usage. These beacon and non-beacon modes of Zigbee can manage periodic (sensors data), intermittent (Light switches), and repetitive data types.

Zigbee Topologies

Zigbee supports several network topologies; however, the most commonly used configurations are star, mesh, and cluster tree topologies. Any topology consists of one or more coordinators. In a star topology, the network consists of one coordinator which is responsible for initiating and managing the devices over the network. All other devices are called end devices that directly communicate with the coordinator. This is used in industries where all the endpoint devices are needed to communicate with the central controller, and this topology is simple and easy to deploy. In mesh and tree topologies, the Zigbee network is extended with several routers where the coordinator is responsible for staring them.

If any node fails, the information is routed automatically to other devices by these topologies. As redundancy is the main factor in industries, hence mesh topology is mostly used. In a cluster-tree network, each cluster consists of a coordinator with leaf nodes, and these coordinators are connected to the parent coordinator which initiates the entire network.

Zigbee Devices

The specification of IEEE 802.15.4 Zigbee mainly includes two devices like Full-Function Devices (FFD) as well as Reduced-Function Devices (RFD). An FFD Device performs different tasks which are explained within the specification & it can adopt any task within the network.

In an IEEE 802.15.4 n/w, the Zigbee devices play three different roles like Coordinator, PAN Coordinator & Device. Here, FFD devices are Coordinator as well as PAN Coordinator whereas the Device is either an RFD/ FFD Device.

Basic Security Model

The basic mechanism to ensure confidentiality is the adequate protection of all keying material. Trust must be assumed in the initial installation of the keys, as well as in the processing of security information. In order for an implementation to globally work, its general correctness (e.g., conformance to specified behaviours) is assumed.

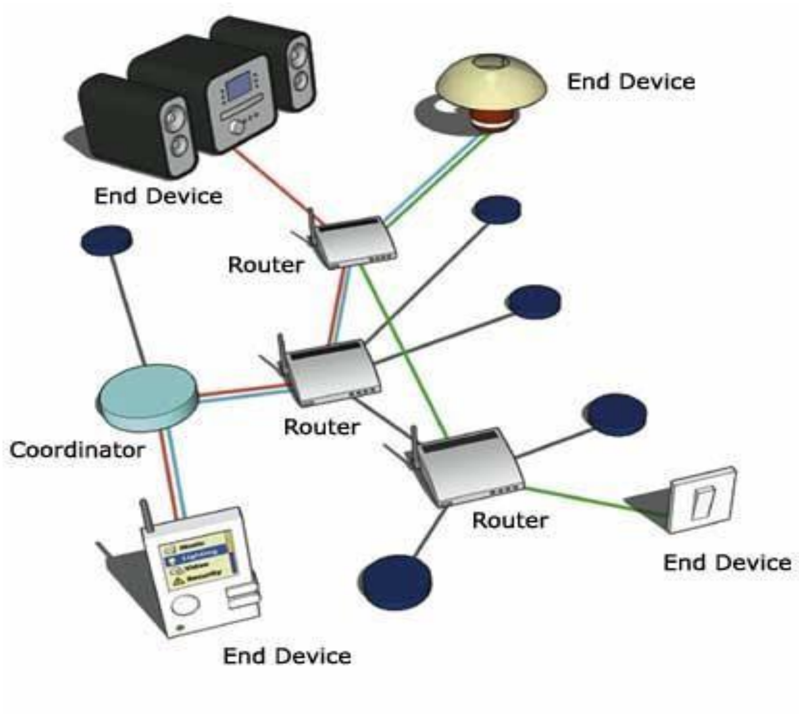


Figure.4.1.11: Basic Security Model

Why Zigbee is Better than Wi-Fi?

In Zigbee, the data transfer speed is less as compared with WiFi, so it's the highest speed is simply 250kbps. It is very less as compared with the less speed of WiFi. One of the most best quality of Zigbee is the rate of power utilization as well as the life of the battery. Its protocol lasts for several months because once it is assembled then we can forget.

What Devices use Zigbee?

The following list of devices supports the ZigBee protocol.

- Belkin WeMo
- Samsung SmartThings
- Yale smart locks
- Philips Hue

- Thermostats from Honeywell
- Ikea Tradfri
- Security Systems from Bosch
- Comcast Xfinity Box from Samsung
- Hive Active Heating & accessories
- Amazon Echo Plus
- Amazon Echo Show

Instead of connecting every Zigbee device separately, a central hub is required for controlling all the devices. The central hub will scan the network for all the supported devices and provides you simple control of the above devices with a central app.

Zigbee Advantages

The advantages of Zigbee include the following.

- This network has a flexible network structure
- Battery life is good.
- Power consumption is less
- Very simple to fix.
- It supports approximately 6500 nodes.
- Less cost.
- It is self-healing as well as more reliable.
- Network setting is very easy as well as simple.
- Loads are evenly distributed across the network because it doesn’t include a central controller
- Home appliances monitoring as well controlling is extremely simple using remote
- The network is scalable and it is easy to add/remote ZigBee end device to the network.

Applications of Zigbee

The applications of ZigBee technology include the following.

- **Industrial Automation:** In manufacturing and production industries, a communication link continually monitors various parameters and critical equipment. Hence Zigbee considerably reduces this communication cost as well as optimizes the control process for greater reliability.
- **Home Automation:** Zigbee is perfectly suited for controlling home appliances remotely as a lighting system control, appliance control, heating, and cooling system control, safety equipment operations and control, surveillance, and so on.
- **Smart Metering:** Zigbee remote operations in smart metering include energy consumption response, pricing support, security over power theft, etc.
- **Smart Grid monitoring:** Zigbee operations in this smart grid involve remote temperature monitoring, fault locating, reactive power management, and so on.

LCD (Liquid Crystal Display)

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. Some of the most common LCDs connected to the controllers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines (RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

Pin	Symbol	I/O	Description
1	GND	-	Ground
2	Vcc	-	+5V power supply
3	VEE	-	Contrast control
4	RS	I	command/data register selection
5	R/W	I	write/read selection
6	E	I/O	Enable
7	DB0	I/O	The 8-bit data bus
8	DB1	I/O	The 8-bit data bus
9	DB2	I/O	The 8-bit data bus
10	DB3	I/O	The 8-bit data bus
11	DB4	I/O	The 8-bit data bus
12	DB5	I/O	The 8-bit data bus
13	DB6	I/O	The 8-bit data bus
14	DB7	I/O	The 8-bit data bus

Table.4.1.2: LCD Pin Description

Description of 16X2

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all-Parallel Ports. It however doesn't show the use of the Status Port as an input. So, what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board. 16×2 LCD is named so because; it has 16 Columns and 2 Rows.

Schematic Diagram

The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore, by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

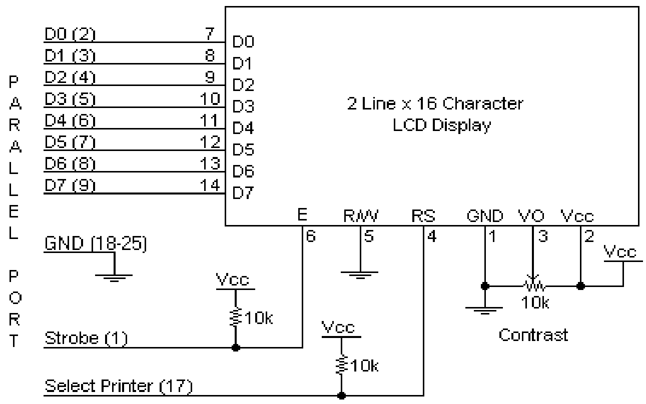


Figure.4.1.12: Schematic Diagram of LCD

Code (hex)	Command to LCD Instruction Register
1	Clear display screen
2	Return home
4	Shift cursor to left
5	Shift display right
6	Shift cursor to right
7	Shift display left
8	Display off, Cursor off
A	Display off, Cursor on
C	Display on, cursor off
E	Display on, cursor blinking
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80	Force cursor to beginning of 1st line
C0	Force cursor to beginning of 2nd line
38	2 lines and 5x7 matrix

Table 4.1.3: LCD commands

Pin Description

- Pin1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.

- Pin2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
- Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
- Pin4 (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).
- Pin5 (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).
- Pin 6 (Enable/Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit & constantly held high.
- Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.
- Pin15 (+ve pin of the LED): This pin is connected to +5V
- Pin 16 (-ve pin of the LED): This pin is connected to GND.

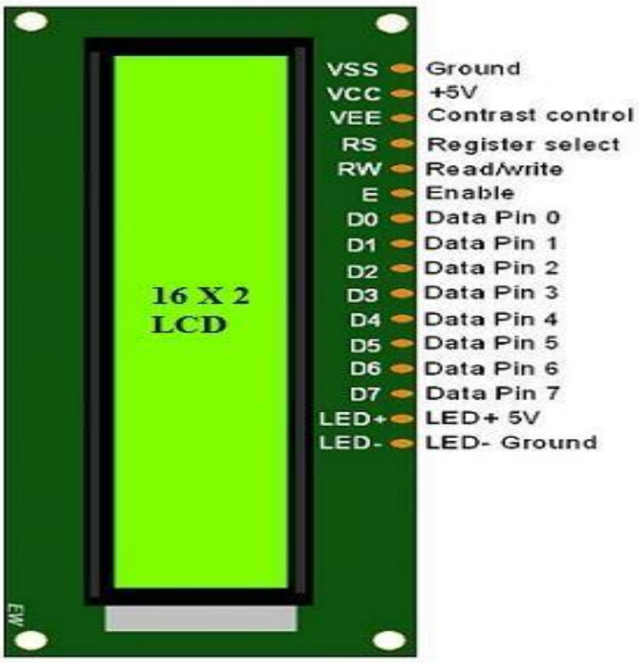


Figure.4.1.13: LCD-16X2 Pin Diagram

Features of LCD16X2

The features of this LCD mainly include the following.

- The operating voltage of this LCD is 4.7V-5.3V
- It includes two rows where each row can produce 16-characters.
- The utilization of current is 1mA with no backlight
- Every character can be built with a 5×8-pixel box
- The alphanumeric LCDs alphabets & numbers
- Is display can work on two modes like 4-bit & 8-bit
- These are obtainable in Blue & Green Backlight.
- It displays a few custom generated characters

Registers of LCD

A 16×2 LCD has two registers like data register and command register. The RS (register select) is mainly used to change from one register to another. When the register set is ‘0’, then it is known as command register. Similarly, when the register set is ‘1’, then it is known as data register.

(i) Command Register

The main function of the command register is to store the instructions of command which are given to the display. So those predefined tasks can be performed such as clearing the display, initializing, set the cursor place, and display control. Here commands processing can occur within the register.

(ii) Data Register

The main function of the data register is to store the information which is to be exhibited on the LCD screen. Here, the ASCII value of the character is the information which is to be exhibited on the screen of LCD. Whenever we send the information to LCD, it transmits to the data register, and then the process will be starting there. When register set =1, then the data register will be selected.

LCD Interfacing with the Arduino Module

The following circuit diagram shows the liquid crystal display with the Arduino module. From the circuit diagram, we can observe that the RS pin of the

LCD is connected to the pin 12 of the Arduino. The LCD of R/W pin is connected to the ground. The pin 11 of the Arduino is connected to the enable signal pin of LCD module. The LCD module & Arduino module are interfaced with the 4-bit mode in this project. Hence there are four input lines which are DB4 to DB7 of the LCD. This process very simple, it requires fewer connection cables and also, we can utilize the most potential of the LCD module.

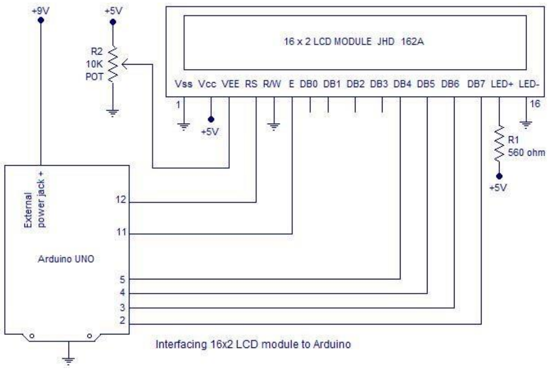


Figure.4.1.14: LCD Interfacing with the Arduino.

The digital input lines (DB4-DB7) are interfaced with the Arduino pins from 5-2. To adjust the contrast of the display here we are using a 10K potentiometer. The current through the back LED light is from the 560-ohm resistor. The external power jack is provided by the board to the Arduino. Using the PC through the USB port the Arduino can power. Some parts of the circuit can require the +5V power supply it is taken from the 5V source on the Arduino board.

Applications

- In most of the applications that's have only small values to show, uses the LCD.
- Most of the commercial meters use this module to represent the data output.
- In the toys and developing projects, it is still vastly in use.
- In black and white printers, it helps to show the printer settings and status.

USB-TTL

This USB-TTL converter is a port-powered bi-directional USB to TTL/CMOS 5V converter, which can be used to convert any standard full-duplex

USB port into a full-duplex TTL port and vice versa. The unit is powered from the USB port and it supports data auto-sensing & self-adjusting, and therefore, no baud rate setting is required. USB to TTL /stc microcontroller programmer / pl2303 in nine upgrades plate with a transparent cover compatible with Arduino, raspberry pi, avr, pic, 8051, etc. The USB to rs232 module based TTL provides the best and convenient way to connect your rs232 TTL devices or demo board to your pc via the USB port.

The purpose of this device to connect your PC to a serial (TTL level) device. The CP2102 is a highly-integrated USB-to-UART Bridge Controller providing a simple solution for updating RS-232 designs to USB using a minimum of components and PCB space. The CP2102 includes a USB 2.0 full-speed function controller, USB transceiver, oscillator, EEPROM, and asynchronous serial data bus (UART) with full modem control signals in a compact 5 x 5 mm QFN-28 package. No other external USB components are required.

Working

The CP2102 is a highly-integrated USB-to-UART Bridge Controller providing a simple solution for updating RS-232 design to USB using a minimum of components and PCB space. Cp2102 Based boards are better, reliable and faster than Cheap PL2303 based boards. This board can be easily used in conjunction with Arduino Pro/ Arduino Por Mini Boards for Programming.

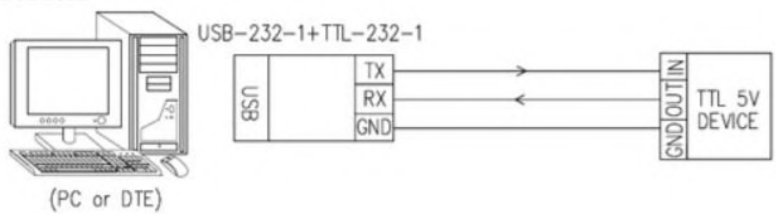


Figure.4.1.15: USB-TTL Converter Connection to PC

Features

- Stable and reliable chipset CP2102
- USB specification 2.0 compliant with full-speed 12Mbps
- Standard USB type A male and TTL 6pin connector.

- 6pins for 3.3V, RST, TXD, RXD, GND & 5V
- All handshaking and modem interface signals
- Baud rates: 300 bps to 1.5 Mbps
- Byte receive buffer; 640 byte transmit buffer
- Hardware or X-On/X-Off handshaking supported
- Event character support Line break transmission
- USB suspend states supported via SUSPEND pins
- Temperature Range: -40 to +85
- Supports Windows 98SE, 2000, XP, Window7, Mac OS 9 & Linux 2.4

Pin Description

- TX = Connect to Transmit Pin (TXD) of Micro controller. This pin is RX pin of CP2102 on board.
- RX = Connect to Receive Pin (RXD) of Micro controller. This pin is TX pin of CP2102 on board.
- RST = Normally Unconnected. Reset Pin for CP2102. Initiate a system reset by driving this pin low for atleast 15 µs.
- GND = Should be common to microcontroller ground.
- 3V3 = Optional output to power external circuit upto 50mA.
- 5V = Optional output to power external circuit upto 100mA

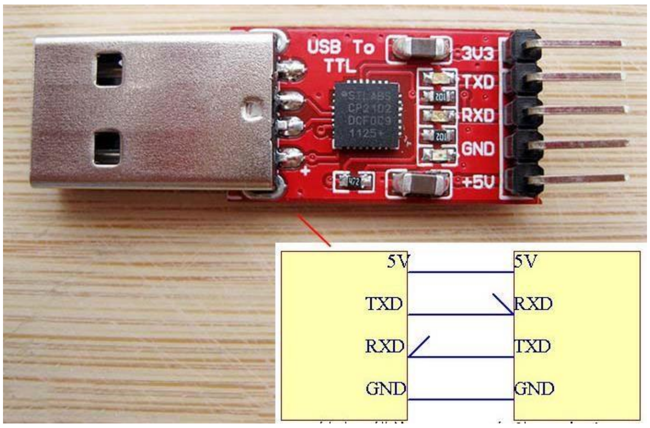


Figure.4.1.16: USB-TTL Converter

Troubleshooting

- Make sure that the USB-TTL converter is connected to your USB port, and the driver is installed correctly (as shown below); otherwise, please reinstall the driver.

- Perform a loopback test by using CommFront's 232Analyzer software: Connect a USB-TTL converter to your USB port, short pin TX (TTL Out) to RX (TTL in) on the termination board, and then send commands from the 232Analyzer software. You should receive an echo of the commands sent. By performing a simple loopback test like this, you can test both the transmitter and receiver of the USB-TTL converter. This is very helpful when you are in doubt about the performance of your converter.

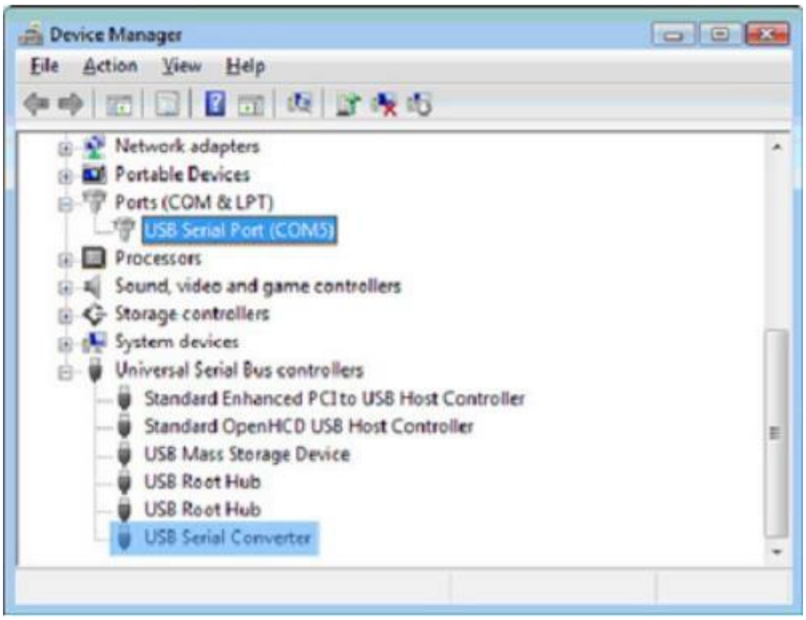


Figure.4.1.17: USB-TTL driver connected to PC

KEYBOARD

A keyboard is one of the primary input devices that allows users to input text into a computer or any other electronic machinery. It is a peripheral device that is the most basic way for the user to communicate with a computer. It consists of multiple buttons, which create numbers, symbols, and letters, and special keys like the Windows and Alt key, including performing other functions. The design of the keyboard comes from the typewriter keyboards, and numbers and letters are arranged on the keyboard in that way, which helps to type quickly.

Keyboards are divided into two main categories based on input method. These categories are:

- **Wired Keyboards**

A wired keyboard means there is a wire connecting your keyboard to your computer. At the end of the wire is a USB plug that goes into a USB port on your computer. Wired keyboards are extremely reliable—there is little that could go wrong with this direct connection. The one drawback to having a wired keyboard is that your distance is limited between the keyboard and your computer; you can only have your keyboard as far from your desktop or laptop as the wire length.

- **Wireless Keyboards**

A wireless keyboard works just like a wireless mouse; you plug a receiver into one of the USB ports on your computer. The receiver then sends a signal to your battery-powered keyboard. Keep in mind, wireless USB keyboards do have a limited range and some must be within the line of sight of your receiver, depending on the type of signal it uses. Also, if your cordless keyboard starts acting sluggish, it probably requires new batteries.

Figure.4.1.18: Keyboard

How it Works

A keyboard consists of two parts: a set of keys that are pushed in sequence by the typist and an encoder that identifies each pressed key and generates a code that uniquely identifies that key. The key set includes the standard alphanumeric keys found on old typewriters and additional keys, such as cursor keys, navigation and function keys, Apple or Windows keys, and a numeric keypad. Keyboards for laptop computers have the minimum number of keys.

The encoder is a microprocessor located in the keyboard that detects each key as it is pressed and released. When the typist presses a key, a connection is made on the grid. If, for example, the connection is in the first row and the third column, the encoder immediately identifies the pressed key and sends a special signal, called a "scanning code," to the computer. The computer translates the scanning code into the appropriate binary code and displays the character on the monitor so that the typist can verify that the correct key was pushed.

The lights on the keyboard (for Caps Lock, Num Lock, Scroll Lock, and so on) are controlled by the computer, not the keyboard. For example, when the typist presses the Caps Lock key, the keyboard encoder sends the code for the Caps Lock key to the computer, and the computer turns on the keyboard's Caps Lock light.

Software Requirements

- Arduino IDE software
- Proteus
- X-CTU

Arduino IDE

The Arduino integrated development atmosphere (IDE) may be a cross-platform application (for Windows, macOS, Linux) that's written within the artificial language Java. it's wont to write and transfer programs to Arduino compatible boards, but also, with the assistance of third-party cores, different trafficker development boards. The ASCII text file for the IDE is free beneath the wildebeest General Public License, version 2. The Arduino IDE supports the languages C and C++ victimisation special rules of code structuring. The Arduino IDE provides a software package library from the Wiring project, that provides several common input and output procedures. User-written code solely needs 2 basic functions, for beginning the sketch and also the main program loop, that are compiled associated connected with a program stub main () into an feasible cyclic programme with the wildebeest tool chain, additionally enclosed with the IDE distribution. The Arduino IDE employs the program argued to convert the feasible code into a document in positional representation system secret writing that's loaded into the Arduino board by a loader program within the board's computer code.

The key highlights are:

- Arduino sheets can understand simple or computerized information signals from various sensors and transform it into a yield, for example, actuating an engine, turning LED on/off, associate with the cloud and numerous different activities.
- You can control your board capacities by sending a lot of guidelines to the microcontroller on the board by means of Arduino IDE (alluded to as transferring programming).
- Dissimilar to most past programmable circuit sheets, Arduino needn't bother with an additional bit of equipment (called a developer) so as to stack another code onto the board. You can basically utilize a USB link.
- Moreover, the Arduino IDE utilizes a rearranged rendition of C++, making it simpler to figure out how to program.
- At long last, Arduino gives a standard structure factor that breaks the elements of the smaller scale controller into an increasingly available bundle.

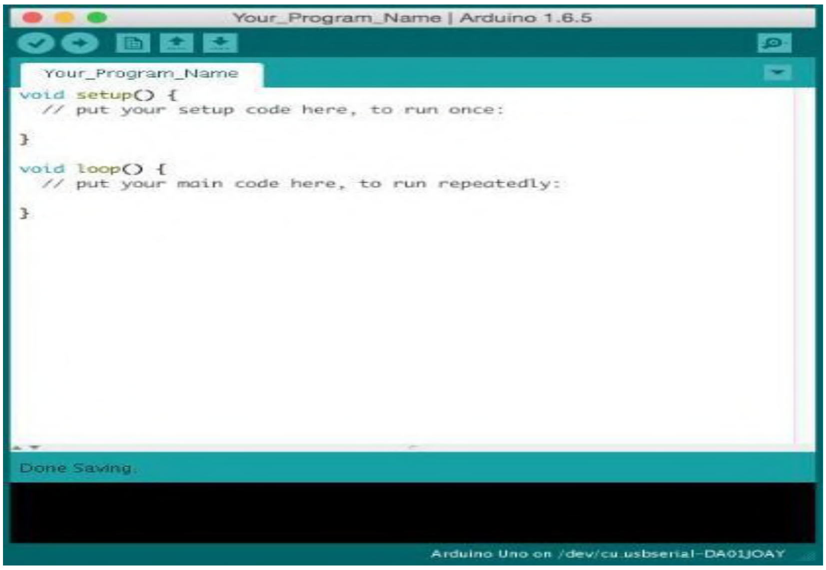


Figure.4.2.1:Program Page

How to Download the Arduino Software (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

Installation

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.4.2.2: USB Cable A plug to B plug

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig.4.2.3: USB Cable A plug to Mini-B plug

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

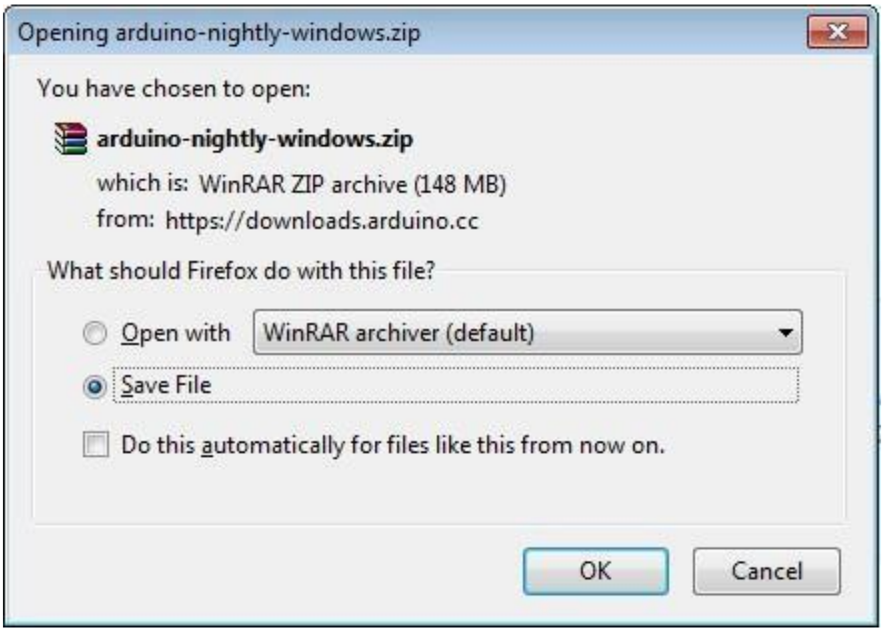


Figure.4.2.4: Unzip of Arduino IDE software

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port. Connect the Arduino board to your computer using the USB cable. The green power LED (labelled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder.

Inside the folder, you can find the application icon with an infinity label. Double-click the icon to start the IDE.

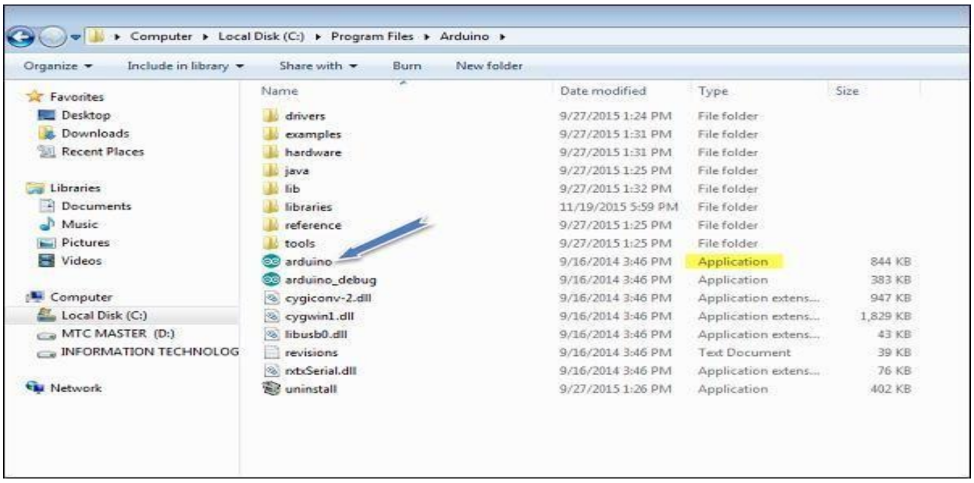


Figure.4.2.5: Open IDE software

Step 5 − Open your first project.

Once the software starts, you have two options:

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.

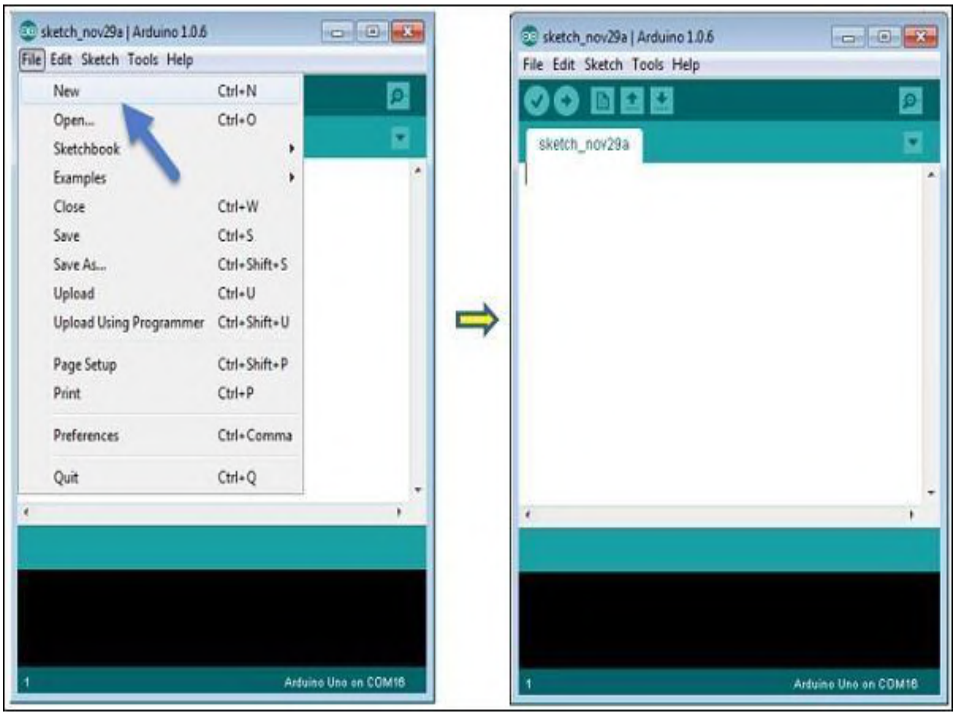


Figure.4.2.6: Open First Project

To open an existing project example, select File → Example → Basics → Blink.

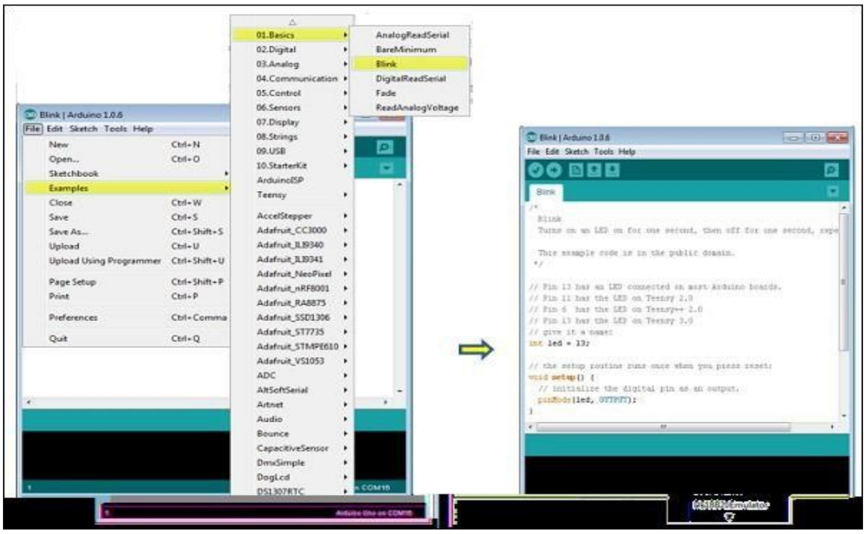


Figure.4.2.7: Open existing Project

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

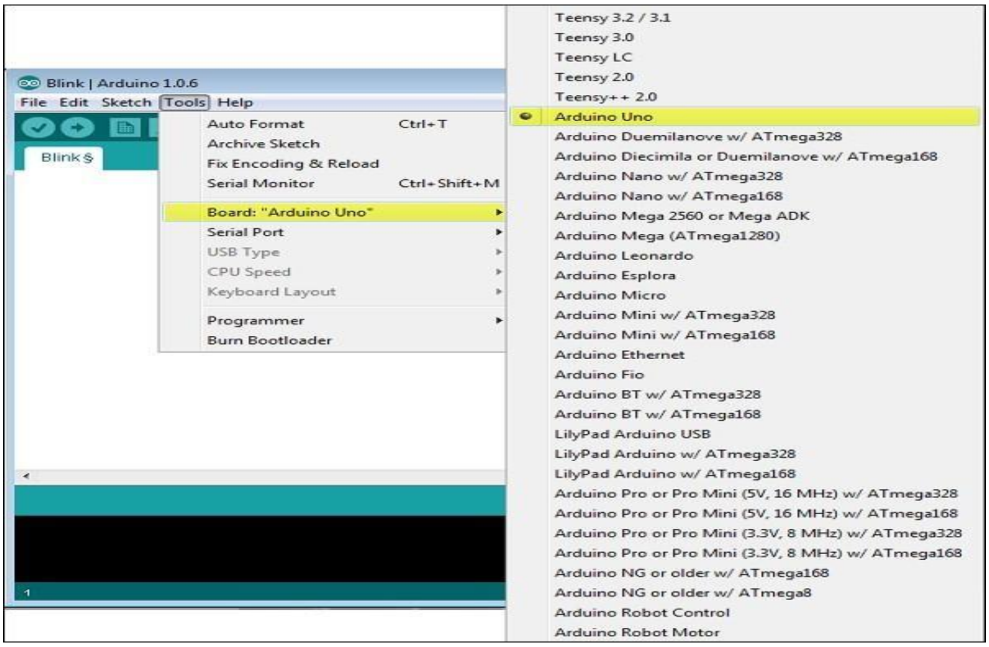


Figure.4.2.8: Selection of Arduino board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

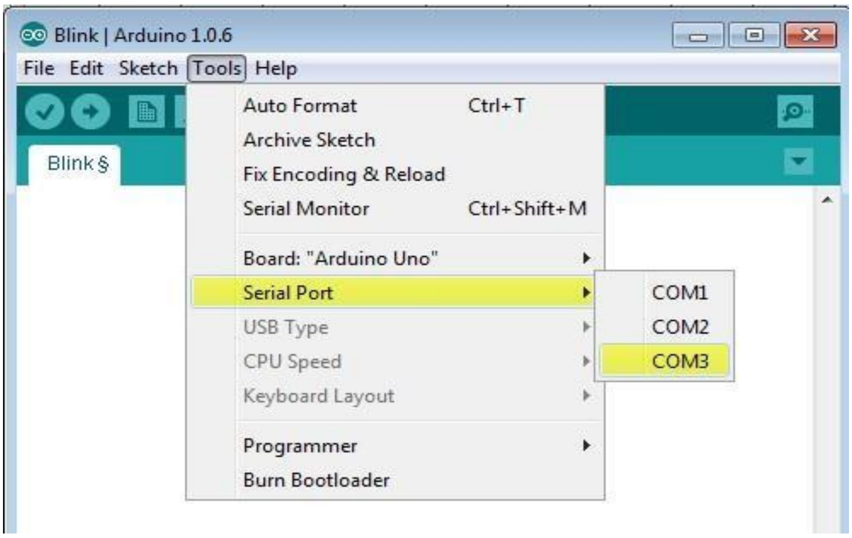


Figure.4.2.9: Selection of Serial Port

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

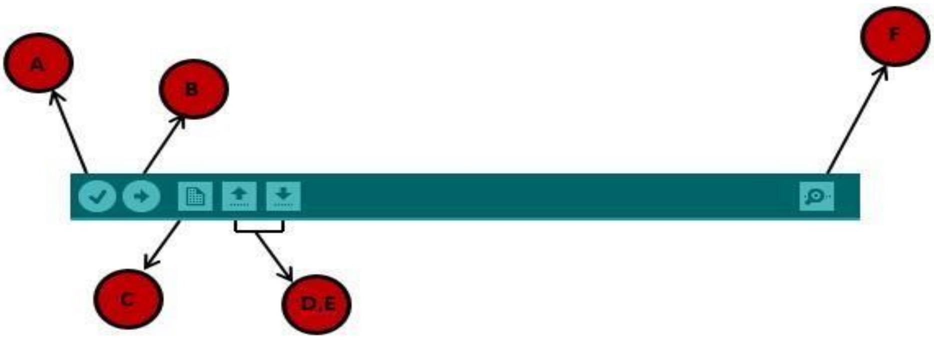


Figure.4.2.10: Uploading the program to board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F – Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Moving on

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called "sketches", but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

Figure.4.2.11: Program in Arduino-speak.

Proteus

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline "From concept to completion".

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc. ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design

Step 1: Open ISIS software and select New design in File menu

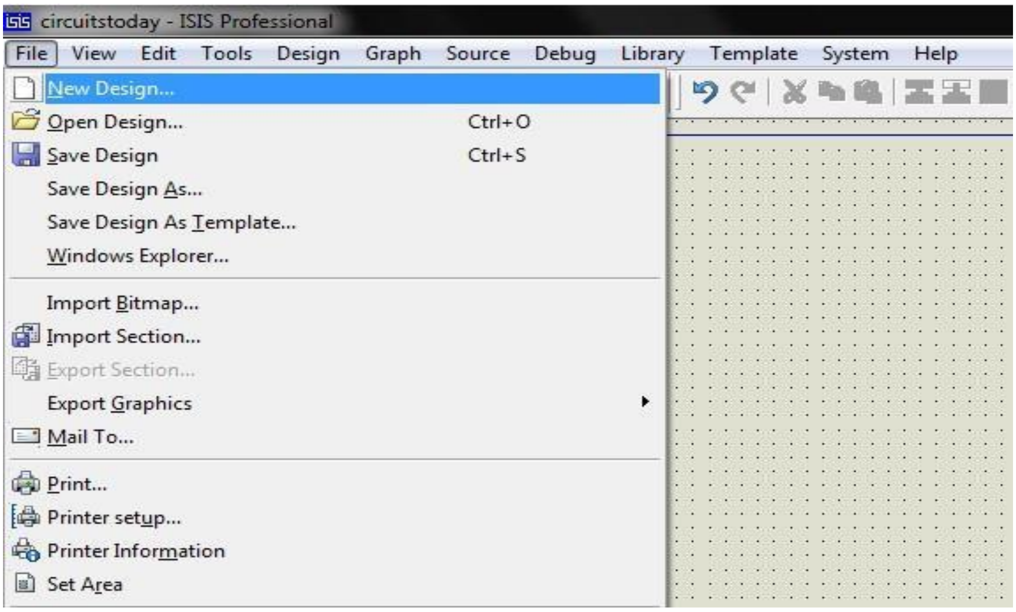


Figure.4.2.12: Proteus file menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

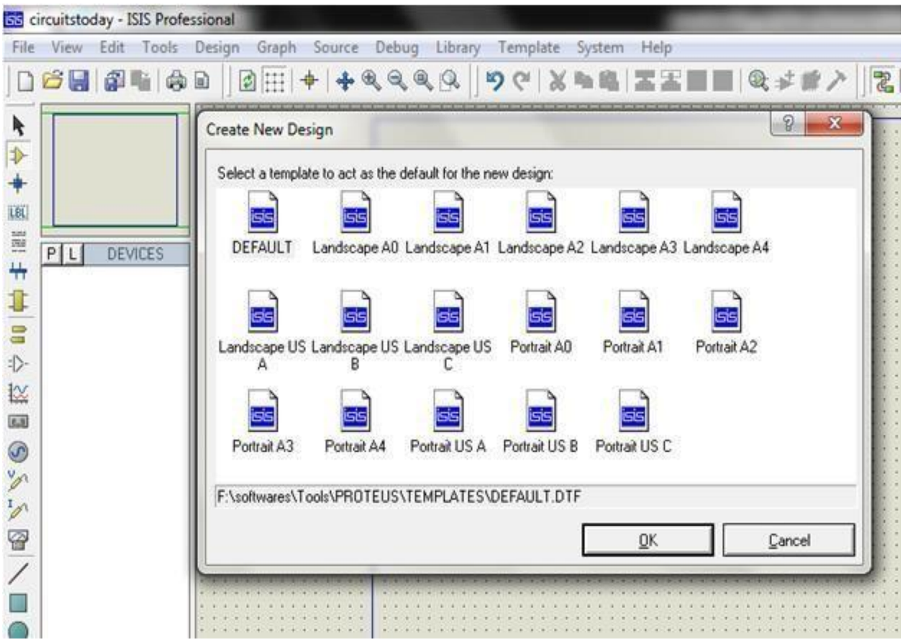


Figure. 4.2.13: Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

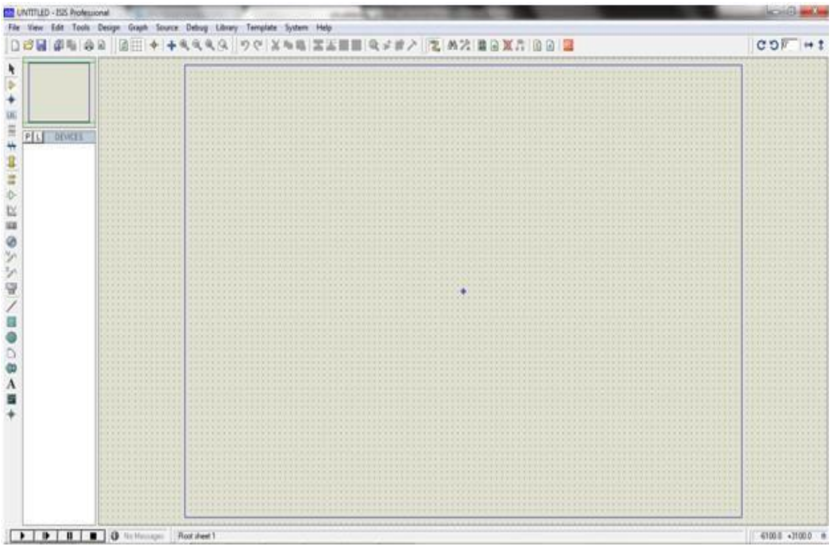


Figure. 4.2.14: Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

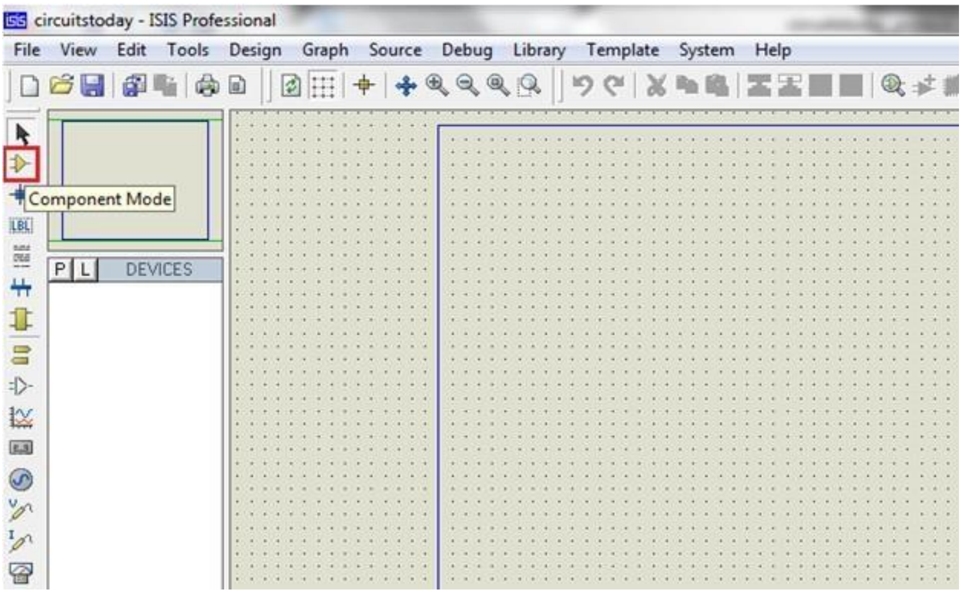


Figure. 4.2.15: Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

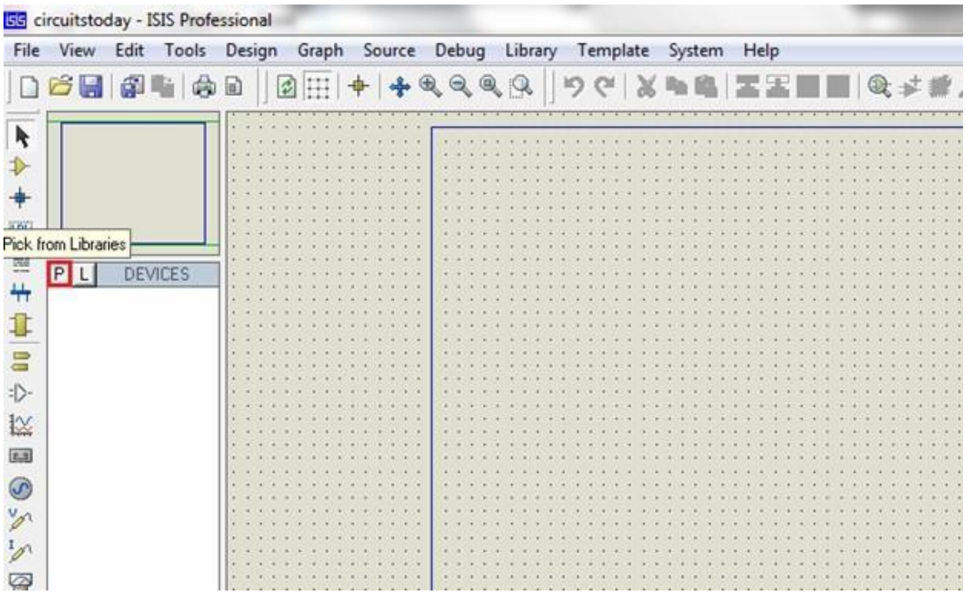


Figure. 4.2.16: Pick from Libraries

Step 6: Select the components from categories or type the part name in Keyword's text box.

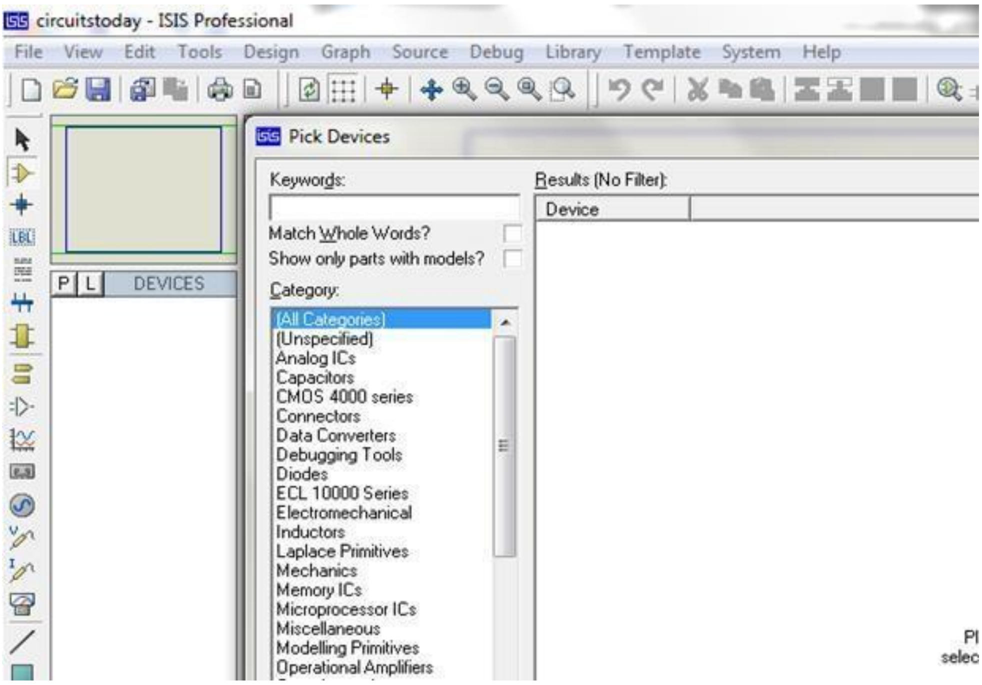


Figure.4.2.17: Keywords Textbox

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

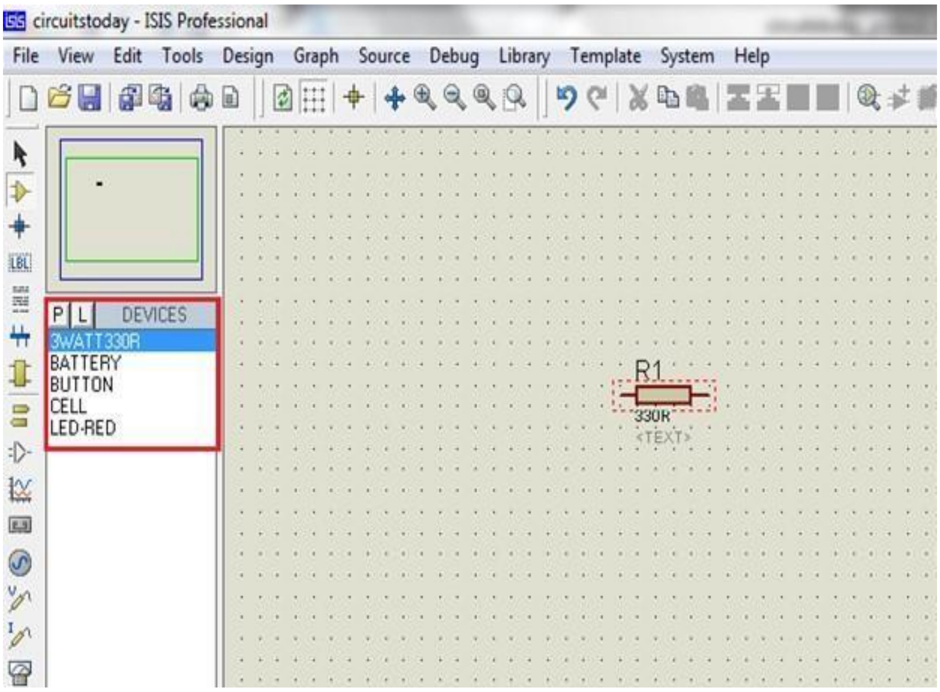


Figure. 4.2.18: Component Selection

Place all the required components and route the wires i.e., make connections. Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

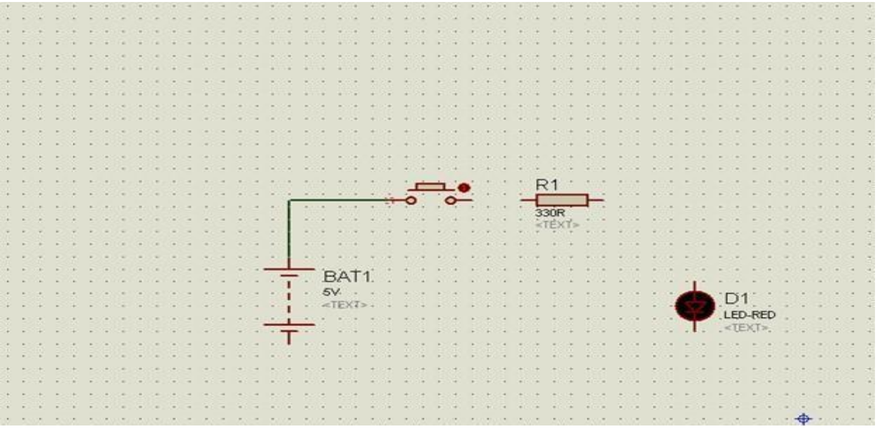


Figure.4.2.19: Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

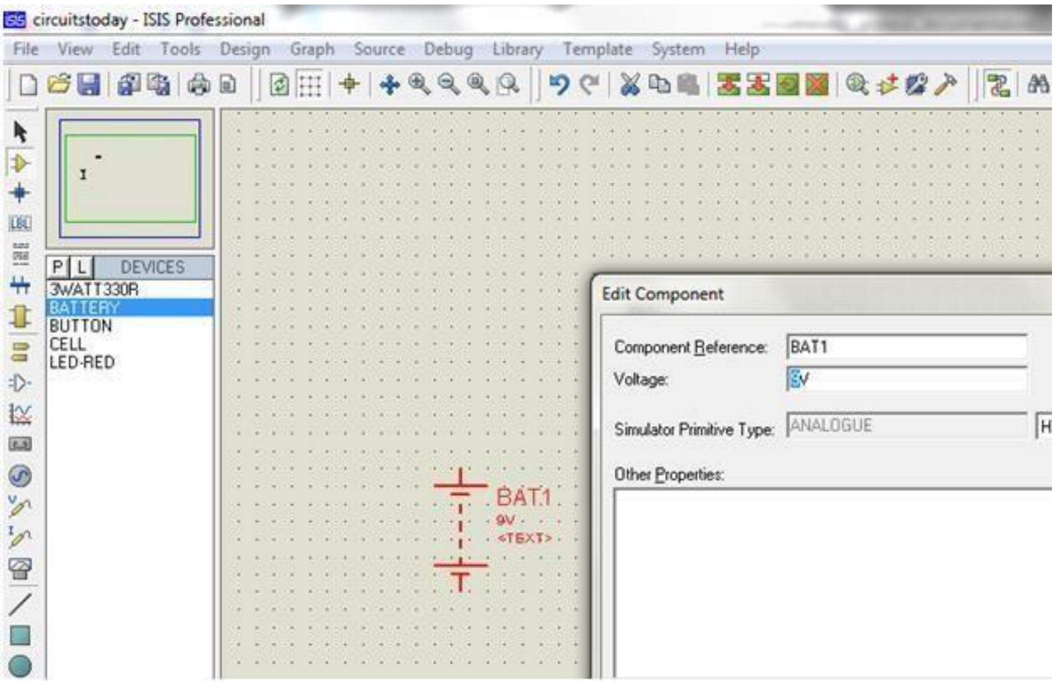


Figure.4.2.20: Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

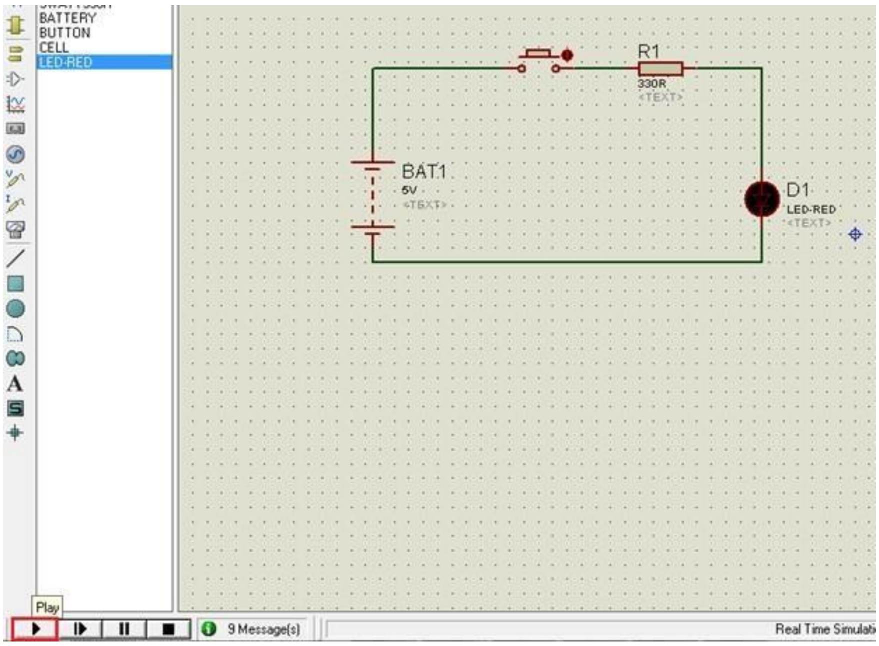


Figure.4.2.21: Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

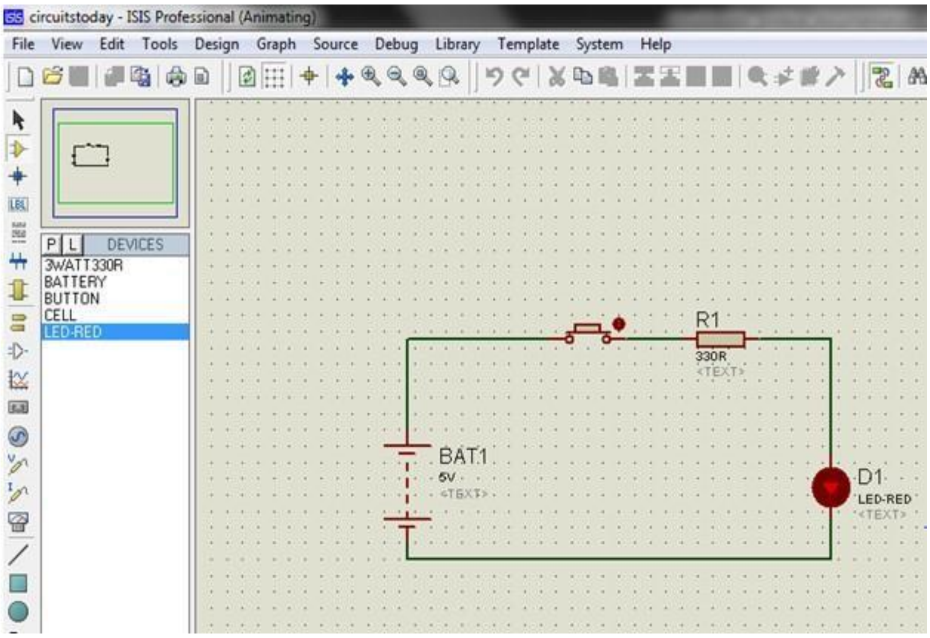


Figure.4.2.22: Simulation Animating

Simulation can be steppe Simulation can be stepped, paused or stopped at any time

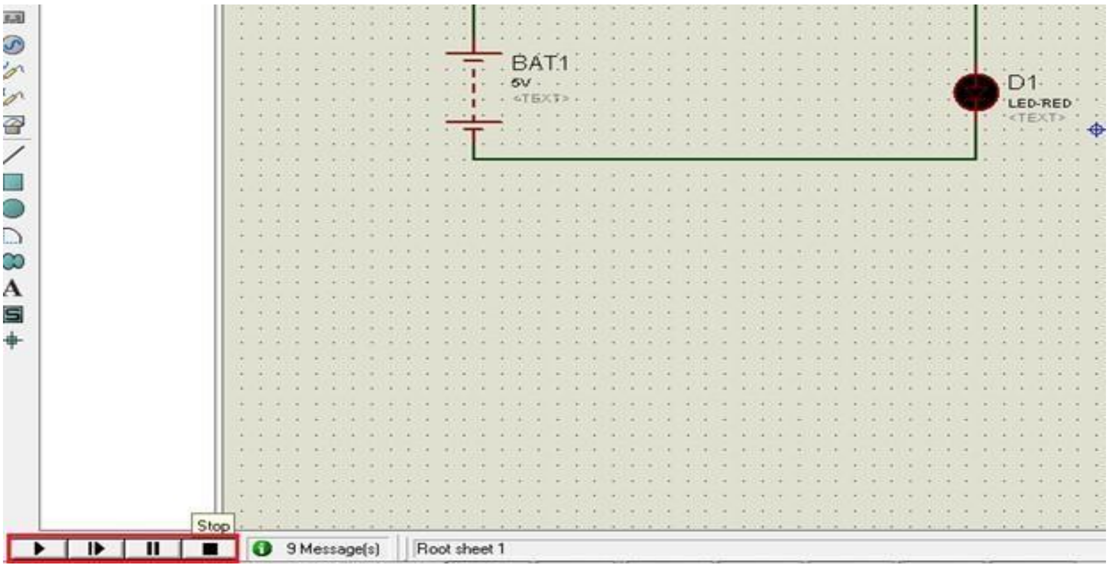


Figure.4.2.23: Simulation Step-Pause-Stop Buttons

X-CTU Software

- XCTU is a free, multi-platform application compatible with Windows, MacOS and Linux
- Graphical Network View for simple wireless network configuration and architecture
- API Frame Builder is a simple development tool for quickly building XBee API frames
- Firmware Release Notes Viewer allows users to explore and read firmware release notes

What is XCTU?

XCTU is a free multi-platform application designed to enable developers to interact with Digi RF modules through a simple-to-use graphical interface. It includes new tools that make it easy to set-up, configure and test XBee® RF modules.

XCTU includes all of the tools a developer needs to quickly get up and running with XBee. Unique features like graphical network view, which graphically represents the XBee network along with the signal strength of each connection, and the XBee API frame builder, which intuitively helps to build and interpret API frames for XBees being used in API mode, combine to make development on the XBee platform easier than ever.

Other highlights of XCTU include the following features:

- You can manage and configure multiple RF devices, even remotely (over-the-air) connected devices.
- The firmware update process seamlessly restores your module settings, automatically handling mode and baud rate changes.
- Two specific API and AT consoles, have been designed from scratch to communicate with your radio devices.
- You can now save your console sessions and load them in a different PC running XCTU.
- XCTU includes a set of embedded tools that can be executed without having any RF module connected:

- Frame’s generator: Easily generate any kind of API frame to save its value.
- **Frame’s interpreter**: Decode an API frame and see its specific frame values.
- **Recovery**: Recover radio modules which have damaged firmware or are in programming mode.
- **Load console session**: Load a console session saved in any PC running XCTU.
- **Range test**: Perform a range test between 2 radio modules of the same network.
- **Firmware explorer**: Navigate through XCTU's firmware library.

- An update process allows you to automatically update the application itself and the radio firmware library without needing to download any extra files.
- XCTU contains complete and comprehensive documentation which can be accessed at any time.

Installation of X-CTU

1. Download X-CTU at www.digi.com/xctu.
2. Browse to the folder to which you saved the above install file.
3. Double-click on the installer file and follow the X-CTU Setup Wizard.
4. When asked if you would like to check Digi's web site for firmware updates, click Yes.
5. After the firmware updates are complete, click Close. Updates may take few minutes, please be patient.
6. Start X-CTU by double-clicking on the X-CTU icon on your desktop, or by selecting Start > Programs Digi > X-CTU.
7. The X-CTU software is now ready to be used.

Zigbee wireless communication Set up

Step 1: Download X-CTU Software.

The X-CTU software is free to download and provides a simple interface to configure and update your XBee transceivers. With this software firmware updates are a breeze and configuration is simple.

Step 2: Configure 1st XBee as a coordinator

When opening the X-CTU software you should see a window like that shown. After selecting the proper COM port click the Test/Query button.

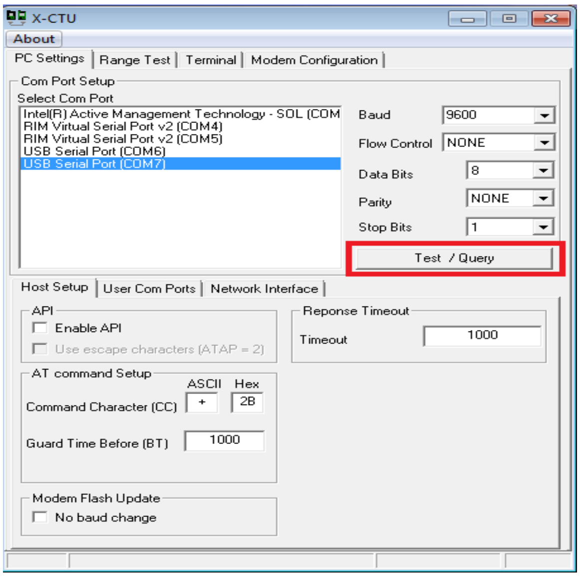


Figure.4.2.24: Configure of 1st Xbee as a transmitter

After selecting the Test/Query button, you should see a dialogue box like the one below. You will want to record the serial number shown as you will need it in a couple minutes.

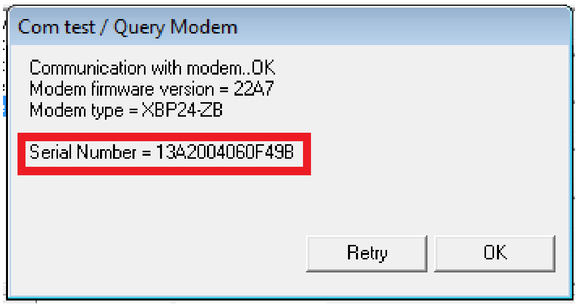


Figure.4.2.25: Com test / query modem

After recording the serial number, you can click OK. Next, select the Modem Configuration tab at the top of the window. Once here, select the read button. This will bring up the current configuration for the connected XBee and will be similar to the following:

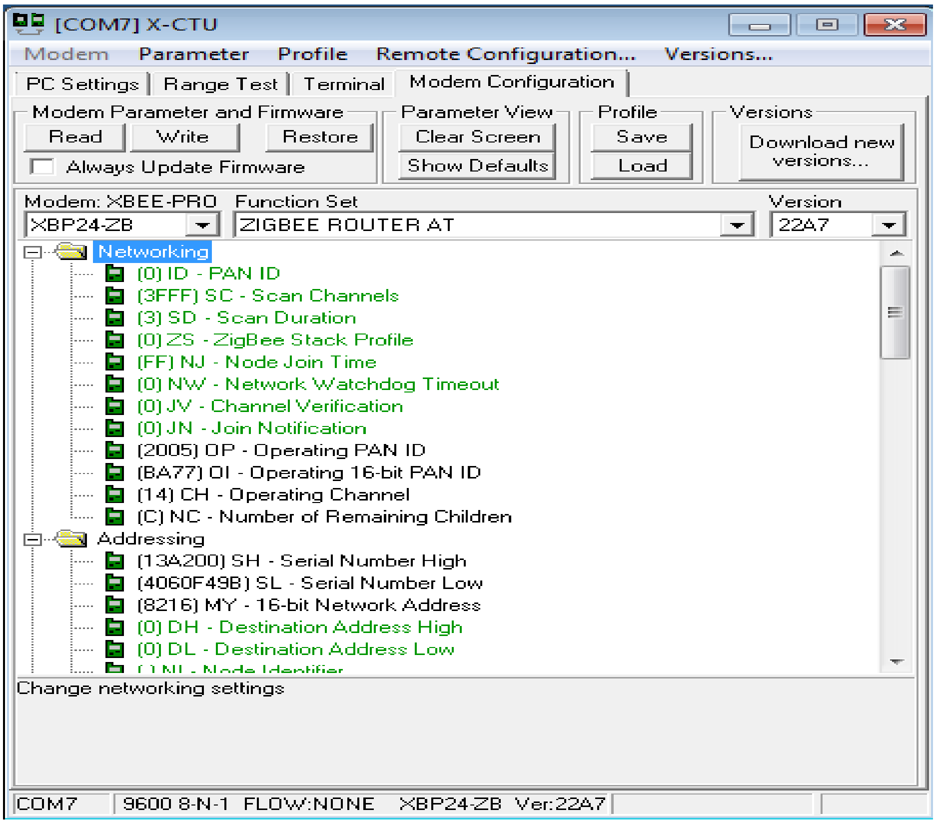


Figure.4.2.26:X-CTU networking

Once here, you want to select Zigbee Coordinator AT in the function set drop down menu.

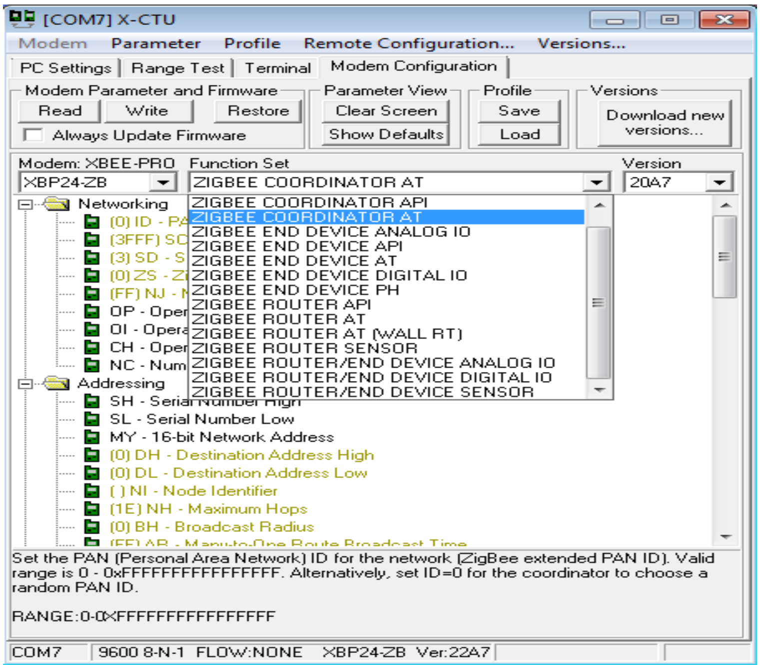


Figure.4.2.27: Zigbee Transmitter

Step 3: Configure 2nd XBee as Router

To configure the 2nd XBee, you will follow the same process as for the coordinator with one difference. In the PC settings tab, again, click the “Test/Query” button and record the serial number. Then in the Modem configuration tab, click the read button to load the current configuration of the XBee and set the PAN ID to the same ID used for the coordinator. The only change will be the function set you choose. For the second XBee we will set this as Zigbee Receiver AT.

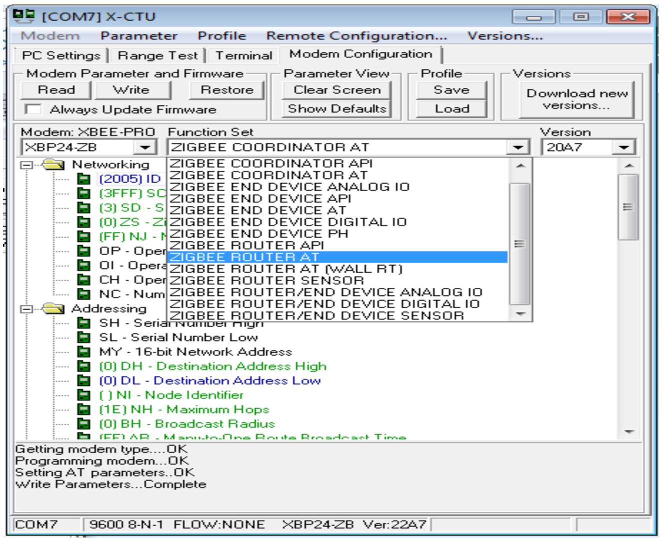


Figure.4.2.28: Zigbee Receiver

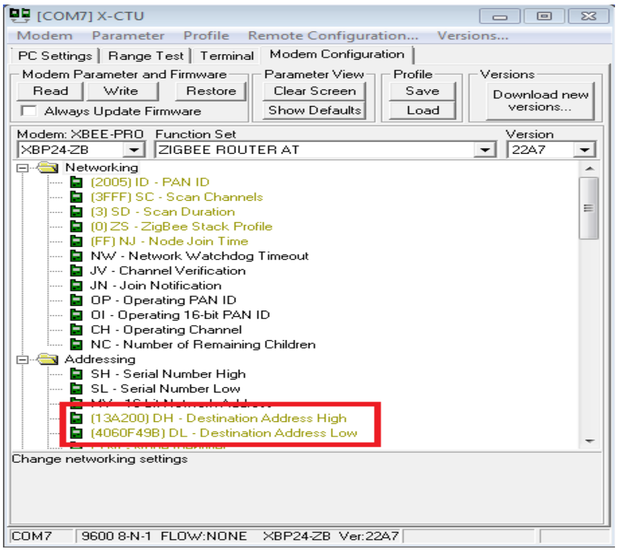


Figure.4.2.29 :X-CTU Receiver

Step 5: Test the configuration

A simple test can be done to be sure the two XBees are communicating properly. You can connect either one of the XBees to the computer. Then, connect the second XBee to 3.3V power and connect the Dout and Din pins together. This will cause the XBee to automatically retransmit any data it receives. When you have both XBees connected go to the Terminal tab in the X-CTU window. Whatever you type in the terminal window will appear in blue font and whatever is received will appear in red font. If the XBees are configured correctly every character you type should be mirrored in red.

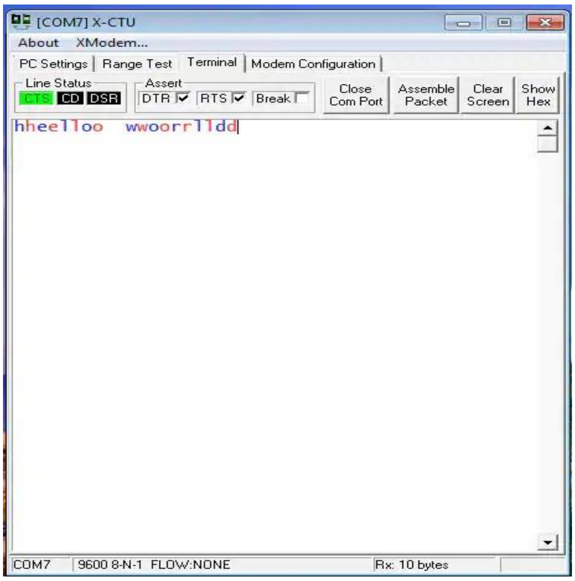


Figure.4.2.30: X-CTU sending information.

When typing single characters, you should see a screen similar to the one above. To send strings of data you can assemble a data packet. To do this click the "Assemble Packet" button and type the wanted string into the box then click send data. This will send the entire packet before receiving the same packet back.

If everything you type is being reflected back in red, you are successfully transmitting and receiving with your XBees!! You are now ready to use them however you wish. You can connect them to any microcontroller and transmit data through the UART peripheral or you can connect each XBee to a different computer and have a chat application. There are many more possibilities that are now at your fingertips with your working XBees.

Figure.4.2.31:Result of information sent.

CHAPTER-5

SOFTWARE TESTING AND CODE IMPLEMENTATION

Software Testing

Software Testing is a process of executing the application with an intent to find any software bugs.It is used to check whether the application met its expectations and all the functionalities of the applicationis working. The final goal of testing is to check whether the application is behaving in the way it is supposed to under specified conditions. All aspects of the code are examined to check the quality of application. The primary purpose of testing is to detect software failures so that defects may be uncoveredand corrected. The test cases are designed in such way that scope of finding the bugs is maximum.

Testing Levels

There are various testing levels based on the specificity of test.

- **Unit testing**: Unit testing refers to tests conducted on a section of code in order to verify the functionality of that piece of code. This is done at the function level.

- **Integration Testing**: Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Its primary purpose is to expose the defects associated with the interfacing of modules.

- **System Testing**: System testing tests a completely integrated system to verify that the system meets its requirements.

- **Acceptance testing**: Acceptance testing tests the readiness of application, satisfying all requirements.

- **Performance testing:** Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or device such as response time or millions ofinstructions per second etc.

System Test Cases

A test case is a set of test data, preconditions, expected results and post conditions, developed for a test scenario to verify compliance against a specific requirement. We have designed and executed a few test cases to check if the project meets the functional requirements.

Code Implementation

Project Code- Transmitter End

```
#include <SoftwareSerial.h>

SoftwareSerial ZIGBEE(2, 3); //TX, RX respetively

#include <LiquidCrystal.h>

LiquidCrystal lcd(13,12,11,10,9,8);


String readvoice;


void setup() {

ZIGBEE.begin(9600);

Serial.begin(9600);


lcd.begin(16,2);

lcd.clear();
```

```
 lcd.setCursor(0,0);
 lcd.print("***WELCOME***");
 ZIGBEE.println("***WELCOME***");
 Serial.println("***WELCOME***");
 delay(3000);
}

void loop() {

  while (Serial.available()){  //Check if there is an available byte to read
  delay(10); //Delay added to make thing stable
 // readvoice +=Serial.read();
  char c = Serial.read(); //Conduct a serial read
  readvoice += c; //build the string- "forward", "reverse", "left" and "right"
         }

 while (ZIGBEE.available()){  //Check if there is an available byte to read
delay(10); //Delay added to make thing stable
 // readvoice +=ZIGBEE.read();
char c = ZIGBEE.read(); //Conduct a serial read
  readvoice += c; //build the string- "forward", "reverse", "left" and "right"
         }
   /* lcd.clear();
   lcd.setCursor(0, 0);
```

```
lcd.print(readvoice);

Serial.println(readvoice); delay(4000);

ZIGBEE.print(readvoice);

delay(4000);*/


if(readvoice.length() > 0) {


 lcd.clear();

 lcd.setCursor(0,1);

 lcd.print(readvoice);

 Serial.println(readvoice); delay(1000);

 ZIGBEE.print(readvoice);

 delay(1000);


readvoice="";

        }


      } //Reset the variable
```

Project Code- Receiver End

```
#include <SoftwareSerial.h>
SoftwareSerial ZIGBEE(2, 3); //TX, RX respetively
#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);
```

```
String readvoice;

 void setup() {
 ZIGBEE.begin(9600);
 Serial.begin(9600);

  lcd.begin(16,2);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("***WELCOME***");
  ZIGBEE.println("***WELCOME***");
  Serial.println("***WELCOME***");
  delay(3000);

}

void loop() {

  if (ZIGBEE.available()){ //Check if there is an available byte to read
  delay(10); //Delay added to make thing stable
  char c = ZIGBEE.read(); //Conduct a serial read
  readvoice += c; //build the string- "forward", "reverse", "left" and "right"
          }



if (Serial.available()){ //Check if there is an available byte to read
delay(10); //Delay added to make thing stable
char c = Serial.read(); //Conduct a serial read
readvoice += c; //build the string- "forward", "reverse", "left" and "right"
          }

   /* lcd.clear();
 lcd.setCursor(0, 0);
```

```
  lcd.print(readvoice);
  Serial.println(readvoice); delay(4000);
  ZIGBEE.print(readvoice);
  delay(4000);*/

   if(readvoice.length() > 0) {


   lcd.clear();
   lcd.setCursor(0,1);
   lcd.print(readvoice);
   Serial.println(readvoice); delay(1000);
   ZIGBEE.print(readvoice);
   delay(1000);



 readvoice="";
}

} //Reset the variable
```

CHAPTER-6

RESULT ANALYSIS

Case 1: LCD displaying at Transmitter Side

Figure.5.1.1: lcd displaying the message that is getting transmitted

- Here we can see the lcd displaying the message that we are going to transmit i.e., 8 bytes of information.
- The maximum possibility of transmission of information is 32 bytes of information. And we will be getting the same image at the receiver side respectively.

Case 2: LCD Displaying at Receiver Side

Figure.5.2.1 : lcd displaying the message that is getting received

- Here we can see the lcd displaying the message that we received i.e., 8 bytes of information.
- The maximum possibility of receiving of information is 32 bytes of information. And we can see the information is the same as the transmitted information.

Case 3: Transmission and Receiving in X-CTU Software

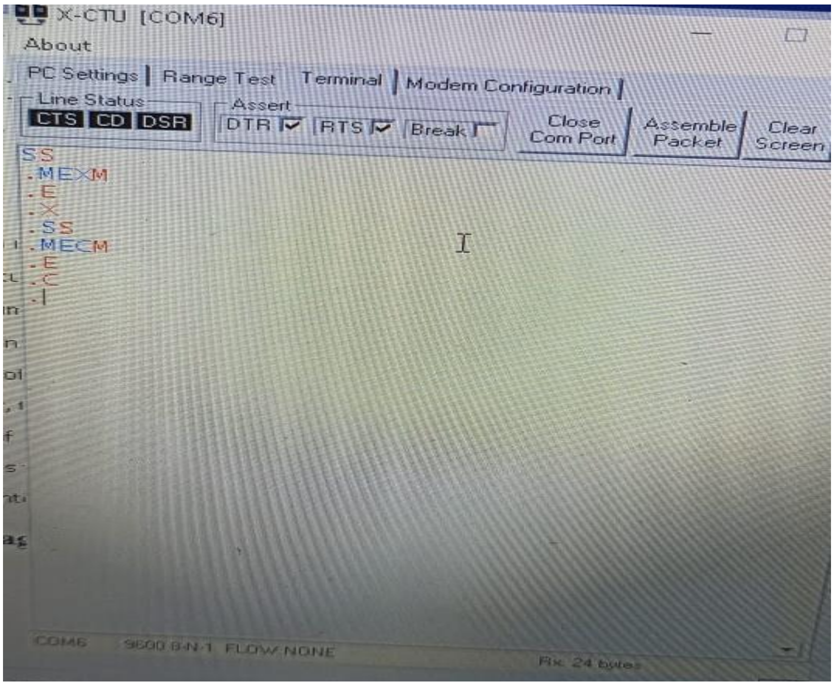


Figure.5.3.1: Transmission and Receiving the Information in X-CTU Software.

- In our project we use X-CTU software for transmission and receiving from a system to system.
- Here blue coloured information is the transmitted information and the orange-coloured information is the received information.
- since were using X-CTU there is no limit for the message size for transmission or receiving.

Case 4: Transmission and Receiving from System to System

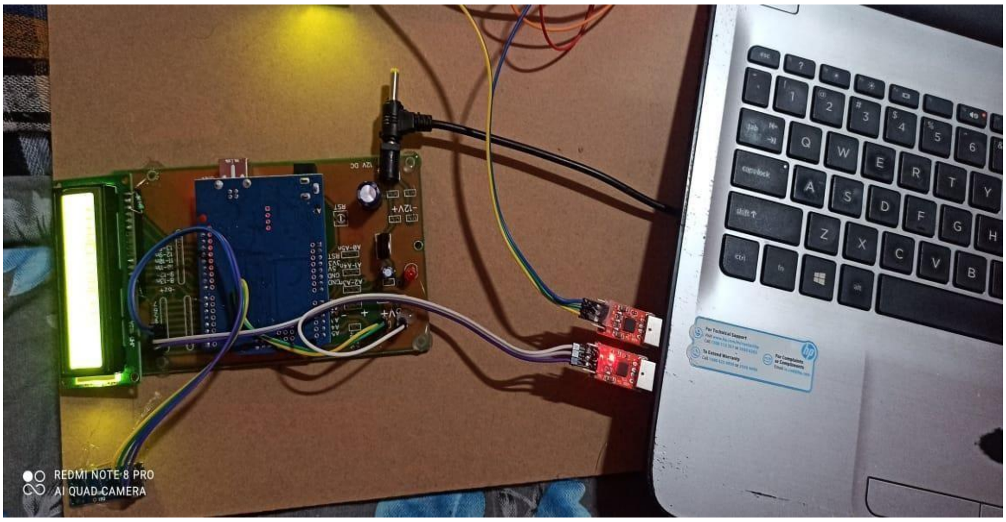


Figure.5.4.1: transmitting and receiving Zigbee’s connected to systems.

- Here we can see the clear connections of our project i.e., the ZigBee is connected to the Arduino along with the lcd.
- Now the Arduino is connected to the system using USB-TTL.
- And now the transmission and receiving of information happens from a system to system through a Zigbee using X-CTU software.

CHAPTER-7

LIMITATIONS AND ADVANTAGES

Limitations

- It is so highly risky to be used for official private information.
- The zigbee has low transmission rate.
- Replacement with zigbee compliant appliances can be costly.
- It does not have many end devices available yet.
- It cannot be used as outdoor wireless communication system dur to it has short coverage limited.
- It is not secure like wifi based secured system.

Advantages

- The Zigbee has flexible network structure.
- It has a very long battery life.
- It is low power consumption.
- It is easy to install.
- It can be easily implemented.
- It supports large number of nodes i.e., 6500 nodes approximately.
- It has a very low cost.
- It is more reliable and self-healing.
- Setting up the network is very simple and easy.

CHAPTER-8

CONCLUSION AND FUTURE WORK

Conclusion

- We can conclude this project by telling that is a ZigBee based wireless communication in order to transmit and receive the message in a secure manner by pairing the devices so that only authorized devices can have the access to the message.
- Zigbee has its own benefits i.e., the communication can happen to long distances when compared to WIFI module and Bluetooth module.
- Also, once you pair the device, the connection is established for longer duration i.e., we don't have to pair the device again and again.

Future Work

- Zigbee has a promising future in front of it. By rapid rise in home networking Zigbee would provide revolutionizing statistics in the upcoming years.
- And we don't need to connect it on a daily basis cause a ZigBee's connection lasts for a lot of time up to a couple of months to be specific.
- Various types of areas such as defense, national security, monitoring and control etc. can be facilitated by devices based on ZigBee standard. Some of its future applications can be:
- Wireless home security.
- Call button for elderly and disabled.
- Wireless smoke detectors.

Industrial automation:

- In manufacturing and production industries a communication link continuously monitors various parameters and critical equipment.

Home automation:

- Lighting system control and appliance control, heating and cooling system control in our home can be controlled using a ZigBee because we have a hub in it and the hub acts as the heart of the appliance automation.

REFERENCES

[1] Li Chunqing, Zhang Jiancheng, “Research of ZigBee's data security and protection”, International Forum on Computer Science-Technology and Applications 2009, IEEE, 2009, pp 298 - 302.

[2] ZigBee Alliance , “ZigBee and wireless radio frequency coexistence”, white paper, published by ZigBee Alliance, June 2007, Retrieved 2011, from http://www.zigbee.org/imwp/idms/popups/pop_download.asp?contentID=11745

[3] Patrick Kinney, “ZigBee technology: wireless control that simply works”, Communications design conference, vol 2, 2 October 2003. retrieved 2011, from http://www.zigbee.org/en/press_kits/latest/documents/white_papers/wp_zigbeetechwireless_final.pdf

[4] Bin Yang, " Study on security of wireless sensor network based on ZigBee standard." International Conference on Computational Intelligence and Security, IEEE, 2009, pp 426 - 430.

[5] Doug Whiting, Russ Housley and Niels Ferguso, “Counter with CBC-MAC (CCM) AES mode of operation”, submitted to NIST, retrieved 2011, from http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/ccm/ccm.pdf

[6] Meng Qianqian and Bao Kejin, “Security analysis for wireless networks based on ZigBee”, IEEE, vol 1, 2009, pp 158 - 160.

[7] Arvinderpal S. Wander, Nils Gura, Hans Eberle, Vipul Gupta, Sheueling Chang Shantz, “Energy analysis of public-key cryptography for wireless sensor networks”, Third IEEE International conference on Pervasive Compting and Communications, IEEE Computer Society, 2005, pp 324-328.

[8] Roy Pereira, “ZigBee and ECC secure wireless networks”, August 09 2004, retrieved 2011, from http://electronicdesign.com/article/embedded-software/zigbee-and-ecc-secure-wireless-networks8369.aspx.

[9] Jerry Krasne, "Using elliptic curve cryptography (ECC) for enhanced embedded security financial advantages of ECC over RSA or Diffie Hellman", Embedded Market Forecasters American Technology International, Inc., November 2004, retrieved 2011, from http://embeddedforecast.com/EMF-ECC-FINAL1204.pdf

[10] ZigBee Alliance, "The enduring value of symmetric encryption", white paper, August 2000, retrieved 2011 from http://www.partnerdata.it/files/WP-Symmetric%20Encryption.pdf.

[11] Zhen-rong Li, Yi-qi Zhuang, Chao Zhang and Gang Jin, "Low-power and area-optimized VLSI implementation of AES coprocessor for ZigBee system", The Journal of China Universities of Posts and Telecommunications (Elsevier), vol 16, Issue 3, June 2009, pp 89-94.

[12] Hyunjue Kim, Chang Hyun Kim and Jong-Moon Chung, "A novel elliptical curve id cryptography protocol for multi-hop ZigBee sensor networks", Wireless Communication and Mobile Computing Wirel. Commun. Mob. Comput. (2010), Wiley Interscience, JohnWiley & Sons, Ltd, 2010.

[13] Tadiwa Elisha Nyamasvisva, Halabi Hasbullah, "Multi-level security algorithm for random ZigBee wireless sensor networks", Information Technology (ITSim), 2010 International Symposium in Kuala Lumpur, IEEE, vol 2, 2010, pp 612 - 617.

[14] Jelena Misic, Vojislav Misic, "Wireless personal area networks performance interconnections and security with IEEE 802.15.4., John Wiley & Sons Ltd, 2008.

[15] NFIPS, "197: Announcing the advanced encryption standard (AES)", Information Technology Laboratory, National Institute of Standards and Technology, November 2001.

[16] S.A Vanston, "Next generation security for wireless: elliptic curve cryptography", Computers and Security, Elsevier, vol 22, no. 5, 2003, pp 412 – 415

[17] Bo Fan "Analysis on the Security Architecture of ZigBee Based on IEEE 802.15.4'', 2017 IEEE 13th International Symposium on Autonomous Decentralized System (ISADS), 22-24 March 2017.

[18] Jing Sun and Xiaofen Zhang, "Study of ZigBee Wireless Mesh Networks'', 2009 Ninth International Conference on Hybrid Intelligent Systems, 12-14 Aug. 2009.

[19] Sun, J., Zhang, X.: Study of ZigBee Wireless Mesh Networks. In: International Conference on Hybrid Intelligent Systems, pp. 264–267. IEEE Computer Society, Los Alamitos (2009) Google Scholar.

[20] IEEE Computer Society, IEEE Std 802.15.4-2006 (2006) Google Scholar.

[21] Gielata, A., Russek, P., Wiatr, K.: AES hardware implementation in FPGA for algorithm acceleration purpose. In: Proceedings of the International Conference on Signals and Electronic Systems, ICSES 2008, pp. 137–140 (2008) Google Scholar.

[22] Park, S.J.: Analysis of AES Hardware Implementations. Department of Electrical and Computer Engineering, Oregon State University (2003) Google Scholar.

[23] Kaps, J.P.: Cryptography for Ultra-Low Power Devices. A Dissertation Submitted to the Faculty of the Worcester Polytechnic Institute (2006) Google Scholar.

[24] AES Lounge, http://www.iaik.tugraz.at/content/research/krypto/AES/IAIK-TUGraz.

[25] Milu Hanna Mathew, Dr.T. Parithimar Kalaignan, "A Literature Survey On Zigbee", International Journal Of Current Engineering And Scientific Research (IJCESR), ISSN (print): 2393-8374, (online): 2394-0697, volume-5, issue-2, 2018.

A

PROJECT REPORT

On

ALCOHOL SENSING DISPLAY WITH ALARM

Submitted by

1)Ms. L Harshasri (18K85A0419)

2)Ms. G Srilekha Reddy (17K81A0477)

3)Mr. K Venkateswar Reddy (18K85A0420)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Mr. Santhosh Singireddy

Assistant professor

DEPARTMENT OF ECE

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

St. MARTINS ENGINEERING COLLEGE

(Affiliated to JNTUH, Hyderabad)

An UGC Autonomous Institute

Dhullapally, Secunderabad-500 100
NBA & NAAC A+ Accredited

Electronics and Communication Engineering

CERTIFICATE

This is to certify that the project entitled **Alcohol sensing display with alarm** is being submitted by**1. Ms. L Harshasri (18K85A0419), 2. Ms. G Srilekha Reddy (17K81A0477) 3. Mr. K Venkateswar Reddy (18K85A0420)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The results embodied in this report have been verified and found satisfactory.

	Head of the Department
Mr. Santhosh Singireddy	Mr. B Hari Krishna
Department of ECE	Department of ECE

Internal Examiner	External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **G. SRILEKA** WITH ROLL NO.**17K81A0477, K. VENKATESHWAR** WITH ROLL NO.**18K85A0420**, **L. HARSHA SRI** WITH ROLL NO.**18K85A0419**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**ALCOHOL SENSING DISPLAY WITH ALARAM**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

DECLARATION

We, the student of **Bachelor of Technology** in Department of Electronics and Communication Engineering, session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled **Alcohol sensing display with alarm** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

L Harshasri (18K85A0419)
G Srilekha Reddy (17K81A0477)
K Venkateswar Reddy (18K85A0420)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P. Santosh Kumar Patra**, **our beloved Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to Dr. B Hari Krishna, **Head &Professor**, the Dept. of Electronics and communication engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept. project coordinator Dr. A. Anand kumar, the Dept. of ECE** for his advice, supervision and for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide, **Mr. Santhosh Singireddy, Assistant professor, Department of ECE** and our class project coordinator **Mr. K Karthik, Assistant professor, Department of ECE** for guidance and encouragement in carrying out this project work

Finally, we are highly grateful and obliged to our college level Project Coordinators **Mr. Santhosh Singireddy, Assistant Professor in ECE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. L Harshasri
2. G Srilekha Reddy
3. K Venkateshwar Reddy

ABSTRACT

This is an important IoT and sensor-based project implemented using microcontrollers. In this project, we have interfaced the MQ series gas sensor with 8051 microcontrollers. We cannot interface the alcohol sensor directly to the 8051 microcontrollers because the output of the gas sensor is in analog format. The 8051 microcontroller recognizes only digital input. The best and the low-cost solution for this is to connect a comparator between the gas sensor and Microcontroller. Comparator contains an operational amplifier comparator has two inputs. One input is from the sensor and another is from the Potentiometer. Whenever the sensor value crosses the threshold limit then the comparator output goes high. In this way the microcontroller comes to know that the alcohol percentage is more than the threshold limit. Potentiometer can be used to vary the trigger level or the threshold limit.

We have provided a piezoelectric buzzer in this project. This Buzzer will be turned on whenever this project detects the alcohol. This Buzzer helps to alert the people that the person undergoing the test for alcohol detection has consumed alcohol. This buzzer is driven through a transistor.

We have also provided an LCD display in this alcohol detector mini project. The LCD display shows two messages. Whenever you turn on the project, the LCD shows the project title. And whenever alcohol is detected, system shows a message as Alcohol sensor crossed limit.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

• AC	Alternating current
• LCD	Liquid crystal display
• DC	Direct current
• Op-amp	Operational amplifier
• BAC	Blood alcohol concentration
• GSM	Global system for mobile communication
• IoT	Internet of things
• DUI	Driving under influence
• PCB	Printed circuit board
• CDS	Computerized documentation service
• ISIS	Integrated set of information systems
• LED	Light emitting diode
• IDE	Integrated development environment

CHAPTER 1
INTRODUCTION

Introduction of the project

Drunk driving is a very dangerous behaviour because excessive consumption of alcohol causes distortion in the thought pattern of drivers. The investigation conducted by the World Health Organization in 2008 shows that about 50%–60% of traffic accidents are related to drunk driving. In present times, the cases of traffic accidents caused by drunk driving have increased rapidly. It has, therefore, become evident that drunk driving does great harm to public security. Different technologies and techniques have been adopted to reduce the incidence of road accidents due to drunk driving by motorists. The alcohol detector is made up of the alcohol sensor, Alternating Current (AC) power supply, LM 393 Operational Amplifier (Op-Amp), and Liquid Crystal Display (LCD) circuitry. This device displays the results of the alcohol sensor as it senses the alcohol molecules in air present around it, and displays when alcohol is detected when it crosses the fixed threshold set by the LM393 Op Amp. The drawback of this system was that it required whosoever to be tested to be close to the AC power outlet due to the system running on AC power. Also, the LM393 op amp which was acting as a comparator in the circuit and came with a preset value 600 for the threshold upon crossing, had no response such as an alarm to warn that the threshold had been crossed. This system required two participants i.e., one person to carry out the testing and note when the threshold is crossed and the other as the person being tested. This system was nowhere close to proving a means of inhibiting a driver if he/she were drunk, not to mention real time implementation.

James and John proposed an alcohol detection system that alerts the driver through his/her cell phone. The major components of this system were the GSM module and the LM358 module. This system was a huge advancement from breath analysers as it was based on GSM technology using the GSM module and dumped the use of an alarm circuit but still employed the LM358 Op-Amp. The system alerted via text messages using a GSM module and had a unique ringtone for such text messages set on the cell phone.

Its major demerit was the lack of an LCD unit and an alarm circuit which could be useful in cases where the driver is not in possession of his/her phone. Another drawback is the presence of the LM358 Op-Amp as a comparator instead of using a microcontroller to allow for flexibility in changing the blood alcohol concentration (BAC) threshold due to probability in changes of body chemistry of the driver. The issue of cell phone batteries running down also comes up implying that the system would be inactive in the state that a cell phone battery is dead. Also, with most drivers in the habit of keeping their cell phones in the vibration or silent mode while driving, this inhibited the alerting property of the work.

Another alcohol detection system was developed based on the PIC16F877A microcontroller. The presence of the microcontroller allowed for ease of manipulation of the threshold depending on body chemistry. The presence of the microcontroller gave room for addition of other features in the future. The only major drawback was the system's inability for a direct real time implementation due to it being powered by an AC power supply, as the alcohol sensor wouldn't have the opportunity to have at least 3 h full run-in time it would get if on DC supply (vehicle battery) to give the sensor the degree of accuracy it requires for its operation. Implementation of our proposed embedded alcohol trigger device enclosed in cars that there might be no accident caused by drunk driving.

The aim of this project is to reduce road accidents related to drunk driving to the barest minimum by using Internet of Things (IoT) technology. With the help of this system, drivers under the influence of alcohol can be detected, monitored, and tracked by relevant law enforcement agencies in the smart city. The Internet of Things (IoT) is the most recent communication display in which the objects of regular day to day existence will be outfitted with microcontrollers, transceivers for digital communication, and appropriate convention stacks that will make them ready to speak with each other and with the clients, turning into a fundamental piece of the Internet.

This system is a very innovative system which will help to keep the roads free from drunk driving related accidents by not only shutting down the vehicle engine upon sensing the drunken state of the driver but would also help road traffic officials to be able to track down the vehicle inhabiting the drunk driver before the vehicle is shut down and allow for quick evacuation of the vehicle and the driver by the authorities so as to prevent traffic on such road.

The system is divided into sections; the interlock section and the monitoring section. The interlock section is made up of the MQ-3 Sensor (alcohol sensor) which senses the alcohol molecules in the air breathed by the driver, an ATMEGA328P microcontroller, a buzzer, an LCD screen, Wireless Fidelity (Wi-Fi) module and mobile cell phone. The monitoring system is a web application built to view the BAC concentration levels of the driver. The system is powered by a 12-V DC source. The system has a threshold value, 600, for communicating the BAC level and coordinates of the vehicle to the monitoring system.

We here propose an alcohol sensing system that measures alcohol intake, displays percentage of alcohol and also sounds an alarm if it is above a particular threshold. Here we use an alcohol sensor circuit along with an LCD display and a buzzer alarm. Our system first uses the alcohol sensor in order to detect alcohol. The sensor provides analog output.

This analog output is now provided to the microcontroller for further processing. Based on the input the microcontroller calculates the percentage of alcohol and displays the same on an LCD display. It also sounds an alarm if the amount of alcohol exceeds a particular amount. Our system thus allows measuring the amount of alcohol and then displaying the percentage of alcohol measured. Also, an alarm is sounded that indicates that measured alcohol is above a particular percentage.

Outcomes of the project

The outcomes of the project are it will measure concentration of alcohol gas taken by the driver when he blows out, when the sensor brought near to him and detects whether the level of alcohol taken was more normal and display it on the lcd.

Gives the result to the mobile user who is connected with the WIFI module’s hotspot and displays the result detected by the WIFI module through a software application called mobile telnet application.

Description of the project

This project is about monitoring the consumption of alcohol by using the technology of the internet of things.

Here this project contains a gas sensor that is used to measure the level of concentration of alcohol that is taken by the automobile driver.

The WIFI module is responsible for transmitting continuous measured results to the online server.

Introduction to embedded systems

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as a combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems include computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer. The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes. The applications software on such processors is sometimes referred to as firmware.

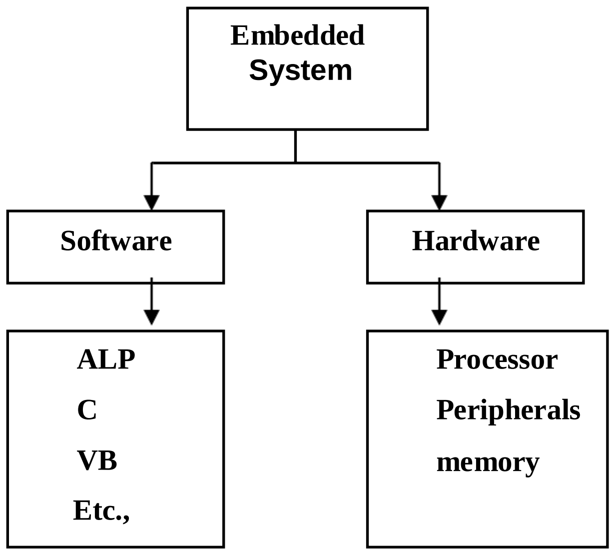


Figure 1.1 Block diagram of Embedded System

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

Memory: It is used to store data or address.

Peripherals: These are the external devices connected

Processor: It is an IC which is used to perform some task

Applications of embedded systems

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

Processors are classified into four types like:

- ➢ Microprocessor (µp)
- ➢ Micro controller (µc)
- ➢ Digital Signal Processor (DSP)
- ➢ Application Specific Integrated Circuits (ASIC)

Micro Processor (µp):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Three basic characteristics differentiate microprocessors:

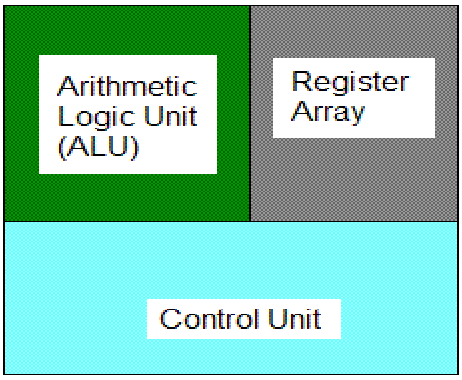
➢ **Instruction set**: The set of instructions that the microprocessor can execute.

➢ **Bandwidth :** The number of bits processed in a single instruction.

➢ **Clock speed :** Given in megahertz (MHz), the clock speed determines how many instructions per second the processor can execute.

In both cases, the higher the value, the more powerful the CPU. For example, a 32-bit microprocessor that runs at 50MHz is more powerful than a 16-bit microprocessor that runs at 25MHz. In addition to bandwidth and clock speed, microprocessors are classified as either RISC (reduced instruction set computer) or CISC (complex instruction set computer).

A microprocessor has three basic elements, as shown above. The ALU performs all arithmetic computations, such as addition, subtraction and logic operations (AND, OR, etc). It is controlled by the Control Unit and receives its data from the Register Array. The Register Array is a set of registers used for storing data. These registers can be accessed by the ALU very quickly. Some registers have specific functions - we will deal with these later. The Control Unit controls the entire process. It provides the timing and a control signal for getting data into and out of the registers and the ALU and it synchronizes the execution of instructions (we will deal with instruction execution at a later date).



Three Basic Elements of a Microprocessor

Micro Controller (μc):

A microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.

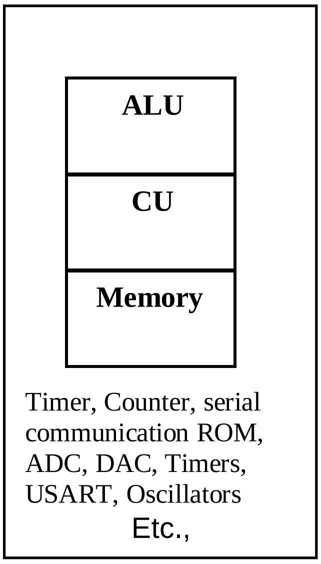


Figure 1.2 Block Diagram of Microcontroller (μc)

Digital Signal Processors (DSPs):

Digital Signal Processors is one which performs scientific and mathematical operations. Digital Signal Processor chips - specialized microprocessors with architectures designed specifically for the types of operations required in digital signal processing. Like a general-purpose microprocessor, a DSP is a programmable device, with its own native instruction code. DSP chips are capable of carrying out millions of floating-point operations per second, and like their better-known general-purpose cousins, faster and more powerful versions are continually being introduced. DSPs can also be embedded within complex "system-on-chip" devices, often containing both analog and digital circuitry.

Application Specific Integrated Circuit (ASIC)

ASIC is a combination of digital and analog circuits packed into an IC to achieve the desired control/computation function

ASIC typically contains

- CPU cores for computation and control
- Peripherals to control timing critical functions
- Memories to store data and program
- Analog circuits to provide clocks and interface to the real world which is analog in nature
- I/Os to connect to external components like LEDs, memories, monitors etc.

Computer Instruction Set

There are two different types of computer instruction set there are:

1. RISC (Reduced Instruction Set Computer) and
2. CISC (Complex Instruction Set computer)

Reduced Instruction Set Computer (RISC)

A RISC (reduced instruction set computer) is a microprocessor that is designed to perform a smaller number of types of computer instruction so that it can operate at a higher speed (perform more million instructions per second, or millions of instructions per second). Since each instruction type that a computer must perform requires additional transistors and circuitry, a larger list or set of computer instructions tends to make the microprocessor more complicated and slower in operation.

Besides performance improvement, some advantages of RISC and related design improvements are:

- A new microprocessor can be developed and tested more quickly if one of its aims is to be less complicated.
- Operating system and application programmers who use the microprocessor's instructions will find it easier to develop code with a smaller instruction set.
- The simplicity of RISC allows more freedom to choose how to use the space on a microprocessor.

Higher-level language compilers produce more efficient code than formerly because they have always tended to use the smaller set of instructions to be found in a RISC computer.

RISC characteristics

- **Simple instruction set:**

In a RISC machine, the instruction set contains simple, basic instructions, from which more complex instructions can be composed.

- **Same length instructions.**

Each instruction is the same length, so that it may be fetched in a single operation.

- **1machine-cycleinstructions.**

Most instructions complete in one machine cycle, which allows the processor to handle several instructions at the same time. This pipelining is a key technique used to speed up RISC machines.

Complex Instruction Set Computer (CISC)

CISC, which stands for **Complex Instruction Set Computer,** is a philosophy for designing chips that are easy to program and which make efficient use of memory. Each instruction in a CISC instruction set might perform a series of operations inside the processor. This reduces the number of instructions required to implement a given program, and allows the programmer to learn a small but flexible set of instructions.

The advantages of CISC

At the time of their initial development, CISC machines used available technologies to optimize computer performance.

- Microprogramming is as easy as assembly language to implement, and much less expensive than hardwiring a control unit.
- The ease of micro-coding new instructions allowed designers to make CISC machines upwardly compatible: a new computer could run the same programs as earlier computers because the new computer would contain a superset of the instructions of the earlier computers.
- As each instruction became more capable, fewer instructions could be used to implement a given task. This made more efficient use of the relatively slow main memory.
- Because microprogram instruction sets can be written to match the constructs of high-level languages, the compiler does not have to be as complicated.

The disadvantages of CISC

Still, designers soon realized that the CISC philosophy had its own problems, including:

- Earlier generations of a processor family generally were contained as a subset in every new version --- so instruction set & chip hardware become more complex with each generation of computers.
- So that as many instructions as possible could be stored in memory with the least possible wasted space, individual instructions could be of almost any length---this means that different instructions will take different amounts of clock time to execute, slowing down the overall performance of the machine.

➢ Many specialized instructions aren't used frequently enough to justify their existence --- approximately 20% of the available instructions are used in a typical program.

➢ CISC instructions typically set the condition codes as a side effect of the instruction. Not only does setting the condition codes take time, but programmers have to remember to examine the condition code bits before a subsequent instruction changes them.

Memory Architecture

There two different type's memory architectures there are:

- Harvard Architecture
- Von-Neumann Architecture

Harvard Architecture

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

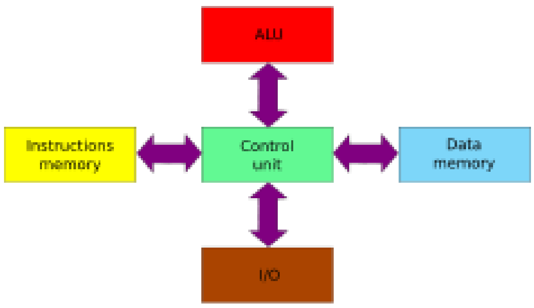


Figure 1.3 Harvard Architecture

Modern uses of the Harvard architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

- Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behaviour must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.
- Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction prefetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

Von-Neumann Architecture

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. The **von Neumann architecture** is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A **stored-program** digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were an advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

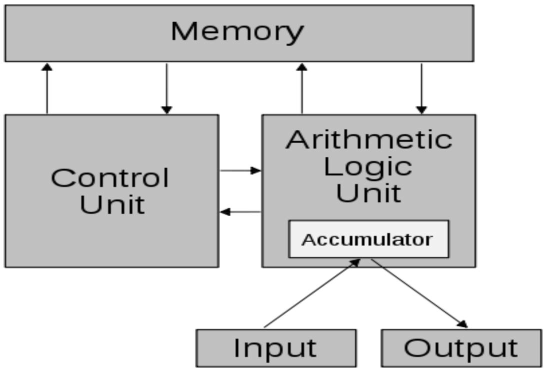


Figure 1.4 Schematic of the Von-Neumann Architecture.

Basic Difference between Harvard and Von-Neumann Architecture

- The primary difference between Harvard architecture and the Von Neumann architecture is in the Von Neumann architecture data and programs are stored in the same memory and managed by the same information handling system.
- Whereas the Harvard architecture stores data and programs in separate memory devices and they are handled by different subsystems.
- In a computer using the Von-Neumann architecture without cache; the central processing unit (CPU) can either be reading an instruction or writing/reading data to/from the memory. Both of these operations cannot occur simultaneously as the data and instructions use the same system bus.
- In a computer using the Harvard architecture the CPU can both read an instruction and access data memory at the same time without cache. This means that a computer with Harvard architecture can potentially be faster for a given circuit

complexity because data access and instruction fetches do not contend for use of a single memory pathway.

➢ Today, the vast majority of computers are designed and built using the Von Neumann architecture template primarily because of the dynamic capabilities and efficiencies gained in designing, implementing, operating one memory system as opposed to two. Von Neumann architecture may be somewhat slower than the contrasting Harvard Architecture for certain specific tasks, but it is much more flexible and allows for many concepts unavailable to Harvard architecture such as self-programming, word processing and so on.

IoT Technology

The Internet of things (IoT) describes the network of physical objects—a.k.a. "Things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet of Things have evolved due to the convergence of multiple technologies, Realtime analytics, machine learning, ubiquitous computing, commodity sensors, and embedded systems. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", including devices and appliances (such as lighting fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers. The IoT can also be used in healthcare systems.

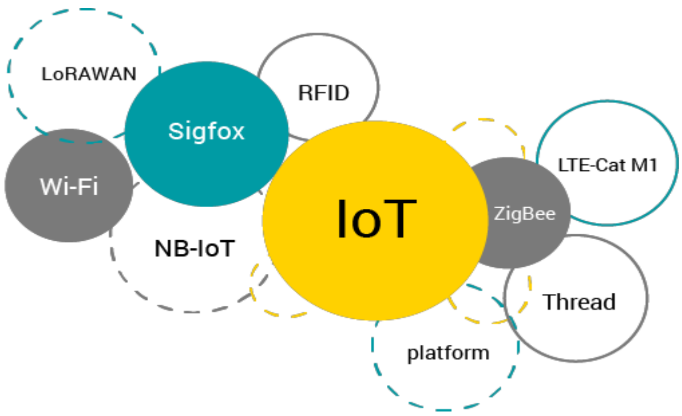


Fig 1.5: IoT Technology

There are a number of serious concerns about dangers in the growth of the IoT, especially in the areas of privacy and security, and consequently industry and governmental moves to address these concerns have begun including the development of international standards.

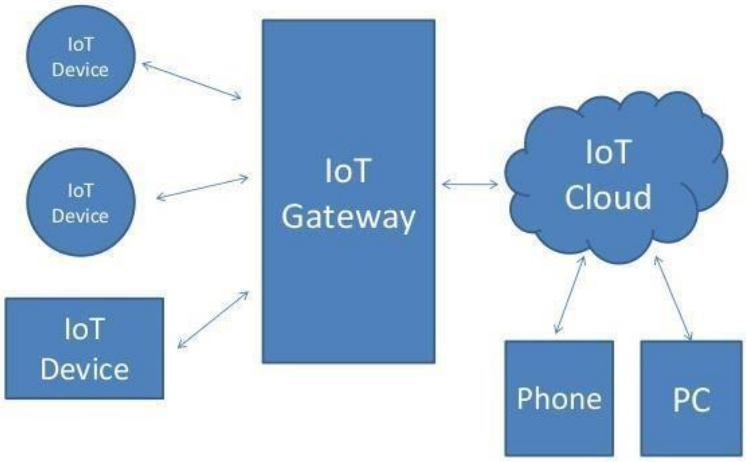


Fig 1.6: IoT Block diagram

History

The main concept of a network of smart devices was discussed as early as 1982, with a modified Coca-Cola vending machine at Carnegie Mellon University becoming the first ARPANET-connected appliance, able to report its inventory and whether newly loaded drinks were cold or not. Mark Weiser's 1991 paper on ubiquitous computing, "The Computer of the 21st Century", as well as academic venues such as UbiComp and PerCom produced the contemporary vision of the IOT. In 1994, Reza Raji described the concept in IEEE Spectrum as "[moving] small packets of data to a large set of nodes, so as to integrate and automate everything from home appliances to entire factories". Between 1993 and 1997, several companies proposed solutions like Microsoft's at Work or Novell's NEST. The field gained momentum when Bill Joy envisioned device-to-device communication as a part of his "Six Webs" framework, presented at the World Economic Forum at Davos in 1999. The concept of the "Internet of Things" and the term itself, first appeared in a speech by Peter T. Lewis, to the Congressional Black Caucus Foundation 15th Annual Legislative Weekend in Washington, D.C, published in September 1985[25]. According to Lewis, "The Internet of Things, or IoT, is the integration of people, processes and technology with connectable devices and sensors to enable remote monitoring, status, manipulation and evaluation of trends of such devices."

The term "Internet of things" was coined independently by Kevin Ashton of Procter & Gamble, later MIT's Auto-ID Centre, in 1999, though he prefers the phrase "Internet for things' '. At that point, he viewed radio frequency identification (RFID) as essential to the Internet of things, which would allow computers to manage all individual things. The main theme of the Internet of Things is to embed short range mobile transceivers in various gadgets and daily necessities to enable new forms of communication between people and things, and between things themselves. Defining the Internet of things as "simply the point in time when more 'things or objects were connected to the Internet than people", Cisco Systems estimated that the IoT was "born" between 2008 and 2009, with the things/people ratio growing from 0.08 in 2003 to 1.84 in 2010.

The IoT technology stack and its Connectivity Solutions

It can prove a hard task if you'd like to find your way through the IoT technological maze given the diversity and sheer numerousness of the technology solutions that surround it. However, for matters of simplicity, we could break down the IoT technology stack into four basic technology layers involved in making the Internet of Things work. These are the following:

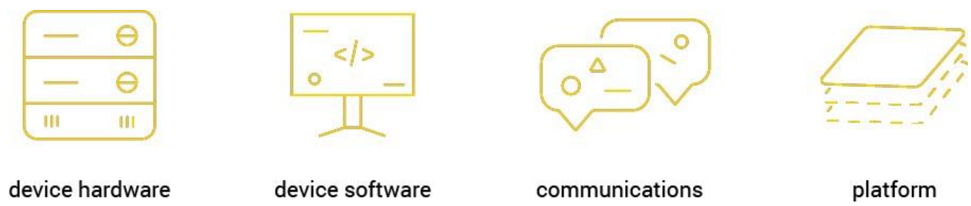


Fig 1.7: IoT technology stack

1. Device Hardware

Devices are objects which actually constitute the 'things' within the Internet of Things. Acting as an interface between the real and the digital worlds, they may take different sizes, shapes and levels of technological complexity depending on the task they are required to perform within the specific IoT deployment. Whether pinhead sized microphones or heavy construction machines, practically every material object (even the animate ones, like animals or humans) can be turned into a connected device by the addition of necessary instrumentation (by adding sensors or actuators along with the appropriate software) to measure and collect the necessary data. Obviously, sensors, actuators or other telemetry gear can also constitute standalone smart devices by themselves. The only limitation to be encountered here is the actual IoT use case and its hardware requirements (size, ease of deployment and management, reliability, useful lifetime, cost-effectiveness)

2. Device Software

This is what actually makes the connected devices 'smart'. Software is responsible for implementing the communication with the Cloud, collecting data, integrating devices as well as performing real-time data analysis within the IoT network. What is more, it is device software that also caters for application-level capabilities for users to visualize data and interact with the IoT system.

3. Communications

Having the device hardware and software in place, there must be another layer which will provide the smart objects with ways and means of exchanging information with the rest of the IoT world. While it is true that communications mechanisms are strongly tied to device hardware and software, it is vital to consider them as a separate layer. Communication layer includes both physical connectivity solutions (cellular, satellite, LAN) and specific protocols used in varying IoT environments (ZigBee, Thread, Z-Wave, MQTT, LwM2M). Choosing the relevant communications solution is one of the vital parts in constructing every IoT technology stack. The technology chosen will determine not only the ways in which data is sent to/received from the Cloud, but also how the devices are managed and how they communicate with third party devices. For the purpose of the present article, we will go into the details of some of the present-day communications solutions later in the text.

4. Platform

As mentioned earlier, thanks to the 'smart' hardware and the software installed the device is able to 'sense' what is going on around it and communicate that to the user via a specific communications channel. An IoT platform is the place where all of this data is gathered, managed, processed, analysed and presented in a user-friendly way. Thus, what makes such a solution especially valuable is not merely its data collection and IoT device management capabilities, but rather its ability to analyse and find useful insights from the portions of data provided by the devices via the communications layer. Again, there are quite a number of IoT platforms on the market, with choice depending on the requirements of the specific IoT project and such factors as architecture and IoT technology stack, reliability, customization properties, protocols used, hardware agnosticism, security and cost-effectiveness. It is also worth mentioning that platforms can be either installed on-premise or cloud-based.

Coiote IoT Device Management platform is a good example of such a platform as it can be deployed on-site as well as in the cloud. The same applies to another IoT platform by AV System — Coiote IoT Data Orchestration.

Connectivity solutions

As many as there are possible real-life applications of the IoT technologies, there is no shortage of connectivity solutions behind them. Depending on the specifications of a given IoT use case, each communications option may offer different service enablement scenarios while having trade-offs between power consumption, range and bandwidth. With this multiplicity and diversity of communication standards and protocols in mind, one may raise a question about the actual need for developing new solutions while there are some well-proven Internet protocols that have been in use already for decades. The reason for this is that existing Internet protocols, such as Transmission Control Protocol Internet Protocol (TCP/IP), are often not effective enough and too power-consuming to be able to work efficiently within the emerging IoT technology applications. This section will present a short overview of the major alternative Internet protocols specially dedicated for use by IoT systems. The overview concerns the most popular IoT radio technologies broken down by radio-frequency range achieved by each of the solutions: short range IoT radio solutions, medium range solutions, and long-range Wide Area Networks solutions.

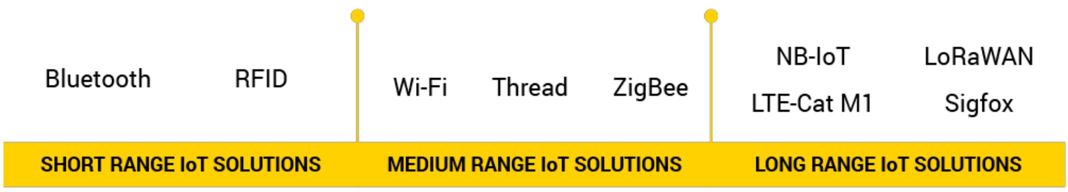


Fig 1.8: Connectivity Solutions

Short range IoT radio solutions

1. Bluetooth

As a well-established short-range connectivity technology, Bluetooth is considered to be the key solution particularly for the future of the wearable electronics market such as wireless headphones or geolocation sensors, especially given its widespread integration with smartphones. Designed with cost-effectiveness and reduced power consumption in mind, the Bluetooth Low-Energy (BLE) protocol requires very little power from the device. Yet, this comes with a compromise: when transferring frequently higher amounts of data, BLE may not be the most effective solution.

2. RFID

Being among the first IoT applications ever implemented, Radio-frequency identification (RFID) offers positioning solutions for IoT applications, especially in supply chain management and logistics, which require the ability of determining the object position inside buildings. The future of RFID technology clearly goes far beyond the simple localisation services, with possible applications ranging from tracking hospital patients to improving efficiency in healthcare to providing real-time merchandise location data to minimize out-of-stock situations for retail stores.

Medium Range Solutions

1. Wi-Fi

Developed based on IEEE 802.11, it remains the most widespread and generally known wireless communications protocol. Its broad usage across the IoT world is mainly limited by higher-than-average power consumption resulting from the need of retaining high signal strength and fast data transfer for better connectivity and reliability. As a key technology in the development of IoT, WiFi provides a wide-ranging ground to a staggering number of IoT solutions, yet it also needs to be managed and used in terms of marketing to yield profits to service providers and users alike. A fine example of a WiFi management platform that offers a value-added service empowering public WiFi access points is Linkify. As one of AVSystem's cutting edge solutions, Linkify allows for practically limitless guest WiFi customization and marketing options.

2.Zigbee

This popular wireless mesh networking standard finds its most frequent applications in traffic management systems, household electronics, and machine industry. Built on top of the IEEE 802.15.4 standard, Zigbee supports low data exchange rates, low power operation, security, and reliability.

Long Range Wide Area Networks (WAN) solutions:

1. NB-IOT

A product of existing 3GPP technologies, Narrowband IoT is a brand-new radio technology standard that ensures extremely low power consumption (10 years of battery power operation) and provides connectivity with signal strength approx. 23 dB lower than in the case of 2G. What is more, it uses existing network infrastructure, which ensures not only global coverage in LTE networks, but also guaranteed signal quality. In many cases, this fact allows for implementing NB-IoT instead of solutions that required the construction of local networks, such as LoRa or Sigfox.

2. Lora WAN

Lora WAN is a low-power Long Range Wide-Area Networking protocol optimized for low-power consumption and supporting large networks with millions of devices. Aiming at wide-area network (WAN) applications, Lora WAN is designed to furnish low-power WANs with features required to support low-cost, mobile and secure bi directional communication within IoT, M2M, smart city, and industrial applications.

3. Sigfox

The concept behind Sigfox is to provide an effective connectivity solution for low power M2M applications requiring low levels of data transfer for which the WiFi range is too short, and cellular range is too expensive and too power-hungry. Sigfox employs UNB, a technology that enables it to handle low data-transfer speeds of 10 to 1,000 bits per second. Consuming up to 100 times less energy compared to cellular communication solutions, it delivers a typical stand-by time of 20 years for a 2.5Ah battery. Offering a robust, energy-efficient and scalable network able to support communication between thousands of thousands of battery-operated devices across areas of several square kilometres, Sigfox proves suitable for various M2M applications, including smart street lighting, intelligent meters, patient monitors, security devices, and environmental sensors

4. LTE-Cat M1

LTE-Cat M1 is a low-power wide-area (LPWA) connectivity standard that connects IoT and M2M devices with medium data rate requirements. It supports longer battery lifecycles and offers enhanced in-building range as compared to cellular technologies such as 2G, 3G, or LTE-Cat 1. Being compatible with the existing LTE network, CAT M1 doesn’t require the carriers to build new infrastructure to implement it. As compared to NB-IoT, LTE Cat M1 proves to be perfect for mobile use cases, as its handling of hand-over between cell sites is significantly better and is very similar to high speed LTE.

IoT vs Embedded Systems

	IoT	Embedded System
Definition	Things that contain computer, software and networking capabilities’	Things that contain computers and software.
Example	A talking doll that accepts voice commands and regularly learns new words.	A talking doll that never learns but that isn’t a privacy concern because it transmits no data.

Table 1.1: IoT vs Embedded system

IoT Technology in Transportation

The IoT can assist in the integration of communications, control, and information processing across various transportation systems. Application of the IoT extends to all aspects of transportation systems (i.e. the vehicle, the infrastructure, and the driver or user). Dynamic interaction between these components of a transport system enables inter- and intra-vehicular communication, smart traffic control, smart parking, electronic toll collection systems, logistics and fleet management, vehicle control, safety, and road assistance.

Motivation of the Project

Vehicles are considered to be the key element in this era. They are being developed on a regular basis to make them more and more safe. But, the one thing that makes people feel afraid of sitting in the vehicle is accidents. These accidents mostly take place under the influence of alcohol, and there is no technology to minimize the chances of accidents happening. To avoid this problem, the Vehicle accident and alcohol sensing alert with engine locking system is introduced.

CHAPTER 2
LITERATURE SURVEY

Literature survey

In many civilized cultures, the use of alcohol is taken as a tradition. The habit is also connected with traditions, used in festivities and different personal parties. A small level of alcohol will change the way humans behave where its bodily behaviour, its actions are diminished. This type of body's inability to control itself can be highly dangerous and can involve car accidents which will risk the persons sitting inside the car also on the persons on the road. The legislation has brought in a number of laws like fines, cancellation of driving licenses etc. so that this can be minimized. The above-mentioned causes show the necessity of a simple, accurate and precise instrument to be used by the automobile manufacturers and vehicle modifiers so that the vehicle will not start due to alcohol content in the air inside the vehicle. The consumption of alcohol is more common in young groups where they drink and cause accidents due to rash driving. The person consuming alcohol changes the blood alcohol concentration in the body thus affects the body's actions. There is a direct connection between blood alcohol and breath alcohol concentration. For the blood alcohol content measurement blood samples have to be taken but for breath alcohol concentration measurement there are sensors available which detect breath. The first method of taking blood samples can be possible by taking on the spot samples by the traffic police which is also a good method. In the second method breath analyzers are used to sense the breath but this itself is not enough, this method can be integrated with the car system. so that any smell of alcohol in the car will force the car to start. The system proposed is developed on embedded applications on Arduino family of boards.

In the modern arena, most of the accidents are caused by drunken driving and driving under the influence (DUI). Drunk driving is the only one reason behind most of the unnatural deaths in the world. In this project, a Novel based IOT (Internet of Things) module is proposed to protect the people from unnecessary deaths caused by road accidents due to drunken driving. The Proposed system makes use of the Internet of things (IOT) device as Raspberry Pi 3 model B as a core. It mainly includes Touch sensor, alcohol concentration detection sensor, Facial recognition, Heart beat rate, to safeguard the drowsy driver. We use different types of safeguarding things such as GPS modules, Triggering an alarm and Automatic ignition off etc.

We here propose an alcohol sensing system that measures alcohol intake, displays percentage of alcohol and also sounds an alarm if it is above a particular threshold. Here we use an alcohol sensor circuit along with an LCD display and a buzzer alarm. Our system first uses the alcohol sensor in order to detect alcohol. The sensor provides analog output. This analog output is now provided to the microcontroller for further processing. Based on the input the microcontroller calculates the percentage of alcohol and displays the same on an LCD display. It also sounds an alarm if the amount of alcohol exceeds a particular amount. Our system thus allows us to measure the amount of alcohol and then display the percentage of alcohol measured. Also an alarm is sounded that indicates that measured alcohol is above a particular percentage.

Problem statement

Previously, there was no technology to lock the engine of the vehicle after sensing the alcohol consumption by the driver, which was considered to be the main cause of the accidents. There was manual checking after a particular distance on the roads or the highways but still these checks were not sufficient to stop the mishaps. So, to avoid these problems, this project vehicle detection and alcohol sensing alert with engine locking system is developed.

CHAPTER 3

PROJECT DESCRIPTION

Block Diagram

In the present day's scenario, with growing technologies and adverse development in the metropolitan cities. This is an important sensor-based project implemented using a microcontroller. In this project, we have interfaced the MQ series gas sensor with 8051 microcontrollers. We cannot interface the alcohol sensor directly to the 8051 microcontrollers because the output of the gas sensor is in analog format. The 8051 microcontroller recognizes only digital input. The best and the low-cost solution for this is to connect a comparator between the gas sensor and Microcontroller. Comparator contains an operational amplifier comparator has two inputs. One input is from the sensor and another is from the Potentiometer. Whenever the sensor value crosses the threshold limit then the comparator output goes high. In this way the microcontroller knows that the alcohol percentage is more than the threshold limit. Potentiometer can be used to vary the trigger level or the threshold limit.

We have provided a piezoelectric buzzer in this project. This Buzzer will be turned on whenever this project detects the alcohol. This Buzzer helps to alert the people that the person undergoing the test for alcohol detection has consumed alcohol. This buzzer is driven through a transistor.

We have also provided an LCD display in this alcohol detector mini project. The LCD display shows two messages. Whenever you turn on the project, the LCD shows the project title. And whenever alcohol is detected, the system shows a message as the Alcohol sensor crosses the limit.

Block Diagram

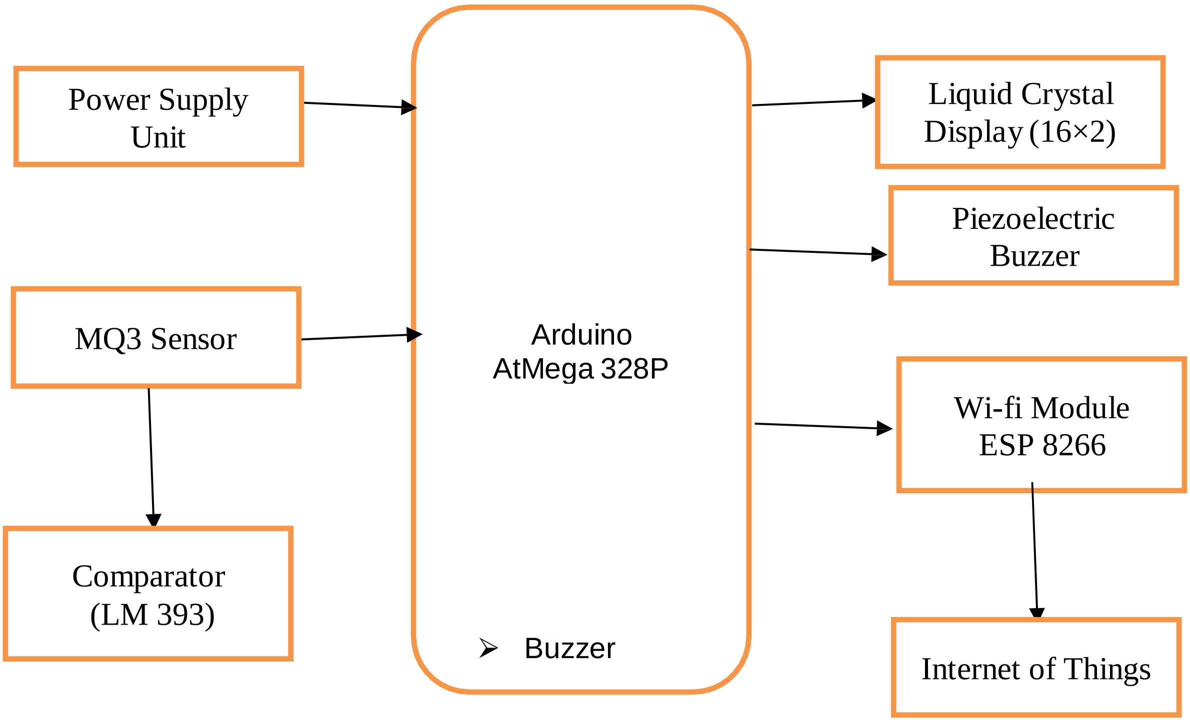


Fig 3.1 Block diagram

The blocks of the circuit are:

1.Arduino UNO

2.Power supply unit

3.MQ3 sensor

4.Comparator

5.Liquid crystal display

6.WIFI module

7.Piezoelectric buzzer

1. Arduino UNO

Arduino is an open-source platform based on easy-to-use hardware and software. Arduino Uno is a microcontroller board, which has an ATmega328P microcontroller in it. It has 14 digital i/o pins and 6 analog input pins, a 16 MHz quartz crystal, a USB connection, a power jack and a reset button. Simply connect to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. Embedded C program is used for coding.

2. Power Supply unit

Every electrical and electronic device that we use in our day-to-day life will require a power supply. In general, we use an AC supply of 230V 50Hz, but this power has to be changed into the required form with required values or voltage range for providing power supply to different types of devices.

3. MQ3 Sensor

MQ-3 module is suitable for detecting Alcohol, Benzine, CH_4, Hexane, LPG, CO. Sensitive material of the MQ-3 gas sensor is SnO_2, which has lower conductivity in clean air. When the target alcohol gas exists, the sensor's conductivity is higher along with the gas concentration rising. MQ-3 gas sensor has high sensitivity to Alcohol, and has good resistance to disturbance of gasoline, smoke and vapor.

This sensor provides an analog resistive output based on alcohol concentration. When the alcohol gas exists, the sensor's conductivity gets higher along with the gas concentration rising. There is a resistance across an A and B inside the sensor which varies on detection of alcohol. More the alcohol, the lower the resistance. The alcohol is measured by measuring this resistance. The sensor and load resistor form a voltage divider, and the lower the sensor resistance, the higher the voltage.

4. Comparator

The IC LM393 has two internally inbuilt operational amplifiers which are internally compensated with frequency. These ICs are specially designed for performing their different tasks using a single power supply. It can also execute its functions properly with a split power supply. The supply of current-drain does not rely on the amount of the power supply. One of the most important features of this IC is, it includes ground in its common-mode input voltage. The applications of this IC mainly include various fields in real life, and also industrial, ADC (analog to digital converters), electrical systems powered by the battery, time-delay generators limit comparators, etc. This article discusses an overview of the LM393 IC and its working.

5. Liquid Crystal Display

The LCD here is used to display the reading. When the sensor is placed near the Automobile driver then it displays either the driver drunk exceeds the input value or the normal value.

6. Wi-Fi Module

The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes preprogrammed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much Wi-Fi-ability.

7. Piezoelectric Buzzer

In simplest terms, a piezo buzzer is a type of electronic device that's used to produce a tone, alarm or sound. It's lightweight with a simple construction, and it's typically a low-cost product. Yet at the same time, depending on the piezo ceramic buzzer specifications, it's also reliable and can be constructed in a wide range of sizes that work across varying frequencies to produce different sound outputs.

Schematic Diagram

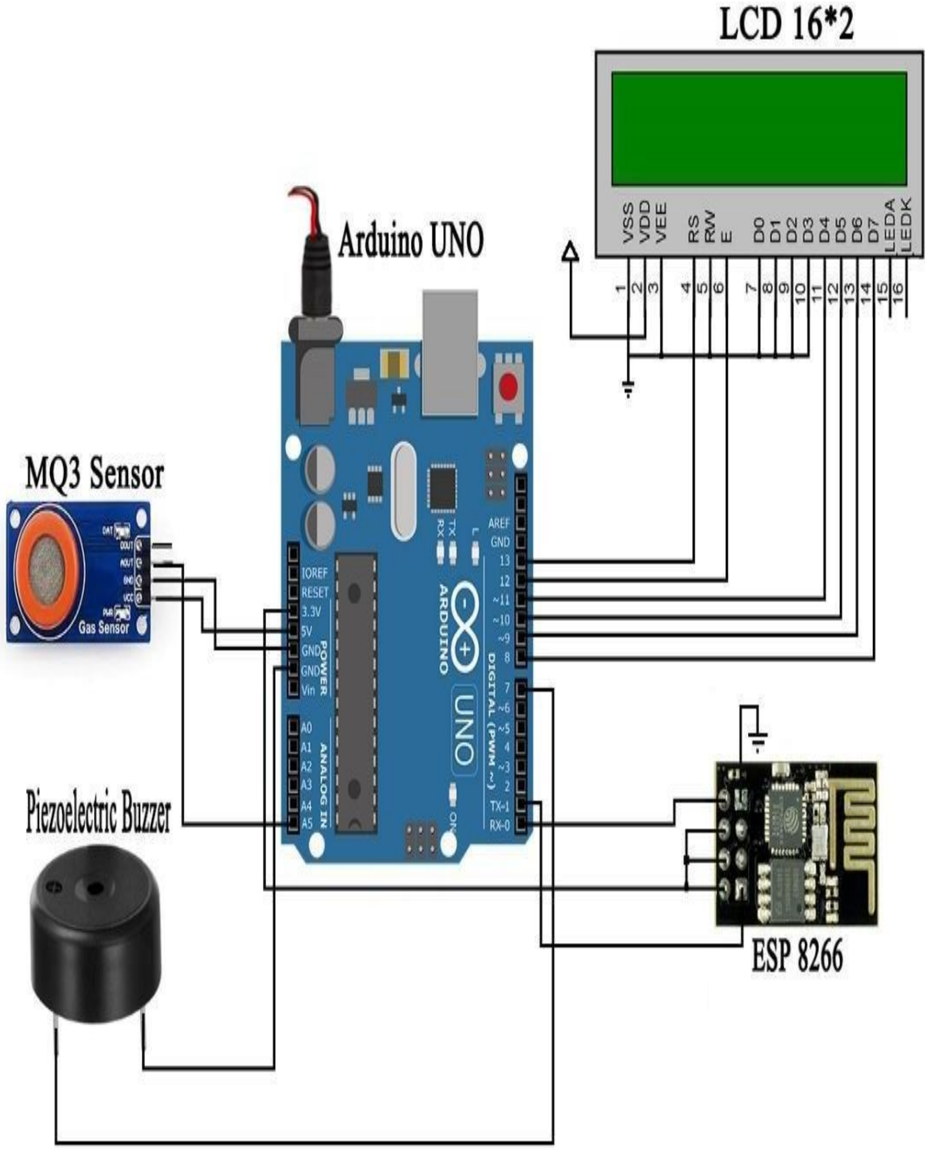


Fig 3.2: Schematic Diagram

Schematic Description

Arduino UNO has 28 pins among which we have 6 analog pins i.e.; A0-A5 to which we connect the input and 14 digital pins i.e.; D0-D15 where the output devices are connected. A MQ3 sensor which is popularly known as an alcohol sensor, is our input device with 4 pins i.e., VCC, D-Out, A-Out and Gnd pins respectively. The VCC pin is connected to the 5V pin of Arduino, Gnd pin to the common Gnd of Arduino Uno and the A-out pin is connected to one of the analog input pins(A0-A5) of Arduino. The D-Out pin of the Alcohol sensor is kept connectionless. The output of the alcohol sensor is given to an analog input pin of Arduino A5.We use two physical output devices i.e.; a piezoelectric buzzer, to give an alert as to whether the alcohol is detected or not. And the other device is a 16×2 Alphanumeric Liquid Crystal Display to show whether the alcohol consumption is detected or normal. We employed an LCD whose 4,6,11,12,13,14 pins are connected to 13,12,11,10,9,8 pins of Arduino Uno. We use a wi-fi module (ESP 8266) which consists of 8 pins, where the tx and rx pins are connected to rx, tx pins of Arduino and gnd pin to gnd, reset and VCC pins to 3.3V pin of Arduino. Internet of Things technology is used for a portable and connectionless interaction to the automobile driver and to the police, in order to generate a friendly interaction and display messages of alcohol content in the mobile, a software application is used through a WIFI module called mobile Telnet application making the user install the application.

CHAPTER 4

SOFTWARE AND HARDWARE DESCRIPTION

A- SOFTWARE

Proteus

Proteus is a simulation and design software tool developed by Lab center Electronics for Electrical and Electronic circuit design. It also possesses 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.

The designer can also develop 2D drawings for the product.

Features

ISIS has a wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semiconductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the footprints of different categories of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in ISIS can be directly transferred to ARES.

Starting New Design

Step 1: Open ISIS software and select new design in File menu

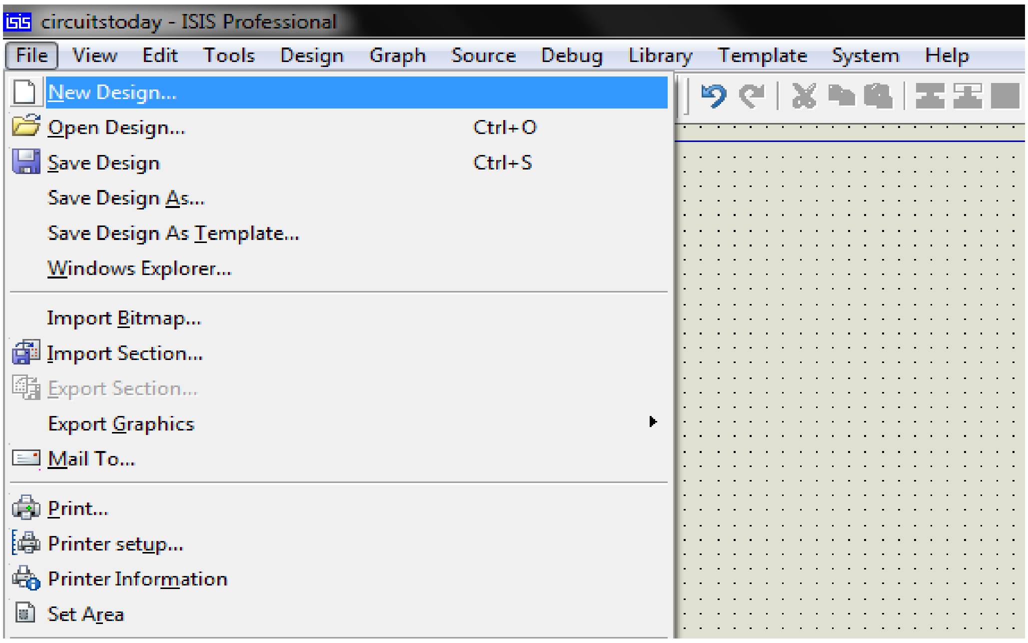


Figure 4.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

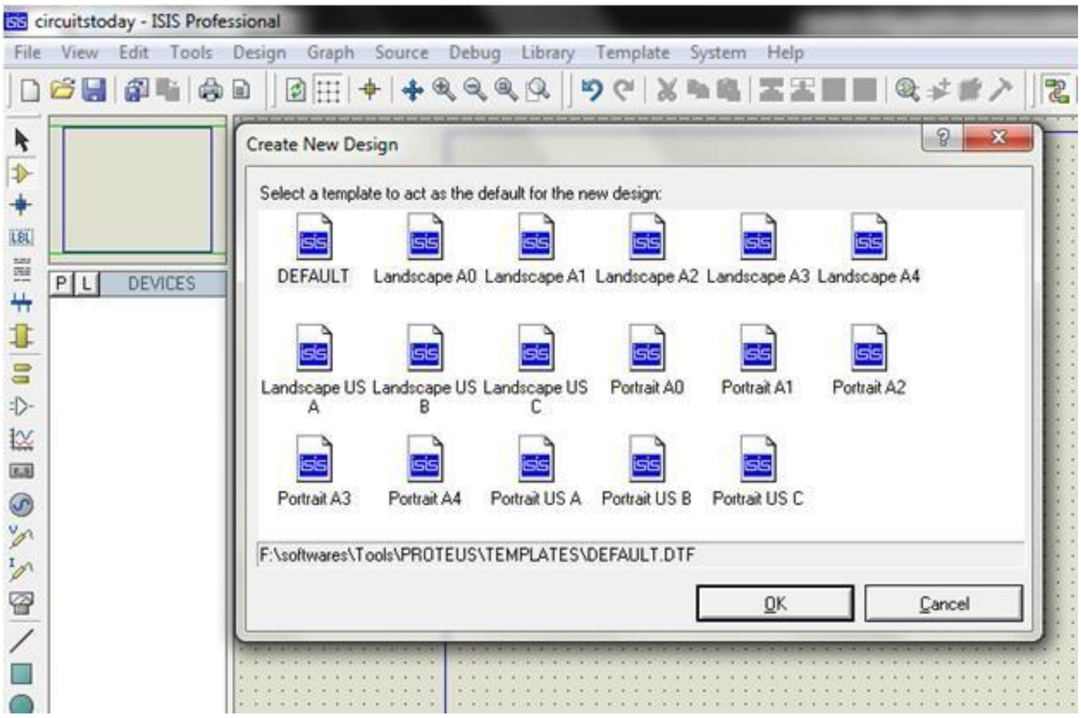


Figure 4.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

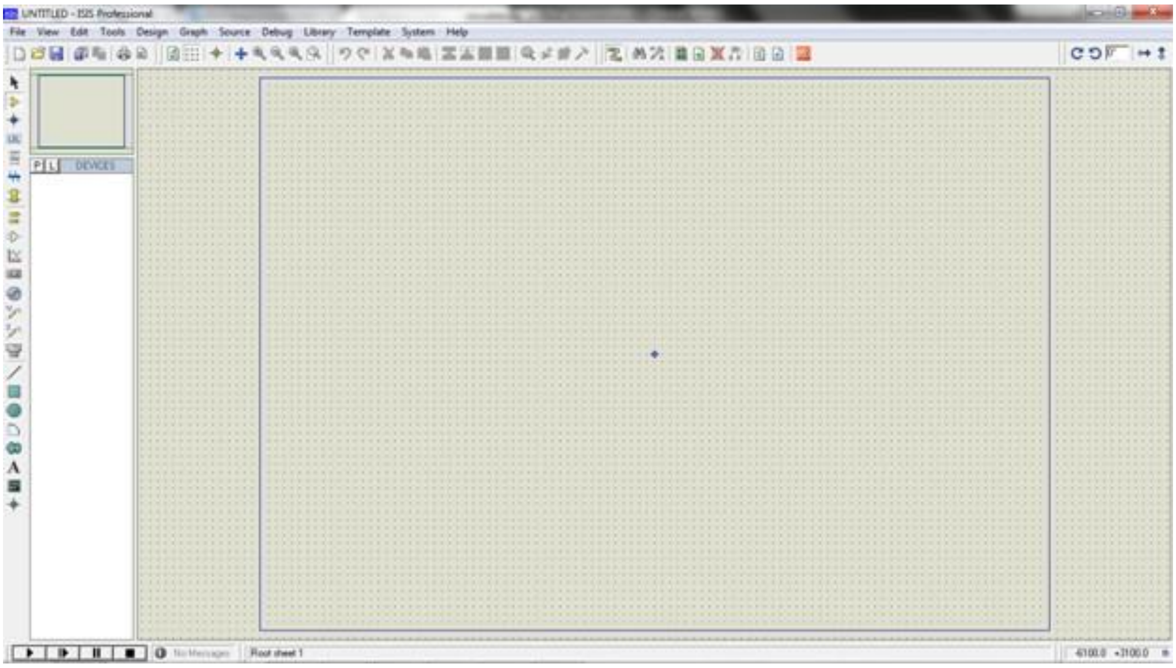


Figure 4.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

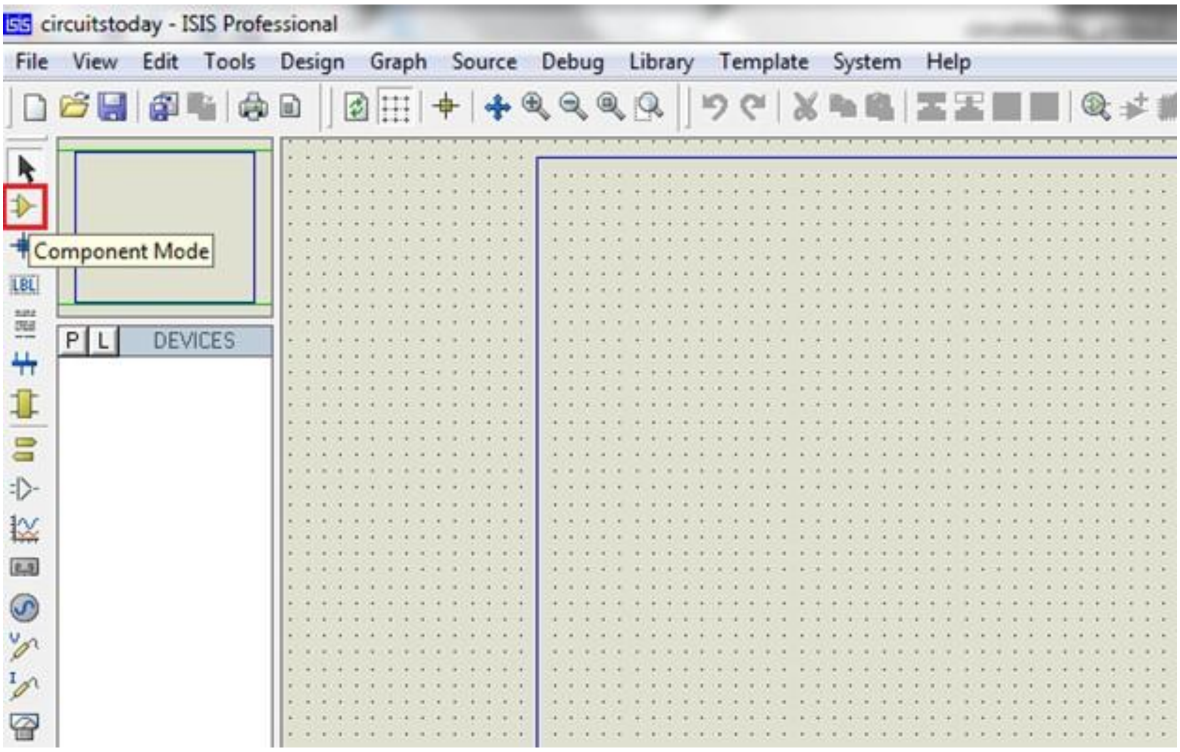


Figure 4.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

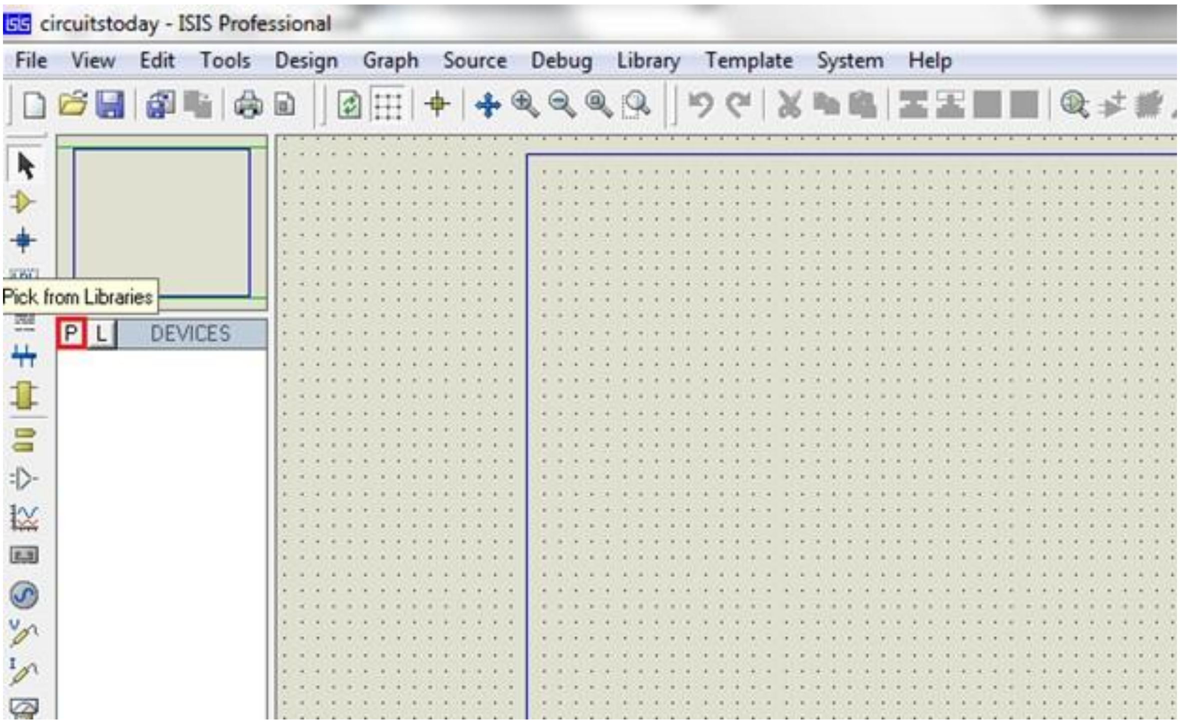


Figure 4.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keyword's text box.

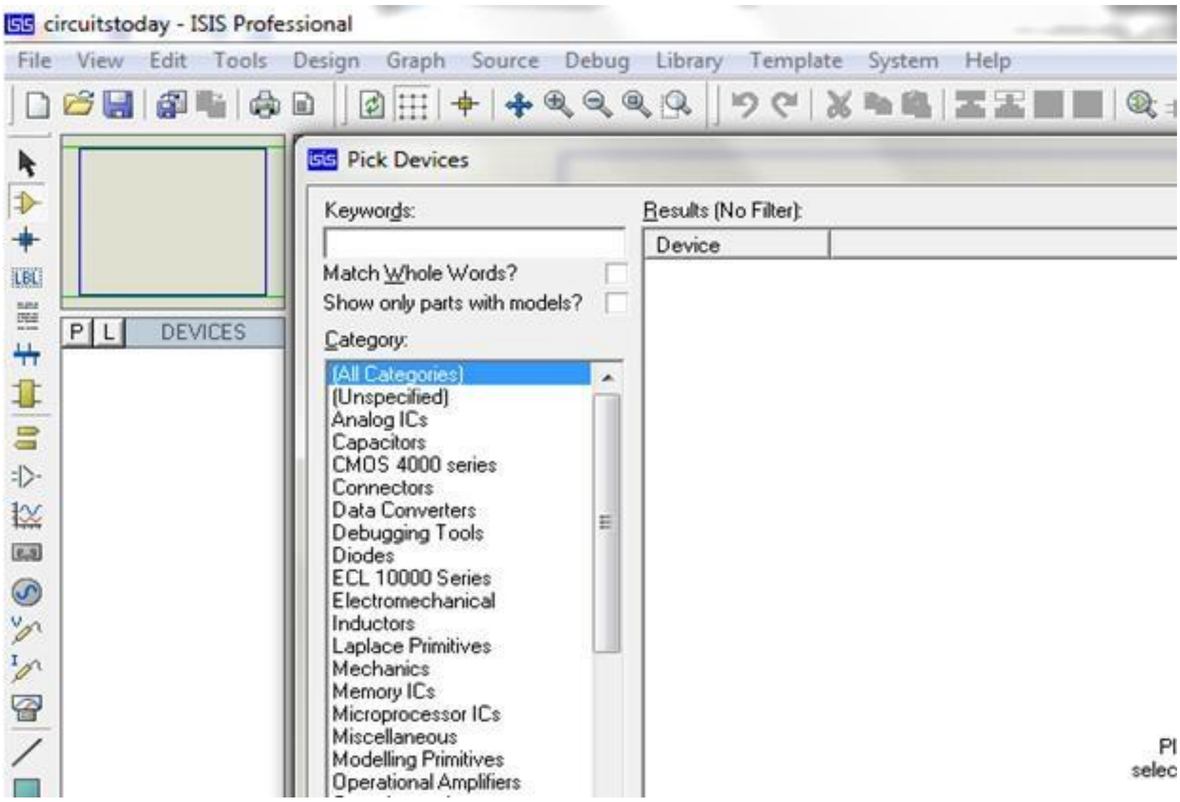


Figure 4.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

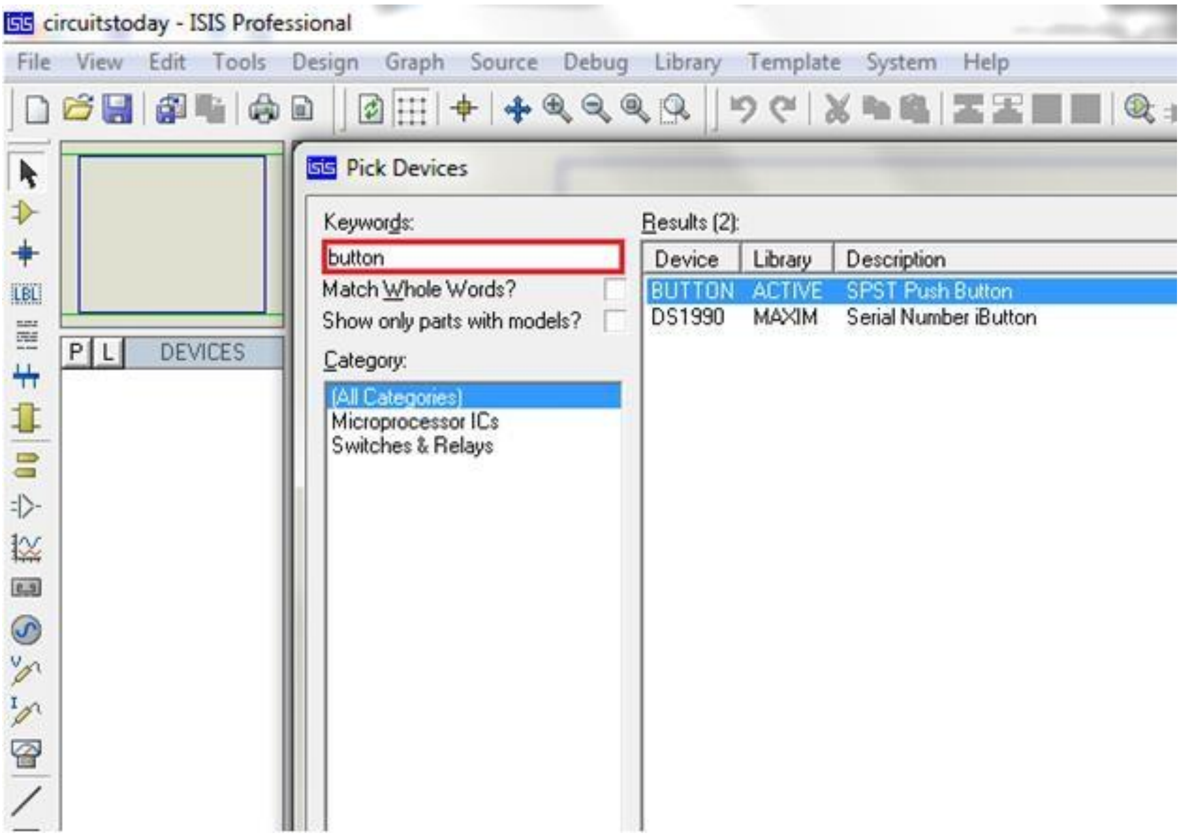


Figure 4.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

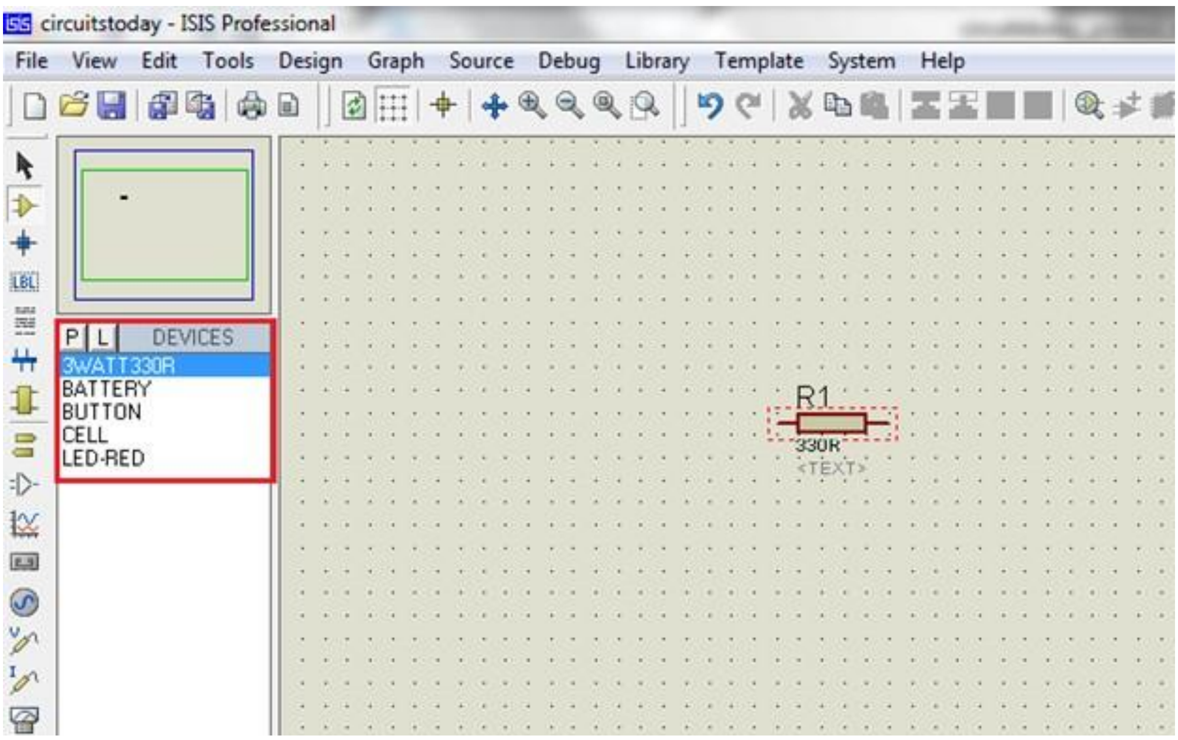


Figure 4.8 Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows it to connect through wires. Left click from one terminal to another to make a connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

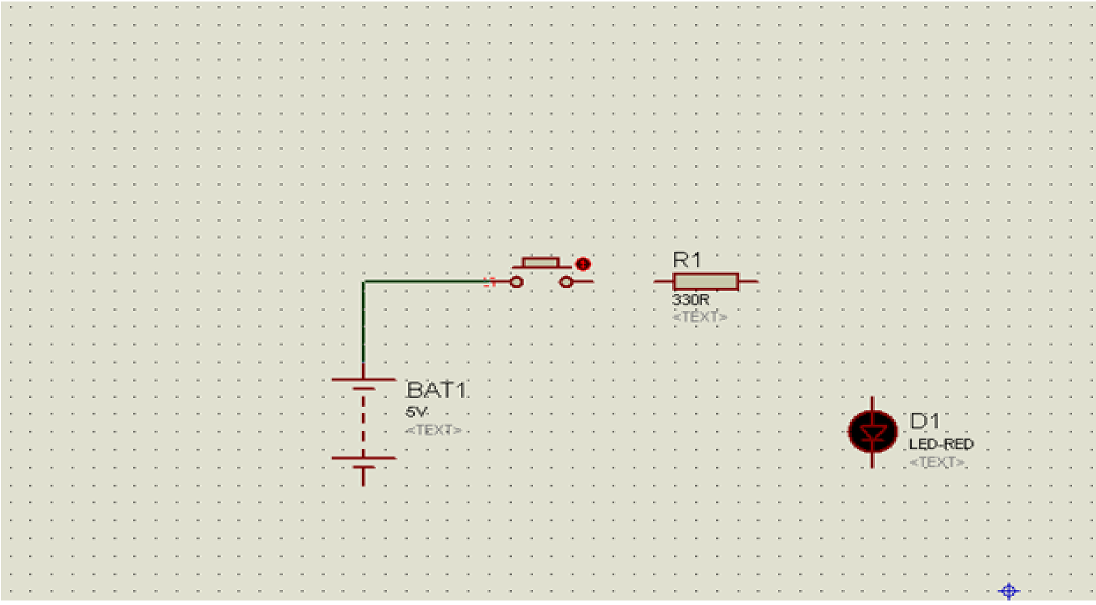


Figure 4.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

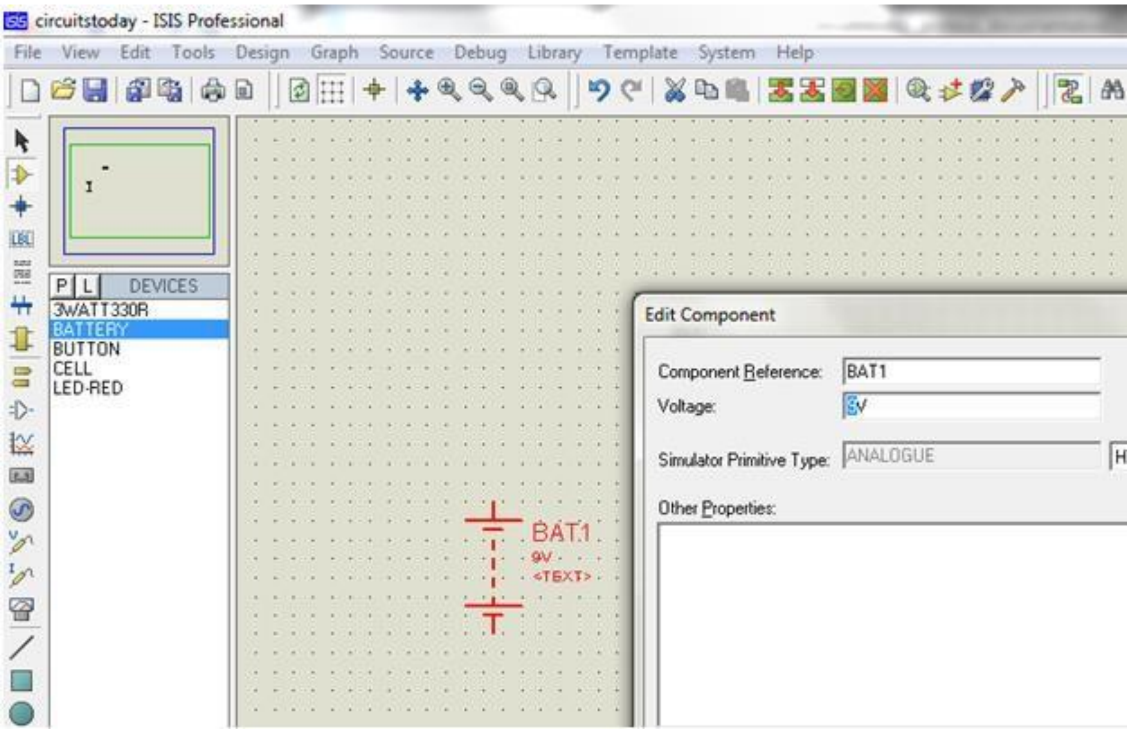


Figure 4.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

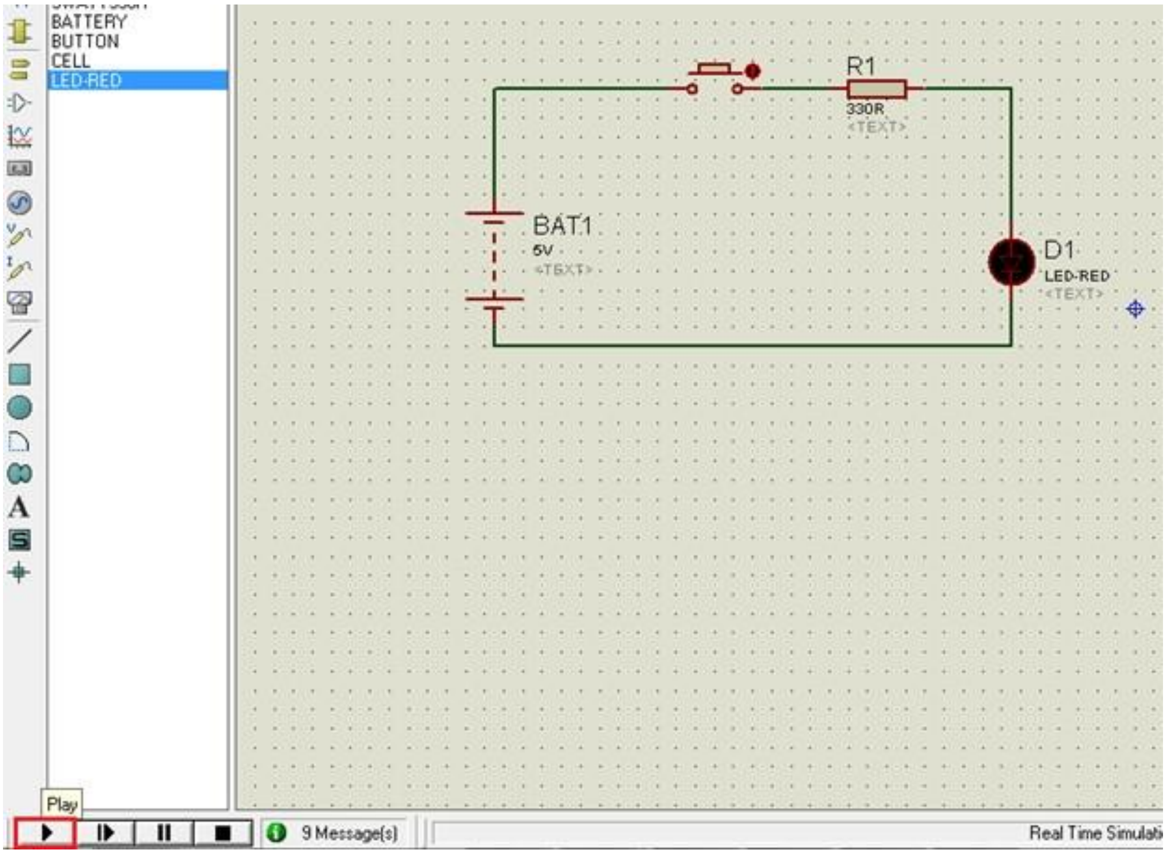


Figure 4.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make the LED glow.

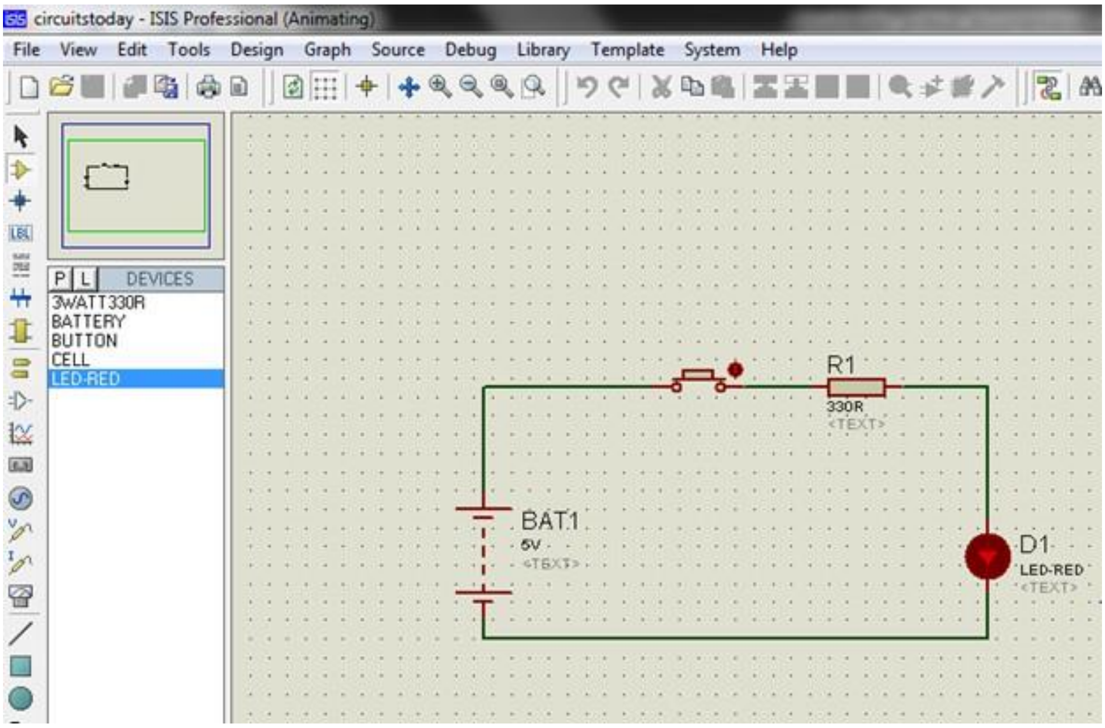


Figure 4.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

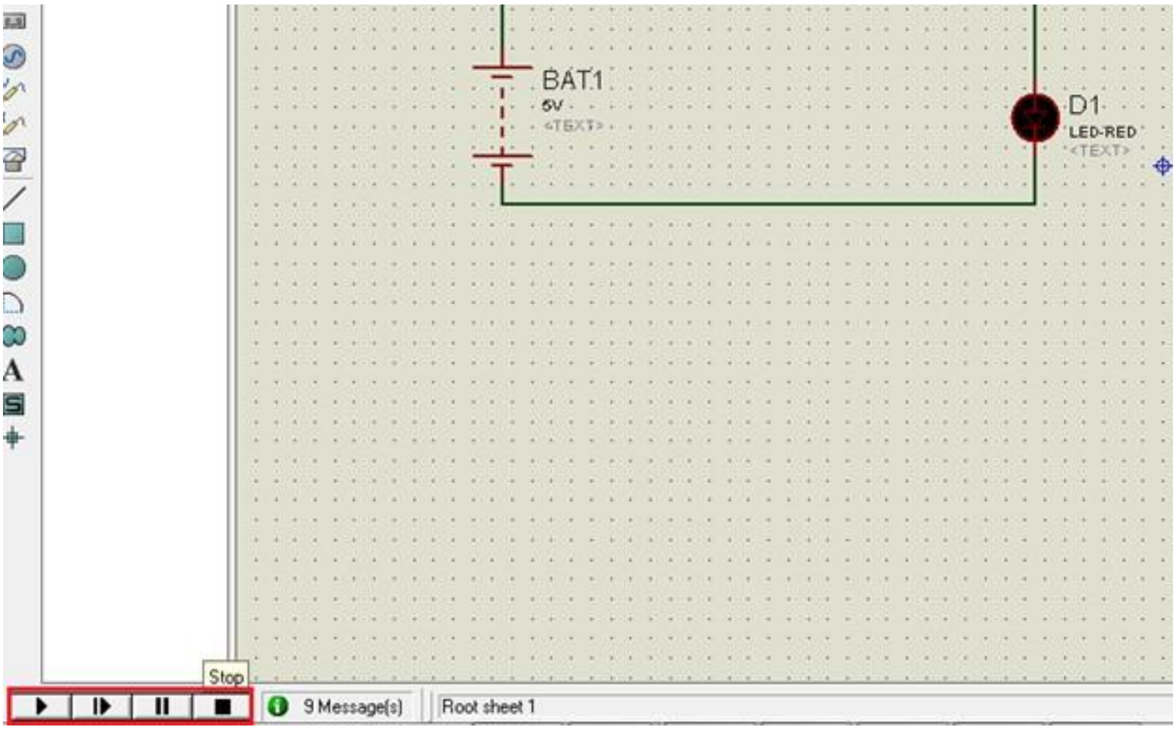


Figure 4.13 Simulation Step-Pause-Stop Buttons

Arduino IDE

The **Arduino Integrated Development Environment (IDE)** is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program argued to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Arduino Pro IDE

Developer(s)	Arduino Software
Preview release	v0.0.2 / 28 October 2019; 3 months ago
Repository	github.com/Arduino/Arduino
Written in	C, C++
Operating system	Windows, macOS, Linux
Platform	IA-32, x86-64, ARM
Type	Integrated development environment
License	LGPL or GPL license
Website	blog.arduino.cc/2019/10/18/arduino-pro-ide-alpha-preview-with-advanced-features/

Table 4.1 Arduino pro IDE

In October 2019 the Arduino organization began providing early access to a new Arduino Pro IDE with debugging[9] and other advanced features.

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

Installation

In this section, we will learn in easy steps how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Figure 4.14 USB cable (B plug)

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Figure 4.15 USB Cable (mini-B plug)

Step 2 – Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

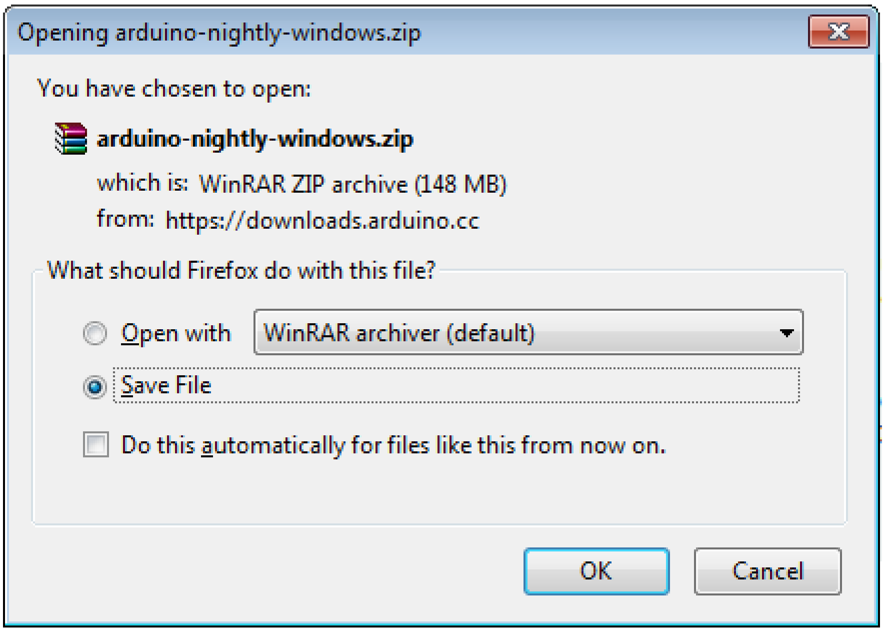


Figure 4.16 downloading Arduino IDE software

Step 3 – Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labelled PWR) should glow.

Step 4 – Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

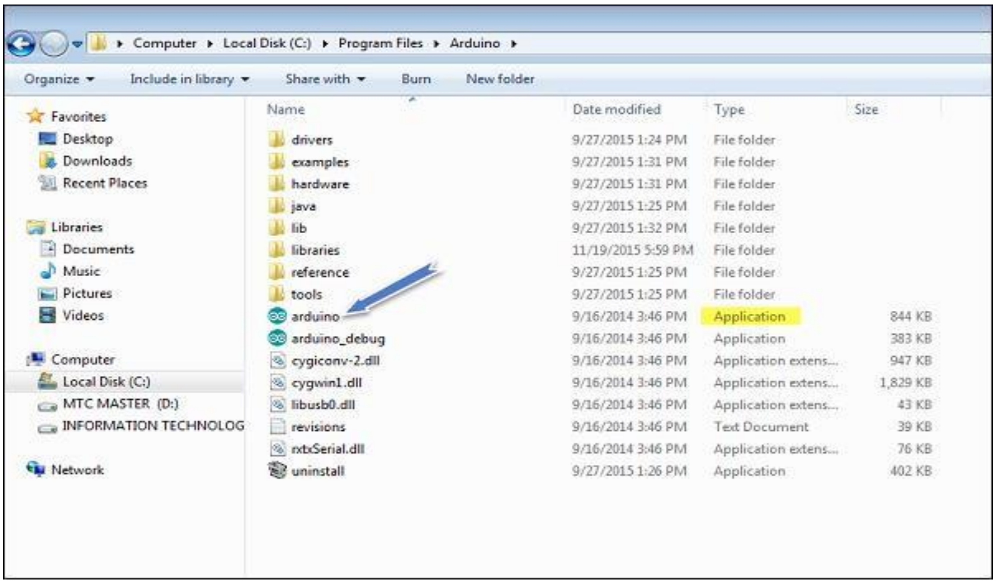


Figure 4.17 launching Arduino

Step 5 – Open your first project.

Once the software starts, you have two options –

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.

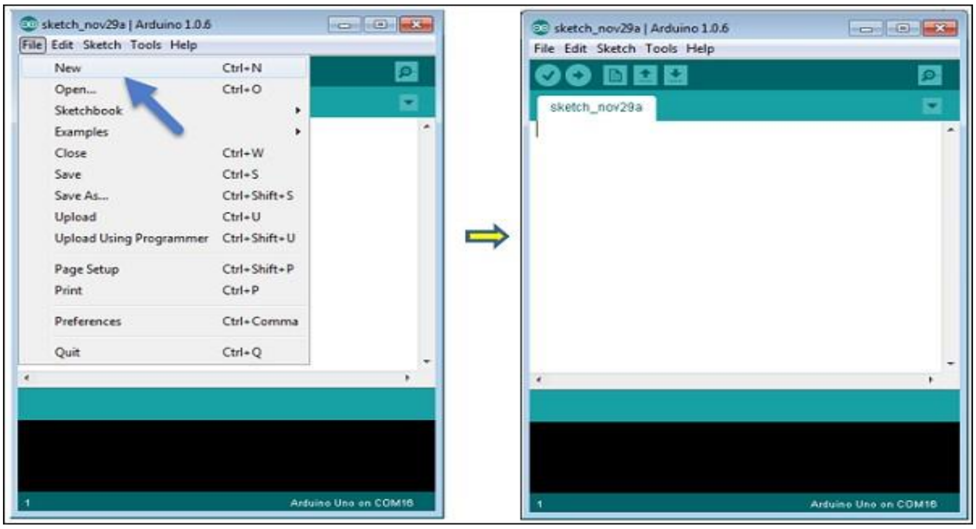


Figure 3.18(a) open new file

To open an existing project example, select File → Example → Basics → Blink.

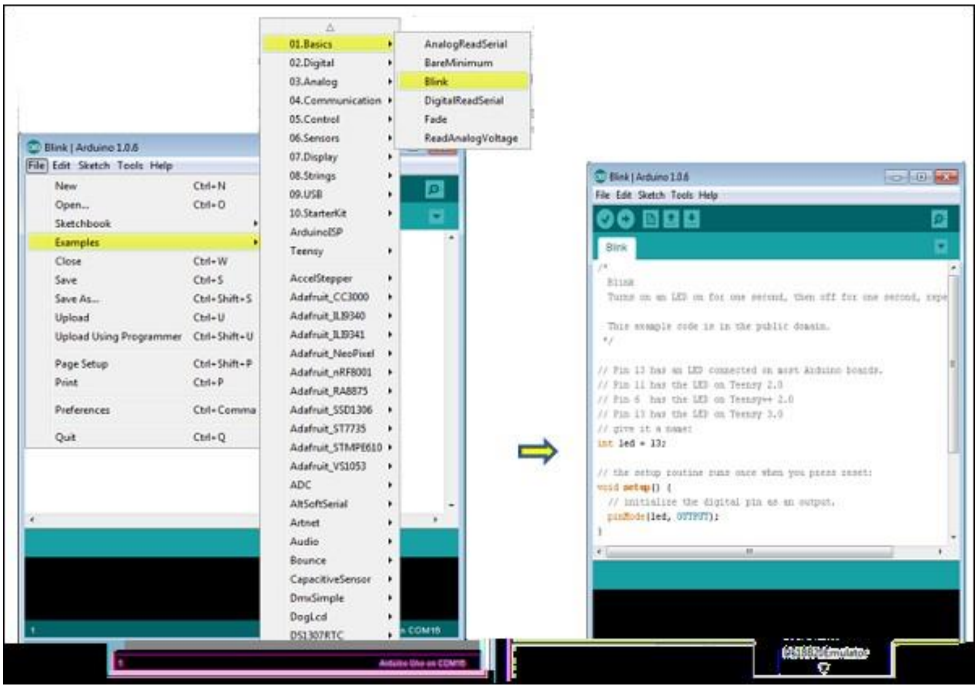


Figure 4.18(b) open existing project

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 – Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

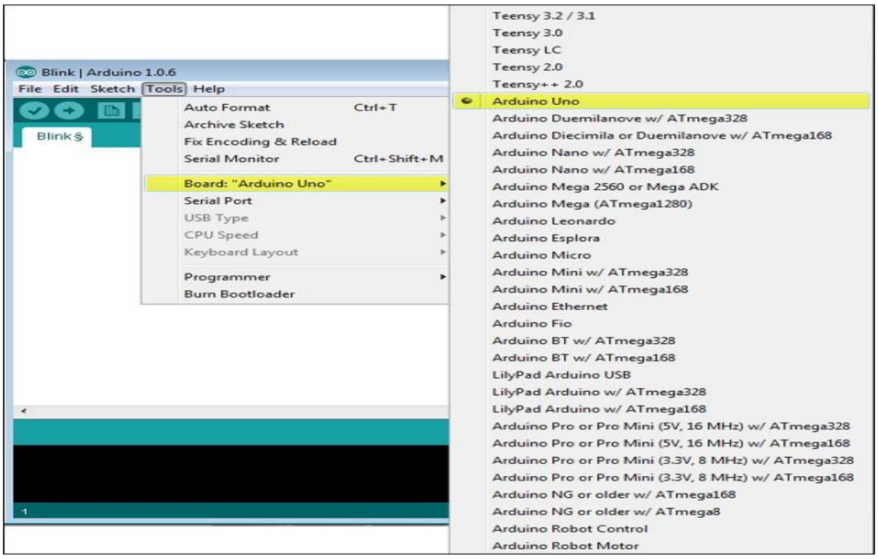


Figure 4.19 Selection of Arduino board

Here, we have selected the Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

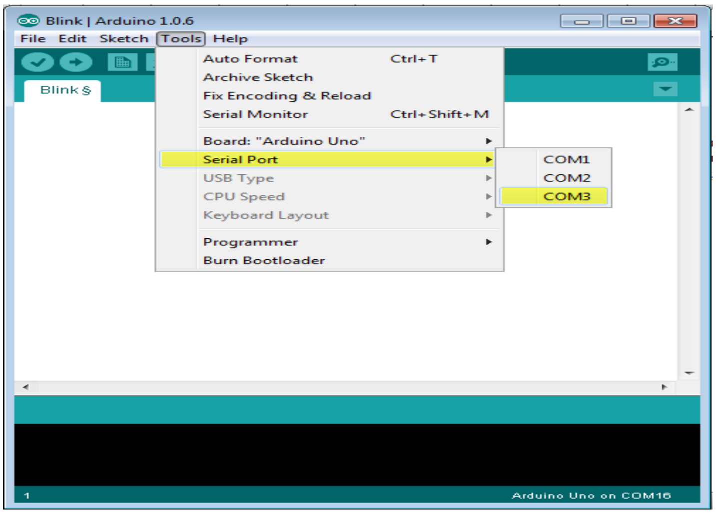


Figure 4.20 selection of serial port

Step 8 – Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

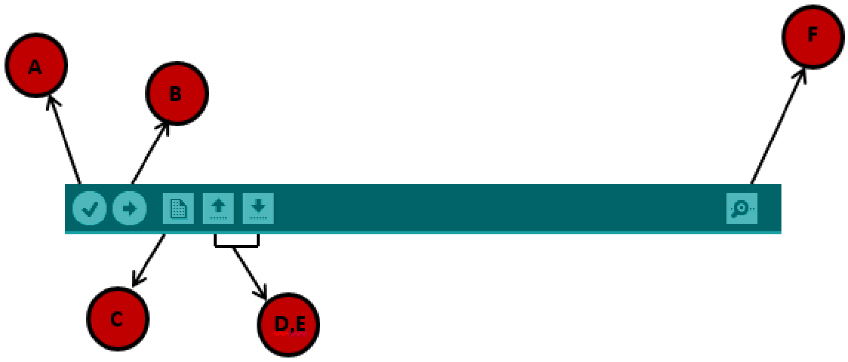


Figure 4.21 upload program

A – Used to check if there is any compilation error.

B – Used to upload a program to the Arduino board.

C – Shortcut used to create a new sketch.

D – Used to directly open one of the example sketch.

E – Used to save your sketch.

F – Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note – If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

B- HARDWARE

Arduino UNO

Arduino is a prototype platform (open-source) based on easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated

Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connecting to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

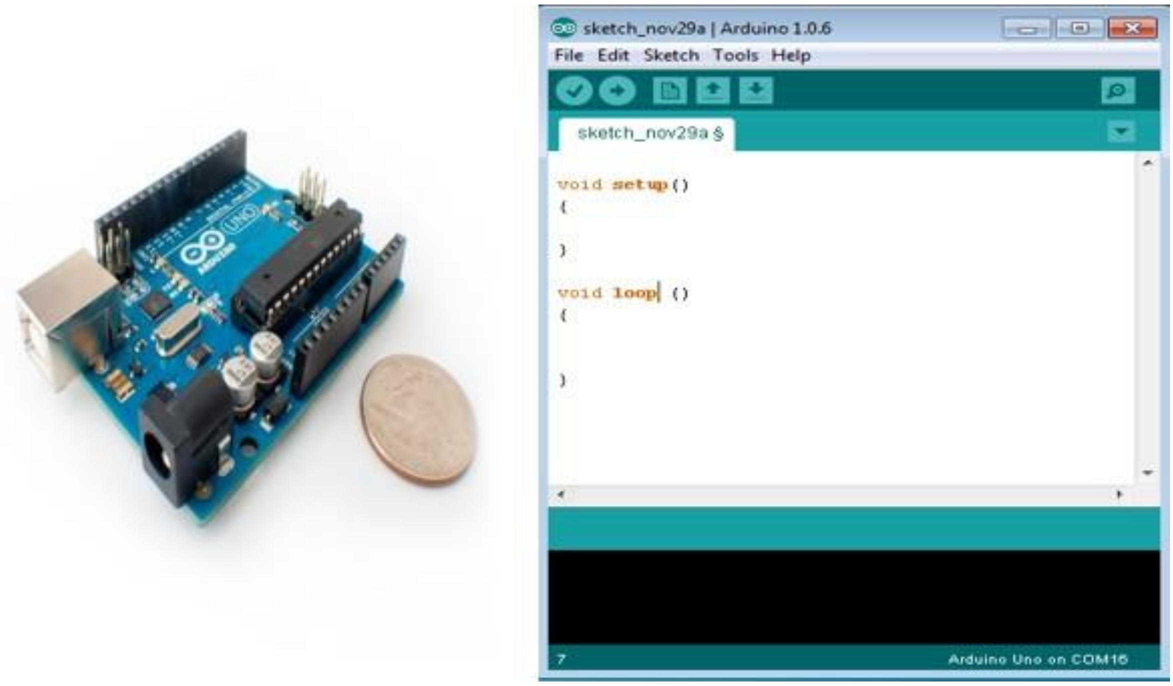


Figure 4.22 Arduino board

Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini-05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini-3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Heade

Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Lilypad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Lilypad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 4.2 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro-3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
Lilypad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 4.3 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 4.4 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table 4.5 Arduino boards based on AT91SAM3X8E microcontroller

Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have the majority of these components in common.

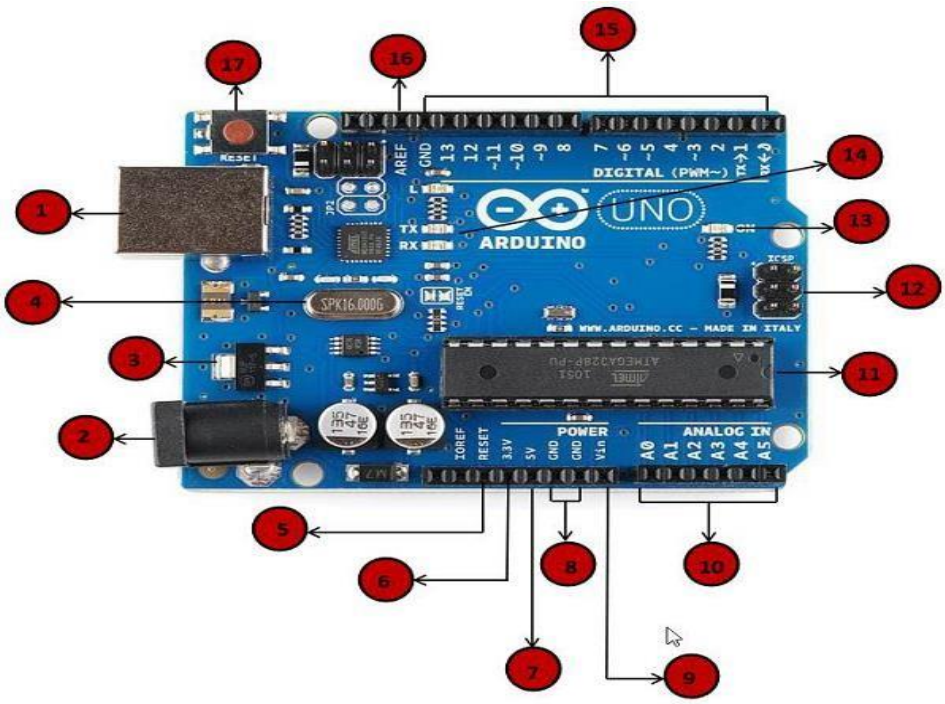


Figure 4.23 Arduino board description

1	**Power USB** An Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) – Supply 3.3 output volt • 5V (7) – Supply 5 output volt • Most of the components used with Arduino boards work fine with 3.3 volts and 5 volts. • GND (8) (Ground) – There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) – This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speeds while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labelled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Table 4.6 Arduino pins description

Liquid Crystal Display

A liquid crystal display (LCD) is a thin, flat display device made up of any number of colour or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to a controller is an LCD display. Some of the most common LCDs connected to the controllers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Shapes and S

available. Line lengths of 8, 16, 20, 24, 32 and 40 characters are all standard, in one, two

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around the LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines (RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

Description Of 16x2:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if not all-Parallel Ports. It however doesn't show the use of the Status Port as an input. So, what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

Schematic Diagram

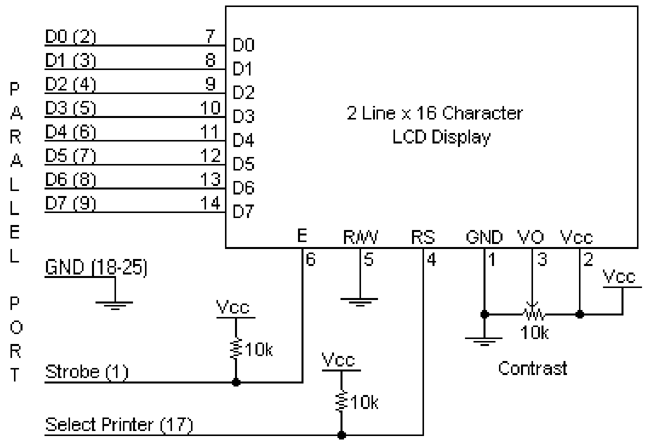


Figure 4.24 Schematic diagram.

- Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore, by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.
- We make no effort to place the Data bus in the reverse direction. Therefore, we hard wire the R/W line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result, we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.
- The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. You can use a bench power supply set to 5v or use an onboard +5 regulator. Remember a few decoupling capacitors, especially if you have trouble with the circuit working properly.

16 x 2 Alphanumeric LCD Module Features:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

16 x 2 Alphanumeric LCD Module Specifications:

PIN	**Symbol**	**Level**	**Function**
1	V_{SS}	-	Power, GND
2	V_{DD}	-	Power, 5V
3	Vo	-	Power, for LCD Drive
4	RS	H/L	Register Select Signal H: Data Input L: Instruction Input
5	R/W	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)
6	E	H, H->L	Enable
7-14	DB0-DB7	H/L	Data Bus; Software selectable 4- or 8-bit mode
15	NC	-	**NOT CONNECTED**
16	NC	-	**NOT CONNECTED**

Table 4.7: 16 x 2 Alphanumeric LCD Module Specifications.

FEATURES:

• 5 x 8 dots with cursor

• Built-in controller (KS 0066 or Equivalent)

• + 5V power supply (Also available for + 3V)

• 1/16 duty cycle

• B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)

• N.V. optional for + 3V power supply

Data can be placed at any location on the LCD. For 16×2 LCD, the address locations are:

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

Figure 4.25 Address locations for a 16×2-line LCD

Even limited to character-based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8,16,20,24,32 and 40 characters are all standard, in one, two and four-line versions.

Several different LC technologies exist. “Supertwist” types, for example, offer Improved contrast and viewing angle over the older “twisted nematic” types. Some modules are available with back lighting, so that they can be viewed in dimly-lit conditions. The back lighting may be either “electro-luminescent”, requiring a high voltage inverter circuit, or simple LED illumination.

PIN DESCRIPTION:

Most LCDs with 1 controller have 14 Pins and LCDs with 2 controllers have 16 Pins (two pins are extra in both for back-light LED connections).

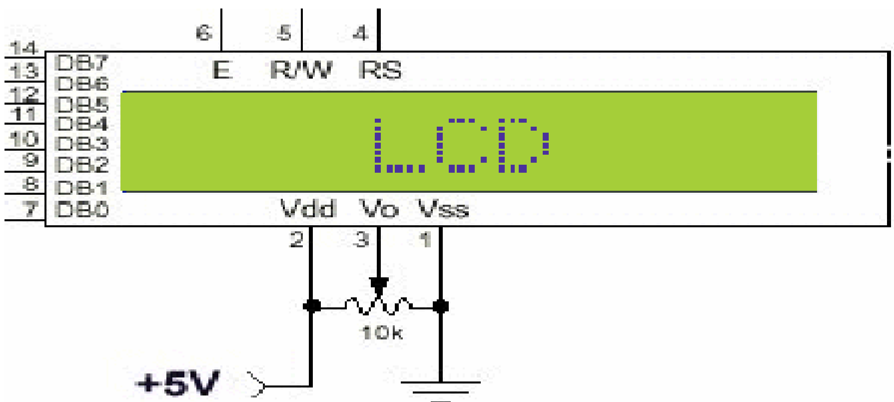


Figure 4.26 Pin diagram of 16x2 lines LCD

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Table 4.8 Pin specifications

CONTROL LINES:

EN: Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS: Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW: Line is the "Read/Write" control line. When RDW is low (0), the information on the data bus is being written to the LCD. When RDW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are written commands, so RW will almost always be low.

Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Logic status on control lines:

• E - 0 Access to LCD disabled

- 1 Access to LCD enabled

• R/W - 0 Writing data to LCD

- 1 Reading data from LCD

• RS - 0 Instructions

- 1 Character

Writing data to the LCD:

1) Set R/W bit to low

2) Set RS bit to logic 0 or 1 (instruction or character)

3) Set data to data lines (if it is writing)

4) Set E line to high

5) Set E line to low

Read data from data lines (if it is reading) on LCD:

1) Set R/W bit to high

2) Set RS bit to logic 0 or 1 (instruction or character)

3) Set data to data lines (if it is writing)

4) Set E line to high

5) Set E line to low

Entering Text:

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the d.i.l. a switch pack with a re-wiring is necessary.

LCD Commands:

There are some present command instructions in the LCD, which we need to send to the LCD through some microcontroller. Some important command instructions are given below:

Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)

06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left
0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

Table 4.9 Commands

ALCOHOL SENSOR

MQ3 is one of the most commonly used sensors in the MQ sensor series. It is a Metal Oxide Semiconductor (MOS) type of sensor. Metal oxide sensors are also known as **Chemiresistors**, because sensing is based on the change of resistance of the sensing material when exposed to alcohol. So, by placing it in a simple voltage divider network, alcohol concentrations can be detected.

Figure 4.27 MQ3 Alcohol sensor

The MQ3 alcohol sensor works on 5V DC and draws around 800mW. It can detect Alcohol concentrations anywhere from 25 to 500 ppm.

Here are the complete specifications

Operating voltage	5V
Load resistance	200KΩ
Heater resistance	33Ω± 5%
Heating consumption	<800mw
Sensing resistance	1 MΩ – 8 MΩ
Concentration scope	25 – 500 ppm
Preheat time	Over 24 hours

Table 4.9 Alcohol sensor specifications

Internal structure of MQ3 Alcohol Sensor

MQ3 is a heater-driven sensor. That's why it is enclosed in two layers of fine stainless-steel mesh called an **Anti-explosion network**. It ensures that the heater element inside the sensor will not cause an explosion, as we are sensing flammable gas (alcohol).

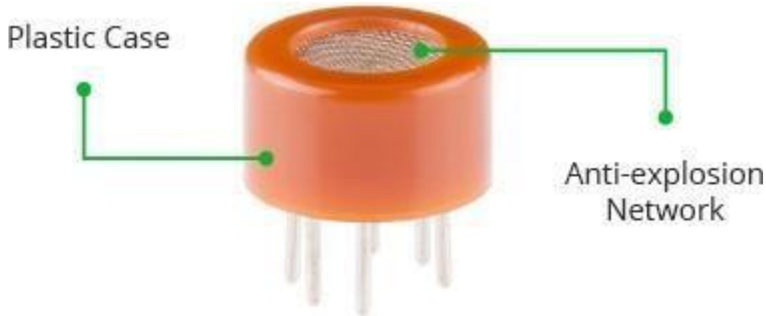


Figure 4.28 external structure of MQ3 alcohol sensor

It also provides protection for the sensor and filters out suspended particles so that only gaseous elements are able to pass inside the chamber.

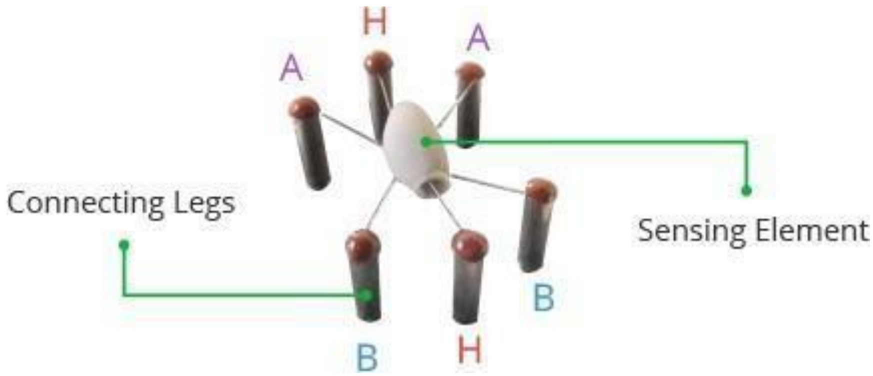


Figure 4.29 internal structure of MQ3 alcohol sensor

This is what the sensor looks like when the outer mesh is removed. The star-shaped structure is formed by the sensing element and six connecting legs that extend beyond the Bakelite base. Out of six, two leads (**H**) are responsible for heating the sensing element and are connected via a **Nickel-Chromium coil** (a well-known conductive alloy).

The remaining four leads (**A** & **B**) responsible for output signals are connected using **Platinum Wires**. These wires are connected to the body of the sensing element and convey small changes in the current that passes through the sensing element.

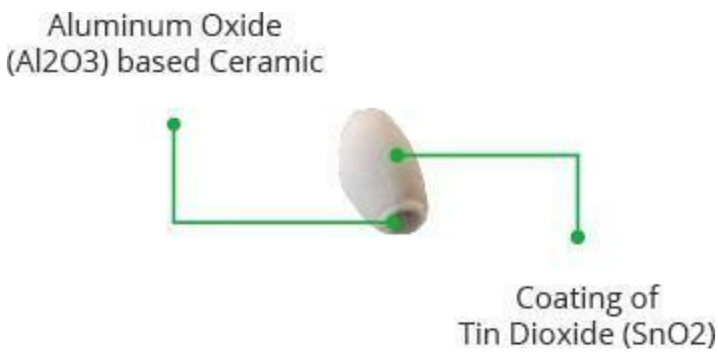


Figure 4.30 sensing element

The tubular sensing element is made up of **Aluminum Oxide** (AL_2O_3) based ceramic and has a coating of **Tin Dioxide** (SnO_2). Tin Dioxide is the most important material being sensitive towards alcohol. However, the ceramic substrate only increases the heating efficiency and ensures that the sensor area is continuously heated to the working temperature.

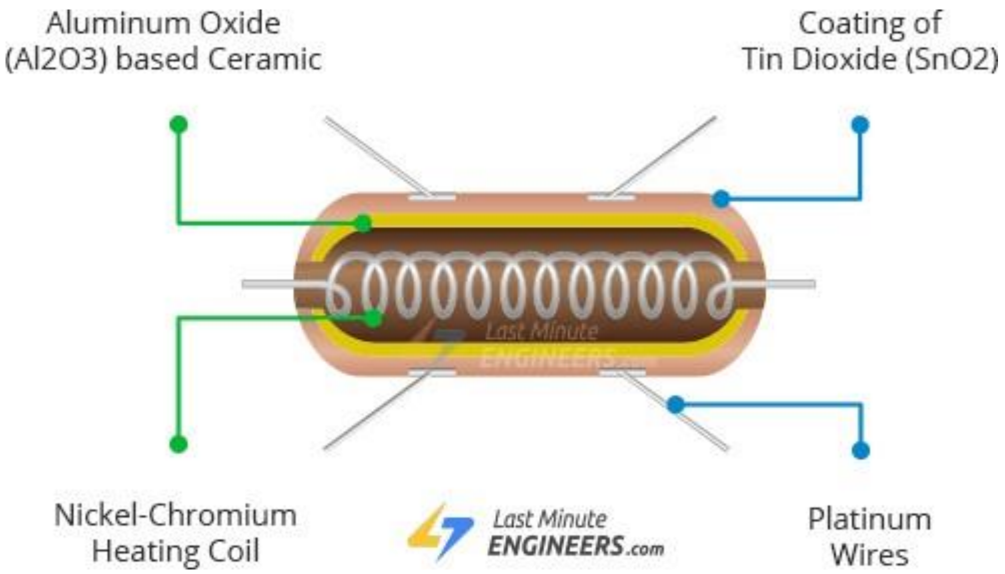


Figure 4.31 tabular sensing element

So, to summarize, the Nickel-Chromium coil and Aluminum Oxide based ceramic forms a **Heating System**; while Platinum wires and coating of Tin Dioxide forms a **Sensing System**.

How MQ3 Alcohol Sensor Works?

When the SnO_2 semiconductor layer is heated at high temperature, oxygen is adsorbed on the surface. In clean air, electrons from the conduction band in tin dioxide are attracted to oxygen molecules. This forms an electron depletion layer just below the surface of SnO_2 particles and forms a potential barrier. As a result, the SnO_2 film becomes highly resistive and prevents electric current flow.

In the presence of alcohol, however, the surface density of adsorbed oxygen decreases as it reacts with the alcohols; which lowers the potential barrier. Electrons are then released into the tin dioxide, allowing current to flow freely through the sensor.

MQ3 Alcohol Sensor Module Hardware Overview

Since the MQ3 alcohol sensor is not breadboard compatible, we recommend this handy little breakout board. It's very easy to use and comes with two different outputs. It not only provides a binary indication of the presence of alcohol but also an analog representation of its concentration in air.

Figure 3.32 MQ3 sensor

The analog output voltage provided by the sensor (at AO pin) varies in proportion to the alcohol concentration. The higher the alcohol concentration in the air, the higher the output voltage; Whereas lower concentration gives lower output voltage. The following animation shows the relationship between alcohol concentration and output voltage.

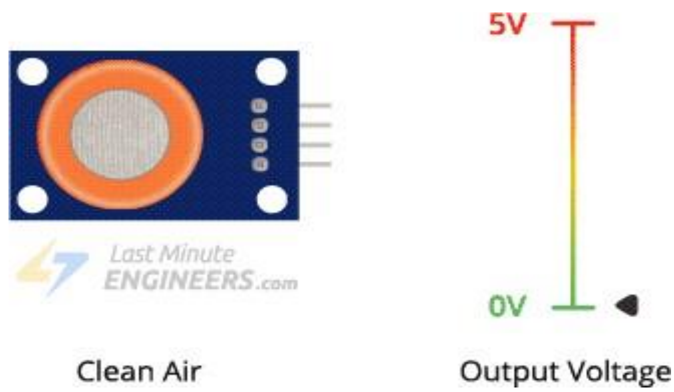


Figure 3.33 Digital Indication of alcohol concentration

The same analog signal is fed to a LM393 High Precision Comparator to digitize it and is made available at the Digital Output (DO) pin.

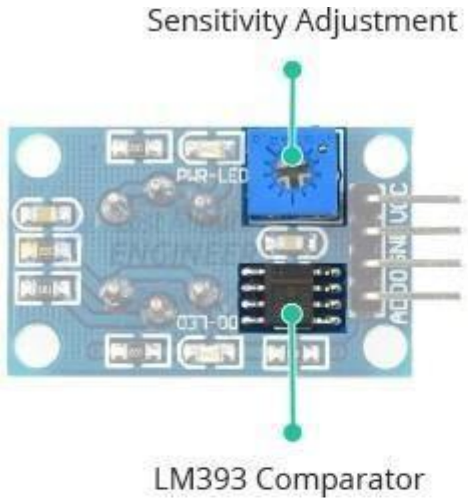


Figure 3.34 Back view of alcohol sensor

The module has a built-in potentiometer for adjusting the sensitivity of the digital output (DO). You can use it to set a threshold; so that when the alcohol concentration exceeds the threshold value, the module will output LOW otherwise HIGH.

This setup is very useful when you want to trigger an action when a certain threshold is reached. For example, when the alcohol concentration in the breath exceeds a threshold, you can tell if someone is drunk.

BUZZER:

Magnetic Transducer

Magnetic transducers contain a magnetic circuit consisting of an iron core with a wound coil and a yoke plate, a permanent magnet and a vibrating diaphragm with a movable iron piece. The diaphragm is slightly pulled towards the top of the core by the magnet's magnetic field. When a positive AC signal is applied, the current flowing through the excitation coil produces a fluctuating magnetic field, which causes the diaphragm to vibrate up and down, thus vibrating air. Resonance amplifies vibration through a resonator consisting of sound hole(s) and cavity and produces a loud sound.

Magnetic Buzzer (Sounder)

Buzzers like the TMB-series are magnetic audible signal devices with built-in oscillating circuits. The construction combines an oscillation circuit unit with a detection coil, a drive coil and a magnetic transducer. Transistors, resistors, diodes and other small devices act as circuit devices for driving sound generators. With the application of voltage, current flows to the drive coil on the primary side and to the detection coil on the secondary side. The amplification circuit, including the transistor and the feedback circuit, causes vibration. The oscillation current excites the coil and the unit generates an AC magnetic field corresponding to an oscillation frequency. This AC magnetic field magnetizes the yoke comprising the magnetic circuit. The oscillation from the intermittent magnetization prompts the vibration diaphragm to vibrate up and down, generating buzzer sounds through the resonator.

Recommended Driving Circuit for Magnetic Transducer

Introduction of Magnetic Buzzer (Transducer)

Specifications:

Rated voltage:	A magnetic buzzer is driven by 1/2 square waves (V o-p).
Operating voltage:	For normal operation. But it is not guaranteed to make the minimum Sound Pressure Level (SPL) under the rated voltage.
Consumption current:	The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.
Direct current resistance:	The direct current resistance is measured by the ammeter directly.
Sound output:	The sound output is measured by a decibel meter. Applying rated voltage and 1/2 square waves, and the distance of 10 cm.
Rated frequency:	A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.
Operating temperature:	Keep working well between -30℃ and +70℃.

How to choose:

Driving methods	AX series with a built in drive circuit will be the best choice when we cannot provide frequency signal to a buzzer, it only needs direct current.
Dimension	Dimension affects frequency, small size results in high frequency.
Voltage	Depend on V o-p (1/2 square waves)

Fixed methods	From the highest cost to the lowest- DIP, wires/ connector, SMD.
Soldering methods	AS series is soldered by hand, the frequency is lower because of the holes on the bottom. On the other hand, we suggest AC series for the reflow soldering, the reliability is better.

How to choose a buzzer

There are many different kinds of buzzers to choose, first we need to know a few parameters, such as voltage, current, drive method, dimension, mounting type, and the most important thing is how much SPL and frequency we want.

Operating voltage	Normally, the operating voltage for a magnetic buzzer is from 1.5V to 24V, for a piezo buzzer is from 3V to 220V. However, in order to get enough SPL, we suggest giving at least 9V to drive a piezo buzzer.
Consumption current	According to the different voltage, the consumption current of a magnetic buzzer is from dozens to hundreds of mill amperes; oppositely, the piezo type saves much more electricity, only needs a few mill amperes, and consumes three times current when the buzzer starts to work.
Driving method	Both magnetic and piezo buzzer have a self-drive type to choose. Because of the internal set drive circuit, the self-drive buzzer can emit sound as long as connecting with the direct current. Due to the different work principle, the magnetic buzzer needs to be driven by 1/2 square waves, and the piezo buzzer needs square waves to get better sound output.
Dimension	The dimension of the buzzer affects its SPL and the frequency, the dimension of the magnetic buzzer is from 7 mm to 25 mm; the piezo buzzer is from 12 mm to 50 mm, or even bigger.
Connecting way	Dip type, Wire type, SMD type, and screwed type for big piezo buzzer are usually seen.
Sound Pressure Level (SPL)	Buzzer is usually tested on the SPL at the distance of 10 cm, if distance doubles, the SPL will decay about 6 dB; oppositely, the SPL will increase 6 dB when the distance is shortened by one time. The SPL of the magnetic buzzer can reach to around 85 dB/ 10 cm; the piezo buzzer can be designed to emit a very loud sound, for example, the common siren is mostly made of piezo buzzer.

Introduction of Piezoelectric Buzzer

Specifications:

Rated voltage	A piezoelectric buzzer is driven by square waves (V p-p).
Operating voltage	For normal operation. But it is not guaranteed to make the minimum SPL under the rated voltage.

Consumption current	The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.
Capacitance	A piezo buzzer can make higher SPL with higher capacitance, but it consumes more electricity.
Sound output	The sound output is measured by a decibel meter. Applying rated voltage and square waves, and the distance of 10 cm.
Rated frequency	A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.
Operating temp.	Keep working well between -30℃ and +70℃.

How to choose:

Driving methods: AZ-xxxxS-x series with built drive circuit will be the best choice when we cannot provide frequency signal to a buzzer, it only needs direct current. Besides, there are different tone natures for you to choose, such as continuous, fast pulse, and slow pulse.

Dimension: Dimension affects frequency, small size results in high frequency.

Voltage: Driven by square waves (V p-p), the higher voltage results in the higher SPL.

Pin Pitch: The numerous spec. for the piezo buzzers lead to the difficulty in finding a spec. in facsimile, therefore we suggest that you can firstly choose a spec. with the same pitch and similar frequency.

Introduction of Micro SpeakerHow to choose:

The factors which affect the SPL: the square measure of diaphragm, the amplitude of vibration, magnetic field intensity, power, impedance, resonant chamber, the pattern and the thickness of diaphragm, and the holes.

Power vs. SPL: Suppose all the conditions are the same, increasing the power does not mean the SPL will increase as well. We need to revise the diaphragm and the sound coil to load the higher power, but it leads to lower SPL instead.

Dimension vs. SPL: A larger speaker can vibrate more air, therefore it provides higher SPL. In addition, the thicker speaker can give wider amplitude of vibration which also leads to higher SPL.

Acoustics: What we request most is how much SPL a micro speaker can output.

Matching: It will be better to provide the power slightly higher than the rated power for the enlarged circuit.

Question for mechanism:

The volume of the resonant chamber: The general problem of the consuming products is that the resonant chambers are not big enough. We can only try to find space to enlarge the volume of the resonant chamber.

Sound Hole: Must be more than 1/8 of the diaphragm's area at least.

Airtight: The front and back sound fields of the speaker should be separated to avoid neutralization.

Shock absorber: When a speaker works the vibration will also happen at the same time. In order to reduce interference, it will do good to have some material between speaker and case to absorb the shock.

Mounting: The speakers are usually fixed on the case. Firmly fixed is important especially for the iron housing or the large size to avoid separating in the drop test.

How to choose the speaker

Dimension: To the micro speaker, size has a decisive influence on its volume. a 5mm difference of diameter might result in a double or half area of diaphragm, therefore the SPL is quite different. Besides, the thicker speaker has more space to vibrate the air, and usually has a bigger magnet, so it will be more powerful to push the air and emit louder sound.

Power: Mainly refers to how much power a speaker can bear, there is no direct relation to the SPL. The speaker with larger power needs to use thicker diaphragm and sound coil to bear larger power, which will lead to lower efficiency (SPL). Therefore, according to the mechanical design, try to select a larger speaker which matches the outputting power from the amplified circuit, then the best SPL would be emitted.

Impedance: Higher impedance can save more electricity; however, the SPL and the loaded power will go down. The reason is that we have to use thinner wire or to coil more, the front makes the power lower, and the after leads to heaviness and low efficiency.

The material of diaphragm: Most speakers (diameter less than 50mm) use mylar diaphragms, which are easily finished, cheaper and waterproof. However, mylar diaphragm is not good at heatproof and the sound is stiff.

The patterns of diaphragm: The speaker with concentric circles diaphragm is good for the speech sounds. Generally, the SPL is good at the frequency before 5-6 KHz, but will dramatically decrease after 6 KHz. On the other hand, the speaker with radiate diaphragm has average frequency response. Supposing other conditions are all the same, the SPL of the radiate diaphragm will be lower than the concentric circles one at the frequency before 6 KHz.

CHAPTER 5

RESULT ANALYSIS

Output

- This is the prototype of the project.
- The main unit of the project is associated alcohol sensor, WIFI module, Arduino uno, buzzer and LCD.
- The IOT application used is Mobile telnet application.
- Following figure shows the working of the kit as power is supplied and input gas is blown to verify the output on both LCD and the online application.

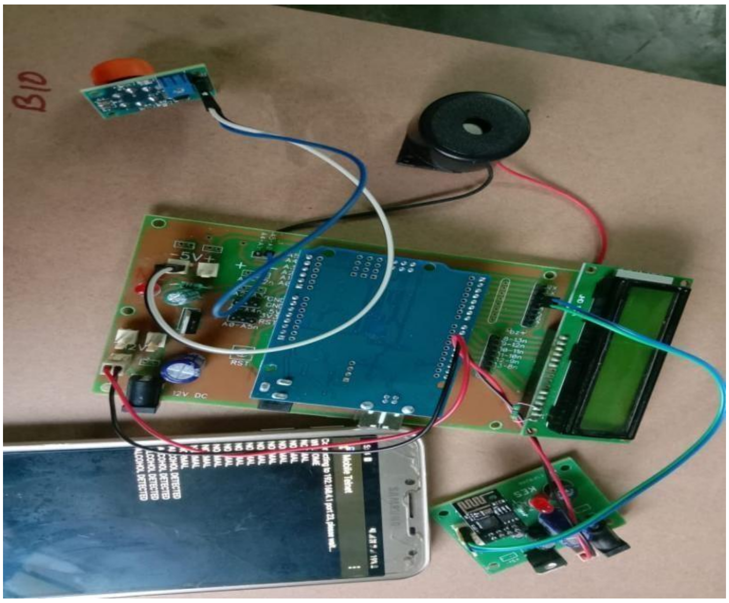


Figure 5.1 Prototype of the project

- The figure 6.2 is showing the display while connecting to the server.

- While connecting to the server it displays “WAITING FOR LINK”.

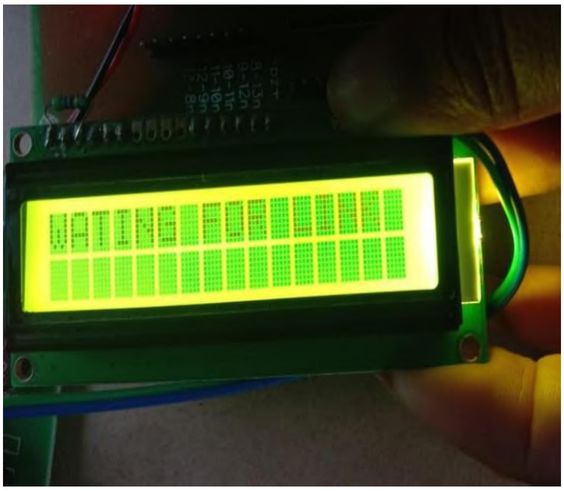


Figure 5.2 Connecting to server

- This figure 6.3 shows the application screen where the information is displayed

- It shows whether the person consumed alcohol or not

- NORMAL in the application screen indicate that there is no alcohol is detected

- ALCOHOL DETECTED in the application screen indicates that the person has consumed the alcohol.

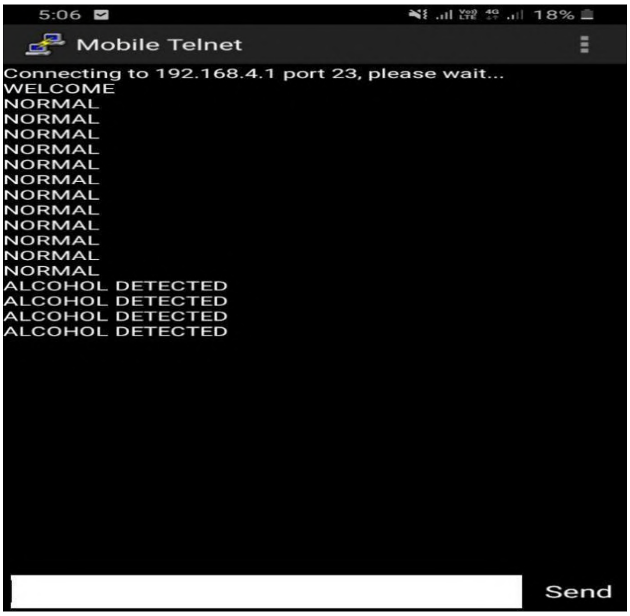


Figure 5.3 Application server

- **The condition used for the code is:**

```
int ad = analogRead(alc);

   lcd.clear();lcd.setCursor(0, 0);lcd.print("ALCOHOL:");lcd.print(ad);

 if(ad > 600 )

 {

     digitalWrite(buz,HIGH);

     lcd.clear();lcd.setCursor(0,1);lcd.print("ALCOHOL DETECTED");

     delay(100);

     sendwifi("ALCOHOL DETECTED \r\n",20);

     delay(2000);

 }
```

```
 else
{
   digitalWrite(buz,LOW);
  lcd.clear(); lcd.setCursor(0,1);lcd.print("NORMAL");
  delay(100);
  sendwifi("NORMAL \r\n",10);
  delay(2000);
```

The figure 5.4 here shows output NORMAL on the lcd screen

- It shows normal when no alcohol is detected by the sensor and the output is low.

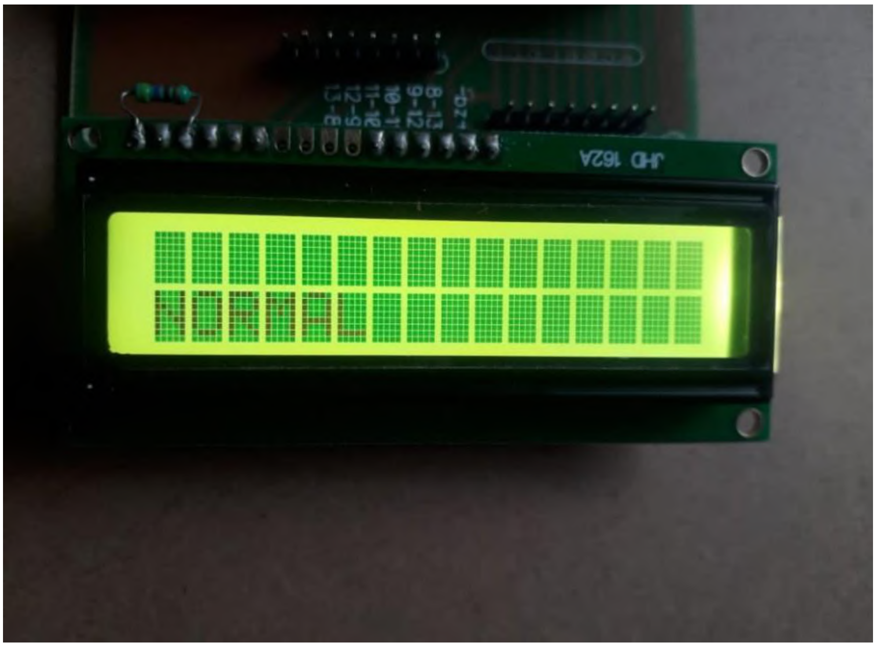


Figure 5.4 LCD displaying output NORMAL

CHAPTER 6

ADVANTAGES AND APPLICATIONS

Advantages

The designed system has several advantages

1. Easy to use.
2. Efficiency of output.
3. Safety of the passenger from accidents.
4. It is very much accurate to detect the presence of alcohol inside the vehicle.

Applications

This device can be used widely in

1. Automobile industry
2. Industries
3. Complex
4. Mall

Limitations

1. Fixing the camera for capturing drunker images is not possible.
2. The program written will not help to show the actual amount of alcohol consumed by the driver

CHAPTER 7

CONCLUSION & FUTURE ENHANCEMENT

Conclusion

In this project, an alcohol detection system was developed for road transportation safety in smart cities using Internet of Things (IoT) technology. This system not only curbs drunk driving by automatically shutting down the vehicle that contains the drunk driver but also allows for traffic authorities to easily locate the shutdown vehicles using the coordinates of the vehicle sent to the web server. The technologies which are used in the proposed system are good enough to ensure the perfect shut down and pick up of the drunk driven vehicle. There are no projects that cannot be improved.

Enhancements have to be carried out so as to improve the efficiency of this system. One of the improvements that could be made on this system in the future is that it should be made smaller. The smaller the system, the more convenient the alcohol system is, the more likely drivers will accept it. There should be proper positioning of the alcohol sensor so as to allow convenient reading of the driver's alcohol consumption quantity with or without the aid of the driver.

Some options for where the sensor can be placed include: (1) an element that can help differentiate the ethanol quantity in alcohol and perfumes should be introduced into the sensor so as to prevent the issue of the system being triggered due to use of large amount of perfume by the driver; (2) a cable can be put near the driver's seat and then connected with the ignition of the car. This means the alcohol detection system can be another key to the car. The driver should blow into the system before he/she starts the car. If the value of the alcohol concentration is above the system's threshold value, the system will stop the car starting. So a drunk driver would not be able to start the car which will prevent the behaviour of drunk driving. It is not only safe to the driver, but can ensure the passengers would not be hit because of the driver's drunk driving.

Future Enhancement

1. The future scope can be added as geofencing can be implemented to the device to get the actual status of the vehicle.
2. The camera can be added for the device for capturing drunker images and reporting it to the police.
3. One more feature can be added as a quick help admin for the women safety in the application.

REFERENCES

1. Altaf SV, Abhinay S, Ansari E, Kaunain Md, Anwer R. Alcohol Detection and Motor Locking System. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. 2017; 6(2): 989-993.
2. Kousikan M, Sundaraj M. Automatic Drunken Drive Prevention System. International Journal of Students Research in Technology and Management. 2014; 2(2): 75-77.
3. Bhuta P, Desai K, Keni A. Alcohol Detection and Vehicle Controlling. International Journal of Engineering Trends and Applications. 2015; 2(2): 92-97.
4. Shafi S., Tanmay NTS, Tarunya D, Vinay G, Reena K. Automatic Vehicle Engine Locking Control System to Prevent Drunken Driving using Virtual Instrumentation. International Journal of Engineering and Technical Research. 2016; 5(1):76-79.
5. Mandalkar RB, Pandora RN, Shinde MB, Godse VD. Alcohol Detection and Accident Avoidance using Locking with Tracking. International Journal of Advanced Research in Computer Science and Management Studies. 2015; 3(9): 142-147.
6. Geeta BS, Marur DR. Smart Drunken Driver Detection and Speed Monitoring System for Vehicles. International Journal of Advanced Technology in Engineering and Science. 2015; 3(3):67-74.
7. Prashanth KP, Padiyar K, Naveen KPH, Kumar KS. Road Accident Avoiding System using Drunken Sensing Technique. International Journal of Engineering

Research and Technology. 2014; 3(10): 818-823.

8. Shah S, Nawgj DD. ARM Based Drunk Driver Identification with Tracking System. International Journal of Modern Trends in Engineering and Research. 2016; 3(4): 302-307.

9. Bindu JH, Reddy GA, Anoosha P, Vinolini KAV. Programmed Engine Locking System by Automatically Detecting Drunken Drivers. International Journal of Innovative Research in Science, Engineering and Technology. 2015; 4(11):11344-11348.

10. Vijay J, Saritha B, Priyadharshini B, Deepika S, Laxmi R. Drunken Drive Protection System. International Journal of Scientific and Engineering Research. 2011; 2(12):1-4.

11. Ogazi C, Edison E. The drunk driving situation in Nigeria. Traffic Injury Prevention. 2012; 13(2):115-9. https://doi.org/10.1080/15389588.2011.645097

CODE

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3); //RX, TX respectively

#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);

int buz = 7,alc=A5;

char res[130];
char buff[130];

void sendwifi(String char,unsigned int len)
{
  mySerial.print("AT+CIPSEND=0,");
  mySerial.println(len-1);
  delay(2000);
  mySerial.print(chr);
  delay(2000);
}

char t;
void setup()
{
  pinMode(buz, OUTPUT);
  digitalWrite(buz,LOW);

  lcd.begin(16,2);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("ALCOHOL SENSING DISPLAY WITH
```

```
ALARM PROJECT");
  delay(500);
  Serial.begin(9600);
  mySerial.begin(115200);
  mySerial.print("AT\r\n");
  delay(1000);
  mySerial.print("ATE0\r\n");
  delay(1000);
    mySerial.print("AT+CWMODE=3\r\n");
    delay(1000);
    mySerial.print("AT+CWSAP=\"Project\",\"project1235\",5,0\r\n\r\n");
    delay(1000);
    mySerial.print("AT+CIPMUX=1\r\n");
    delay(1000);
    mySerial.print("AT+CIPSERVER=1,23\r\n");
    delay(1000);
    lcd.clear();lcd.setCursor(0, 0);lcd.print("WAITING FOR LINK");
    while(1)
    {
      if(mySerial.available())
      {
         //if(Esp.find("0,LINK"))
        if(mySerial.find("0,CONNECT"))
        {
            lcd.clear();lcd.setCursor(0, 0);lcd.print("LINK FOUND");
          break;
        }
      }
    }
    delay(1000);
    sendwifi("WELCOME \r\n",11);
    delay(1000);
}
char tt;

void loop()
{
    int ad = analogRead(alc);
     lcd.clear();lcd.setCursor(0, 0);lcd.print("ALCOHOL:");lcd.print(ad);
  if(ad > 600 )
  {
      digitalWrite(buz,HIGH);
      lcd.clear();lcd.setCursor(0,1);lcd.print("ALCOHOL DETECTED");
      delay(100);
      sendwifi("ALCOHOL DETECTED \r\n",20);
      delay(2000);
  }
  else
 {
      digitalWrite(buz,LOW);
```

```
    lcd.clear(); lcd.setCursor(0,1);lcd.print("NORMAL");
    delay(100);
    sendwifi("NORMAL \r\n",10);
    delay(2000);
  }
}//loop
```

A Major Project report on

ARDUINO BASED SYSTEM TO MEASURE SOLAR POWER

Submitted in partial fulfilment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

IN

Electronics & Communication Engineering

Submitted

By

D. Kiran Reddy (17K81A0474)

M. Sudhakar (17K81A0491)

P. Sharath (18K85A0411)

Under the Guidance of

Mrs. Sri Lakshmi

M. Tech

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100.

NBA &NAAC A+ Accredited

2020-2021

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the major-project work entitled **"Arduino Based System to Measure Solar Power**" is a bonafide work carried out by D. Kiran Reddy (17K81A0474), M. Sudhakar (17K81A0491), P. Sharath (18K85A0411) in partial fulfilment of the requirements for the Degree of Bachelor of Technology in Electronics & Communication Engineering by the Jawaharlal Nehru

Technological University, Hyderabad during the academic year 2020-2021.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree.

INTERNAL GUIDE

Mrs. Sri Lakshmi
Assistant Professor,
Department of ECE.

HEAD of THE DEPARTMENT, ECE

Dr. B. Hari Krishna,
Professor.

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We sincerely thank the management of our college **St. MARTIN'S ENGINEERING COLLEGE** for providing required facilities during our project work.

We derive our great pleasure in expressing our sincere gratitude to our principal **Dr. P. Santosh Kumar Patra** for his timely suggestions, which helped us to complete the project work successfully.

It is the very auspicious moment to express our gratitude to **Dr. B. Hari Krishna**, Head of the Department of ECE for his consistent encouragement during the progress of this project. We take it as a privilege to thank our project coordinator **Dr. A. Anand** Professor, department of ECE for the ideas that led to complete the project work and we also thank them for their continuous guidance, support and unfailing patience, throughout the course of this work.

We sincerely thank our project Internal guide **Mrs. Sri Lakshmi,** Assistant Professor**,** Department of ECE for guidance and encouragement in carrying out this project work.

DECLARATION

We, the students of **"Bachelor of Technology in Department of Electronics and Communication Engineering"**, session: 2017 – 2021, **St. Martin's Engineering College**, hereby declare that the project work entitled "**Arduino Based System to Measure Solar Power**" is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

D. Kiran Reddy	**M. Sudhakar**	**P. Sharath**
17K81A0474	**17K81A0491**	**18K85A0411**

ABSTRACT

The designed project measures different solar cell parameters like light intensity, voltage, current and temperature by using multiple sensor data acquisition. The project uses a solar panel to monitor sunlight and Arduino board which has ATmega family microcontroller attached to it. The project requires an LDR sensor for measuring light intensity, a voltage divider to measure voltage and a temperature sensor to measure the temperature. These measurements are then displayed by the microcontroller to a LCD screen. Thus, this system allows user to effectively monitor solar parameters using this system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1. Introduction to the project:

Rising fossil fuel and burning fuel such as coal, global warming and severe weather conditions have compelled many nations to look for alternative sources to reduce reliance on fossil based fuels. Solar energy is one of the most promising renewable sources that is currently being used worldwide to contribute for meeting rising demands of electric power. [1] Solar power is a conversion of sunlight into electricity, sunlight was collect either directly by using photovoltaics or indirectly using concentrated of solar energy. [2] Photovoltaics was initially use as a power source for a small and medium-size applications from the calculator powered by a single solar cell to a remote homes powered by an off-grid rooftop photovoltaics system. As the cost of solar electricity has fallen, the number of grid-connected solar photovoltaics systems has grown into the millions and utility scale solar power stations with hundreds of megawatts are being built. Solar photovoltaic is becoming inexpensive, low-carbon technology to harness renewable energy from the sun. [3] This paper presented by Arindam Bose et. al [4] describe a potential a solar system using two set of stepper motor, the light sensor and the concave mirror. The purpose of this project to improve the power collection efficiency 65% with developing the track of solar panel perpendicular. This paper presented by Mohsen Taherbaneh et. al [5] proposed the method based on simulation of two fuzzy controllers in order to maximize the generate the output power of solar panel in a photovoltaic system. The output I-V curve from this project are the maximum current is 1.56A nad the maximum voltage are 20V with solar irradiance = 500W/m2 and temperature = 34.5°C. The results of the fuzzy-based maximum power point tracking with 23W with 51% and 11W with 24.5% from the nominal output power. On the other hand, the combination of fuzzy-based maximum power point tracking and fuzzybased sun tracking is 35W with 78% of the nominal output power. Yi-Hua Liu et.al in [6] presented the maximum power point tracking (MPPT) method for low power photovoltaic system, 87W PV system. The MPPT circuit using the low power operational amplifier (OPAMP) with high and low irradation line and an analog switch. The simple DC-DC converter using such as the interface the voltage from the PV system. The type of multi-crystalline solar cell KC85T with the nominal open-circuit voltage of 21.7V and the nominal voltage at the maximum power point of 17.4V. The simulation results shows with 1000W/m2 irradiance, 25°C and the maximum power is 87.46W. From experimental results with

1000W/m2 , the maximum power is 87.39W. The advantages of analog MPPT are very simple, the lowpower analog component, the fast speed and can using to all types of power convertes. This paper [7] propose the method the Fibonnaci search technique for narrow down possible locations. The results shows that 1000W/m2 is 112.4W with 13% efficiency of panel solar. Mohammad H. Moradi presented in [8] to improve the performance of the maximum power point tracking in solar panel. The PV panel with Aria Solar are 60W, with 25°C,1000W/m2 and the current is 2.5A and the voltage is 23.1V. The proposed algorithm are the set point calculation and the fine turning loops. The method was simulated with Matlab/Simulink and also development of the prototype. Concentration of solar energy are concentration of the light from the sun to a point by using lens or mirror and tracking system. The light from the sun also produce heat and if we concentrate at a point. The point will experience the heat produce from the sun [9-10]. This project are focus on measure the solar power using Arduino [10]. This design project are to measured parameters: light intensity, voltage and current and temperature using multiple sensor. The main part in this project are the solar panel, the light sensor, the temperature sensor, a voltage divider, the current sensor and the LCD screen to display.

CHAPTER 2
LITERATURE SURVEY

2.1 Literature Survey:

Title 1 - Design and Research of High Voltage Power Conversion System for Space Solar Power Station

Abstract:

To meet the needs of its characteristics, this paper studies the space high voltage power conversion system's circuit topology and the optimization design of its structure. In the first part, the Common space solar power station power management and distribution method is analyzed. And based on that, a new hybrid power system structure is presented, in which a power transmission bus-bar structure suited for space solar arrays is constructed and the modular multi-converter with serial-parallel combination control strategy is proposed. By applying this new structure, the high transformation ratio in the solar cell arrays and sub-arrays, the low power loss, the large power conversion with high power density and the high efficiency can all be achieved. In the second part, a new soft-switched isolated full-bridge converter is also proposed. And research is made based on the new converter, which includes the study on the transformer modular magnetic integration, the space temperature field and the thermal design. Finally, the simulation and experiment prove the correctness of the theoretical analysis.

Published on: 2018 IEEE International Power Electronics and Application Conference and Exposition (PEAC)

Title 2 - Research on Reliability Evaluation of Power Generation System with Solar Thermal Power

Abstract:

The reliability assessment of hybrid power generation systems with new energy is of great significance for optimizing energy allocation. At present, most of the researches on reliability assessment focus on wind power, photovoltaics and their hybrid power generation, but there is little research on the reliability assessment of hybrid power generation systems including solar thermal power generation (STP). By expanding the reliability evaluation research of wind power and traditional wind power photovoltaic hybrid power generation systems, a reliability evaluation model including STP is established. Considering the operation

characteristics of thermal storage devices in STP system, a sequential Monte Carlo evaluation framework is adopted and random production simulation is performed for two intermittent power generation modes (wind power and STP). Therefore, the credible capacities of STP system with and without heat storage are evaluated when accessed to the grid separately or under hybrid power generation state.

Published on: 2020 5th Asia Conference on Power and Electrical Engineering (ACPEE).

Title 3 -A modelling case study of an island mode solar power plant

Abstract:

This paper deals with efficiency analysis of a solar power plant that is considered to be installed in island mode with 2kWp rated power. The test bed designed in Simulink © consists of solar panel, buck converter, and H-bridge inverter models all based to analytical concepts. The solar panel model is designed according to equation of a solar cell, and all the required parameters such as open circuit voltage, short circuit current etc are considered in order to modify the output power regarding to a commercial photovoltaic (PV) panel. The dc-dc conversion is performed with a buck converter where the optimal duty cycle is dynamically updated by a proportional-integral (PI) controlled generator. The settling time and dc voltage ripples are minimized to increase the efficiency of the dc-dc converter. On the other hand, the single-phase inverter is constructed with an H-bridge that is the most basic multilevel topology known ever. The main intention during the inverter operation was decreasing the total harmonic distortion (THD) ratios as much as possible. Therefore, the modulation scheme is particularly applied to decrease THD by eliminating side band harmonics. The line voltage is properly fixed to 220V/50Hz that is common in northern and central Europe. The THD ratio of the line voltage is measured at 0.03%, while the THD of the current is 0.82%. The power factor of the entire system is measured as 0.99 considering the 8.1° phase displacement between current and voltage waveforms. The analysis results verify the efficiency of the proposed solar power plant and its components.

Published on: 2015 7th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)

Title 4 - Possibilities and limitations of measuring the energy of direct solar radiation

Abstract:

Previous research shows that the most of the energy needs in the near future is going to be satisfied mainly from conventional forms of energy, but it is important to take into account,

that in the period of preparation, design and implementation of power plants that use conventional forms of energy occurs, as a rule, a number of serious problems, limitations and conflict resolution around the many harmful effects on the environment. Energy analysis point to the fact that, except of work at the discovering of additional reserves and improving the procedures and technologies in the use of conventional energy sources, it is necessary to do much more researches on possibility for better and massive use of new, so-called alternative forms of energy, especially energy of direct solar radiation, wind and biomass. In order to avoid unnecessary loss of time and resources, with more intensive use of technological developments it is necessary that in each specific area and example make a quality research, measurement, evaluation, calculation and analysis of available resources, and only based on them, to plan the next steps for optimal satisfaction of fast-growing energy needs. In this paper, the possibilities of the most commonly used means and methods for measuring the energy of direct solar radiation, and their main advantages and disadvantages in specific conditions will be analyzed. Based on the review and analysis of available literature it will be carried out detailed evaluation and analysis of ways of selection suitable measurement methods for measurement locations and the use of different measurement parameters, necessary devices and equipment, taking into account the specific conditions of application. It will be given the results of measurements in Montenegro which will be compared with other methods for assessment of solar power at the specific location. According to those comparisons appropriate analysis will be conducted. Also, the aim of this work is to determine the optimum angles for the installation of solar collectors and photovoltaic cells in order to obtain maximal intensity of radiation at their surfaces.

Published on: 4th International Conference on Power Engineering, Energy and Electrical Drives

Title 5 - Solar electric power for a better tomorrow

Abstract:

The promise of solar electricity based on the photovoltaic (PV) effect is well known, but these systems are not common all over the world. Consumers in the USA are well-known for their attraction to new technology, but PV power systems are still not appearing on roof-tops in the US. The reason may be that grid-connected roof top systems are too difficult to acquire, too difficult to integrate with the grid, too difficult to measure the energy produced and too expensive. The author argues that it is essential that PV power systems are made user-friendly, while reducing the component and system costs. Elegant PV technology must be reduced to practical systems that can be used by the average person everywhere. Choosing the right problem will help achieve the promise of photovoltaics. This paper describes a methodology to choose the right problems as an approach to PV development. Solar cell and PV power system examples are presented.

Published on: Conference Record of the Twenty Fifth IEEE Photovoltaic Specialists Conference - 1996.

CHAPTER 3
INTRODUCTION ABOUT EMBEDDED SYSTEMS

Introduction:

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.

The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes.

The applications software on such processors is sometimes referred to as firmware.

The simplest devices consist of a single microprocessor (often called a "chip”), which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine.

As the embedded system is the combination of both software and hardware

Block diagram of embedded system:

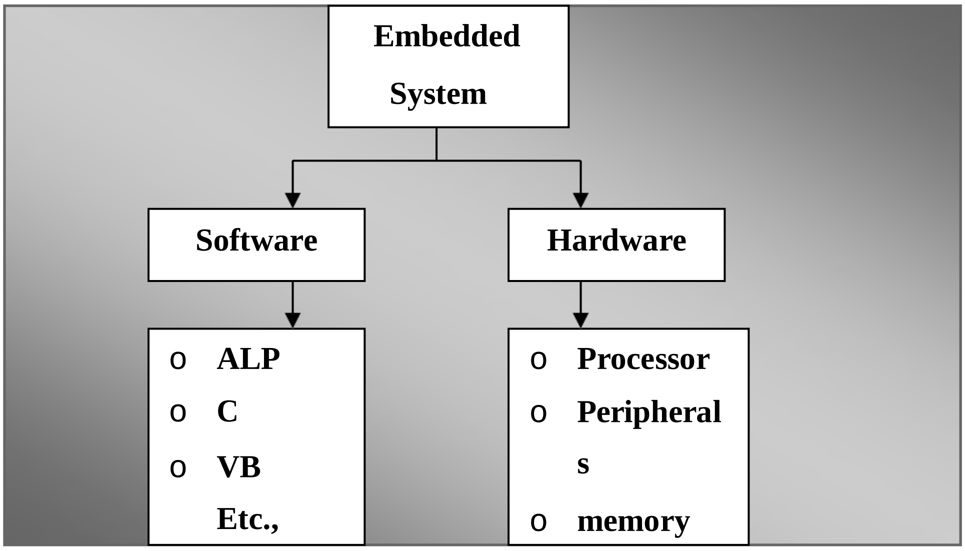


Figure 3.2.1 Block diagram of embedded system

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

Applications of embedded systems:

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

Micro Processor (μp):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Processors are classified into four types like:

- Micro Processor (µp)
- Micro controller (µc)
- Digital Signal Processo1r (DSP)
- Application Specific Integrated Circuits (ASIC)

Three Basic Elements of a Microprocessor:

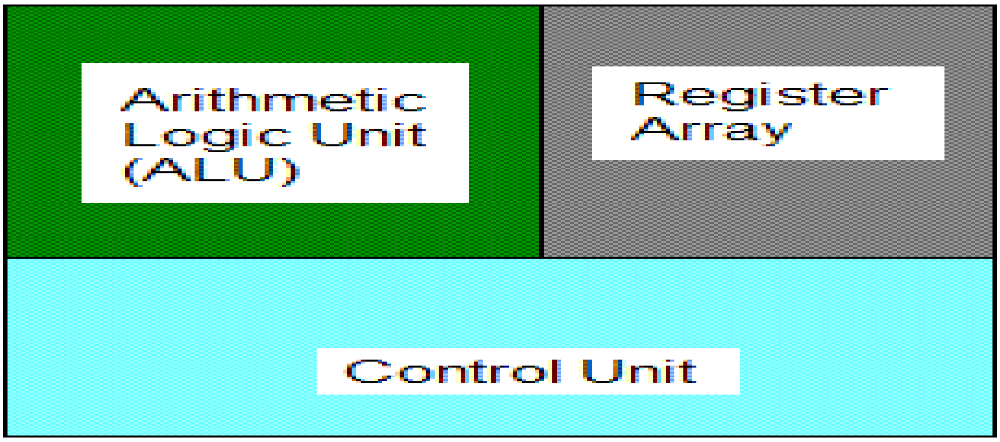


Figure 3.5.1 Three basic elements of a microprocessor

Harvard Architecture:

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

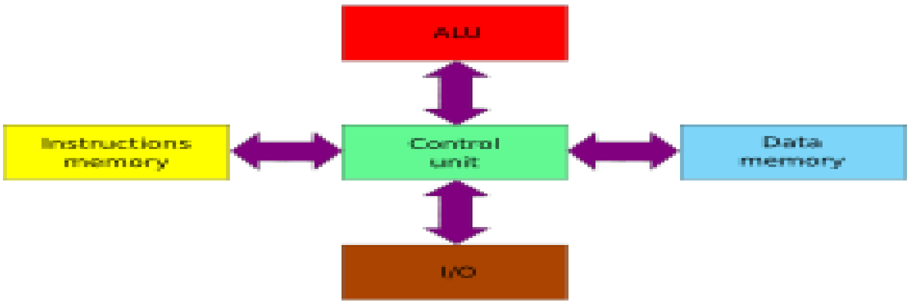


Figure 3.6.1 Harvard Architecture

Uses of the Harvard architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

- Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behaviour must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.
- Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

Von-Neumann Architecture:

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. They cannot be performed at the same time. The von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A stored-program digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

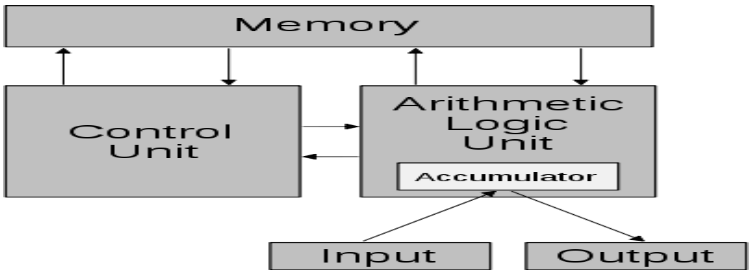


Figure 3.6.2.1 Schematic of the Von-Neumann Architecture

CHAPTER 4
MICRO CONTROLLER UNIT

ATMEGA328:

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The ATmega8 microcontroller contains 32 general purpose working registers. As shown in the below figure these registers are directly connected to ALU. Two registers can carry one single instruction consequently in one clock cycle.

Specifications:

Microcontroller	ATmega328P – 8-bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 Ma
DC Current on 3.3V Pin	50 Ma
Flash Memory	32 KB (0.5 KB is used for Boot loader)

SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

Table 4.1.1 Atmega328 specifications

Arduino:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

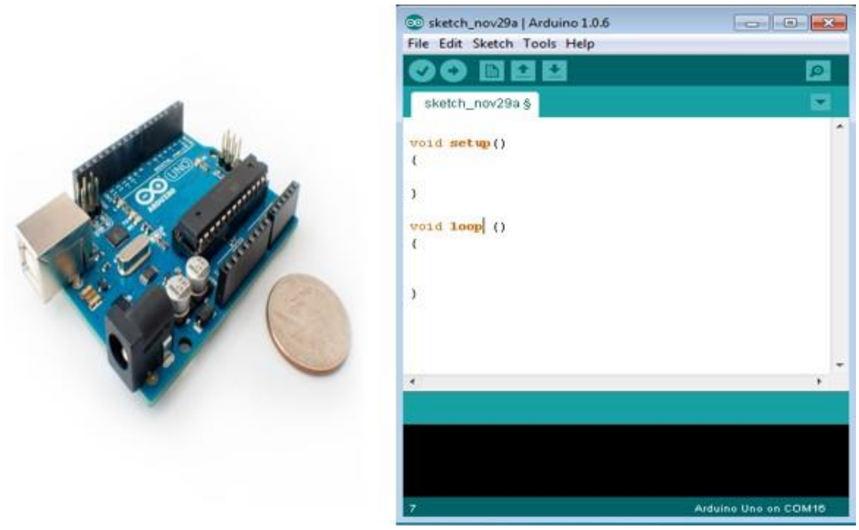


Figure 4.1.1 Arduino Uno

Board Types:

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2

Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header

LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compati
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 4.2.1 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

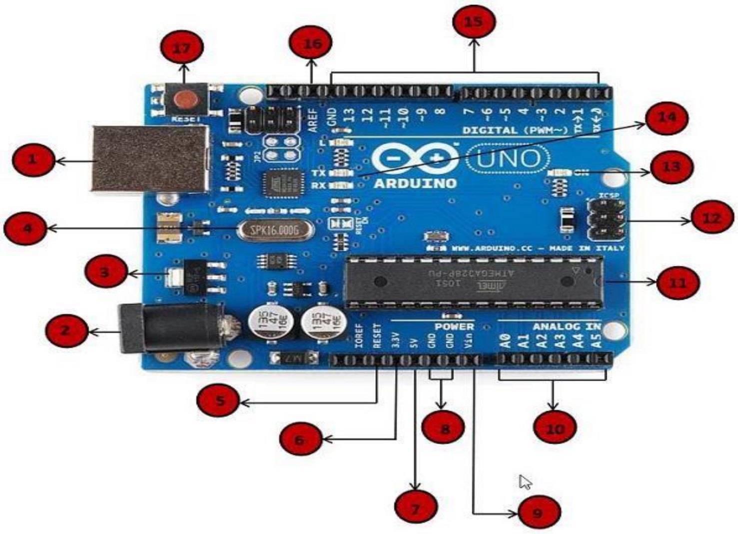
Table 4.2.2 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 4.4.3 Arduino boards based on AT91SAM3X8E microcontroller

4.2.1 Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.



1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7,	**Pins (3.3, 5, GND, Vin)** 3.3V (6) − Supply 3.3 output volt

8, 9	5V (7) − Supply 5 output volt Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The

	TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Arduino Family:

Arduino makes several different boards, each with different capabilities. In addition, part of being open-source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

Figure 4.3.1 Arduino Family

Shields:

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

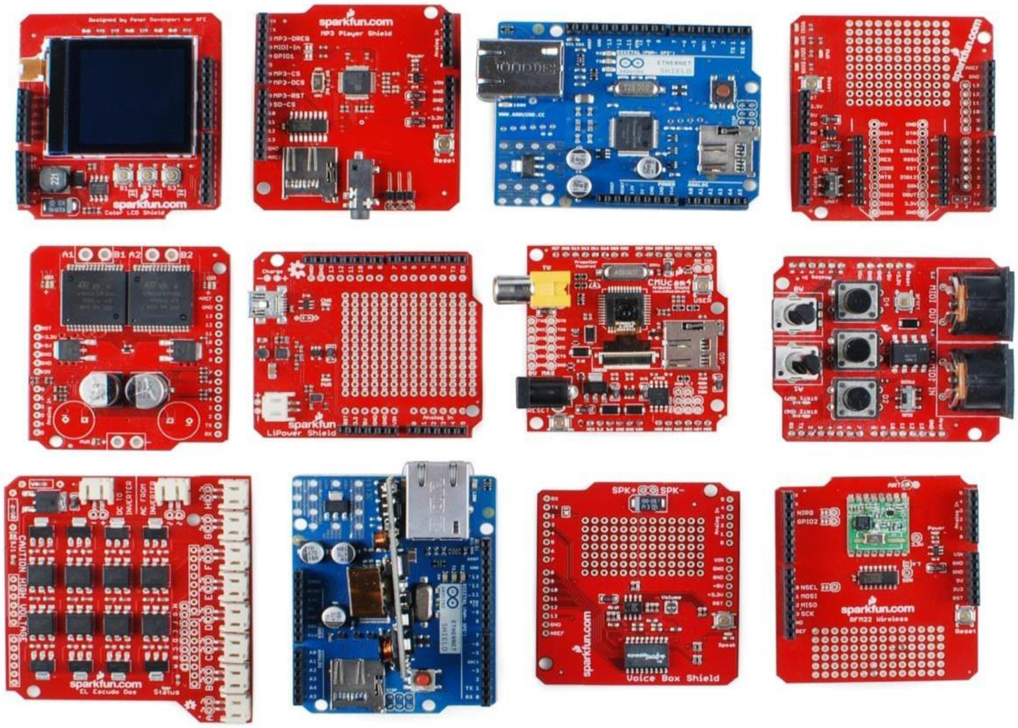


Figure 4.4.1 Arduino Shields

Pin Description of ATMEGA328:

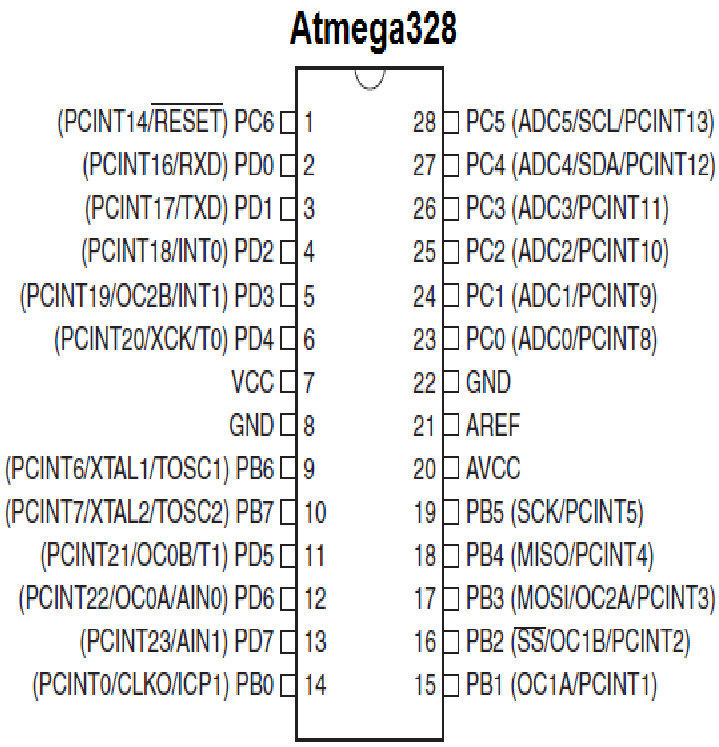


Figure 4.5.1 Pin description of ATMEGA328

Advantages of Arduino:

- It is cheap
- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
- For beginners, it is very simple to use.

Applications:

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model.

CHAPTER 5

POWER SUPPLY UNIT

Introduction:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as “Regulated D.C Power Supply”.

Block Diagram of Power Supply:

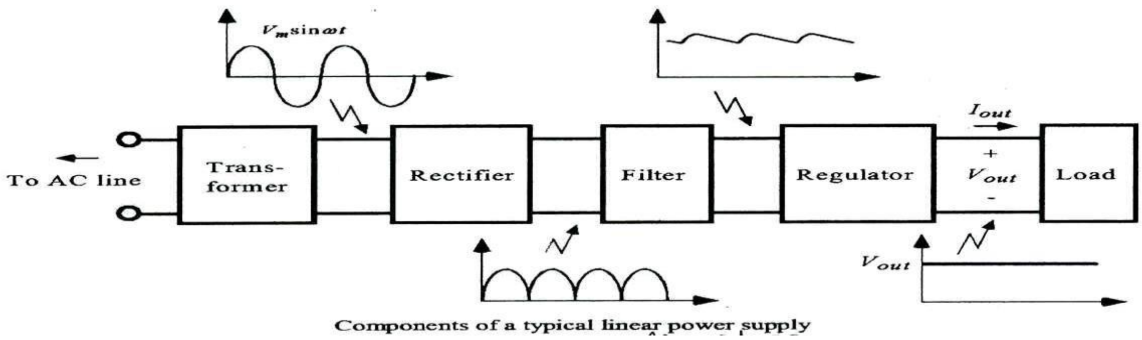


Figure 5.1.1.1 Block Diagram of Power Supply

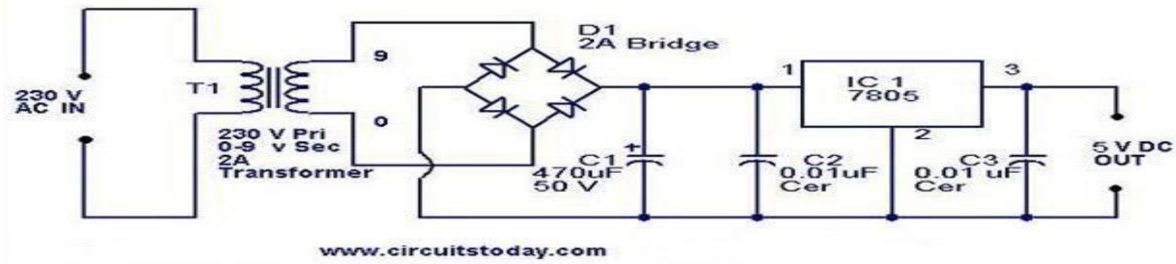


Figure 5.1.1.2 Schematic Diagram of Power Supply

Description of Power Supply:

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a a different form of energy – such as solar, mechanical, or chemical - into electrical energy.

A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered. For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

Transformer:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

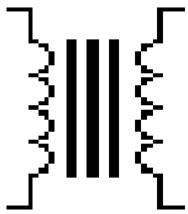


Figure 5.2.1 Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

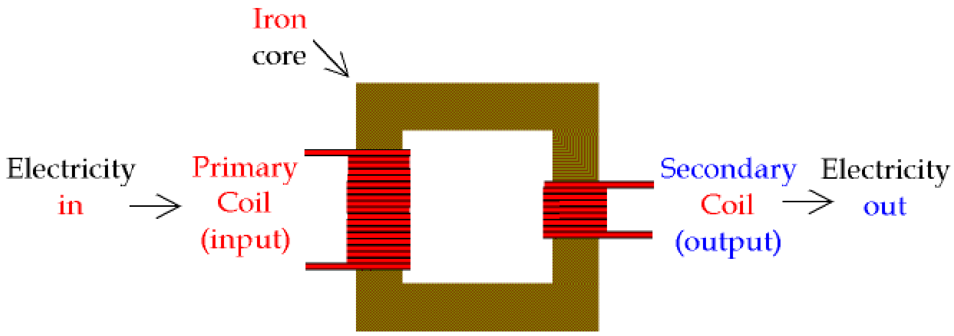


Figure 5.2.2 Transformer

Basic Principle of Transformer:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

From Faraday's Law

$$\frac{V_S}{V_P} = \frac{N_S}{N_P}$$

For ideal transformer

The voltage ratio is equal to the turns ratio, and power in equals power out.

From conservation of energy

$$P_P = V_P I_P = V_S I_S = P_S$$

Working of Transformer:

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

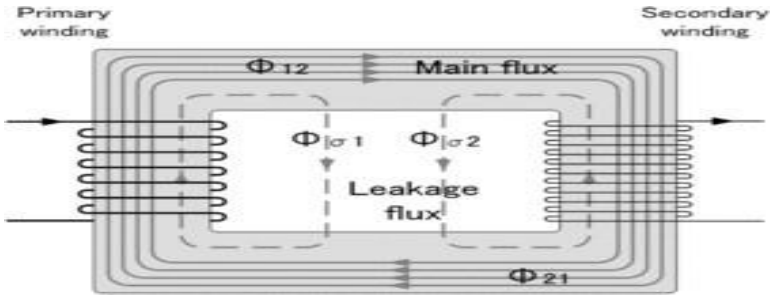


Figure 5.2.1.1 Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced E.M.F. The iron core is laminated to prevent 'eddy currents' flowing in the core.

These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

Classification of Transformer:

- Step-Up Transformer
- Step-Down Transformer

Step-Down Transformer:

Step down transformers is designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step-down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers is made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn's ratio of the two sets of windings determines the amount of voltage transformation.

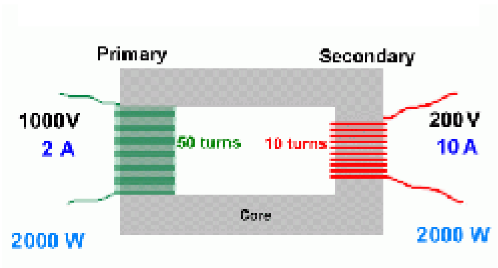


Figure 5.2.2.1 Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts, then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single phase steps down transformers 1kv and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase steps up or step-down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

Step-Up Transformer:

A step-up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step-up transformer is needed to use a 220v product in a country with a 110v supply. A step-up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So, a step-up transformer increases the voltage and a step-down transformer decreases the voltage.

The primary components for voltage transformation are the step-up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, Nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

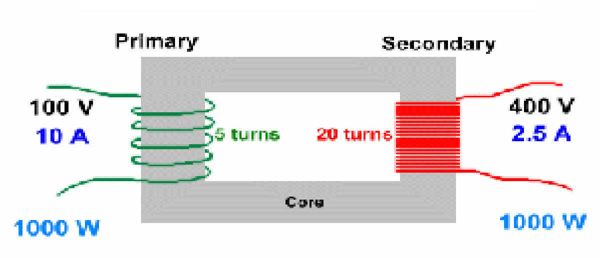


Figure 5.2.2.2 Step-Up Transformer

Diodes:

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

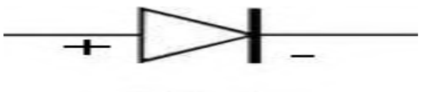


Figure 5.3.1 Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

Rectifier:

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

The Half-Wave Rectifier:

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure.

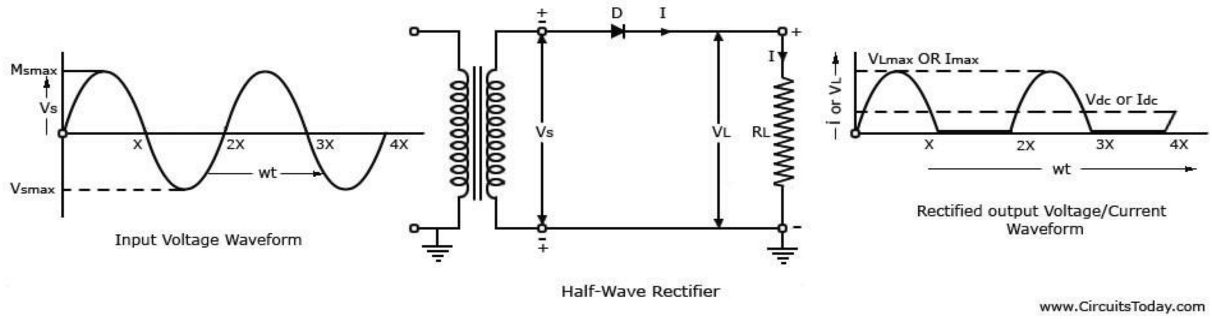


Figure 5.4.1 Half Wave Rectifier

Figure 5.4.1 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

The Full-wave Rectifier:

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC.

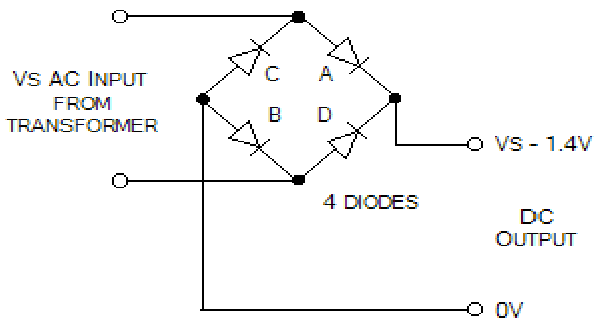


Figure 5.4.2 Full-Wave Rectifier

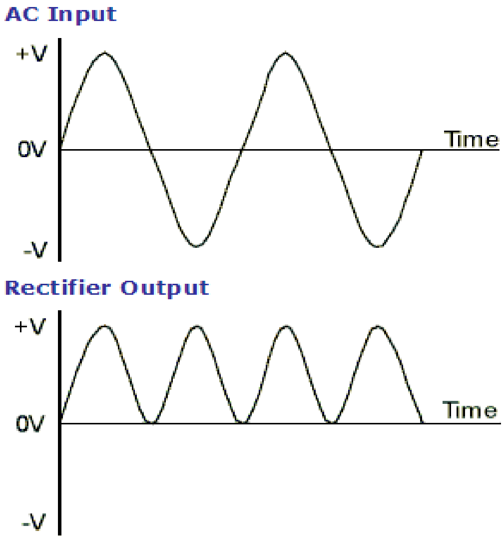


Figure 5.4.3 Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased. While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

Capacitor Filter:

The capacitor-input filter, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

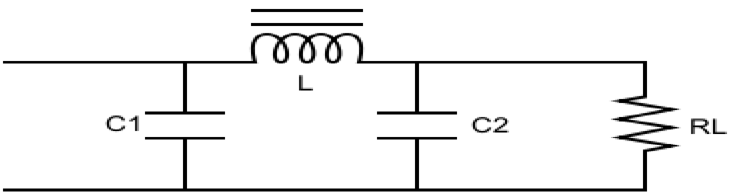


Figure 5.5.1 Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result, the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result, the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result, only the DC component appears across the load RL.

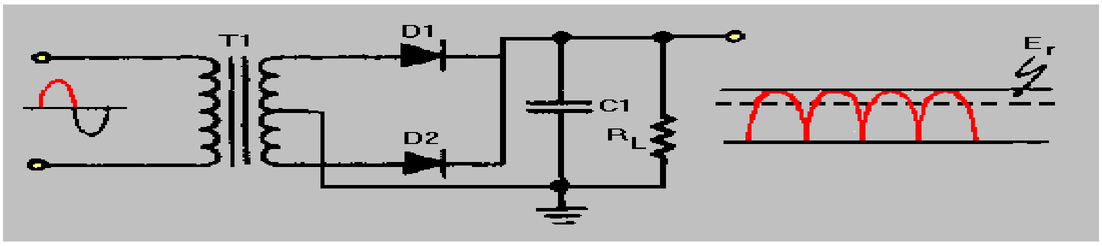


Figure 5.5.2 Centered Tapped Full-Wave Rectifier with a Capacitor Filter

Voltage Regulator:

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulators are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx' indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx' indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consist the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

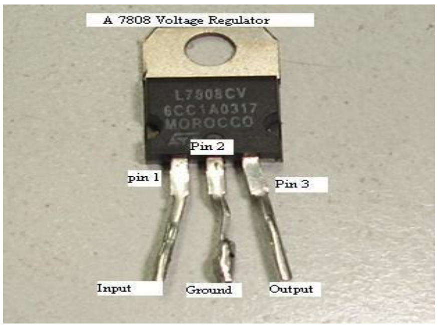


Figure 5.6.1 Regulator

CHAPTER 6

PROJECT DISCRIPTION

Hardware Requirement:

- Battery
- LCD
- L293D
- Motors
- Soil sensor
- Arduino uno
- Relay
- Submersible Motor

Software specification:

- Arduino IDE

Block Diagram of Project:

Block Diagram:

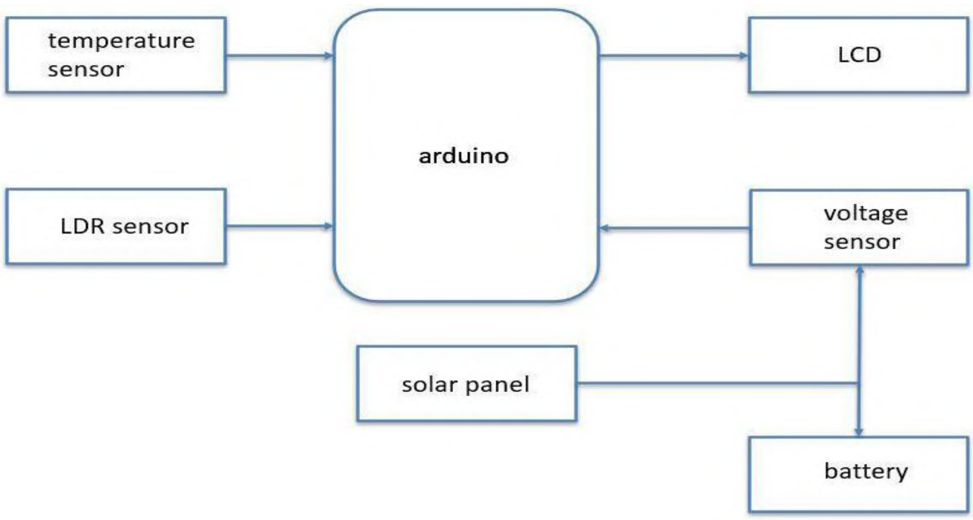


Figure 6.2.1 Block diagram of the project

HARDWARE

Arduino

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analogue or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

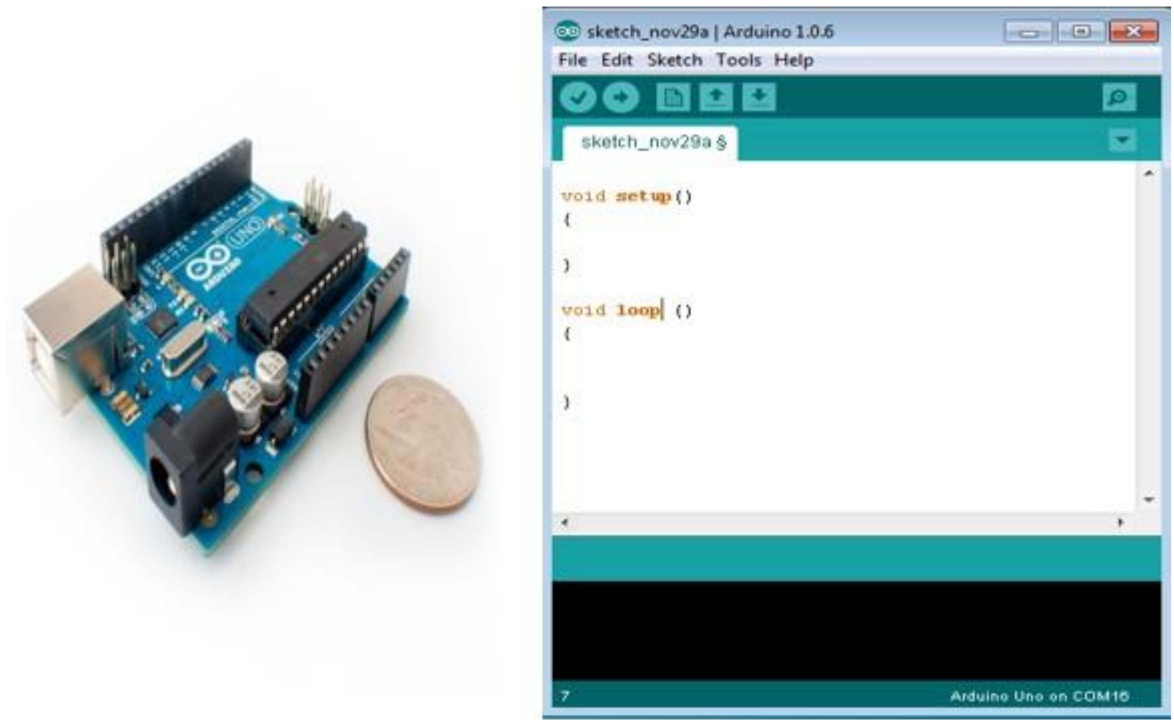


Figure 6.3.1 Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Board Name	**Operating Volt**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header

Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 6.3.2 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 6.3.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2 B

Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 6.3.4 Arduino boards based on AT91SAM3X8E microcontroller

Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

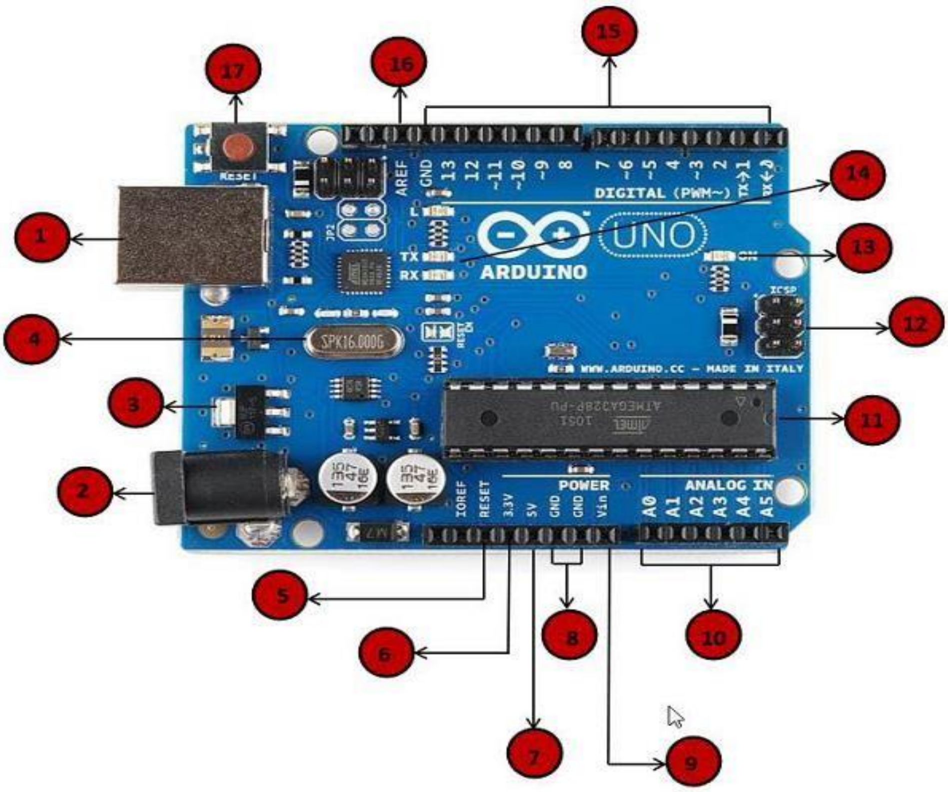


Figure 6.3.5 Arduino Board

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** 3.3V (6) − Supply 3.3 output volt 5V (7) − Supply 5 output volt Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

The term **solar panel** is used colloquially for a photo-voltaic **(PV) module**.

A PV module is an assembly of photo-voltaic cells mounted in a frame work for installation. Photo-voltaic cells use sunlight as a source of energy and generate direct current electricity. A collection of PV modules is called a PV Panel, and a system of Panels is an Array. Arrays of a photovoltaic system supply solar electricity to electrical equipment.

The most common application of solar energy collection outside agriculture is solar water heating systems.[1]

Photovoltaic modules use light energy (photons) from the Sun to generate electricity through the photovoltaic effect. Most modules use wafer-based crystalline silicon cells or thin-film cells. The structural (load carrying) member of a module can be either the top layer or the back layer. Cells must be protected from mechanical damage and moisture. Most modules are rigid, but semi-flexible ones based on thin-film cells are also available. The cells are connected electrically in series, one to another to a desired voltage, and then in parallel to increase amperage. The wattage of the module is the mathematical product of the voltage and the amperage of the module.

A PV junction box is attached to the back of the solar panel and functions as its output interface. External connections for most photovoltaic modules use MC4 connectors to facilitate easy weatherproof connections to the rest of the system. A USB power interface can also be used.

Module electrical connections are made in series to achieve a desired output voltage or in parallel to provide a desired current capability (amperes) of the solar panel or the PV system. The conducting wires that take the current off the modules are sized according to the ampacity and may contain silver, copper or other non-magnetic conductive transition metals. Bypass diodes may be incorporated or used externally, in case of partial module shading, to maximize the output of module sections still illuminated.

Some special solar PV modules include concentrators in which light is focused by lenses or mirrors onto smaller cells. This enables the use of cells with a high cost per unit area (such as gallium arsenide) in a cost-effective way.

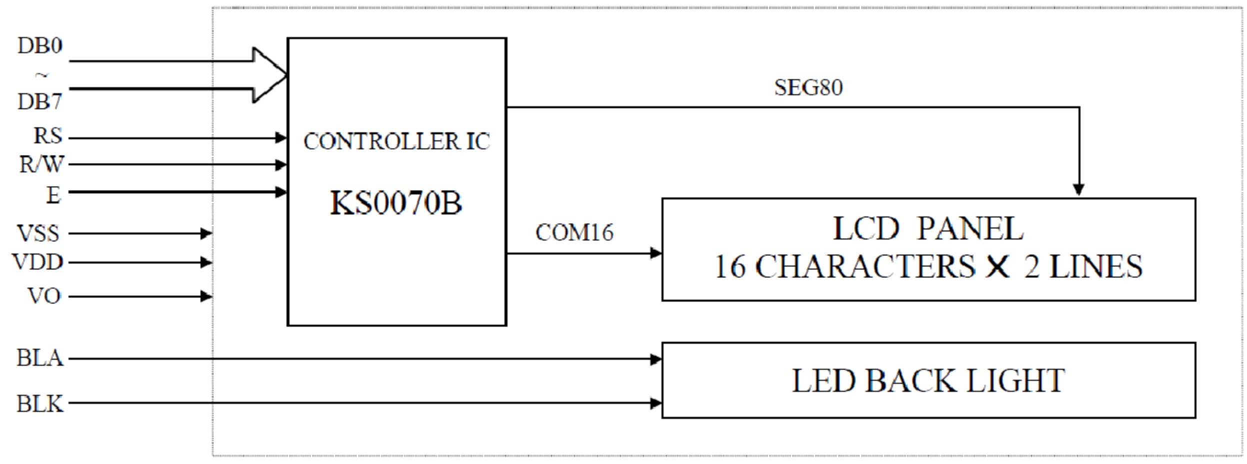
Solar panels also use metal frames consisting of racking components, brackets, reflector shapes, and troughs to better support the panel structure.[2]

16 * 2 Alphanumeric LCD

Description

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

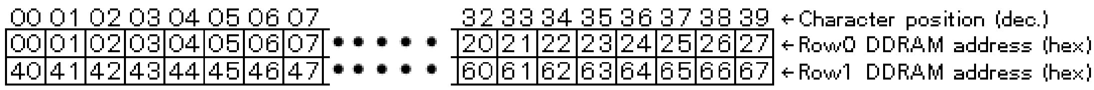
Block Diagram



Display Data RAM

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.



DDRAM Address for 2 Line LCD

Character Generator ROM

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. Userdefined character patterns are also available by mask-programmed ROM.

Busy Flag

Busy Flag is a status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

Instruction Register (IR) and Data Register (DR)

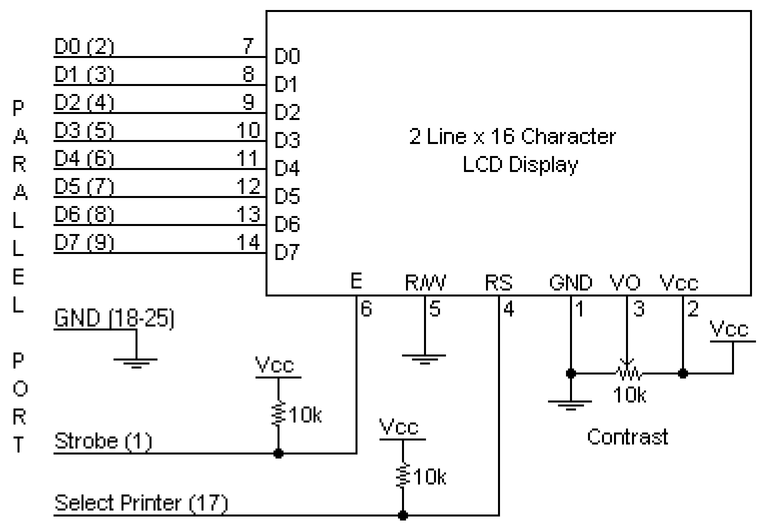
There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD.

Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

16 x 2 Alphanumeric LCD Module Features

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

Schematic



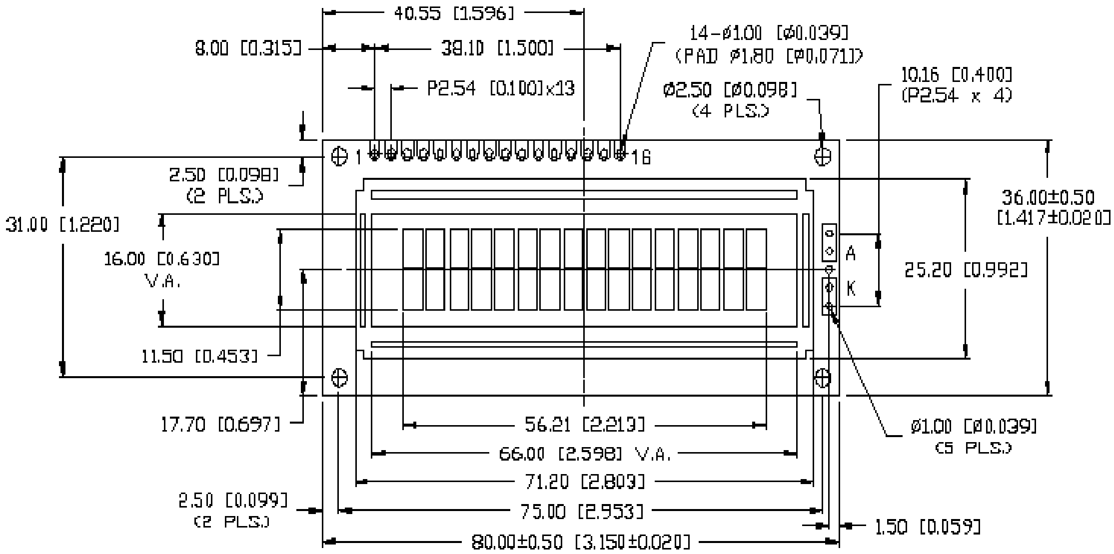
Specifications

Connector Pin Assignment:

Pin	*Symbol*	*Function*	*Pin*	*Symbol*	*Function*
14	DB7	Data Bus Line	6	E	Enable
13	DB6	Data Bus Line	5	R/W	Read/Write
12	DB5	Data Bus Line	4	RS	Register select input
11	DB4	Data Bus Line	3	Vcontrast	Contrast (0V for max contrast)
10	DB3	Data Bus Line	2	**GND (0V)**	Power supply
9	DB2	Data Bus Line	1	**Vcc +5V**	Power supply
8	DB1	Data Bus Line			
7	DB0	Data Bus Line			

Note1: Pin 1 is +5V and pin 2 is GND! This is different from most other HD44780 compatible models.

Note2: Pin 14 is marked with a “1” on the back of the PCB. Do not get confused by this.



Circuit Description

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore, by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore, we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result, we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly. The 2-line x 16-character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

Battery:

Batteries are a collection of one or more cells whose chemical reactions create a flow of electrons in a circuit. All batteries are made up of three basic components: an anode (the '-' side), a cathode (the '+' side), and some kind of electrolyte (a substance that chemically reacts with the anode and cathode).

When the anode and cathode of a battery is connected to a circuit, a chemical reaction takes place between the anode and the electrolyte. This reaction causes electrons to flow through the circuit and back into the cathode where another chemical reaction takes place. When the material in the cathode or anode is consumed or no longer able to be used in the reaction, the battery is unable to produce electricity. At that point, your battery is "dead." Batteries that must be thrown away after use are known as **primary batteries**. Batteries that can be recharged are called **secondary batteries**.

Figure 6.3.6 Lithium Battery

Lithium polymer batteries, for example, can be recharged without batteries, your quadcopter would have to be tethered to the wall, you would have to hand crank your car, and your Xbox controller would have to plugged in all the time (like in the good old days). Batteries offer a way to store electrical potential energy in a portable container.

Figure 6.3.7 Batteries come in a variety of shapes, sizes, and chemistries

History

The Term Battery

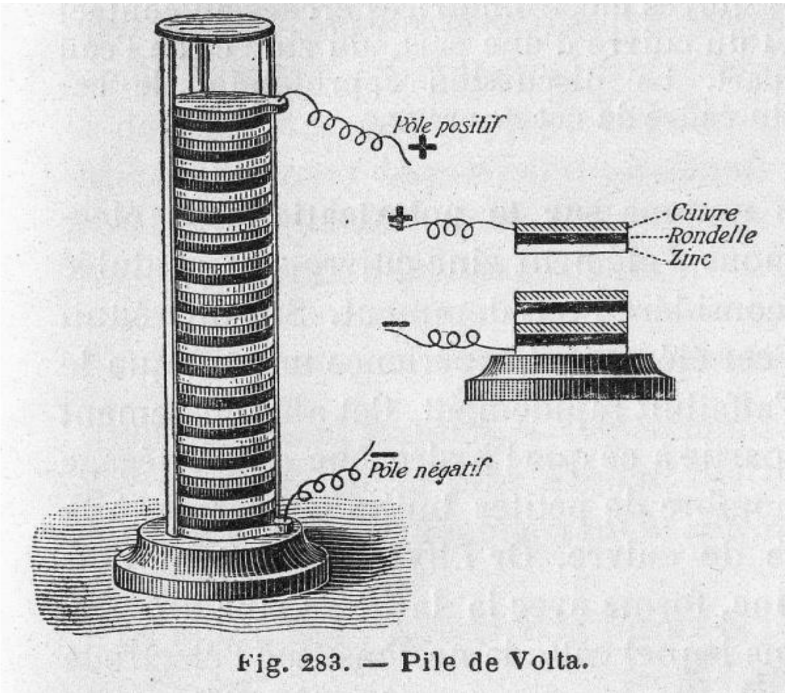
Historically, the word "battery" was used to describe a "series of similar objects grouped together to perform a function," as in a battery of artillery. In 1749, Benjamin Franklin first used the term to describe a series of capacitors he had linked together for his electricity experiments. Later, the term would be used for any electrochemical cells linked together for the purpose of providing electric power.

Battery of Leyden Jar "capacitors" linked together (Image courtesy of Alvin rune of Wikimedia Commons)

Invention of the Battery

One fateful day in 1780, Italian physicist, physician, biologist, and philosopher, Luigi Galvani, was dissecting a frog attached to a brass hook. As he touched the frog's leg with an iron scapel, the leg twitched. Galvani theorized that the energy came from the leg itself, but his fellow scientist, Alessandro Volta, believed otherwise.

Volta hypothesized that the frog's leg impulses were actually caused by different metals soaked in a liquid. He repeated the experiment using cloth soaked in brine instead of a frog corpse, which resulted in a similar voltage. Volta published his findings in 1791 and later created the first battery, the voltaic pile, in 1800.



The voltaic pile consisted of a stack of zinc and copper plates separated by cloth soaked in brine. Volta's pile was plagued by two major issues: the weight of the stack caused the electrolyte to leak out of the cloth, and the particular chemical properties of the components resulted in a very short life span (about an hour). The next two hundred years would be spent perfecting Volta's design and solving these issues.

Fixes to the Voltaic Pile

William Cruickshank of Scotland solved the leakage problem by laying the voltaic pile on its side to form the "trough battery."

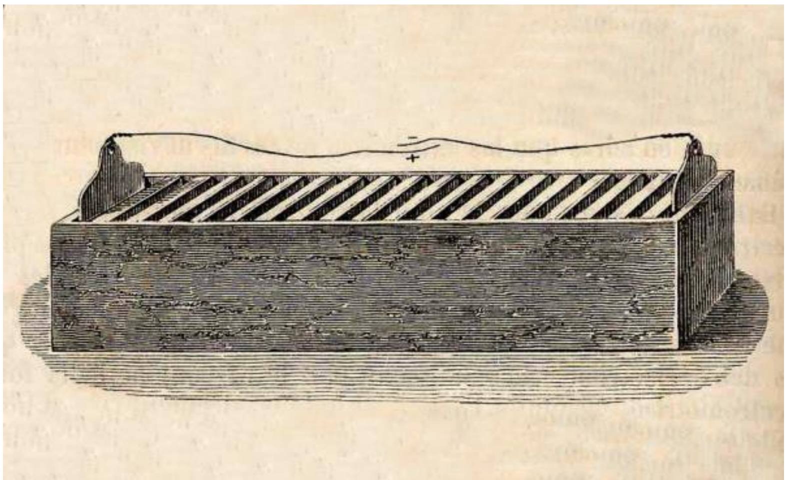


Figure 6.3.8 The trough battery solved the leakage problem of the voltaic pile

The second problem, short life span, was caused by the degradation of the zinc due to impurities and a build-up of hydrogen bubbles on the copper. In 1835, William Sturgeon discovered that treating the zinc with mercury would prevent degradation.

The British chemist John Frederic Daniell used a second electrolyte that reacted with the hydrogen, preventing buildup on the copper cathode. Daniell's two-electrolyte battery, known as the "Daniell cell," would become a very popular solution to providing power to the budding telegraph networks.

Figure 6.3.9 A collection of Daniell cells from 1836

The First Rechargeable Battery

In 1859, the French physicist Gaston Planté created a battery using two rolled sheets of lead submerged in sulfuric acid. By reversing the electrical current through the battery, the chemistry would return to its original state, thus creating the first rechargeable battery.

Later, in 1881, Camille Alphonse Faure improved Planté's design by forming the lead sheets into plates. This new design made the batteries easier to manufacture, and the lead acid battery saw wide-spread use in automobiles.

-> The design for the common "car battery" has been around for more than 100 years (Image courtesy of Emilian Robert Vicol of Wikimedia Commons).

The Dry Cell

Up until the late 1800s, the electrolyte in batteries was in a liquid state. This made battery transportation a very careful endeavor, and most batteries were never intended to be moved once attached to the circuit.

In 1866, Georges Leclanché created a battery using a zinc anode, a manganese dioxide cathode, and an ammonium chloride solution for the electrolyte. While the electrolyte in the Leclanché cell was still a liquid, the battery's chemistry proved to be an important step for the invention of the dry cell.

Carl Gassner figured out how to create an electrolyte paste out of ammonium chloride and Plaster of Paris. He patented the new "dry cell" battery in 1886 in Germany.
These new dry cells, commonly called "zinc-carbon batteries," were massed produced and proved hugely popular until the late 1950s. While carbon is not used in the chemical reaction, it performs an important role as an electrical conductor in the zinc-carbon battery.

-> 3V zinc-carbon battery from the 1960s (Image courtesy of PhFabre of Wikimedia Commons)

<- In the 1950s, Lewis Urry, Paul Marsal, and Karl Kordesch of the Union Carbide company (later known as "Eveready" and then "Energizer") replaced the ammonium chloride electrolyte with an alkaline substance, based on the battery chemistry formulated by Waldemar Jungner in 1899. Alkaline dry cell batteries could hold more energy than zinc carbon batteries of the same size and had a longer shelf life.

Alkaline batteries rose in popularity in the 1960s, overtook zinc-carbon batteries, and have since become the standard primary cell for consumer use.

-> Alkaline batteries come in many shapes and sizes (Image courtesy of Aney~commonswiki of Wikimedia Commons).

20th Century Rechargeable Batteries

In the 1970s, COMSAT developed the nickel-hydrogen battery for use in communication satellites. These batteries store hydrogen in a pressurized, gaseous form. Many man-made satellites, like the International Space Station, still rely on nickel-hydrogen batteries.

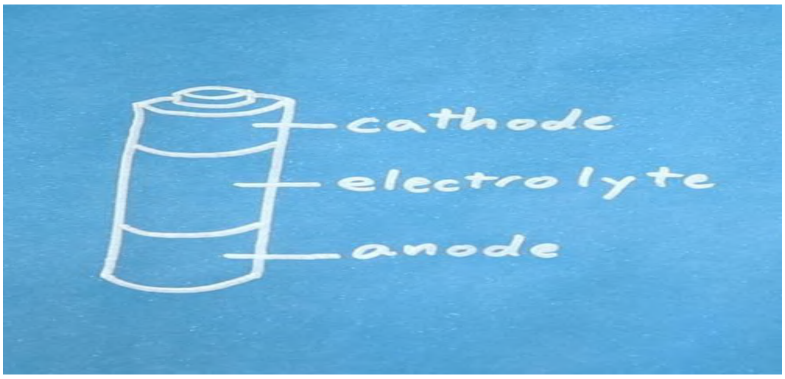
The research of several companies since the late 1960s resulted in the creation of the nickel-metal hydride (NiMH) battery. NiMH batteries were released to the consumer market in 1989, and provided a smaller, cheaper alternative to the rechargeable nickel-hydrogen cells.

Asahi Chemical of Japan built the first lithium-ion battery in 1985, and Sony created the first commercial lithium-ion battery in 1991. In the late 1990s, a soft, flexible casing was created for lithium-ion batteries and gave rise to the "lithium polymer" or "LiPo" battery.

The chemical reactions in the lithium polymer battery are essentially the same as those in the lithium-ion battery. Obviously, many more battery chemistries have been invented, manufactured, and become obsolete. If you would like to read more about modern, popular battery technologies, check out our Battery Technologies tutorial.

Components

Batteries are made up of three basic components: an **anode**, a **cathode**, and an **electrolyte**. A **separator** is often used to prevent the anode and cathode from touching, if the electrolyte is not sufficient. In order to store these components, batteries usually have some kind of **casing**.



OK, most batteries are not actually divided up in three equal sections, but you get the idea. A better cross-section of an alkaline cell can be found on Wikipedia.

Both the anode and cathode are types of **electrodes**. Electrodes are conductors through which electricity enters or leaves a component in a circuit.

Anode

Electrons flow out from the anode in a device connected to a circuit. This means that conventional "current" flows into an anode.

Figure 6.3.10 On batteries, the anode is marked as the negative (-) terminal

In a battery, the chemical reaction between the anode and electrolyte causes a build up of electrons in the anode. These electrons want to move to the cathode, but cannot pass through the electrolyte or separator.

Cathode

Electron's flow into the cathode in a device connected to a circuit. This means that conventional "current" flows out from a cathode.

Figure 6.3.11 On batteries, the cathode is marked as the positive (+) terminal

In batteries, the chemical reaction in or around the cathode uses the electrons produced in the anode. The only way for the electrons to get to the cathode is through a circuit, external to the battery.

Electrolyte

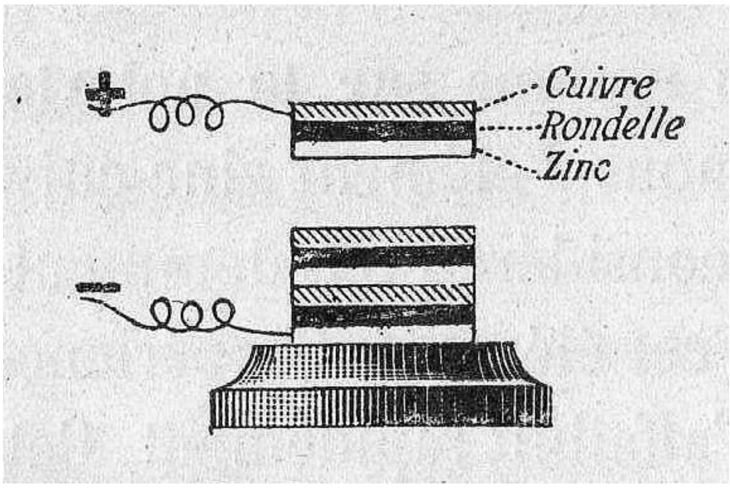
The electrolyte is the substance, often a liquid or gel, that is capable of transporting ions between the chemical reactions that happen at the anode and cathode. The electrolyte also inhibits the flow of electrons between the anode and cathode so that the electrons more easily flow through the external circuit rather than through the electrolyte.

-> Alkaline batteries can leak their electrolyte, potassium hydroxide, if subjected to high heat or reverse voltage (Image courtesy of Wiliam Davies of Wikimedia Commons)

<- The electrolyte is crucial in the operation of a battery. Because electrons cannot pass through it, they are forced to travel through electrical conductors in the form of a circuit that connect the anode to the cathode.

Separator

Separators are porous materials that prevent the anode and cathode from touching, which would cause a short circuit in the battery. Separators can be made from a variety of materials, including cotton, nylon, polyester, cardboard, and synthetic polymer films. Separators do not chemically react with either the anode, cathode, or electrolyte.



The voltaic pile used cloth or cardboard (separator) soaked in brine (electrolyte) to keep the electrodes apart. Ions in the electrolyte can be positively charged, negatively charged, and can come in a variety of sizes. Special separators can be manufactured that allow some ions to pass but not others.

Casing

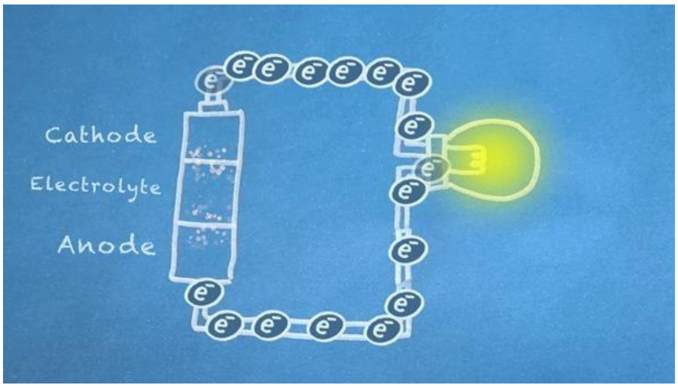
Most batteries need a way to contain their chemical components. Casings, otherwise known as "housings" or "shells," are simply mechanical structures meant to hold the battery's internals.

Figure 6.3.12 This lead-acid battery has a plastic casing

Battery casings can be made of almost anything: plastic, steel, soft polymer laminate pouches, and so on. Some batteries use a conducting steel casing that is electrically connected to one of the electrodes. In the case of the common AA alkaline cell, the steel casing is connected to the cathode.

Operation

Batteries generally require several chemical reactions in order to operate. At least one reaction occurs in or around the anode and one or more reactions occur in or around the cathode. In all cases, the reaction at the anode produces extra electrons in a process called **oxidation**, and the reaction at the cathode uses the extra electrons during a process known as **reduction**.



When the switch is closed, the circuit is complete, and electrons can flow from the anode to the cathode. These electrons enable the chemical reactions at the anode and cathode.

In essence, we are separating a certain kind of chemical reaction, a reduction-oxidation reaction or redox reaction, into two separate parts. Redox reactions occur when electrons are transferred between chemicals. We can harness the movement of electrons in this reaction to flow outside the battery to power our circuit.

Anode Oxidation

This first part of the redox reaction, oxidation, occurs between the anode and electrolyte, and it produces electrons (marked as e^-).

Some oxidation reactions produce ions, such as in a lithium-ion battery. In other chemistries, the reaction consumes ions, like in the common alkaline battery. In either case, ions are able to flow freely through the electrolyte where electrons cannot.

Cathode Reduction

The other half of the redox reaction, reduction, occurs in or near the cathode. Electrons produced by the oxidation reaction are consumed during reduction.

In some cases, like lithium-ion batteries, positively charged lithium ions produced during the oxidation reaction are consumed during reduction. In other cases, like alkaline batteries, negatively charged ions are produced during reduction.

Electron Flow

In most batteries, some or all of the chemical reactions can occur even when the battery is not connected to a circuit. These reactions can impact a battery's shelf life.

For the most part, the reactions will only occur at full force when an electrically conductive circuit is completed between the anode and cathode. The less resistance between the anode and cathode, the more electrons are allowed to flow, and the quicker the chemical reactions occur.

Sensor:

A sensor (also called detectors) is a device that measures a measurable attribute and converts it into a signal which can be read by an observer or by an instrument. For example, a mercury-in-glass thermometer converts the measured temperature into expansion and contraction of a liquid which can be read on a calibrated glass tube. A thermocouple converts temperature to an output voltage which can be read by a voltmeter.

Temperature Sensor:

Temperature Sensor which converts temperature value into electrical signals. We used IC called LM 35 as a temperature sensor. LM35 series sensors are precision integrated-circuit temperature sensors whose output voltage is linearly proportional to the Celsius temperature. The LM35 requires no external calibration since it is internally calibrated. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°C over a full −55 to +150°C temperature range.

The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air.

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteed (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts

- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±1⁄4°C typical
- Low impedance output, 0.1 W for 1 mA load

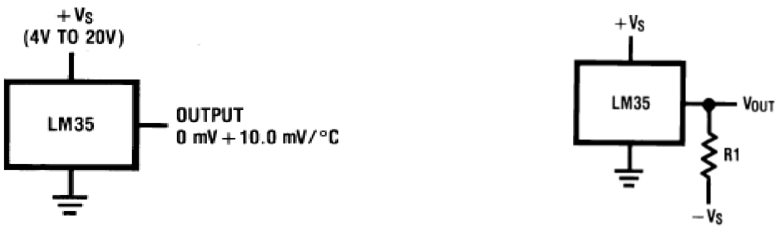


Figure 6.3.13 Temperature Sensor

LM35:

- ➢ LM35 is precision integrated circuit temperature sensor. Its output voltage is linearly proportional to temperature (in Celsius).
- ➢ The LM35 thus has an advantage over linear temperature sensors calibrated in° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°C over a full −55 to +150°C temperature range
- ➢ It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air.

The characteristic of this LM35 sensor is:

For each degree of centigrade temperature, it outputs 10milli volts. ADC accepts the output from LM35 and converts that data into digital form which is sent to microcontroller for further processing.

Temperature Sensing Circuit

The methods of temperature measurement may be divided into two main classes according as the exchange of heat between the testing body and the hot system takes place by contact, or by radiation across a space. In the contact methods, thermometers or thermocouples are used and they are immersed in solids or liquids. The thermodynamic equilibrium between the hot body and the testing body is established by material contact. In the non-contact methods, the thermodynamic equilibrium is established by the radiation emitted as excited atom and molecules in the hot body return to the ground state.

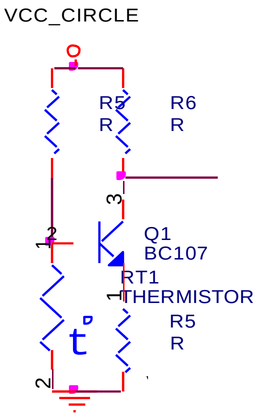


Figure 6.3.14 Temperature Sensing Circuit

Light Dependent Resistor

A photoresistor or light dependent resistor or cadmium sulfide (CdS) cell is a resistor whose resistance decreases with increasing incident light intensity. It can also be referred to as a photoconductor.

A photo resistor is made of a high resistance semiconductor. If light falling on the device is of high enough frequency, photons absorbed by the semiconductor give bound electrons enough energy to jump into the conduction band. The resulting free electron (and its hole partner) conduct electricity, thereby lowering resistance.

A photoelectric device can be either intrinsic or extrinsic. An intrinsic semiconductor has its own charge carriers and is not an efficient semiconductor, e.g. silicon. In intrinsic devices the only available electrons are in the valence band, and hence the photon must have enough energy to excite the electron across the entire bandgap. Extrinsic devices have impurities, also called dopants, added whose ground state energy is closer to the conduction band; since the electrons do not have as far to jump, lower energy photons (i.e., longer wavelengths and lower frequencies) are sufficient to trigger the device. If a sample of silicon has some of its atoms replaced by phosphorus atoms (impurities), there will be extra electrons available for conduction. This is an example of an extrinsic semiconductor.

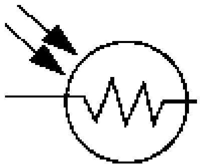


Figure 6.3.15 The symbol for a photoresistor

Applications:

Photoresistors come in many different types. Inexpensive cadmium sulfide cells can be found in many consumer items such as camera light meters, street lights, clock radios, alarms, and outdoor clocks.

They are also used in some dynamic compressors together with a small incandescent lamp or light emitting diode to control gain reduction.

Lead sulfide (PbS) and indium antimonide (InSb) LDRs (light dependent resistor) are used for the mid infrared spectral region. Ge:Cu photoconductors are among the best far-infrared detectors available, and are used for infrared astronomy and infrared spectroscopy.

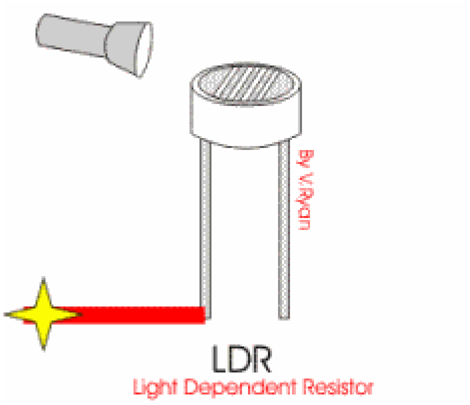
Transducers are used for changing energy types.

Figure 6.3.16 A light dependent resistor

A light dependent resistor is a small, round semiconductor. Light dependent resistors are used to re-charge a light during different changes in the light, or they are made to turn a light on during certain changes in lights. One of the most common uses for light dependent resistors is in traffic lights. The light dependent resistor controls a built in heater inside the traffic light, and causes it to recharge overnight so that the light never dies. Other common places to find light dependent resistors are in: infrared detectors, clocks and security alarms.

LDRs or Light Dependent Resistors are very useful especially in light/dark sensor circuits. Normally the resistance of an LDR is very high, sometimes as high as 1000 000 ohms, but when they are illuminated with light resistance drops dramatically.

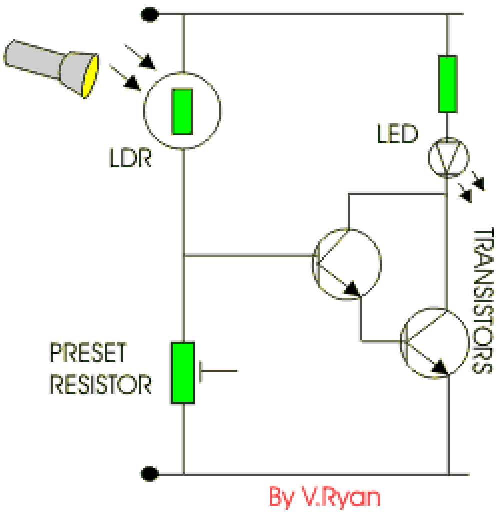
The animation opposite shows that when the torch is turned on, the resistance of the LDR falls, allowing current to pass through it.



Circuit Wizard software has been used to display, the range of values of a ORP12, LDR.

When a light level of 1000 lux (bright light) is directed towards it, the resistance is 400R (ohms).

When a light level of 10 lux (very low light level) is directed towards it, the resistance has risen dramatically to 10.43M (10430000 ohms).



This is an example of a light sensor circuit:

When the light level is low the resistance of the LDR is high. This prevents current from flowing to the base of the transistors. Consequently, the LED does not light.

However, when light shines onto the LDR its resistance falls and current flows into the base of the first transistor and then the second transistor. The LED lights.

The present resistor can be turned up or down to increase or decrease resistance, in this way it can make the circuit more or less sensitive.

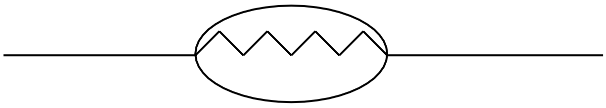


Figure 6.3.17 Schematic Symbol of LDR

The photo resistor, or Light Dependent Resistor, finds many uses as a low cost photo sensitive element and was used for many years in photographic light meters as well as other applications. Such as flame, smoke, and burgler detectors, card readers and lighting controls for street lamps.

Units for the light intensity are Lux or Lumence.

Basic structure:

Although there are many ways in which LDR's or photo resistors can be manufactured, ther are naturally a few more common methods that are seen. Essentially the LDR or photo resistor consists of a resistive material sensitive to light that is exposed to light. The photo resistive element comprises section of material with contacts at either end. Although many of the material used for light dependent resistors are semiconductors, when used as photo resistors, they are used only as a resistive element and there are no p-n junctions. Accordingly the devices purely passive.

A typical structure for a Light Dependent Resistor uses an active semiconductor layer that is deposited on an insulating substrate. The semiconductor is normally lightly doped to enable it to have the required level of conductivity. Contacts then placed either side of the exposed area. In many instances the area between the contacts is in the form of zig zag, or inter digital pattern. This maximizes the exposed area and by keeping the distance between the contacts small it enhances the gain.

It also possible to use a poly crystalline semiconductor that is deposited onto a substrate such as ceramic. This makes for a very low cost light dependent resistor.

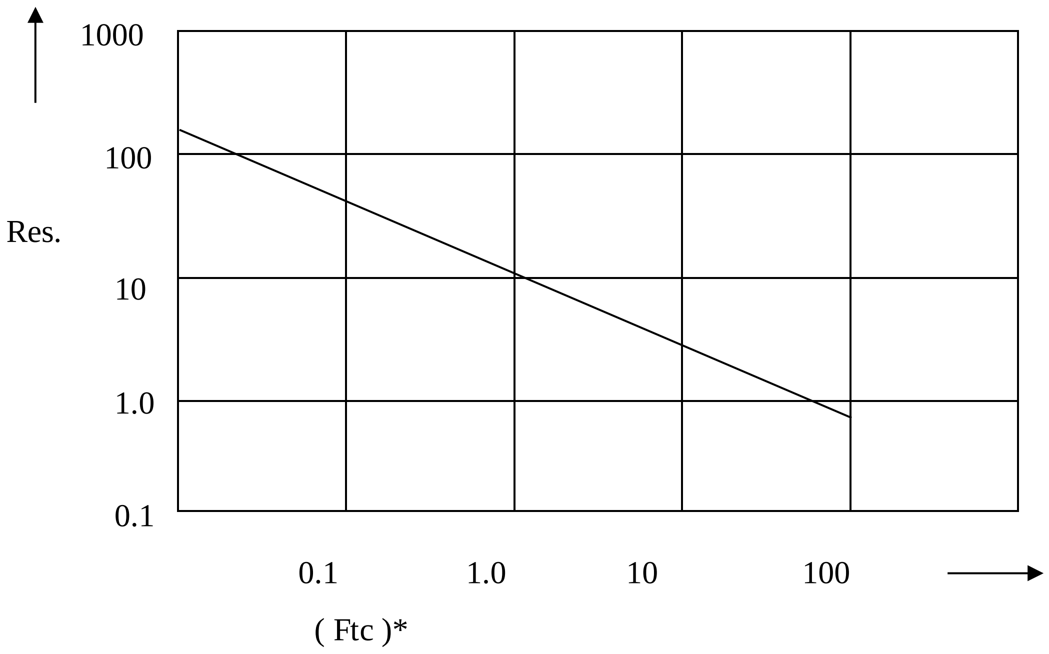
Operation:

Light Dependent Resistor made of a high resistance semiconductor, if light falling on the is of high enough efficiently, photon absorbed by the semiconductor give bound electrons enough energy to jump into the conduction band. The resulting free electron (and its hole partner) conduct electricity, thereby lowering resistance.

In intrinsic devices, the only available electrons are in the valence band, and hence the photon must have enough energy to excite the electrons across the entire band gap. Extrinsic devices have impurities added, which have a ground state energy closer to the conduction band, since the electrons don't have so far to jump, lower energy photons (i.e. longer wavelengths and lower frequencies) will suffice to trigger the device.

Characteristics of LDR:

The characteristics of LDR are shown below. Here the resistance variations are shown as a function of illumination. The resistance of LDR decreases with increasing incident light



*1 Ftc = 10.764 lumens

Figure 6.3.18 Characteristics of LDR

LDR Applications:

LDR’s are very useful especially in light/dark sensor circuits. Normally the resistance of LDR is very high, sometimes as high as 1000k ohms, but when they are illuminated with light, resistance drops immediately.

1. Camera light meters.
2. Clock radios.
3. Security alarms.
4. Optical switches.
5. Far infrared detector.
6. Streetlights.

Testing circuit of LDR:

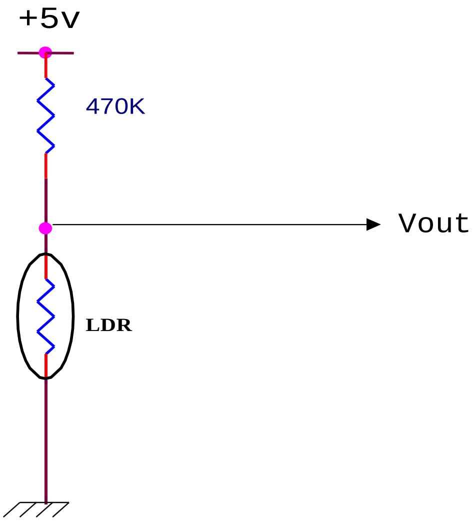


Figure 6.4.1 Testing Circuit of LDR

Voltage Monitoring Circuit

Transformer failures have many causes and one of the main causes is over voltage. The primary of the distribution transformer or any other transformer primary is designed to operate at certain specific voltage, if that voltage is more than the rated voltage, then immediately the transformer primary may burn because of over voltage. To protect the transformer, burning due to over voltage, this voltage monitoring circuit is used in this project work.

In this project work for generating high voltage, autotransformer is used so that the line voltage can be increased to more than 240V. For monitoring the line voltage, a step-down transformer of 3V secondary is used as a line voltage sensor. As this transformer primary voltage increases, according to that secondary voltage also raises, and this secondary voltage is rectified, filtered and it is applied to the analog to digital converter for converting the analog information in to the digital information.

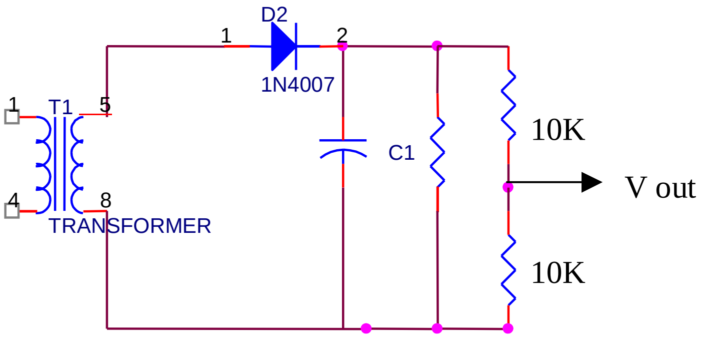


Figure 6.4.2 Schematic of Voltage Monitoring Circuit

CHAPTER 7
SOFTWARE EXPLINATION

Software Explanation:

Software Requirements

- Proteus simulation
- Arduino software
- Programming language

Arduino software:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals
- has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low-cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step-by-step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers c an make their own

version

- Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money
- Getting Started with Arduino and Genuino products:-
- Install the Arduino Software (IDE) on Windows PCs-

This document explains how to install the Arduino Software (IDE) on Windows machines.

- ✓ Download the Arduino Software (IDE)
- ✓ Proceed with board specific instructions.

How to Download the Arduino Software (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

Installation:

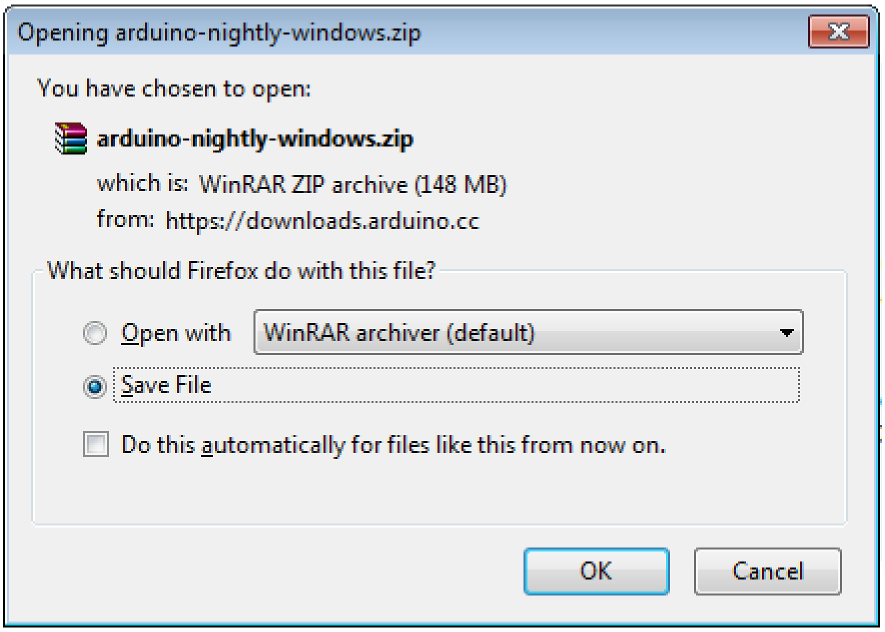
In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



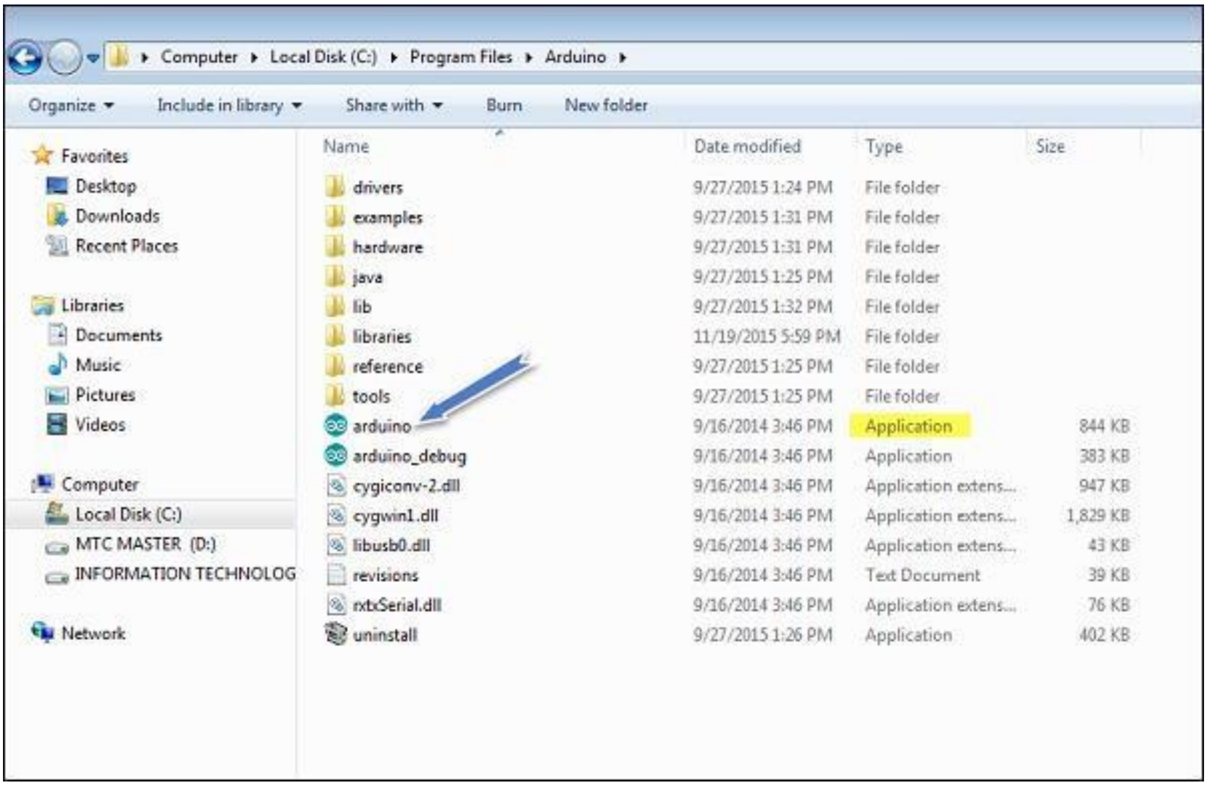
Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

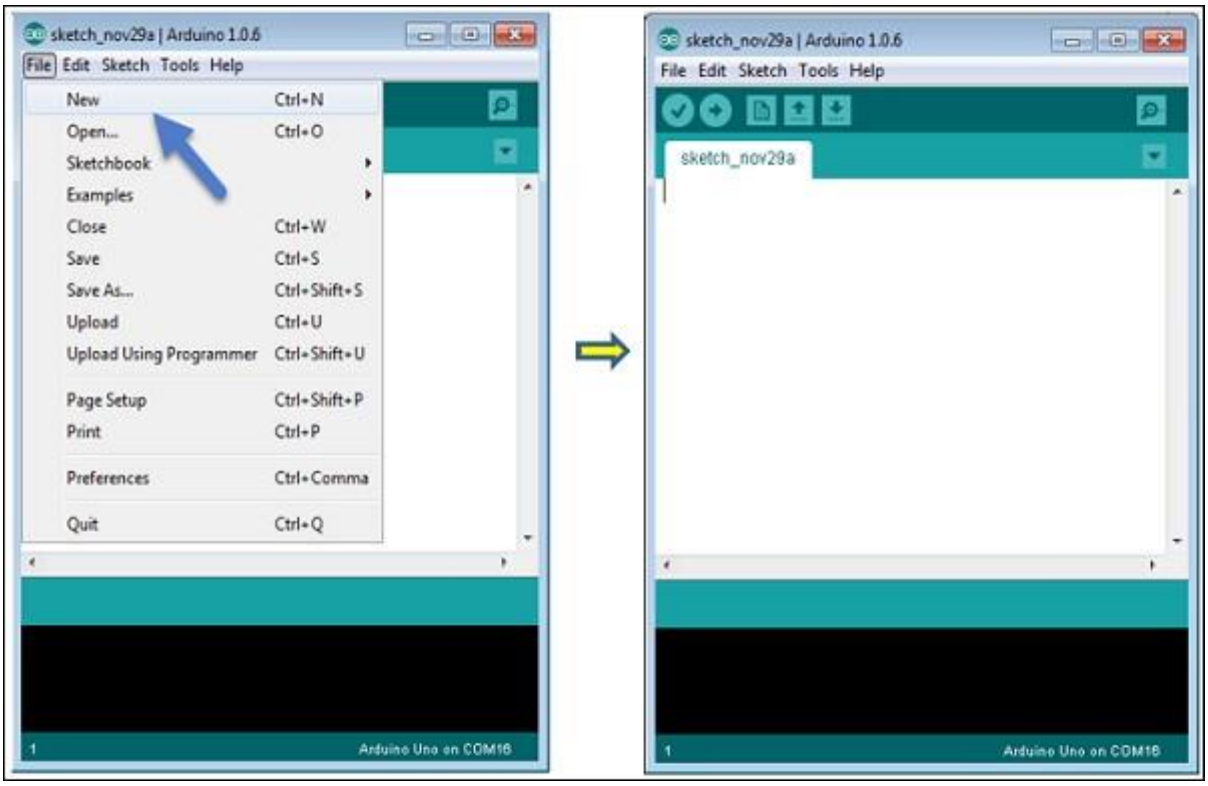


Step 5 − Open your first project.

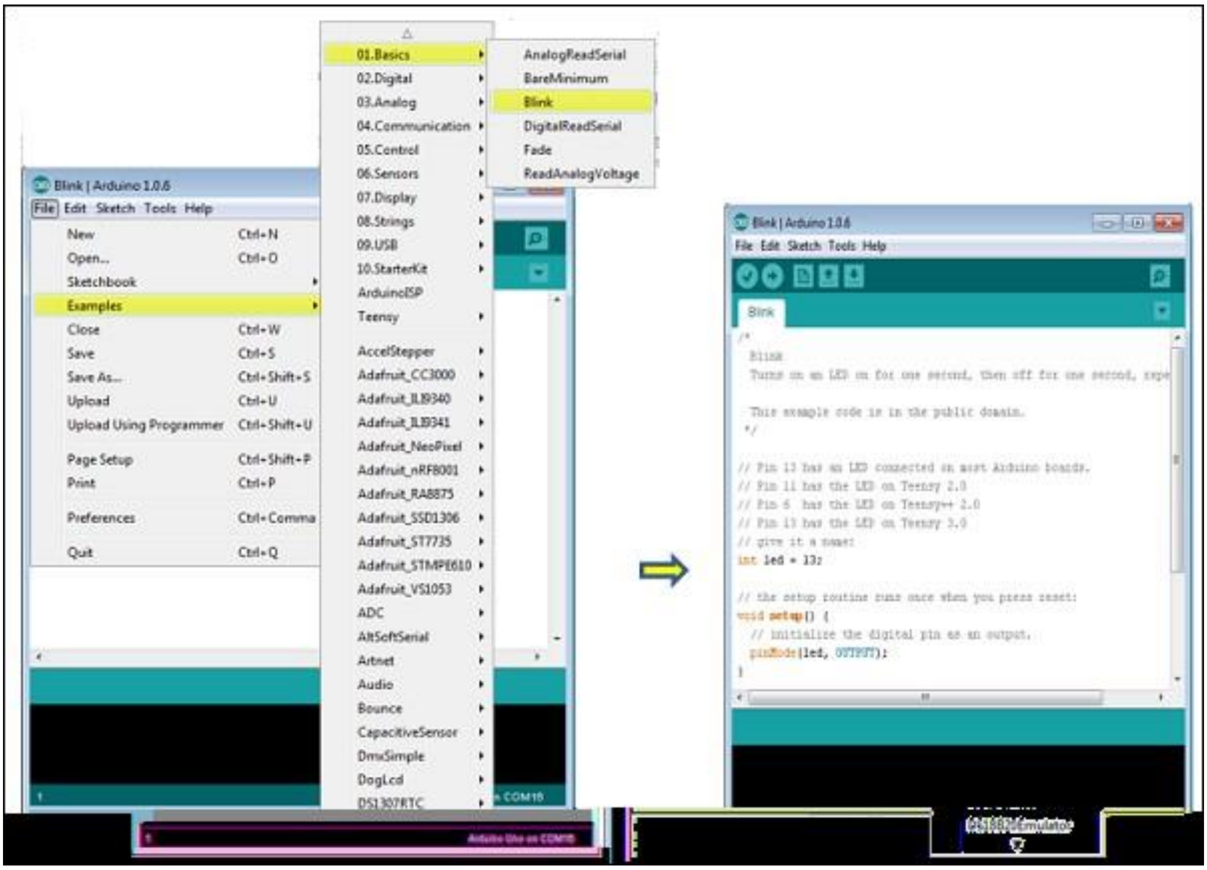
Once the software starts, you have two options

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.



To open an existing project example, select File → Example → Basics → Blink.

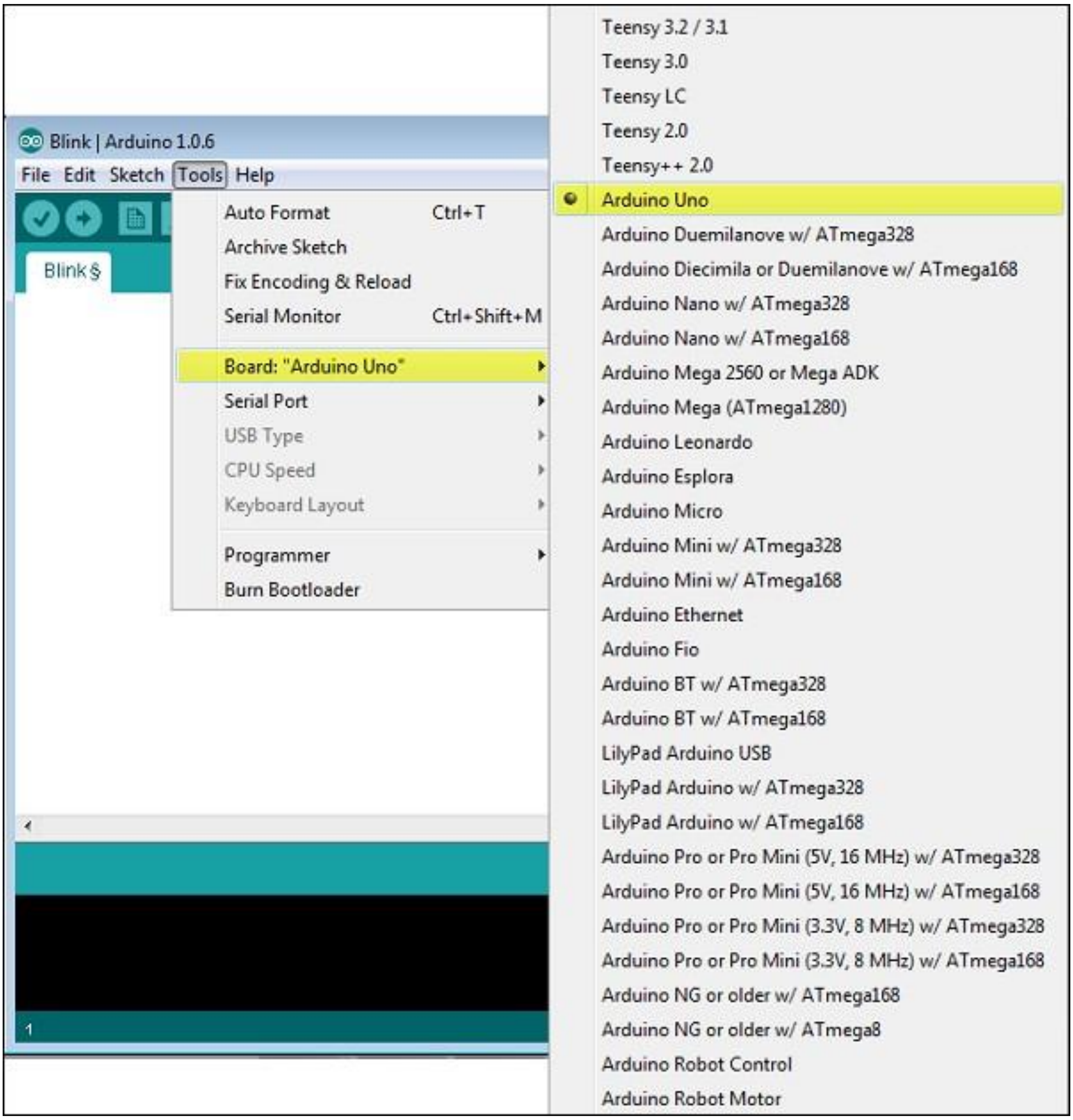


Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

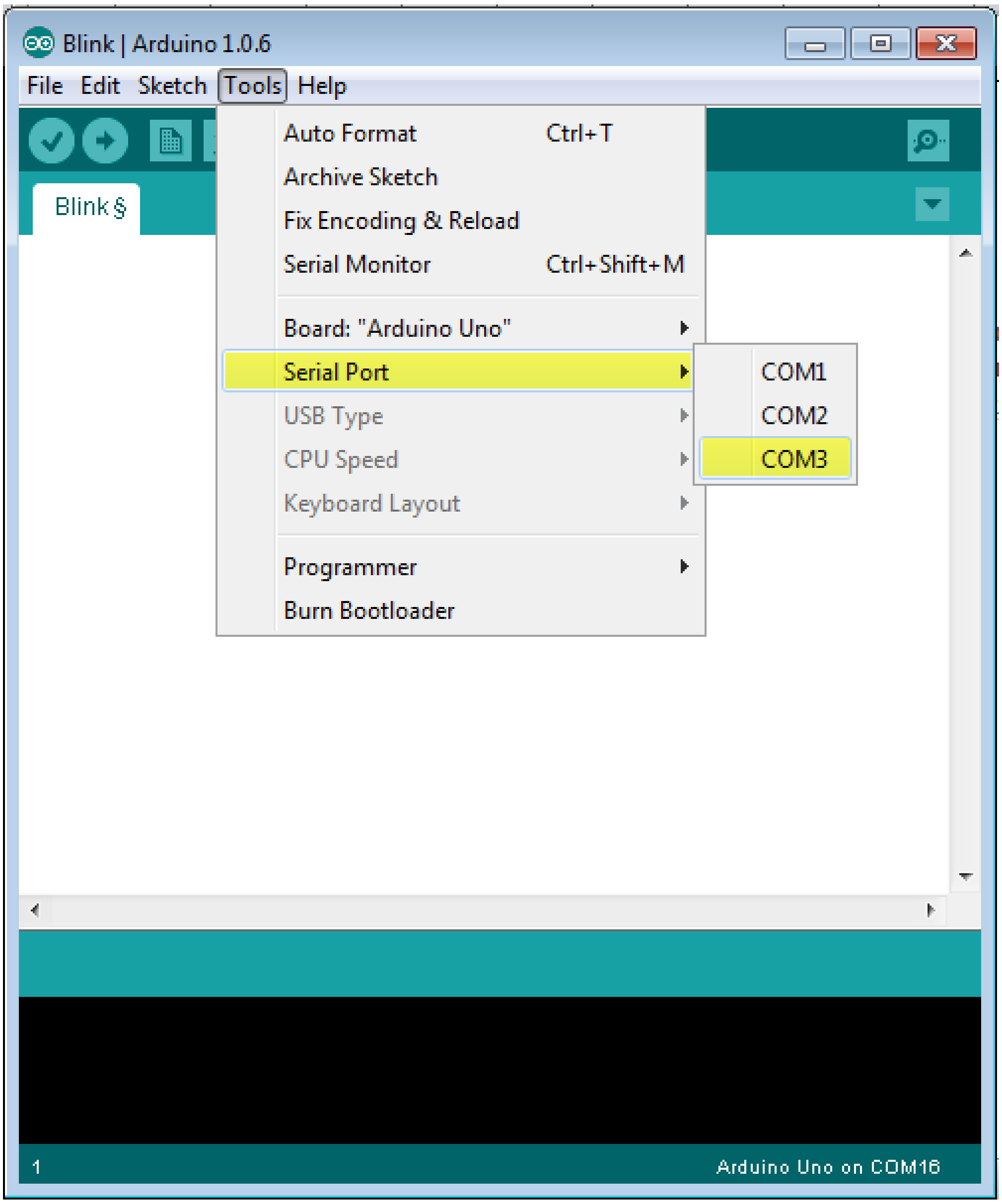


Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

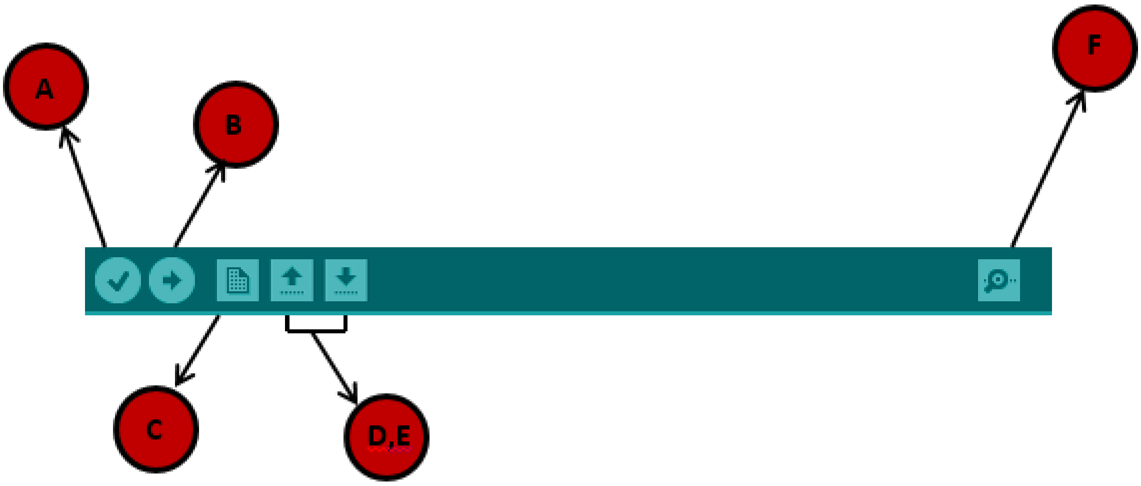
Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports).

To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.



Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.



A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketches.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24 V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

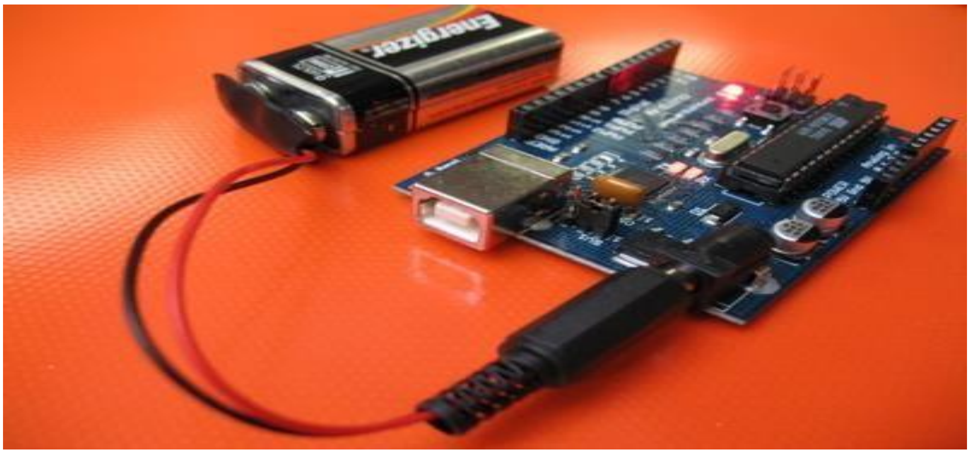


Figure 7.1.1.1 Arduino with battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called “sketches”, but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

MC Programming Language: Embedded C

This is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements. Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

Proteus:

Proteus:

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline "From concept to completion".

About Proteus:

It is a software suite containing schematic, simulation as well as PCB designing.

ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.

The designer can also develop 2D drawings for the product.

Features:

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design

Step 1: Open ISIS software and select new design in File menu

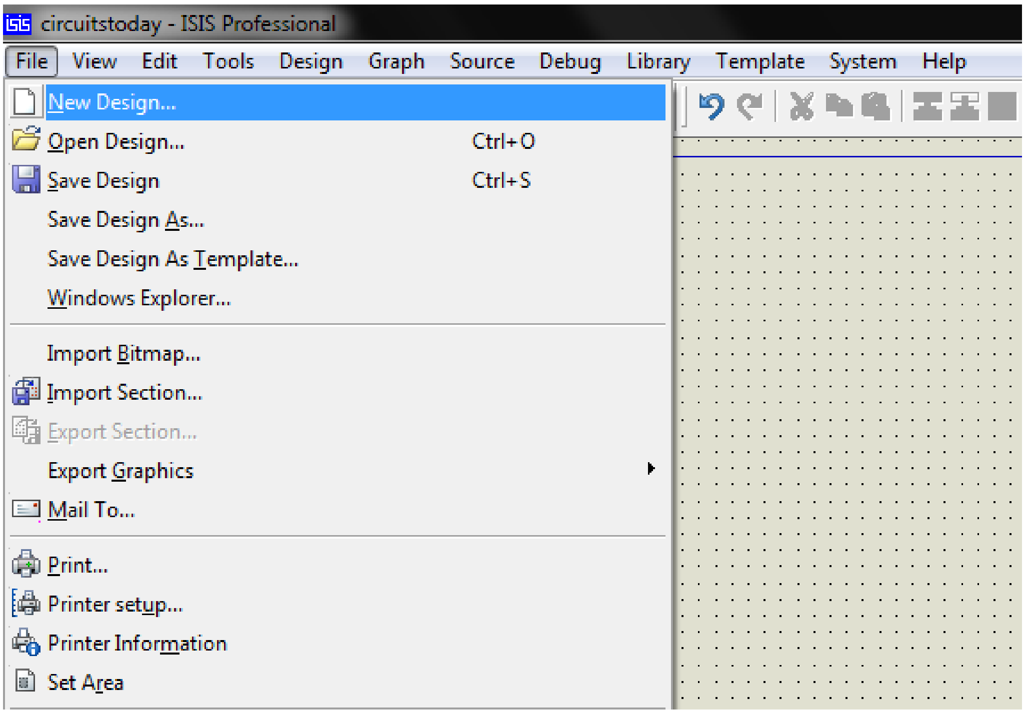


Figure 7.3.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

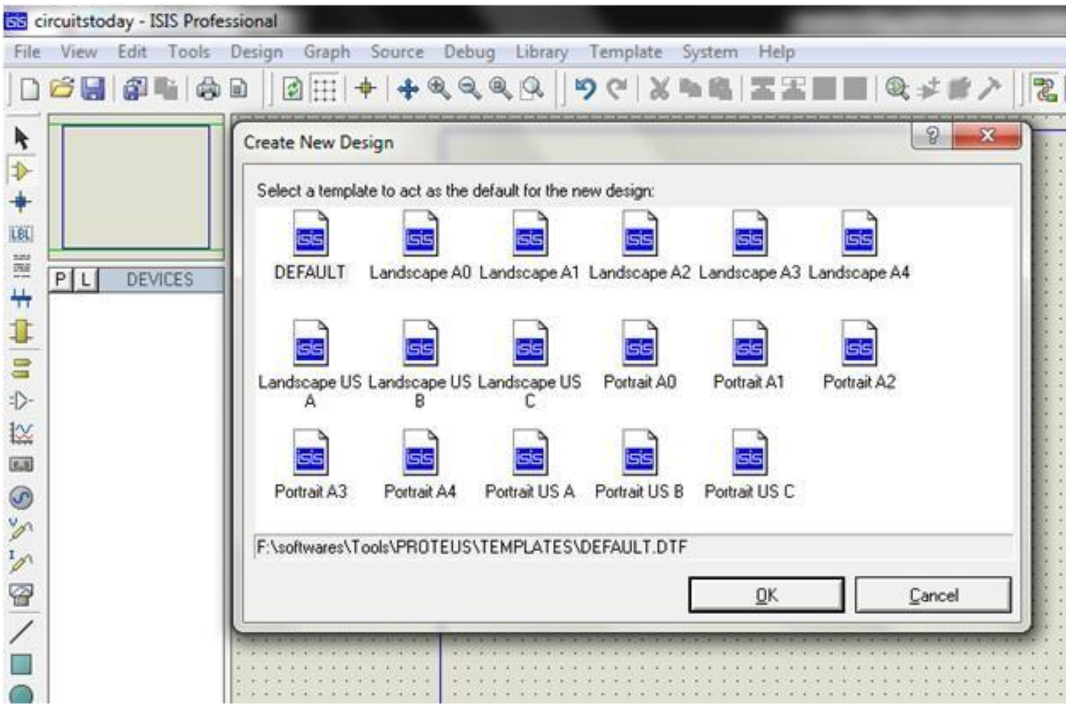


Figure 7.3.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

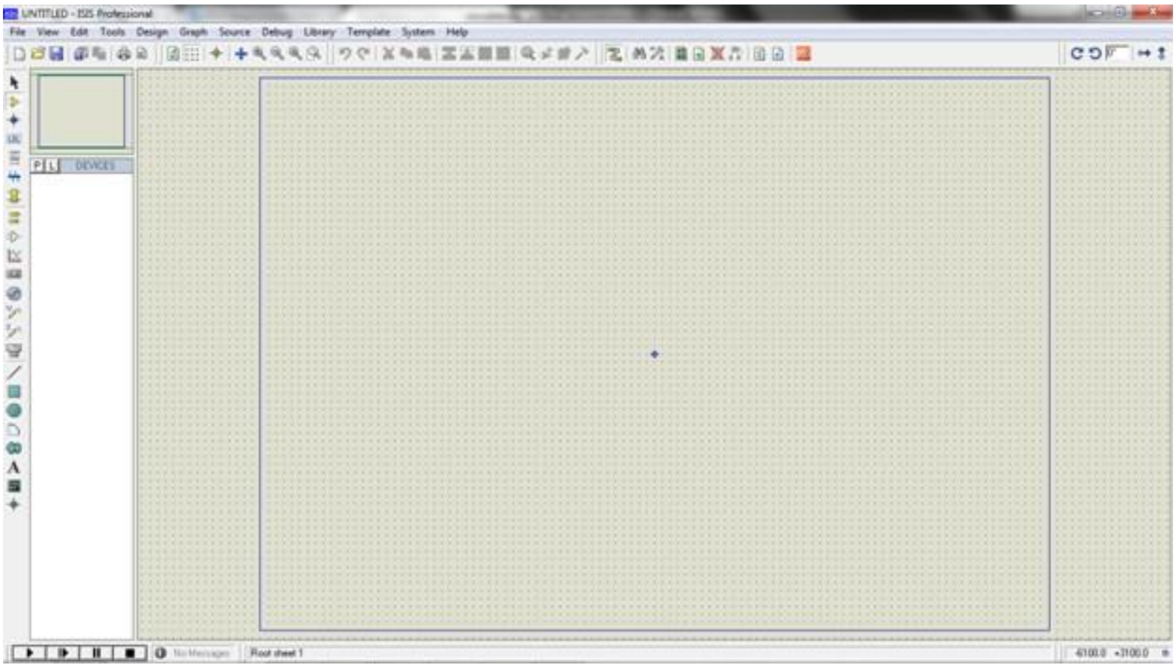


Figure 7.3.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

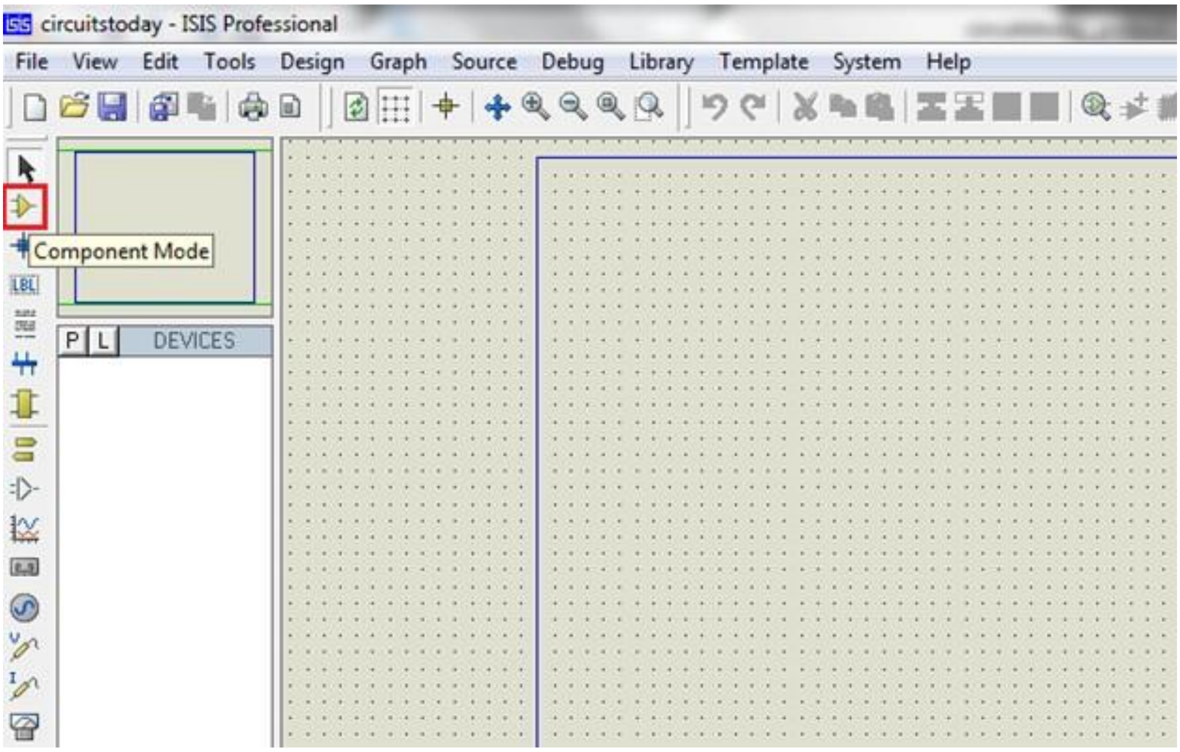


Figure 7.3.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

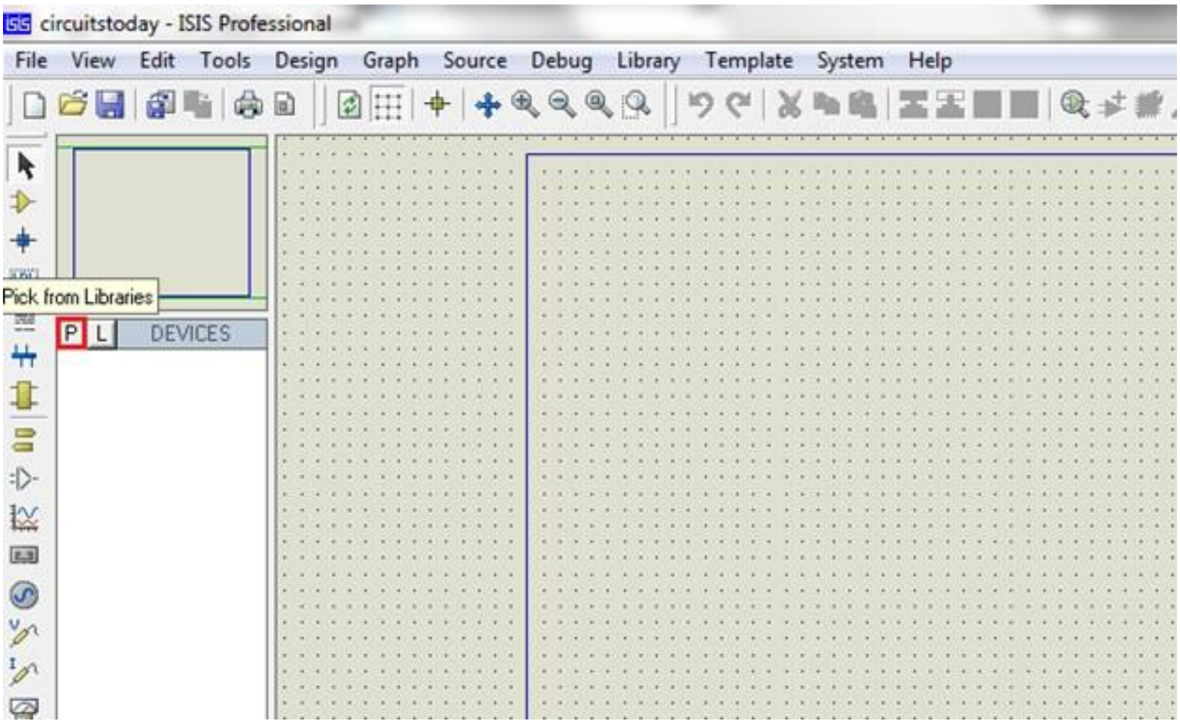


Figure 7.3.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keyword's text box.

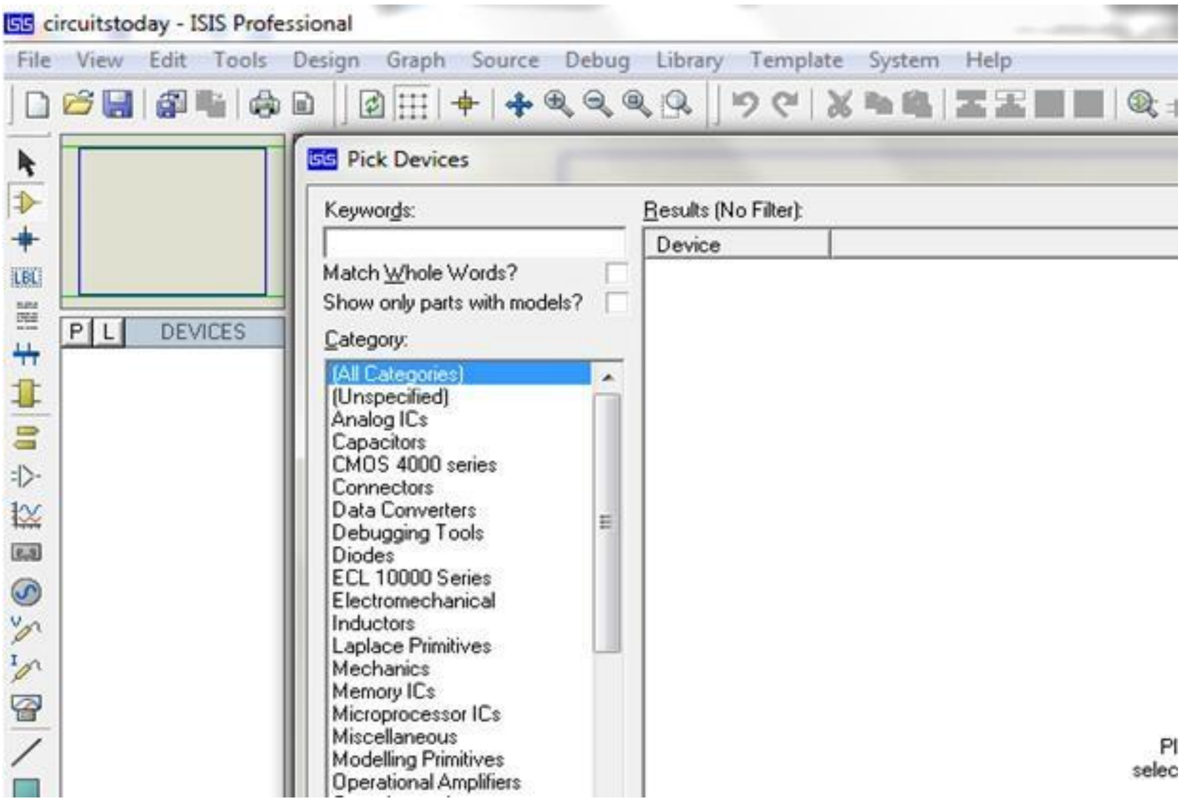


Figure 7.3.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

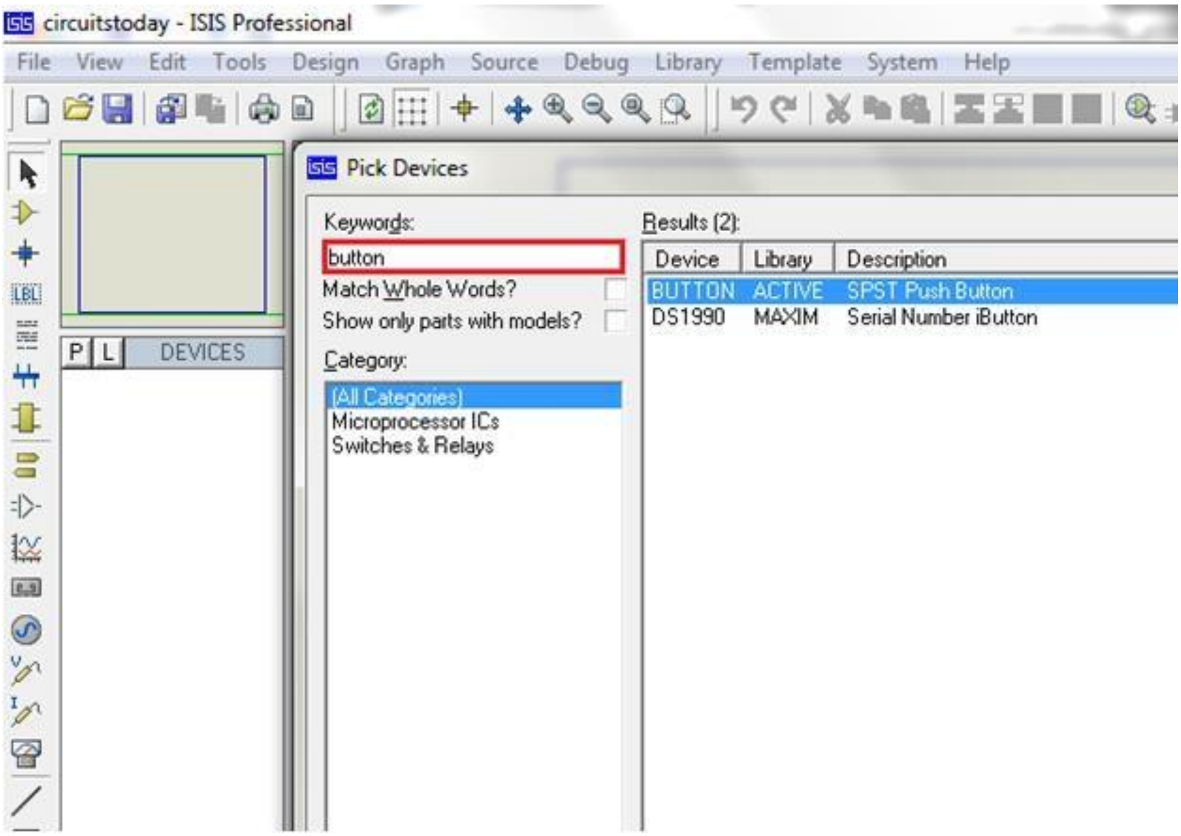


Figure 7.3.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

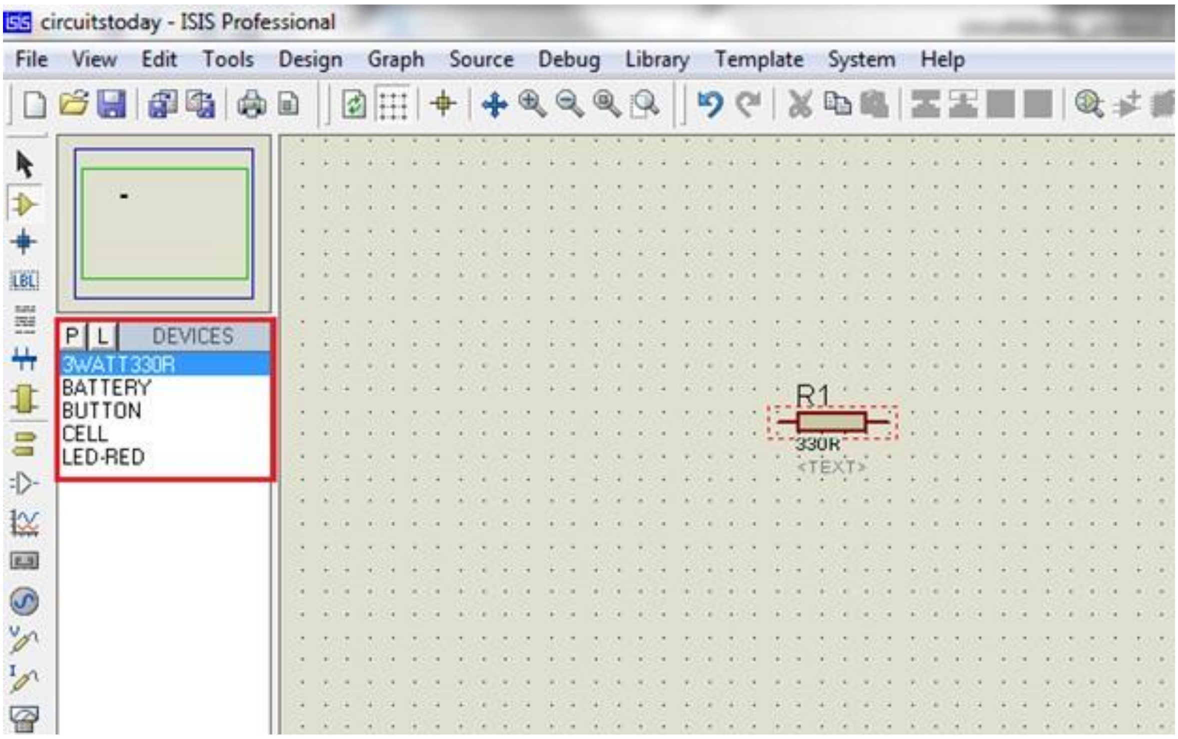


Figure 7.3.8 Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

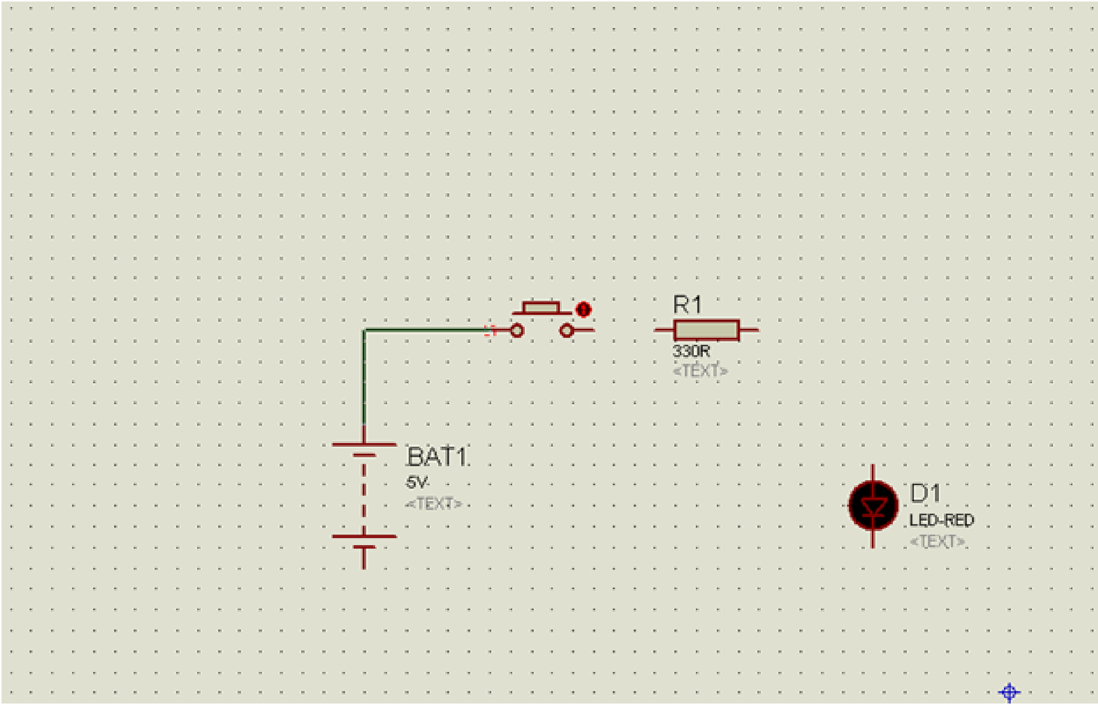


Figure 7.3.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

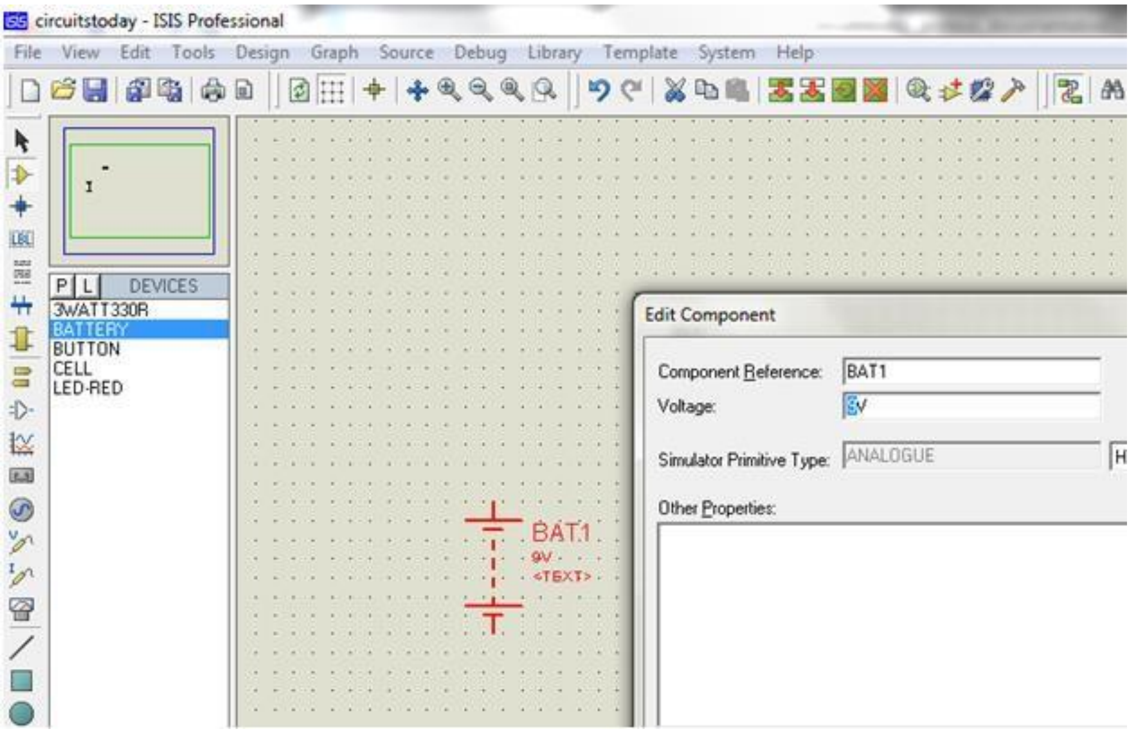


Figure 7.3.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

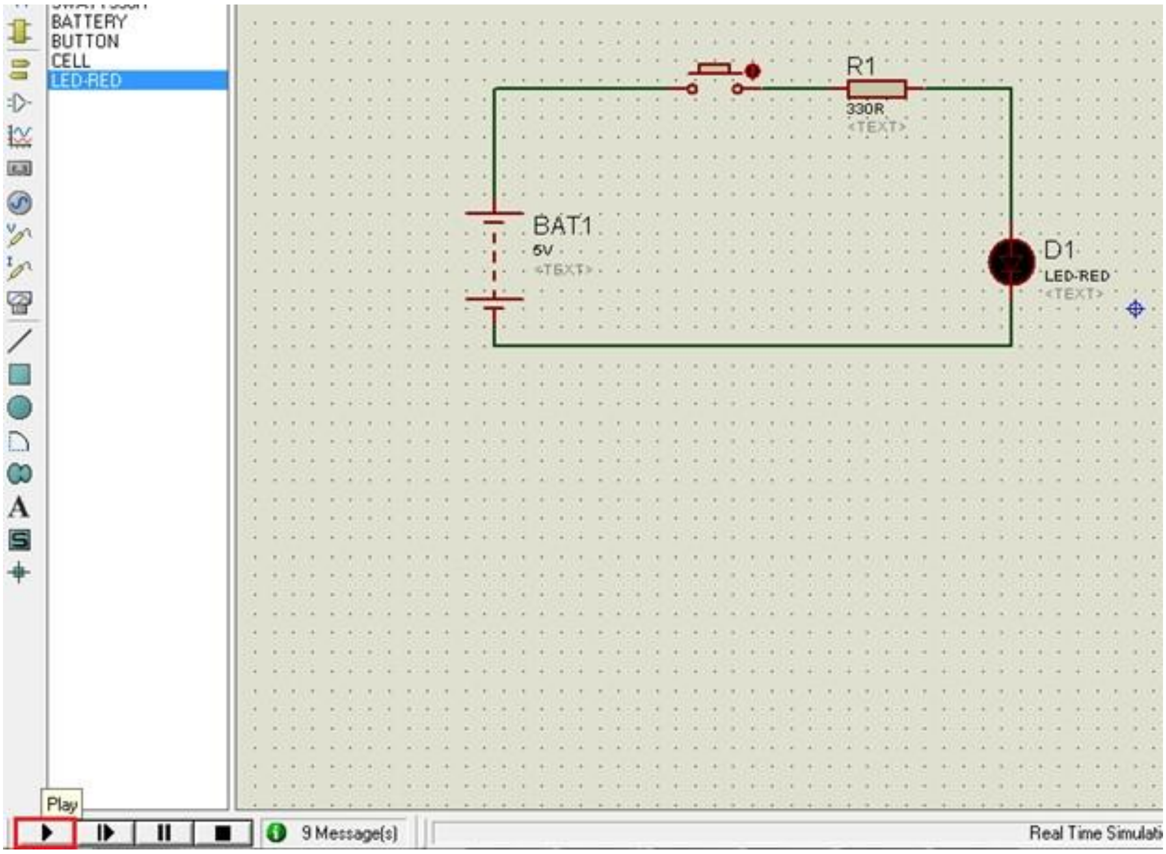


Figure 7.3.12 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

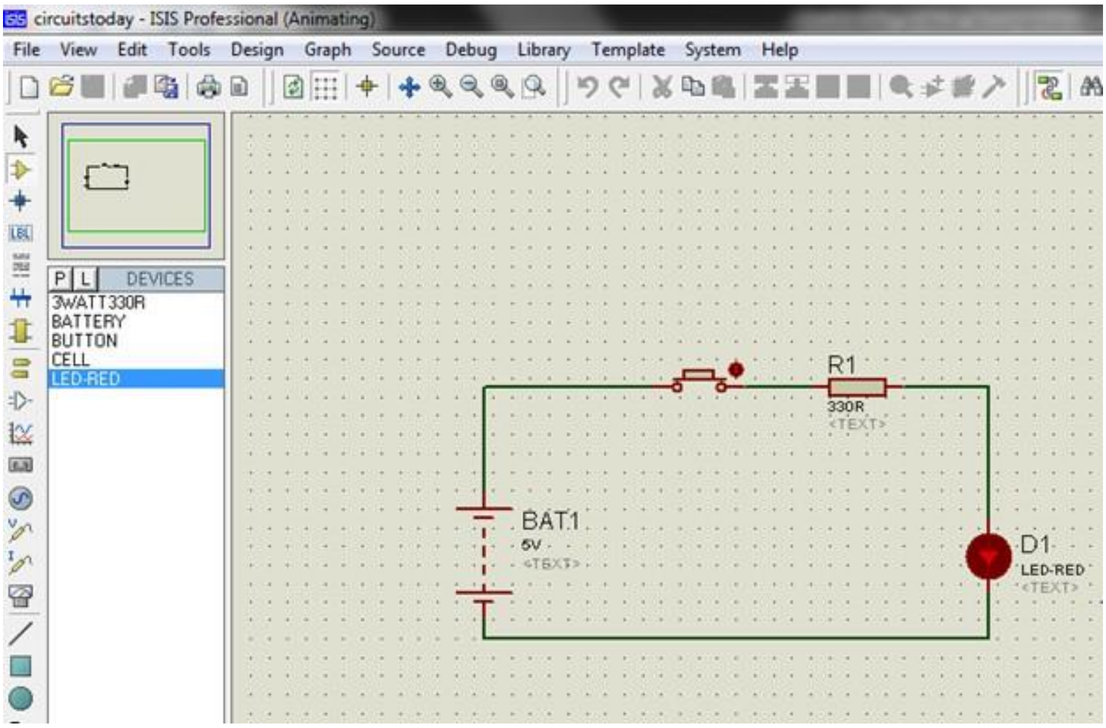


Figure 7.3.13 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

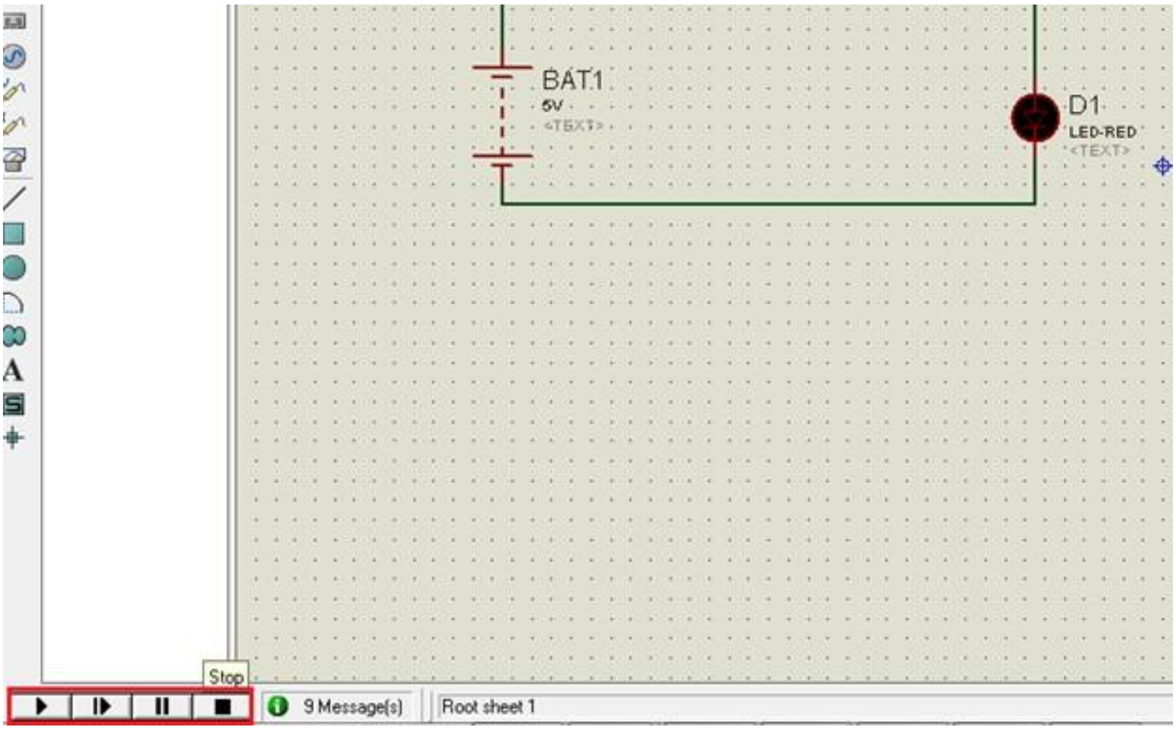


Figure 7.3.14 Simulation Step-Pause-Stop Buttons

Project Code:

```
#define TempPin A4
int TempValue;
#include<LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);

int ldr=A3;

const int voltageSensor = A5;
float vOUT = 0.0;
float vIN = 0.0;
float R1 = 30000.0;
float R2 = 7500.0;
int value = 0;
unsigned int count=0;
void setup()
{
  lcd.begin(16,2);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Solar Energy");
  lcd.setCursor(0,1);
  lcd.print("Tracking System");
  pinMode(ldr,INPUT);
  delay(3000);
}
int ldr_s;
void loop()
{
  ldr_s=analogRead(ldr);

  TempValue = analogRead(TempPin); // Getting LM35 value and saving it in variable
  float TempCel = ( TempValue/1024.0)*500; // Getting the celsius value from 10 bit analog
value
  float TempFarh = (TempCel*9)/5 + 32; // Converting Celsius into Fahrenhiet
```

```
  value = analogRead(voltageSensor);
  vOUT = (value * 5.0) / 1024.0;
  vIN = vOUT / (R2/(R1+R2));
  lcd.clear();lcd.setCursor(0,0);
  lcd.print("Vtg=");lcd.print(vIN);
  lcd.setCursor(10,0);lcd.print("L:");lcd.print(ldr_s);
  lcd.setCursor(0,1);lcd.print("Temp:");lcd.print(TempCel);lcd.print((char)223);
  lcd.print("C");
  delay(1000);
}
```

CHAPTER 8

RESULT & CONCLUSION

Application of The Project:

- Used in industries.
- It is used solar power stations.
- Used in house hold solar inverters.

Advantages of The Project:

- Renewable Energy Source.
- Reduces Electricity Bills.
- Diverse Applications.
- Low Maintenance Costs.
- Technology Development.

Result:

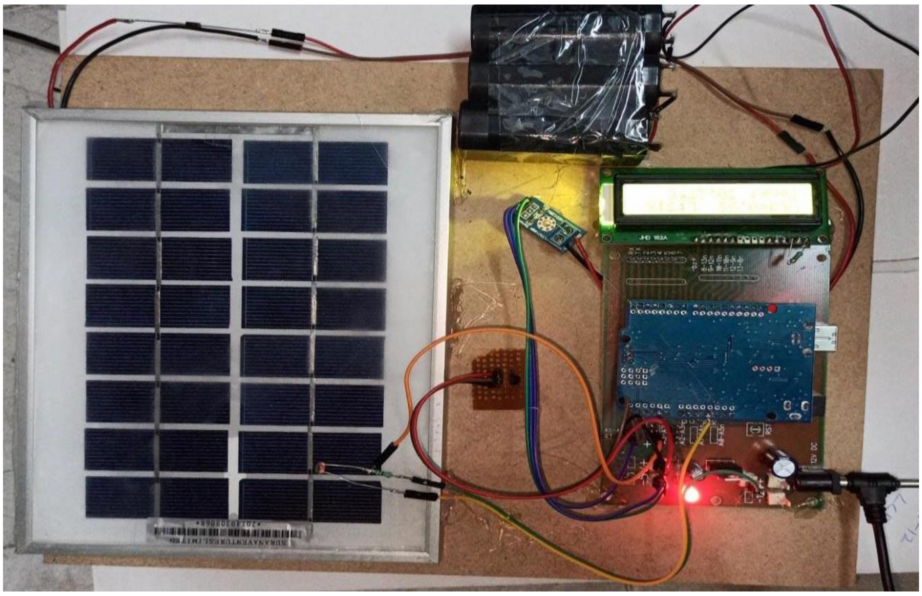


Figure 8.3.1 Prototype of the Project

Figure 8.3.1 Output 1

Initially when the Arduino board is turned on, the following message will be appeared as shown in figure 8.3.1.

Figure 8.3.2 Output 2

After measuring all the solar parameters, the values are displayed accordingly as shown in figure 8.3.2.

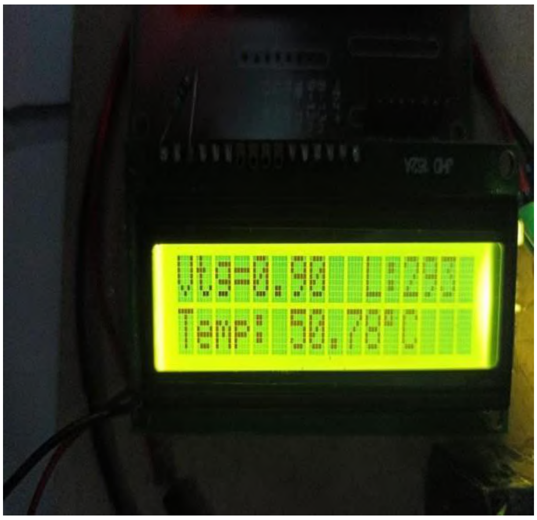


Figure 8.3.3 Output 3

The solar panel parameters will change for every 30 seconds. The varied values are shown in the figure 8.3.3.

Conclusion:

The proposed system stores the data from the solar photovoltaic system continuously, so it keeps track of the solar photovoltaic system and daily or monthly analysis becomes easy and efficient.

Using the analysis, it is possible to detect any fault occurring in the system as there would be inconsistency in the data generated by the system.

By solar tracking the solar panel is operated at its maximum efficiency all day.

Future Scope:

The controller requires an external supply to work but using the power generated from solar panel itself the controller's input power supply can be met.

For very large solar panel dual axis solar panel tracking can be done.

By analyzing the data, it is possible to predict the future values of parameters.

Artificial intelligence can be implemented using various machine learning algorithms that the system can become smart enough to take decisions about data and performance.

CHAPTER 9

REFERENCE

1. M.R. Al Rashidi, M.F. Al Hajri, K.M. ElNaggar, A.K. Al-Othman, “A new estimation approach for determining the I–V characteristics of solar cells”, Electrical Engineering Department, College of Technological Studies (PAAET), Shuwaikh, Kuwait, (2011).

2. V. Ryan, what is solar energy? (2016, September 20) Retrieved from http://www.technologystudent.com/energy1/sol ar1.htm.

3. John Balfour, “Introduction to Photovoltaic”, United States of America, (2013).

4. Arindam Bose, Sounak Sarkar, Sayan Das, “Helianthus-a Low Cost High Efficient Solar Tracking System Using AVR Microcontroller”, International Journal of Scientific & Engineering Research, Volume 3, Issue 10, October (2012).

5. Mohsen Taherbaneh, A. H. Rezaie, H. Ghafoorifard, K. Rahimi and M. B. Menhaj, “Maximizing Output Power of a Solar Panelvia Combination of Sun Tracking and Maximum Power Point Tracking by Fuzzy Controllers”, Hindawi Publishing Corporation, International Journal of Photoenergy, Volume 2010, (2010).

6. Yi-Hua Liu, Jia-Wei Huang, “ A fast and low cost analog maximum power point tracking method for a low power photovoltaic systems”, Department of Electrical Engineering, National Taiwan University of Science and Technology, 13 September (2011).

7. R. Ramaprabha, M. Balaji, B. L. Mathur. “ Maximum power point tracking of partially shaded solar PV system using modified Fibonacci search method with fuzzy controller”, Department of EEE, SSN Collage of Engineering Chenmai, India, 10 July (2012).

8. Mohammad H. Moradi, Ali Reza Reisi, “A hybrid maximum power point tracking method for photovoltaic systems”, Department of Electrical Engineering, Faculty of Engineering, Bu Ali Sina University, Hamedan, Iran, 14 September (2011).

9. Mybotic (2010) Solar cell 12V 250mA (2016, November 20), retrieve from http://www.mybotic.com.my/webshaper/store/v iewProd.asp?pkProductItem=831

10. Chua Hock-Chuan (2015) Getting Started with Arduino (2016, October 30), retrieve from https://www.ntu.edu.sg/home/ehchua/program ming/arduino/Arduino.html

11. Electrical 4U.com (2011) Light Dependent Resistor / LDR and working Principle of LDR (2016, November 2) retrieve from http://www.electrical4u.com/light-dependentresistor-ldr-working-principle-of-ldr/

12. Taxes Instrument (2016) LM35 precision centigrade temperature sensors (2016, November 4) retrieve from http://www.ti.com/lit/ds/symlink/lm35.pdf

13. SparkFun (2015) Voltage Divider (2016, November 6) retrieve from https://learn.sparkfun.com/tutorials/voltagedividers

14. Engineer Garage (2012) LCD (2016, November 7) retrieve from http://www.engineersgarage.com/electroniccomponents/16x2-lcd-module-datasheet

15. Wikipedia (2016) Proteus Design Suite (2016, November 21) retrieve from https://en.wikipedia.org/wiki/Proteus_Design_S uite

16. Arduino (2016) Overview of the Arduino, (2016, November 21) retrieve from https://www.arduino.cc/en/Main/ArduinoBoard Uno

17. Mohd Rashid B. Mohd Kamlon, “Solar Power Measurement System”, Universiti Teknikal Melaka Malaysia, (2007).

18. Ganiyu R. A., Arulogun O. T. and Okediran O. O, (2014), Development of A MicrocontrollerBased Traffic Light System for Road Intersection Control, Journal of Scientific and Technology Research, vol. 3, p.1-4.

19. Vulnerable space blogspot (2016) Arduino Welcome to World, (2016, December 14) retrieve from http://vulnerablespace.blogspot.my/2016/01/ard uino-welcome-to-world-of.html.

20. Wikipedia (2016) Current sensor module 5A, (2016, December 22) retrieve from https://www.elecrow.com/wiki/index.php?title= ACS712_Current_Sensor-_5A.

21. Siti Amely Jumaat, Firdaus Mohammad, Shamsul Aizam Zulkifli, Development of Portable Case Solar Battery Charger, Electricial and Electronics Engineering 2016, 6(4) : 55-61, DOI : 1-.59923/j.eee.20160604.01 (2016).

22. Idris. I, Robian. M.S, Mahamad. A.K, Saon. S, 'Arduino based maximum power point tracking for photovoltaic system', APRN Journal of Engineering and Applied Sciences, Volume 11, Issues 14, (2016), pp 8805-8809.

23. Zulkifli. S.A, Hussin M.N, Saad A.S, 'MATLAB-Arduino as a low-cost microcontroller for 3-phase inverter', 2014 IEEE Student Conference on Research and Development, SCOReD 2014.

A

PROJECT REPORT
ON

LIQUID LEVEL MONITORING SYSTEM USING IOT WITH ULTRSONIC SENSOR

Submitted by

D. Sai Nikitha	**(17K81A0475)**
K. Sitha Ram Reddy	**(17K81A0481)**
A. Naga Vamshi Reddy	**(17K81A0463)**

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION

Under The Guidance Of

Ms.E.Parvathi

ASSISTANT PROFESSOR

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

ST. MARTIN’S ENGINEERING COLLEGE

(An Autonomus Institute)

Dhulapally, Secunderabad-500 100.

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the project entitled **LIQUID LEVEL MONITORING SYSTEM USING IOT WITH ULTRASONIC SENSOR**, is being submitted by1.**D.Sai Nikitha (17K81A0475), 2. K. Sitha Ram Reddy (17K81A0481), 3. A. Naga Vamshi Reddy (17K81A0463)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OFTECHNOLOGY IN ELECTRONICS AND COMMUNICATION** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Signature	Head of the Department
Ms.E.Parvathi	Dr.B.HariKrishna
Department of ECE	Department of ECE

Internal Examiner	External Examiner

Place:

Date:

TUESDAY, 15 June 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **A. NAGA VAMSHI REDDY** WITH ROLL NO.17K81A0463**, D. SAI NIKITHA** WITH ROLL NO.17K81A0475**, K. SITHA RAM REDDY** WITH ROLL NO.17K81A0481, OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD H AVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "LIQUID LEVEL MONOTORING WITH IOT SYSTEM USING ULTRASONIC SENSOR " AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAINING & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad50014
Email:contact@lasyainfotech.com,ov@lasyainfotech.com
Website :www.lasyainfotech.com | **contact: 7330666881/82/83/84/86**

DECLARATION

We, the student of **Bachelor of Technology** in Department of ‘Electronics and Communication’, session: 2017 – 2021, St. Martin’s Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled LIQUID LEVEL MONITORING SYSTEM USING IOT WITH ULTRASONIC SENSOR is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

1.D. Sai Nikitha (17K81A0475)

2.K.Sitha Ram Reddy (17K81A0481)

3.A.Naga Vamshi Reddy (17K81A0463)

ACKNOWLEDGEMENT

While bringing out this project report to its final form, we came across a number of people whose contributions in various ways helped my field of research and they deserve special thanks. It is a pleasure to convey our gratitude to all of them.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B.HariKrishna**, **Head &Professor**, the Dept. of Electronics and Communication for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our Dept .Project Coordinator Dr.A.Anand Kumar** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide **Ms.E.Parvathi, Assistant Professor**, Department of ECE and our class project coordinator **Mr.Karthik, Assistant Professor,** Department of ECE for guidance and encouragement in carrying out this project work.

Finally,We are highly grateful and obliged to our **college level Project Coordinators Dr.Malayadri**, **Professor in IT** and **Dr.R.Santosh Kumar** , **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words..

1. D. Sai Nikitha
2. K. Sitha Ram Reddy
3. A. Naga Vamshi Reddy

ABSTRACT

Liquid Level Monitoring System Using IOT With Ultrasonic Sensor is a very innovative system which will inform the users about the level of liquid and will prevent it from overflowing. For this the system uses ultrasonic sensors placed over the containers to detect the liquid level and compare it with the container's depth. The system makes use of AVR family microcontroller, LCD screen, WIFI modem for sending data and a Ultrasonic sensor. The LCD screen is used to display the status of the level of liquid in the containers. Whereas an app is built to show the status to the user monitoring it. The app gives the level or percentage view of the containers and highlights the liquid level order to show the level of liquid. The LCD screen shows the status of the liquid level. Thus this system helps to prevent the wastage of water by informing about the liquid levels of the containers by providing message to the user through app(Telnet).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

Overview Of The Project:

The present invention relates to the field in Liquid Level Monitoring. The goal of this research is to develop a smart system for estimating accurately the liquid level and prevent overflowing. The recent advancements in the area of IOT has provided solutions to measure the quantity and level of liquid in liquid transmission system using sensor based wireless technology. Our research team has developed a smart system that challenges the current system and provides the continuous monitoring of liquid level to the user.

This project IOT Liquid Level Monitoring system is a very innovative system which will inform the users about the level of liquid and will prevent it from overflowing. To demonstrate this the system makes use of ultrasonic sensors placed over the containers to detect the liquid level and compare it with the container's depth. The system makes use of AVR family microcontroller, LCD screen, Wifi modem for sending data. The system is powered by a 12V transformer. The LCD screen is used to display the status of the level of liquid in the containers. Whereas an app(TELNET) is built to show the status to the user monitoring it. The app gives a percentage or level view of the containers and highlights the liquid level in percentage in order to show the level of liquid. The LCD screen shows the status of the liquid level.. Thus this system helps to prevent the wastage of water by informing about the liquid levels of the containers by providing information to TELNET app.

The system proposes an IOT-enabled liquid monitoring system for all users that are connected to system. The liquid monitoring system monitors the level of the liquid by detecting the amount of liquid in the tank during the transfer of the liquid such as water, oil, milk etc.

Earlier in olden days a physical person has to stay beside the tank in order to monitor whether the tank is filled with water completely or not. The person might waste his precious time in order to do this. When it comes to petrol tanks and diesel tanks, the person need to climb the tank and check whether the tank is filled with petrol/diesel or not. When it comes to chemical industries the person need to climb the top of the tank by ladder which is at great distance and need to monitor physically. Even though a person is physically observed still there might be overflowing of liquid due to negligence of the person. In order to prevent all these we came up with an innovatie idea which is called liquid level monitoring using IOT.

IOT based Liquid level Monitoring project is designed in a way that it can be used to remotely monitor the level of liquid indoors and out, like measuring tanks, wells, ponds, canals, and streams. This system tells (or) informs user about the level of liquid and prevent it from overflowing.IOT based Liquid level monitoring project enables the user in level detection of any liquid and continuously monitor it – over the internet. The user can monitor continously in the app which is called TELNET about the status of liquid of container/tank.

OBJECTIVE OF THE STUDY:

The main aim of this system is to monitor the water level at rural areas so that they help in detecting the wastage of water and measures can be taken to avoid unnecessary overflowing of water in the areas where monitoring is a difficult task and the water can be saved for future generations.

One of the major problems faced by most of the countries is the issue of water scarcity in the world and wastage during transmission has been identified as a major culprit; this is one of the motivations for this research, to deploy computing techniques in creating a barrier to wastage in order to not only provide more financial gains and help the environment as well as the water cycle which in turn ensures that we save water for our future. IOT based Water Level Monitoring system is an innovative system which will inform the users about the level of liquid and will prevent it from overflowing.

Water Level Monitoring systems of course record the usage of water and the amount of liquid or water present in container or tanks..Thus this system helps to prevent the wastage of water by informing about the liquid levels of the containers to users through TELNET app.

SCOPE OF THE STUDY:

Technology nowadays has become an integrated part of people's lives. It has, and continues to influence many aspects of daily life and has allowed better social interaction, ease of transportation, the ability to indulge in entertainment and media and has helped in the development in medicine. The creation of many devices such as mobile phones and computers have caused many people to rely on technology to communicate with their friends, store information such as pictures, movies, documents, and music.

The internet has become a common interface that many devices use in order to simplify the daily life of many people giving the ability to search for information, store their own information in the cloud while also giving them better ways of managing information. From the time of its introduction, the number of people that use mobile phones and the internet to communicate with other people has increased dramatically to become one of the major means of communication

People with the help of smartphones can now connect to the internet withoutthe need for a computer, while still offering the same functionality but through different means. With the introduction of advanced software and hardware devices, smartphones are now powerful devices and have become an important part of people’s daily lives. A major aspect is how the Smartphone is able to connect and communicate with other devices.

A field that is recently gaining popularity is Water Level Monitoring which can also use smartphones as information or functionality hubs.

Im this system we use WIFI-module which acts as internet connectivity between the user and system. It acts as a channel for means of communication between user and system. The amount of liquid level data is send to user with the help of wifi module by using certain commands for transfer of data.

MATERIAL/COMPONENT REQUIREMENT:

HARDWARE REQUIREMENT:

- Arduino uno
- UltraSonic Sensor
- WIFI module
- Liquid Crystal Display

SOFTWARE REQUIREMENTS:

- ARDUINO IDE
- PROTEUS SOFTWARE

CHAPTER-2: LITERATURE SURVEY

1. Scarcity of water is one of the biggest issues revolving across the globe and water crisis is reaching the alarming level day by day. So, water conservation in one or the other way is gaining a significant importance. Mostly, now a days in urban as well as in rural areas water tank system is available. The biggest disadvantage of this system is the overflow of water from overhead tank and overrunning of water pump. Hence, in this work, it is tried to design an automatic water tank level and pump control system, which ensures several benefits. The sensor devices used in our system detects and controls the water level in the overhead tank and even in the pump. As per the level of water present in the overhead tank, the sensor senses the levels and sends different signals to the Arduino and the signals are used for switching ON and OFF the motor pump as per requirements. The Arduino will control the pump by preventing it from dry running. A buzzer is attached to the circuit, which will ring when the water reaches the critical level in the overhead tank.

2. The quality and quantity of water supplied by high-level water tank directly affect people's daily life. Most traditional water tanks use manual detection, which has high detection cost and low data accuracy. In order to ensure the safety and continuity of high-level water tank water supply, this paper designs a set of high-level water tank monitoring system based on Internet of things. The system adopts TDS sensor and ultrasonic sensor to monitor the water quality and water level of the water tank respectively, and transmits the data back to the monitoring center for real-time monitoring through ZigBee -a wireless communication protocol. The system can control water level automatically and alarm when water is polluted. It has strong practical value and significance that ensures the safety and stability of household water, and provides data support for the improvement of water tank water supply in the future.

3. This project IOT Liquid Level observance system could be a very innovative system that will inform the users regarding the amount of liquid and can stop it from overflowing. To demonstrate this system makes use of four containers. For this, the system uses ultrasonic sensors placed over the containers to observe the liquid level and compare it with the container’s depth. The system makes use of AVR family microcontroller, LCD screen, wireless local area network electronic equipment for causing knowledge and a buzzer. The system is steam-powered by a 12V electrical device. The liquid crystal display LCD digital display alphanumeric display screen is employed to display the standing of the amount of liquid within the containers. Whereas an online page is made to indicate the standing to the user observance it. The net page provides a graphical read of the containers and highlights the liquid level in color order to indicate the amount of the liquid. The liquid crystal display screen shows the standing of the liquid level. The system puts on the buzzer once the amount of liquid collected crosses the set limit. So this method helps to stop the wastage of water by informing regarding the liquid levels of the containers by providing a graphical image of the containers via an internet page.

4.This paper shows non-contact water level monitoring system implemented using LabVIEW and arduino. Water depth of the tank is measured by Ultrasonic sensor. Hence the water level present in the tank is known. Depending on the sensor reading LabVIEW program sends the data to arduino and through arduino board the pump is switched ON if the water level in the upper tank is low and pump is switched OFF if the water level is completely filled in the upper tank. The input to arduino board is from ultrasonic sensor. Arduino read the height and reports the water depth of the tank and the same is displayed in LabVIEW front panel. Advantage of this over other existing automatic system is it provides non-contact water level measurement using ultrasonic sensor, but existing automatic system using ss(stainless steel) sensor which is a contact type water level sensor and also these sensors quickly corroded by some chemicals. The same is present in LabVIEW using the graphical user interface for visualization, LabVIEW communicate with add on devices like arduino, ultrasonic sensor, pump through makerhub.

CHAPTER-3: INTRODUCTION ABOUT EMBEDDED SYSTEM

INTRODUCTION:

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.

The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes.The applications software on such processors is sometimes referred to as firmware.

The simplest devices consist of a single microprocessor (often called a "chip”), which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine..

BLOCK DIAGRAM OF EMBEDDED SYSTEM:

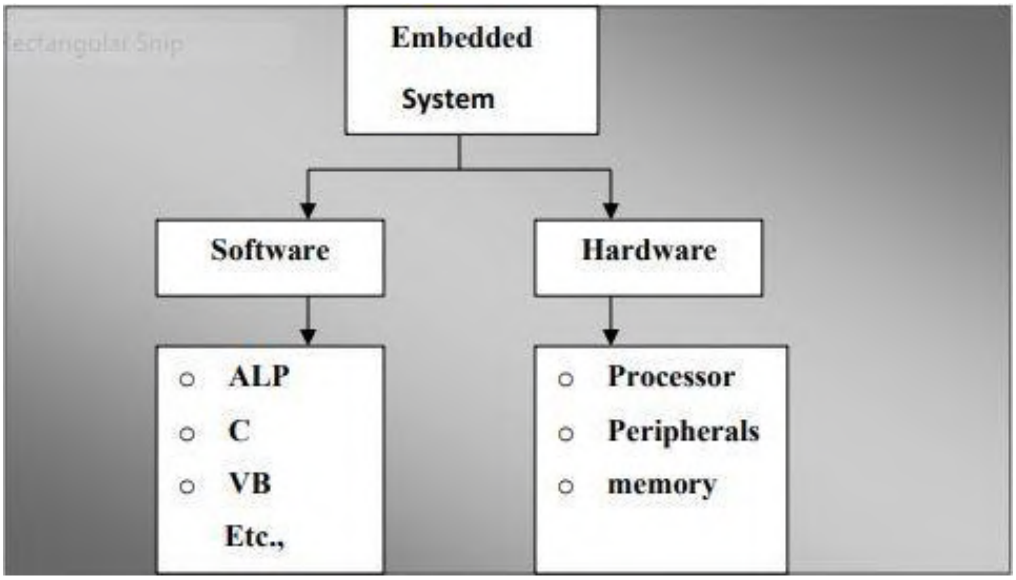


Fig 3.2.1 BLOCK DIAGRAM OF EMBEDDED SYSTEM

Software deals with the languages like ALP, C, and VB etc., and **Hardware** deals with Processors, Peripherals, and Memory

APPLICATIONS OF EMBEDDED SYSTEM:

• Manufacturing and process control

• Construction industry

• Transport

• Buildings and premises

• Domestic service

• Communications

• Office systems and mobile equipment

• Banking, finance and commercial

• Medical diagnostics, monitoring and life support

• Testing, monitoring and diagnostic systems

Micro Processor(μp):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Processors are classified into four types:

- Micro Processor (μp)
- Micro controller (μc)
- Digital Signal Processo1r (DSP)
- Application Specific Integrated Circuits (ASIC)

Three Basic Elements of a Microprocessor:

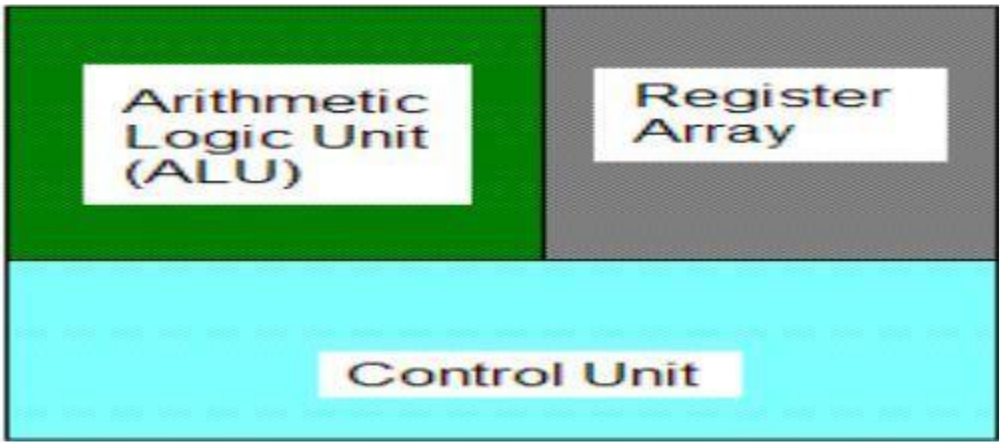


Fig 3.5.1 BASIC ELEMENTS OF MICRO PROCESSOR

Harvard Architecture:

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

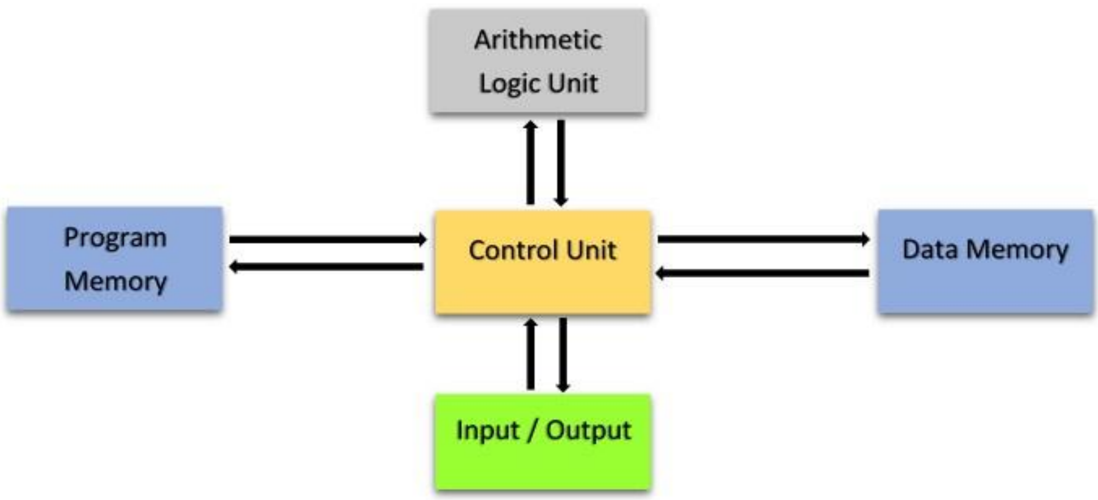


Fig 3.6.1 HARVARD ARCHITECTURE

Uses of the Harvard architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

- Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behaviour must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.
- Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

Von-Neumann Architecture

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. They cannot be performed at the same time. The von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A stored-program digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

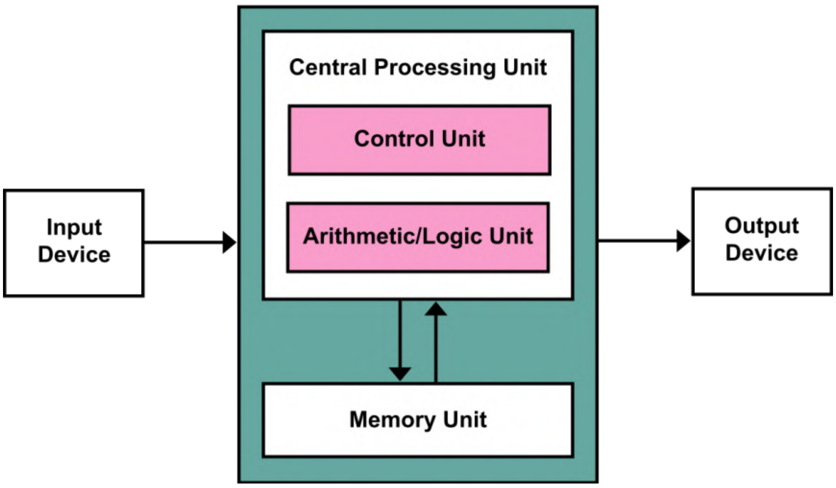


FIG3.6.2.1: VON-NEUMANN ARCHITECTURE

CHAPTER-4

HARDWARE DESCRIPTION

MICROCONTROLLER-ATMEGA328:

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The ATmega8 microcontroller contains 32 general purpose working registers. As shown in the below figure these registers are directly connected to ALU. Two registers can carry one single instruction consequently in one clock cycle.

Specifications:

Microcontroller	Atmel ATmega328
Operating Voltage (logic level)	5v
Input Voltage (recommended)	7v-12v
Input Voltage (limits)	6v-20v
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40mA
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2KB
EEPROM	1KB
Clock Speed	16MHz
Dimensions	0.73" x 1.70"
Length	45 mm
Width	18 mm
Weight	5g

Table 4.1.1 ATMEGA SPECIFICATIONS

Arduino:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board. The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software)
- .Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

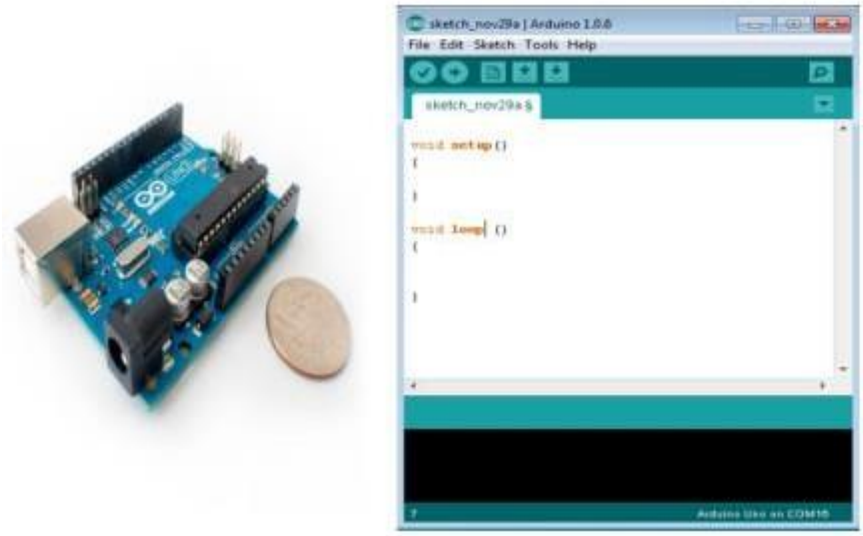


FIG 4.2.1 ARDUINO UNO

Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

4.3.1 Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

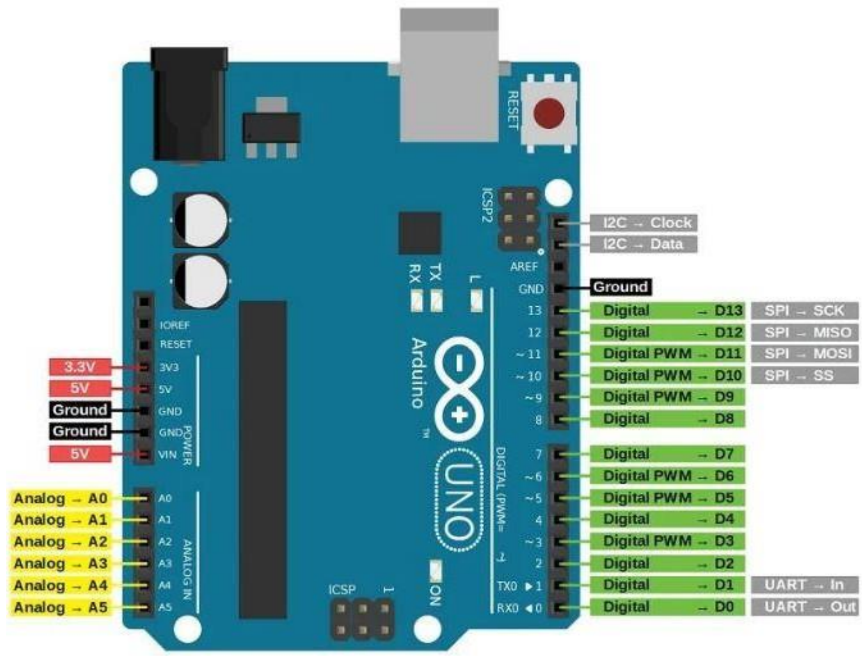


Fig 4.3.1.1 ARDUINI UNO BOARD

Vin: This is the input voltage pin of the Arduino board used to provide input supply from an external power source.

5V: This pin of the Arduino board is used as a regulated power supply voltage and it is used to give supply to the board as well as onboard components.

V: This pin of the board is used to provide a supply of 3.3V which is generated from a voltage regulator on the board

GND: This pin of the board is used to ground the Arduino board.

Reset: This pin of the board is used to reset the microcontroller. It is used to Resets the microcontroller.

Analog Pins: The pins A0 to A5 are used as an analog input and it is in the range of 0-5V.

Digital Pins: The pins 0 to 13 are used as a digital input or output for the Arduino board.

Serial Pins: These pins are also known as a UART pin. It is used for communication between the Arduino board and a computer or other devices. The transmitter pin number 1 and receiver pin number 0 is used to transmit and receive the data resp.

External Interrupt Pins: This pin of the Arduino board is used to produce the External interrupt and it is done by pin numbers 2 and 3.

PWM Pins: This pins of the board is used to convert the digital signal into an analog by varying the width of the Pulse. The pin numbers 3,5,6,9,10 and 11 are used as a PWM pin.

SPI Pins: This is the Serial Peripheral Interface pin, it is used to maintain SPI communication with the help of the SPI library. SPI pins include:

1. SS: Pin number 10 is used as a Slave Select
2. MOSI: Pin number 11 is used as a Master Out Slave In
3. MISO: Pin number 12 is used as a Master In Slave Out

4. SCK: Pin number 13 is used as a Serial Clock

LED Pin: The board has an inbuilt LED using digital pin-13. The LED glows only when the digital pin becomes high.

AREF Pin: This is an analog reference pin of the Arduino board. It is used to provide a reference voltage from an external power supply.

ARDUINO FAMILY:

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project , check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

Fig 4.4.1 ARDUINO FAMILY

SHIELDS

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities - controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

PIN DESCRIPTION OF ATMEGA328

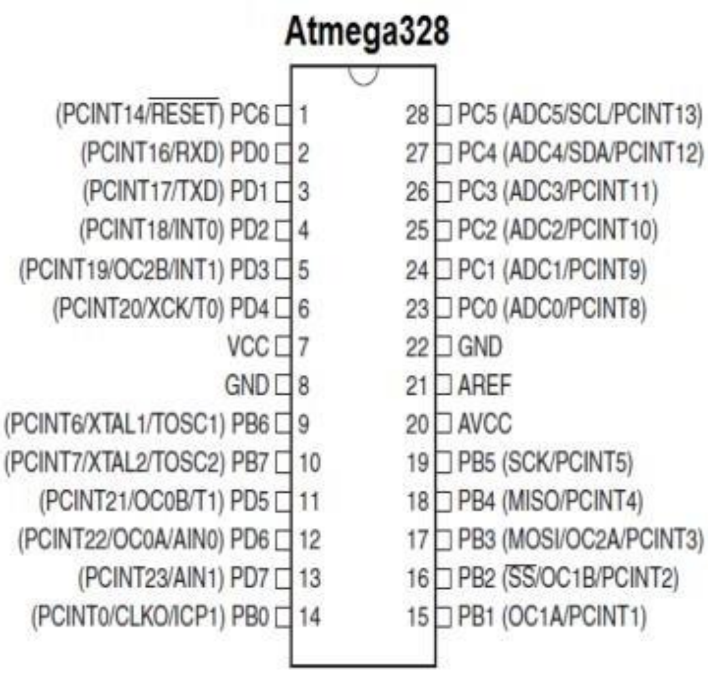


Fig4.6 .1 .PIN DIAGRAM OF ATMEGA

ADVANTAGES OF ARDUINO

- It is cheap
- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
- For beginners, it is very simple to use.

APPLICATIONS

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

POWER SUPPLY UNIT

INTRODUCTION:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as “Regulated D.C Power Supply”.

BLOCK DIAGRAM OF POWER SUPPLY:

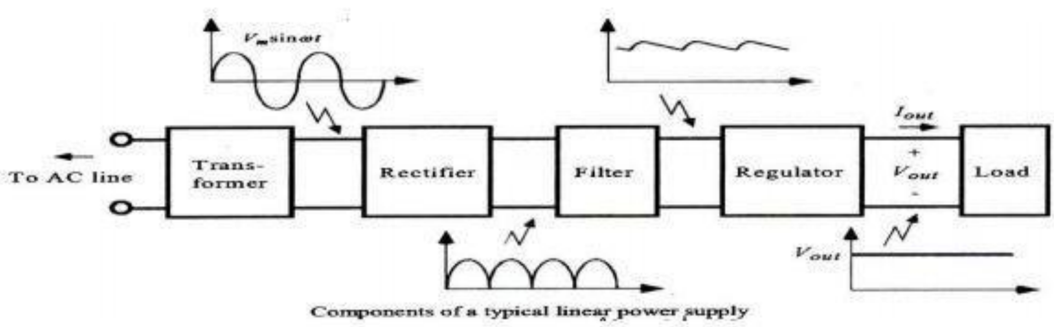


Fig 4.9.2.1 block diagram of power supply

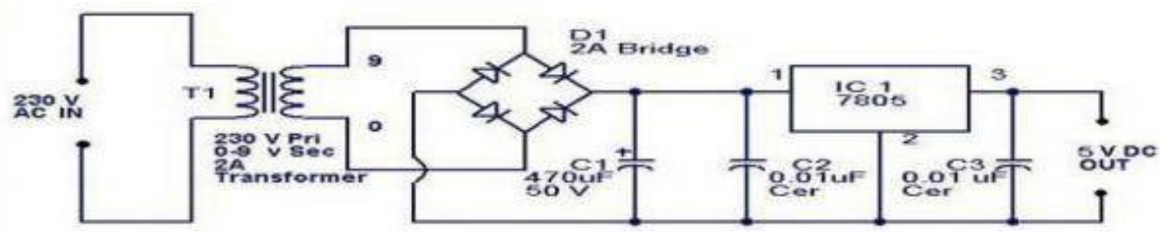


Fig 4.9.2.2 schematic diagram of power supply

DESCRIPTION OF POWER SUPPLY:

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a a different form of energy – such as solar, mechanical, or chemical - into electrical energy. A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered. For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

TRANSFORMER:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or primary winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the secondary winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

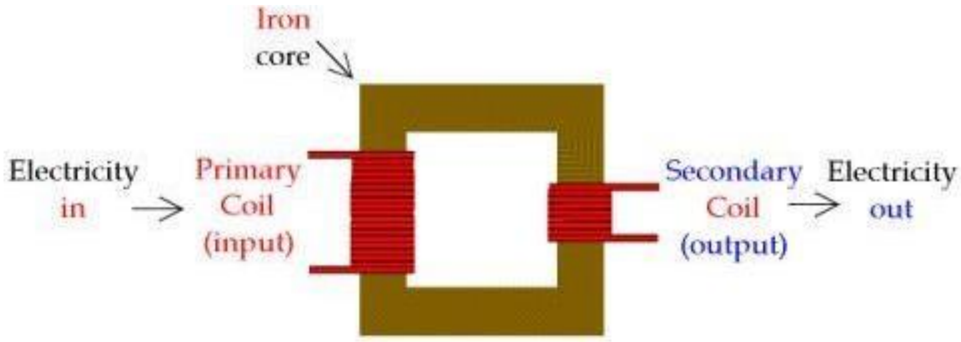


Fig 4.11.1 transformer

Basic Principle of Transformer:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

Working of Transformer:

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f. The iron core is laminated to prevent 'eddy currents' flowing in the core

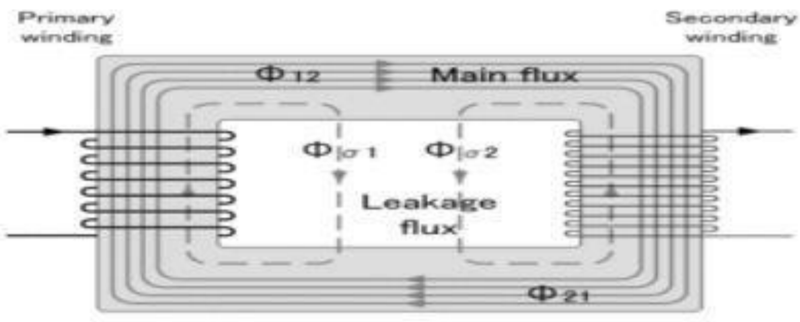


Fig 4.11.2.1 basic transformer

Classification of Transformer:

- Step-Up Transformer
- Step-Down Transformer

Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn's ratio of the two sets of windings determines the amount of voltage transformation.

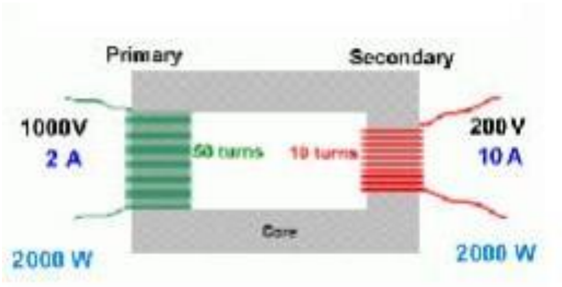


Fig 4.12.1.1 step down transformer

Step-Up Transformer:

A step up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply. A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years

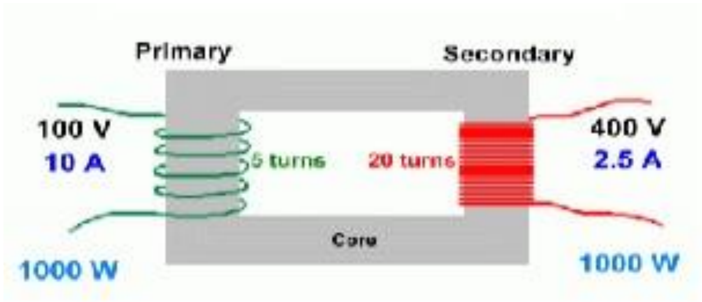


Fig 4.12.2.2 step up transformer

Rectifier:

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as 'half-wave' and 'full-wave' rectifiers. Both use components called diodes to convert AC into DC.

VOLTAGE REGULATOR:

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- ➢ Positive Voltage Series (78xx) and
- ➢ Negative Voltage Series (79xx)

78xx:’78’ indicate the positive series and ‘xx’indicates the voltage rating. Suppose 7805 produces the maximum 5V.’05’indicates the regulator output is 5V.

79xx:’78’ indicate the negative series and ‘xx’indicates the voltage rating. Suppose 7905 produces the maximum -5V.’05’indicates the regulator output is -5V.

LIQUID CRYSTAL DISPLAY(16 * 2)

INTRODUCTION:

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below

Display Data RAM:

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Busy Flag :

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact ammount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process

Instruction Register(IR) and Data Register(DR):

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD.

Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

16 x 2 Alphanumeric LCD Module Features:

Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing

• 61 x 15.8 mm viewing area

• 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line

• Can display 224 different symbols

• Low power consumption (1 mA typical)

• Powerful command set and user-produced characters

• TTL and CMOS compatible

• Connector for standard 0.1-pitch pin headers

SCHEMATIC DIAGRAM:

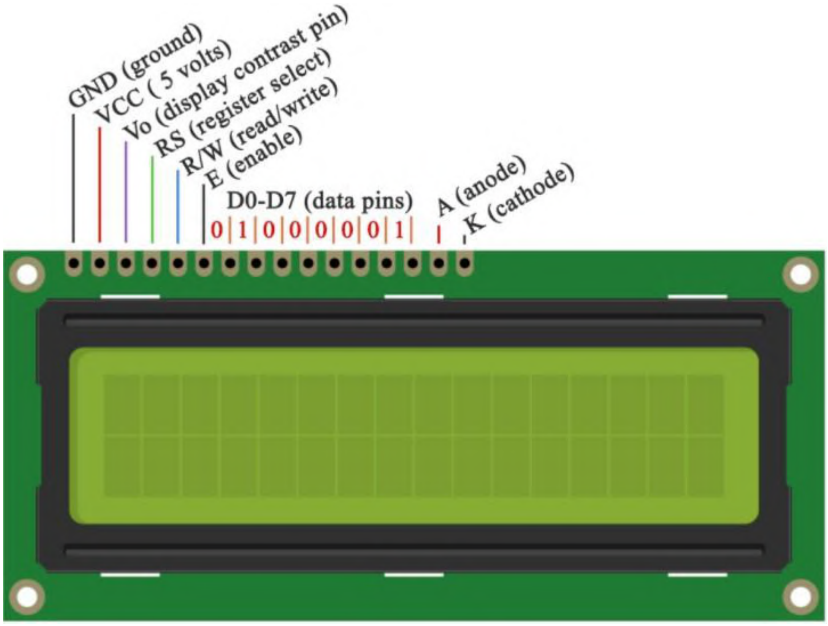


FIG 4.15.3.1: LCD

SPECIFICATIONS:

Connector Pin Assignment:

Pin	*Symbol*	*Function*	*Pin*	*Symbol*	*Function*
14	DB7	Data Bus Line	6	E	Enable
13	DB6	Data Bus Line	5	R/W	Read/Write
12	DB5	Data Bus Line	4	RS	Register select input
11	DB4	Data Bus Line	3	Vcontrast	Contrast (0V for max contrast)
10	DB3	Data Bus Line	2	**GND (0V)**	Power supply
9	DB2	Data Bus Line	1	**Vcc +5V**	Power supply
8	DB1	Data Bus Line			
7	DB0	Data Bus Line			

Table 4.15.4.1 connector pin segments

ULTRASONIC SENSOR

INTRODUCTION

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

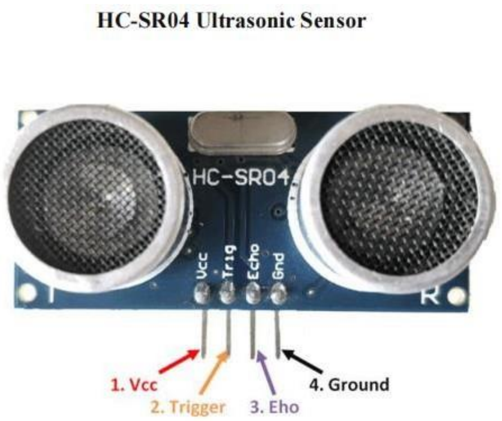


FIG 4.16 . 1: HCSR04 ULTRASONIC SENSOR

1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

Table 4.16.1: Ultrasonic sensor pin configuration

HC-SR04 Sensor Features

• Operating voltage: +5V

• Theoretical Measuring Distance: 2cm to 450cm

• Practical Measuring Distance: 2cm to 80cm

• Accuracy: 3mm

• Measuring angle covered: <15°

• Operating Current: <15mA

• Operating Frequency: 40Hz

HCSR04 WORKING：

As shown above the **HC-SR04 Ultrasonic (US) sensor** is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below

FIG: ULTRASONIC RECEIVER MODULE

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

How to use the HC-SR04 Ultrasonic Sensor:

HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used.

Power the Sensor using a regulated +5V through the Vcc ad Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor.

The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information the distance is measured as explained in the above heading.

Applications

•Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.

• Used to measure the distance within a wide range of 2cm to 400cm

• Can be used to map the objects surrounding the sensor by rotating it

• Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water.

CHAPTER 5

PROJECT IMPLEMENTATION

BLOCK DIAGRAM

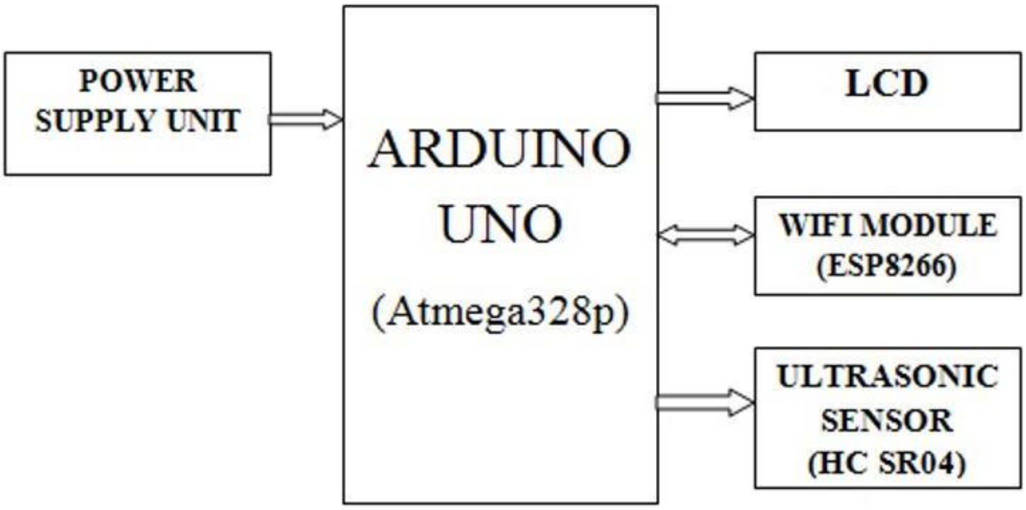


FIG 5.1:BLOCK DIAGRAM

SCHEMATIC DESIGN

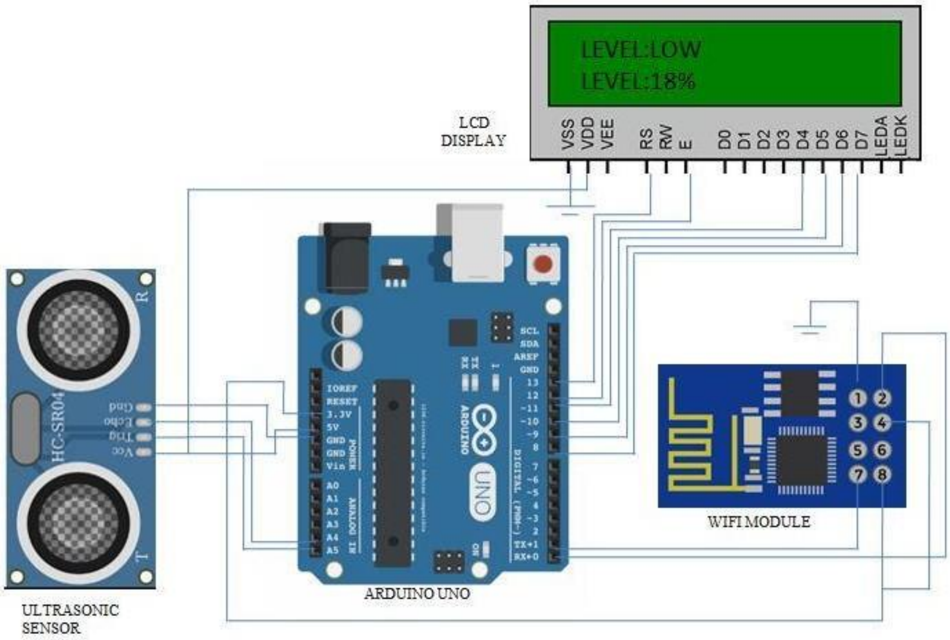


FIG 5.2 :SCHEMATIC DESIGN

- In the circuit the digital pins D13,D12,D11,D10,D9,D8 are connected to the liquid crystal display(LCD).
- The pins D2,D3 are connected to transmitter(TX) and receiver(RX) of wifi module respectively .
- The analog pins A5 and A4 are connected to rigger pin and echo pin of ultrasonic sensor respectively.
- The vcc of 5V is given to ultrasonic sensor and lcd display and vcc of 3.3V is given to wifi module.
- The pins D2 and D3 are connected to wifi module in order to send the usable data through internet.
- The obtained data can be reviewed and monitored through Telnet app.
- The wifi module transmits the data to telnet app by using some commands. These commands we are going to interface in aruduino ide.

SOFTWARE WORKFLOW

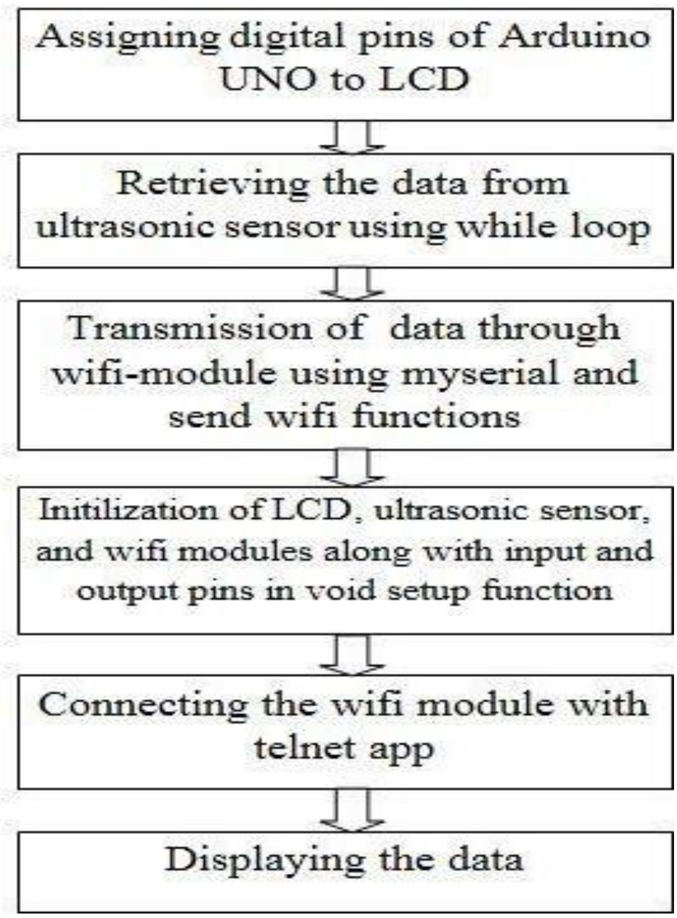


FIG 5.3:SOFTWARE WORKFLOW

The above workflow states what are the steps that we need to implement in software(ArduinoIDE) part to interface with hardware components.

FLOWCHART

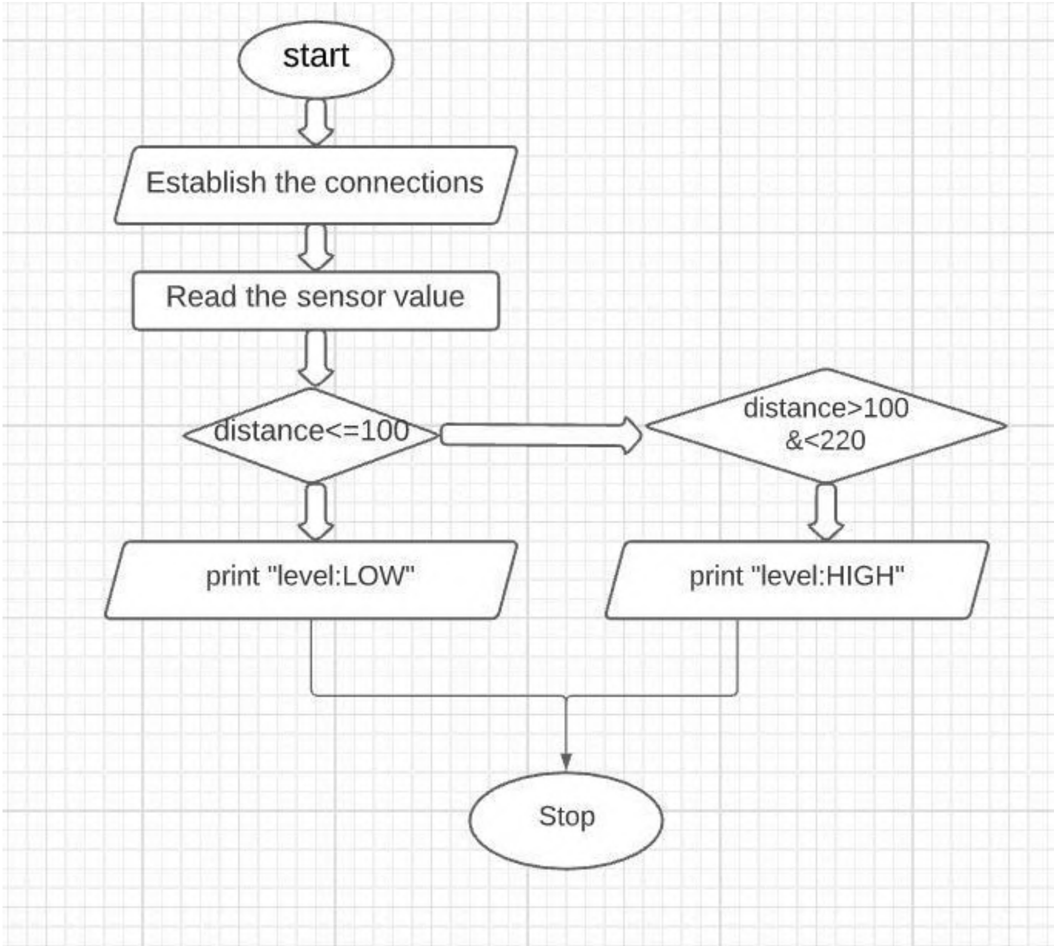


FIG 5.4.1:FLOWCHART

In above flow chart we can see that there are two conditions

If distance<=100 it is going to print low which means the level of liquid is low

If distance>100 &<200 it is going to print high which means the level of liquid is at peak stage.

Based on these two conditions the user will able to know whether the tank is full or not.

CHAPTER -6

SOFTWARE EXPLINATION

SOFTWARE EXPLANATION:

Software Requirements

- Proteus simulation
- Arduino software
- Programming language

Arduino software:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals.

- has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

•Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50

• Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

• Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

• Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based.

•Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers c an make their own version

• Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money

• Getting Started with Arduino and Genuino products:-

Install the Arduino Software (IDE) on Windows PCs-

How to Download the Arduino Software (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

Installation:

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

FIG 6.4.1 : A TO B USB CABLE

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig 6.4.2:A to Mini-B cable

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

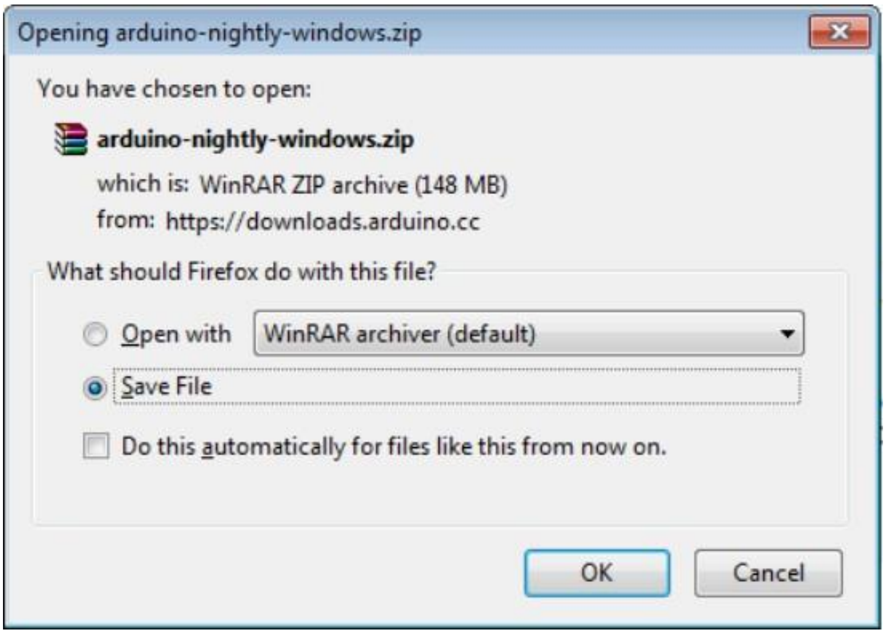


Fig 6.4.3: Opening Arduino

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper,a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

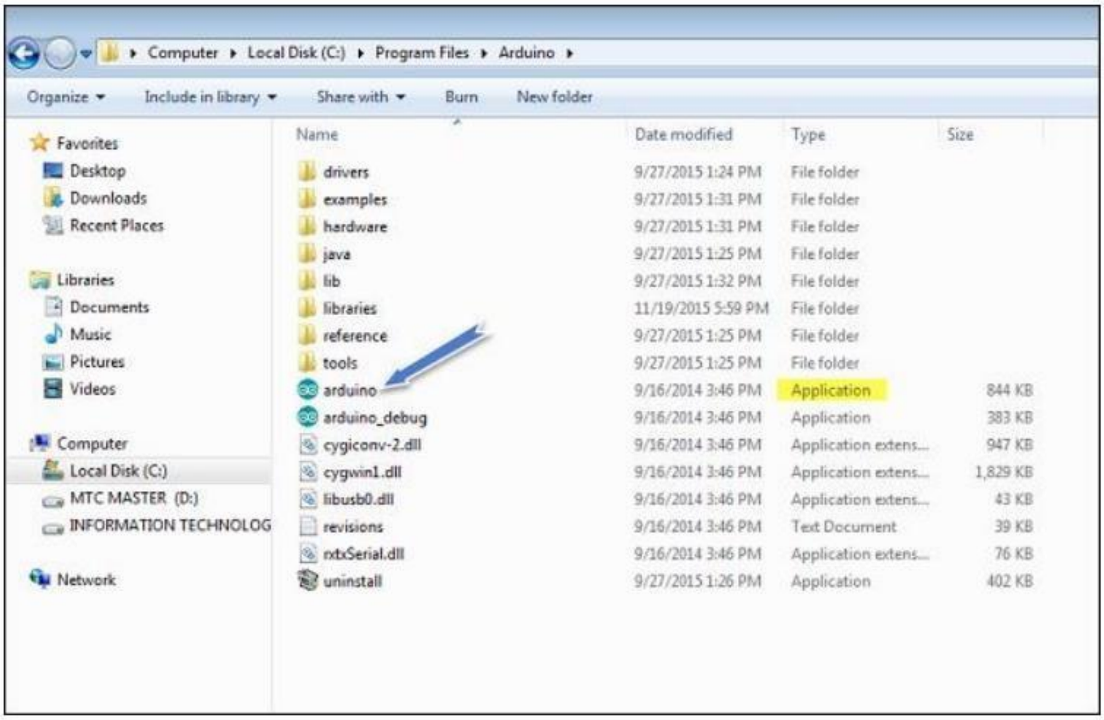


Fig 6.4.4:Arduino Application File

Step 5 − Open your first project.

Once the software starts, you have two options −

• Create a new project.

• Open an existing project example.

To create a new project, select File → **New**

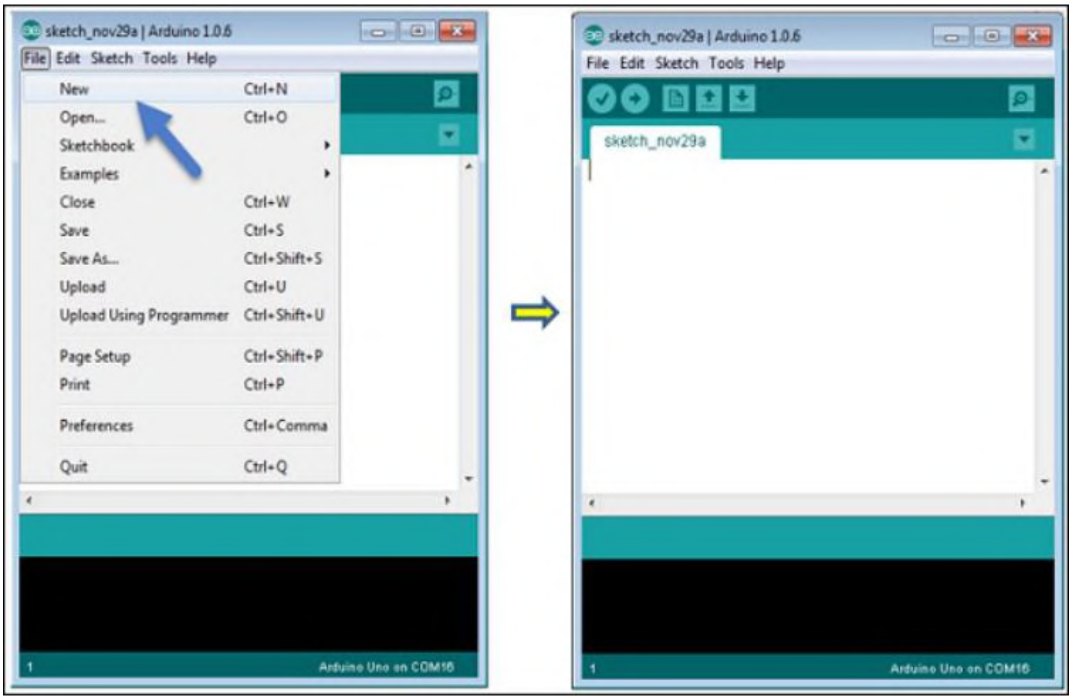


Fig 6.4.5: Opening a new file

To open an existing project example, select File → Example → Basics → Blink.

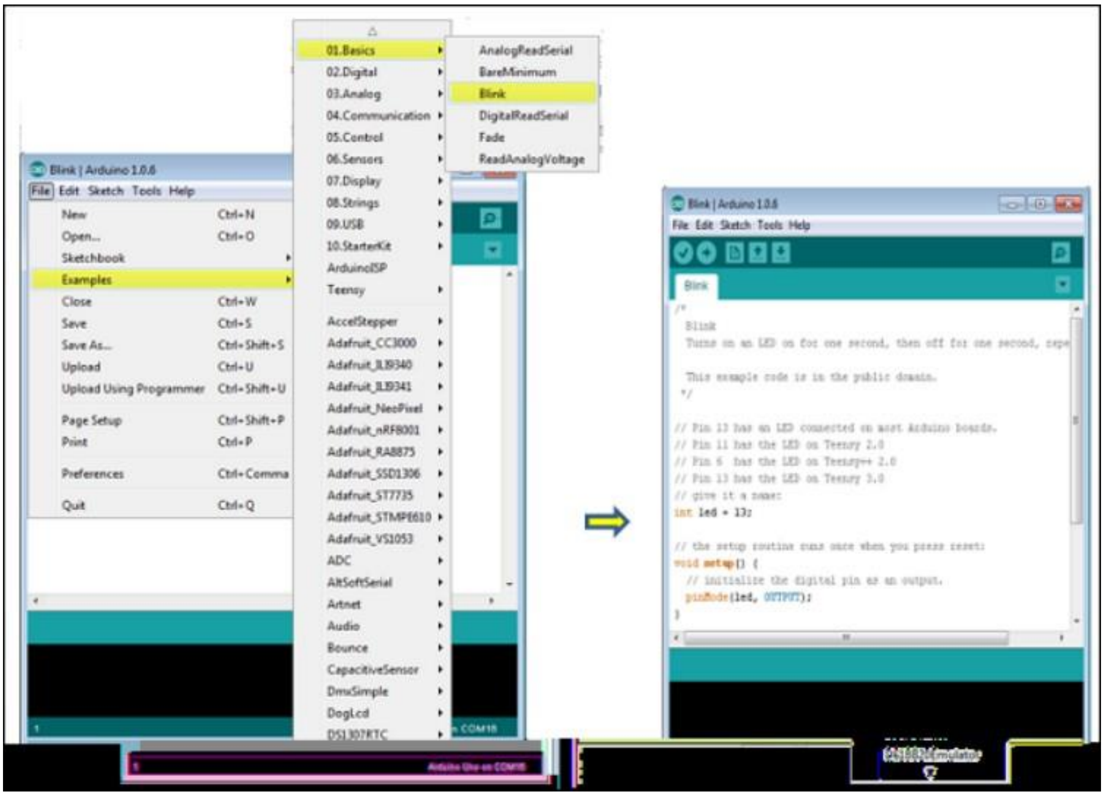


Fig 6.4.6:Examples related to Arduino

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer. Go to Tools → Board and select your board.

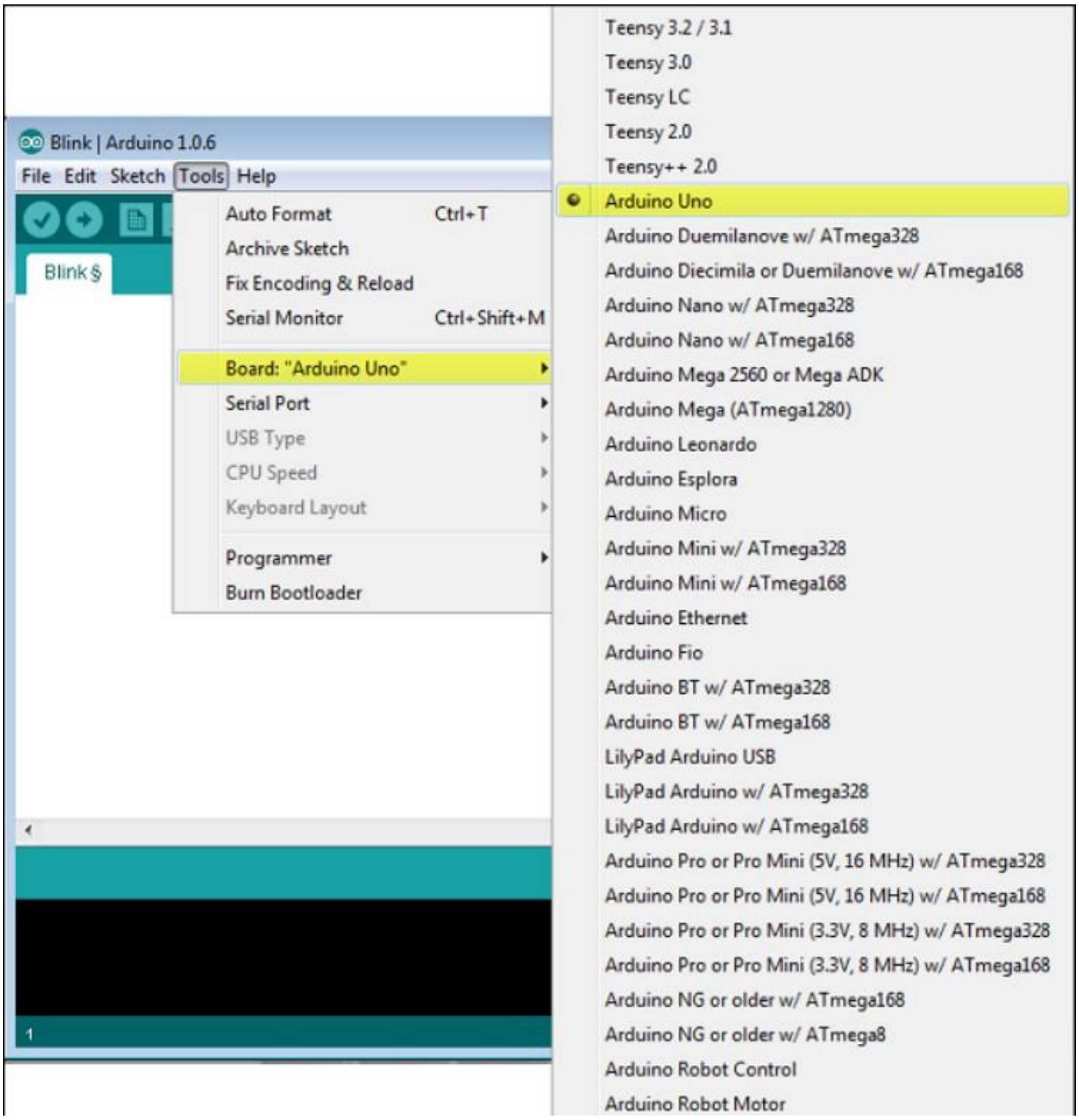


Fig 6.4.7:Basic Arduino board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

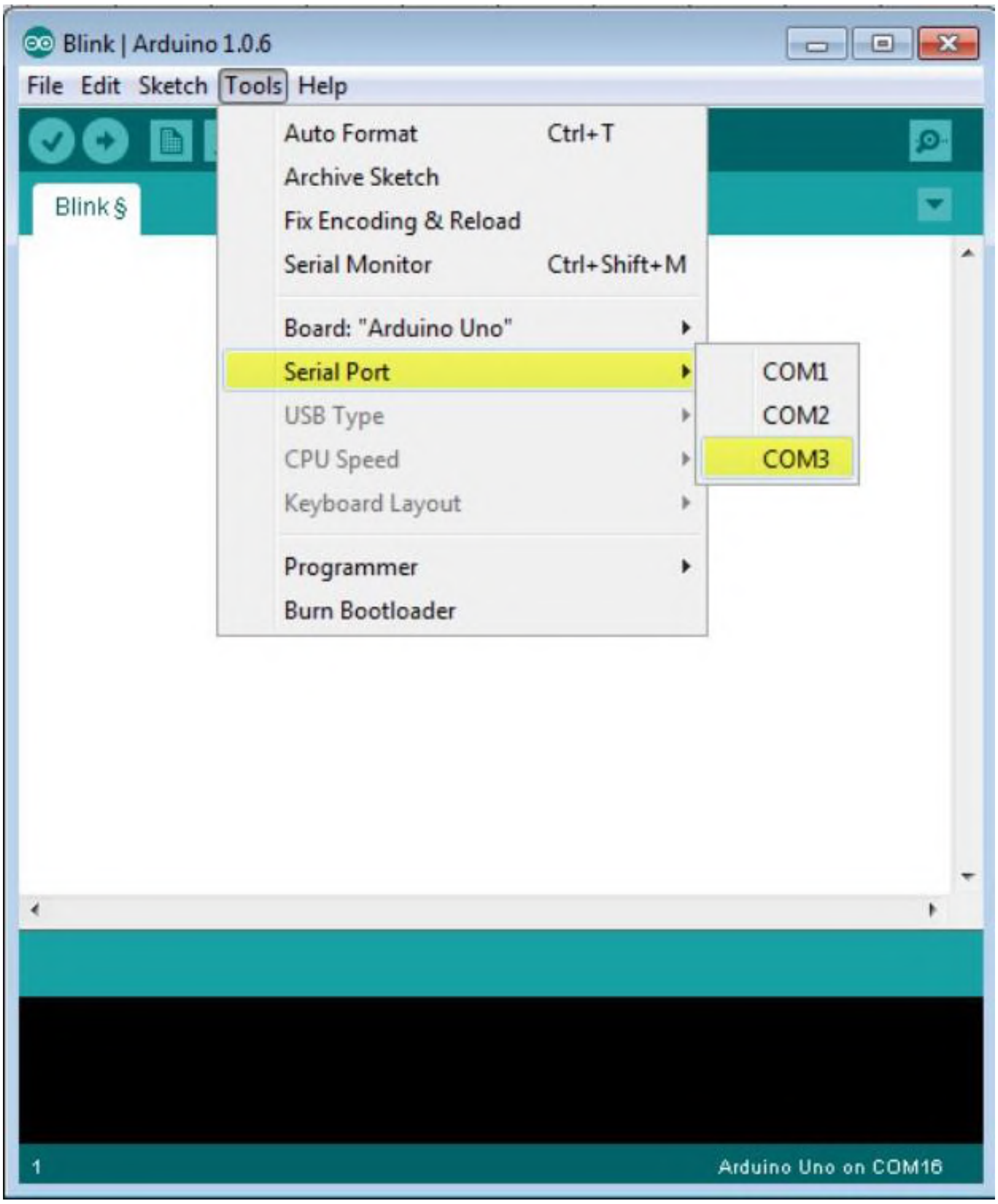


Fig 6.4.8 : Serial Port-COM3

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

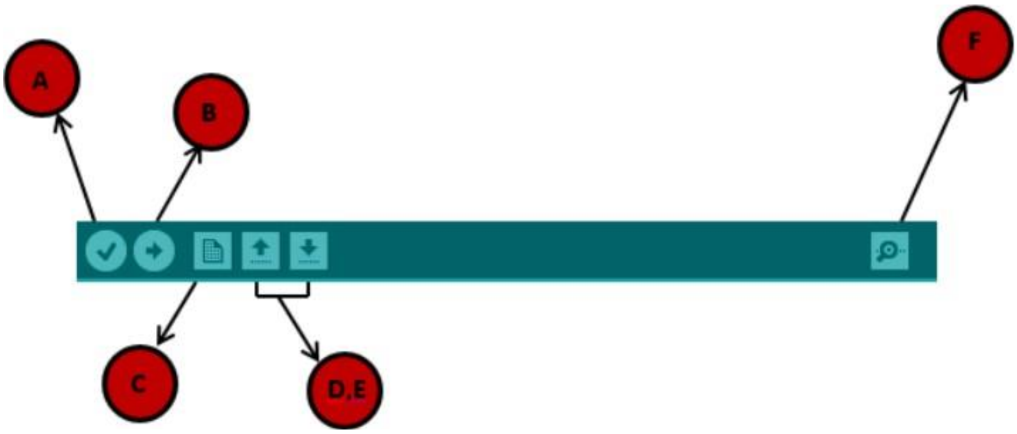


FIG 6.4.9: Uploading the program on board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24 V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

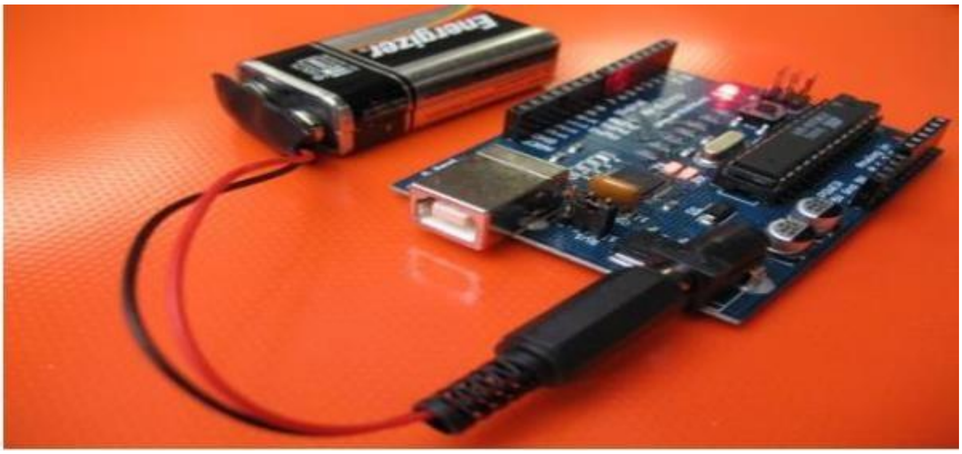


FIG 6.5.1:ARDUINO WITH BATTERY

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called "sketches", but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()

{

Serial.begin(9600);


Serial.println("Hello World");

}

void loop() {}
```

Fig 6.6.1 Arduino Programming Sketch

MC Programming Language: Embedded C

This is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements. Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

Proteus:

Proteus is a simulation and design software tool developed by Lab centre

Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing.

ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc. ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design

Step 1: Open ISIS software and select New design in File menu

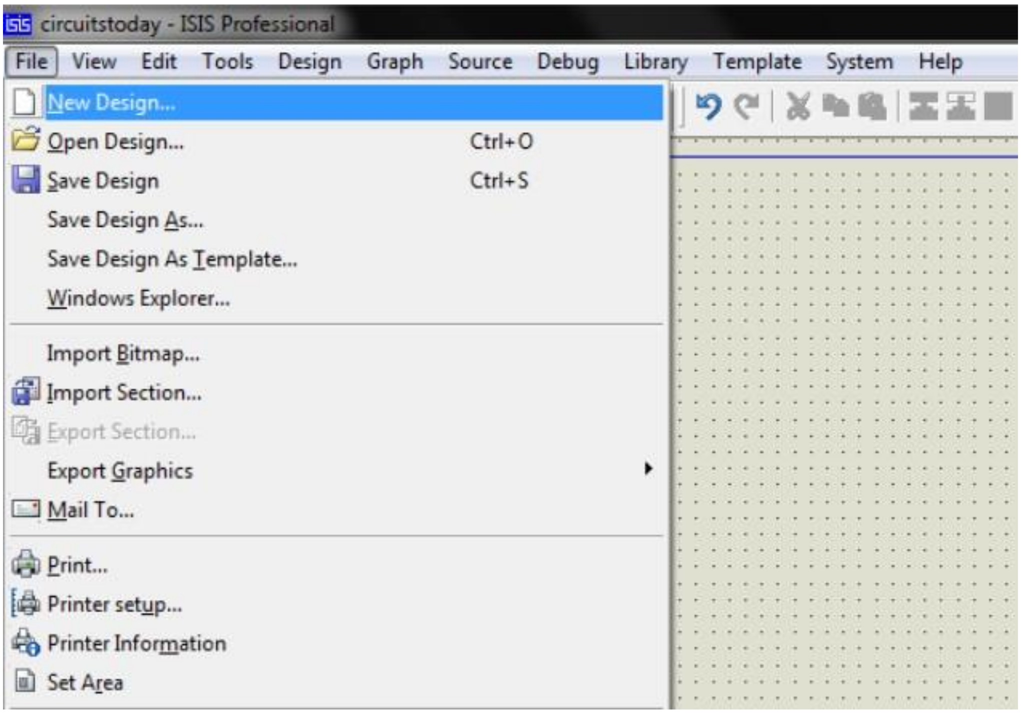


Fig 6.8.3.1:Proteus file menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a PopUp appears asking to select the template. It is similar to selecting the paper size while printing.

For now, select default or according to the layout size of the circuit.

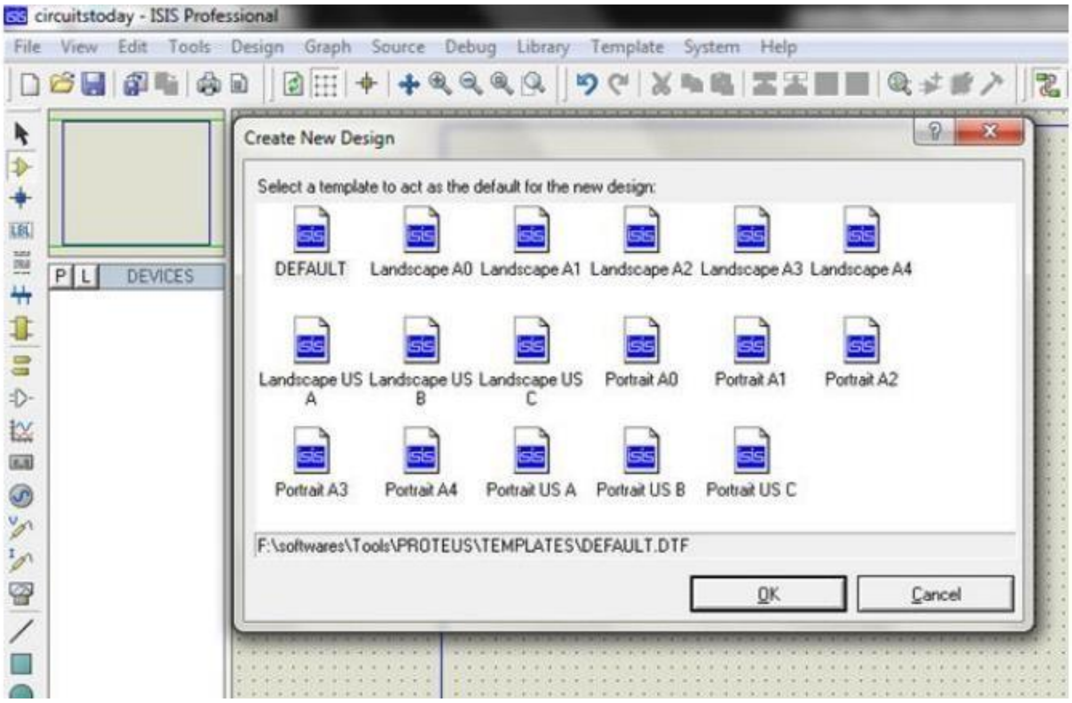


Fig 6.8.3.2:Proteus Default Select Template

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory

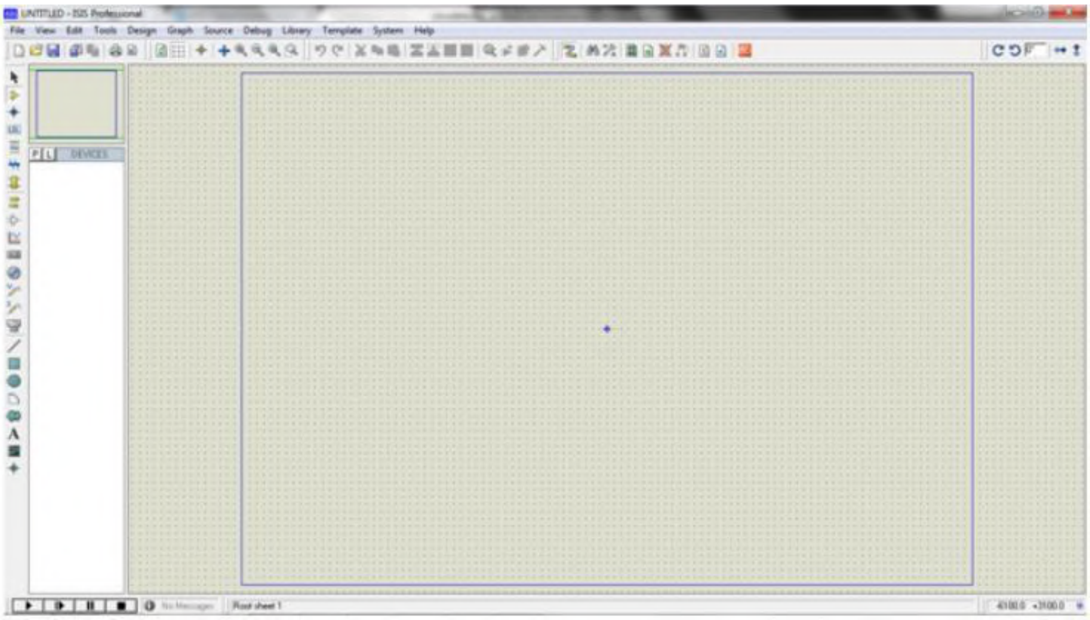


Fig6.8.3.3: Proteus design sheet

Step 4: To Select components, Click on the component mode button.

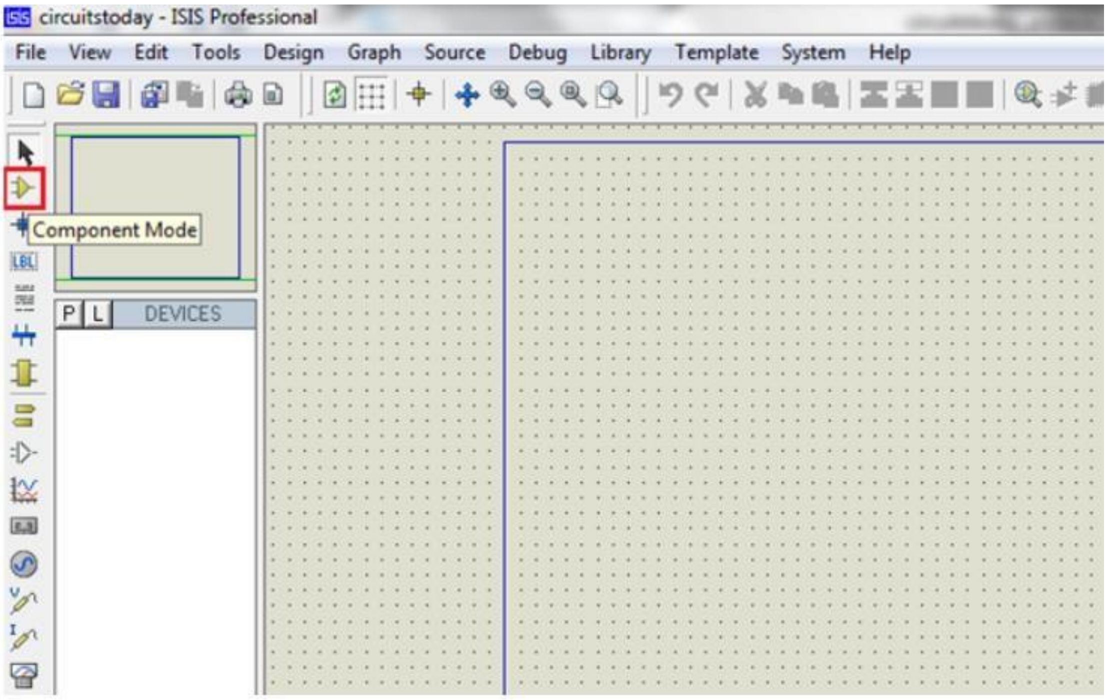


Fig 6.8.3.4: Component mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

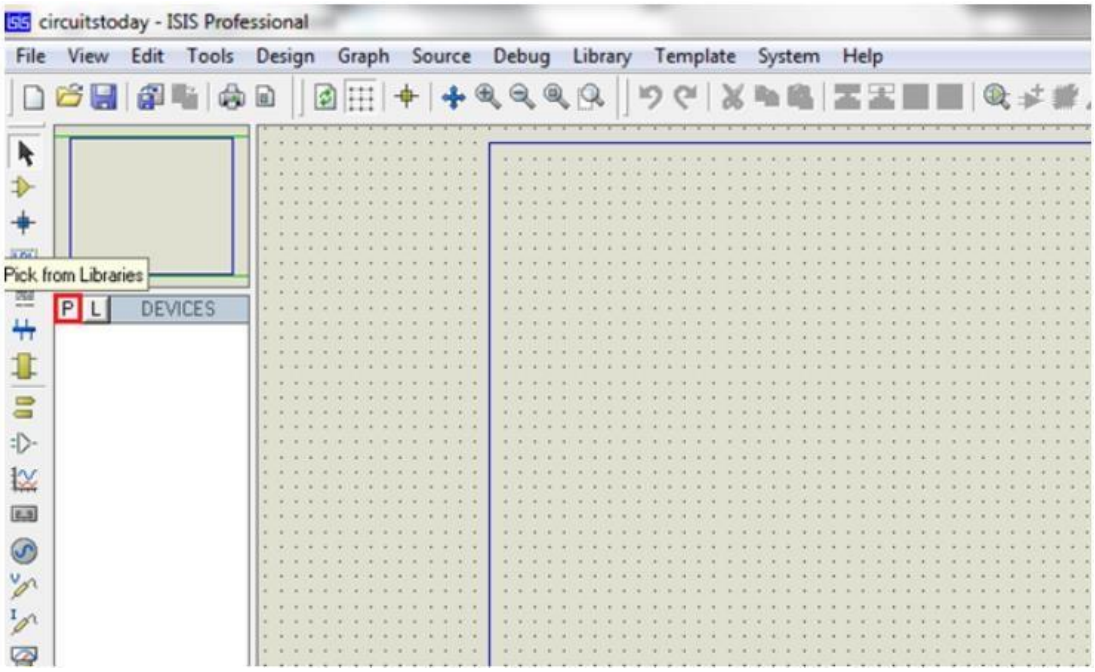


Fig6.8.3.5: Pick from libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

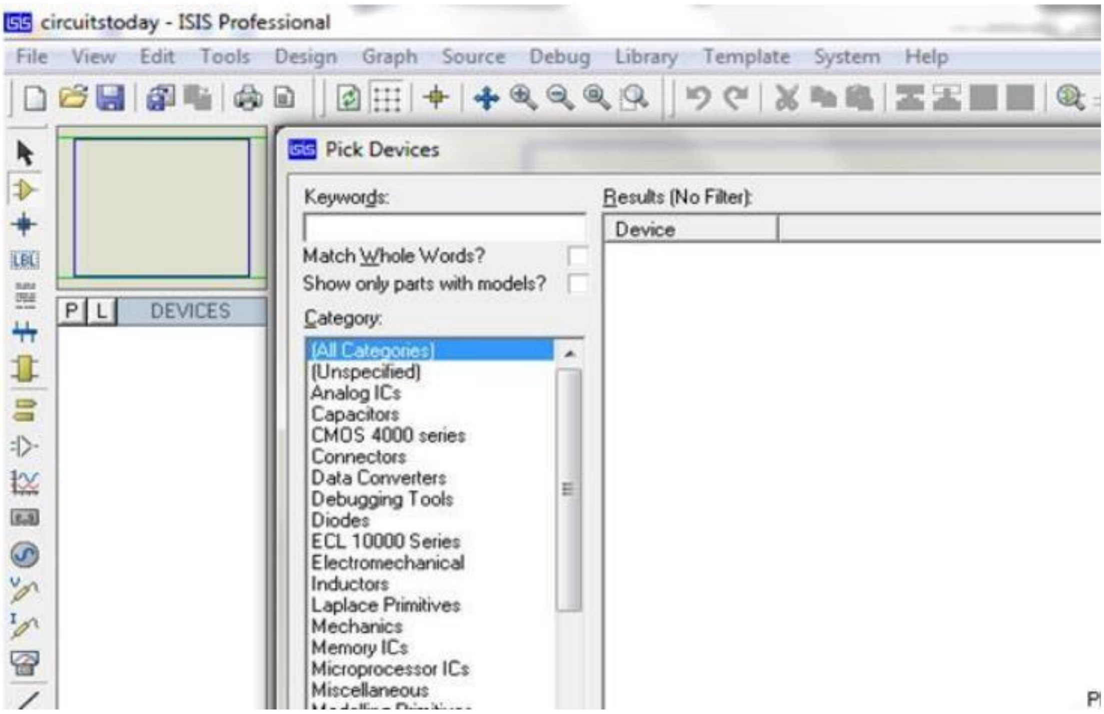


Fig 6.8.3.6:Keywords Textbooks

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

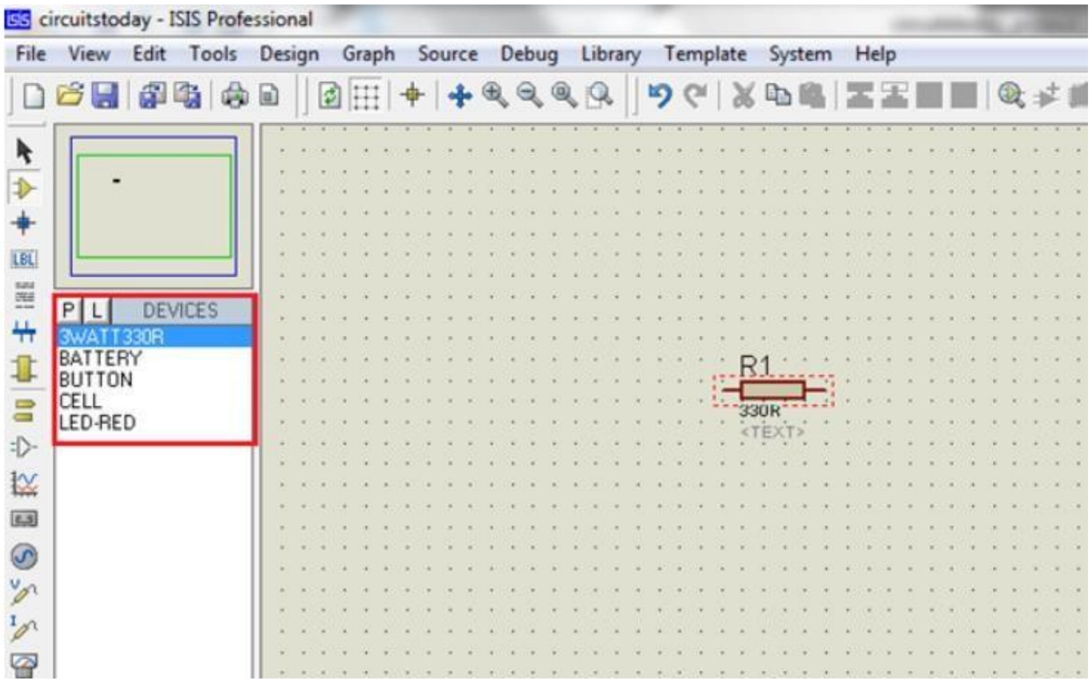


Fig 6.8.3.7:Component selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to

connect through wires. Left click from one terminal to other to make connection. Double rightclick on the connected wire or the component to remove connection or the component respectively.

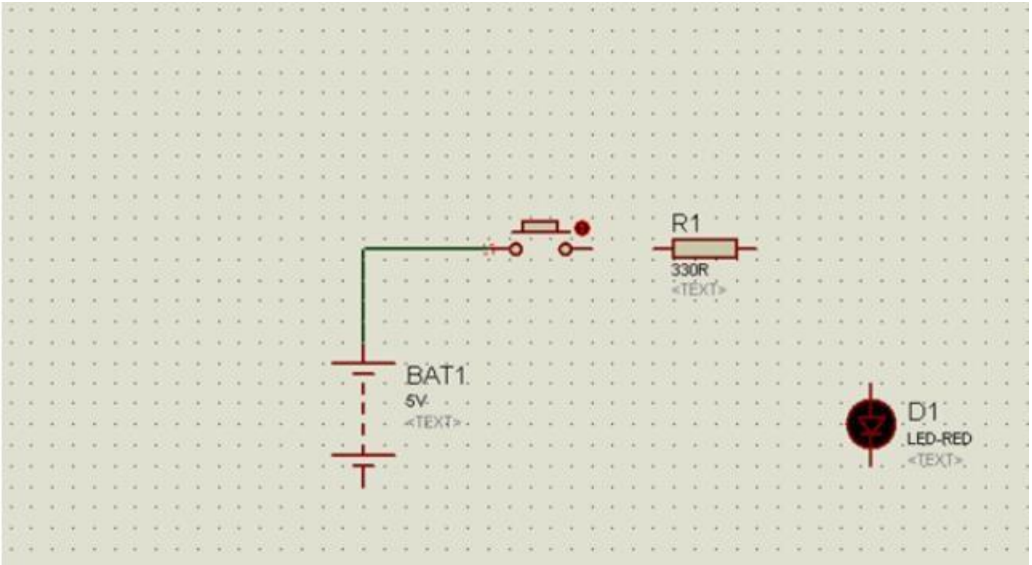


Fig 6.8.3.8: Component Properties selection

Double click on the component to edit the properties of the components and click on Ok.

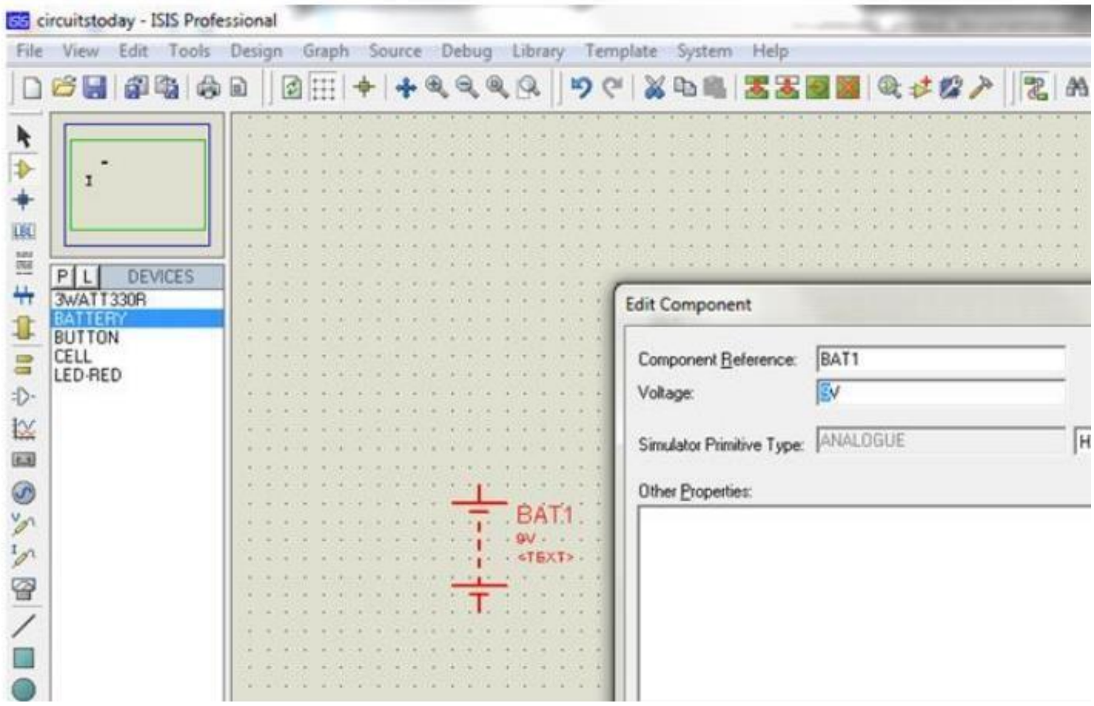


Fig 6.8.3.9: Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

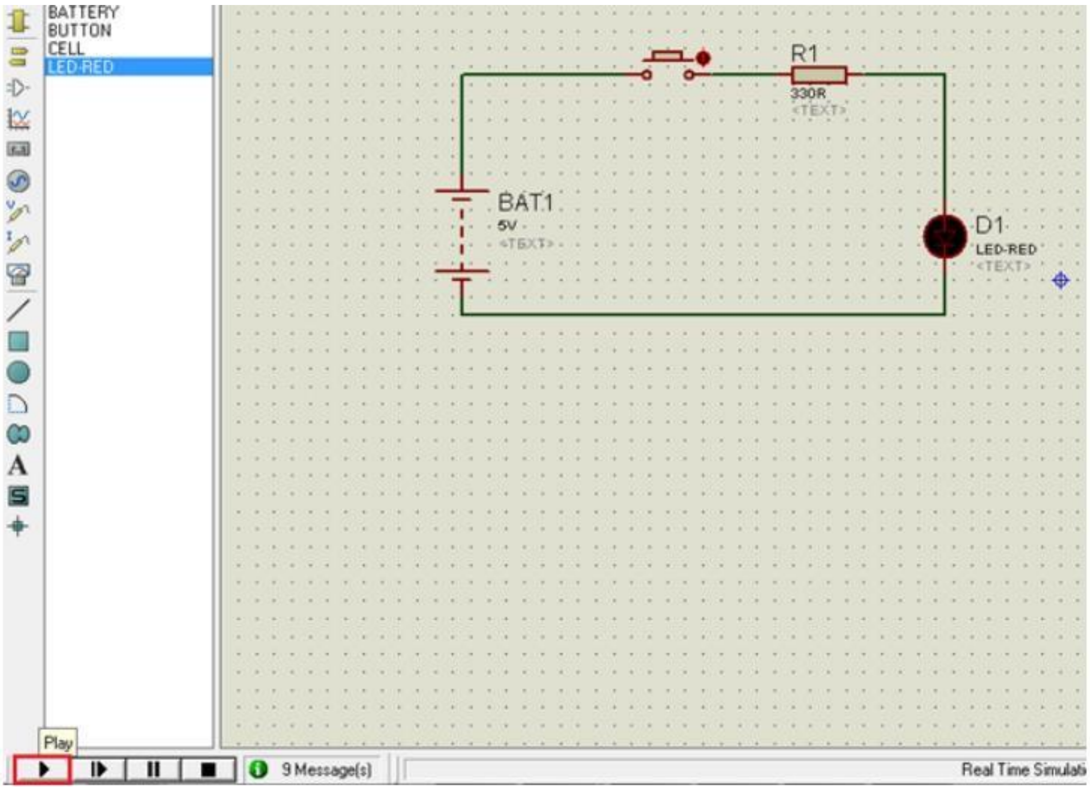


Fig 6.8.3.10:Simulation run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

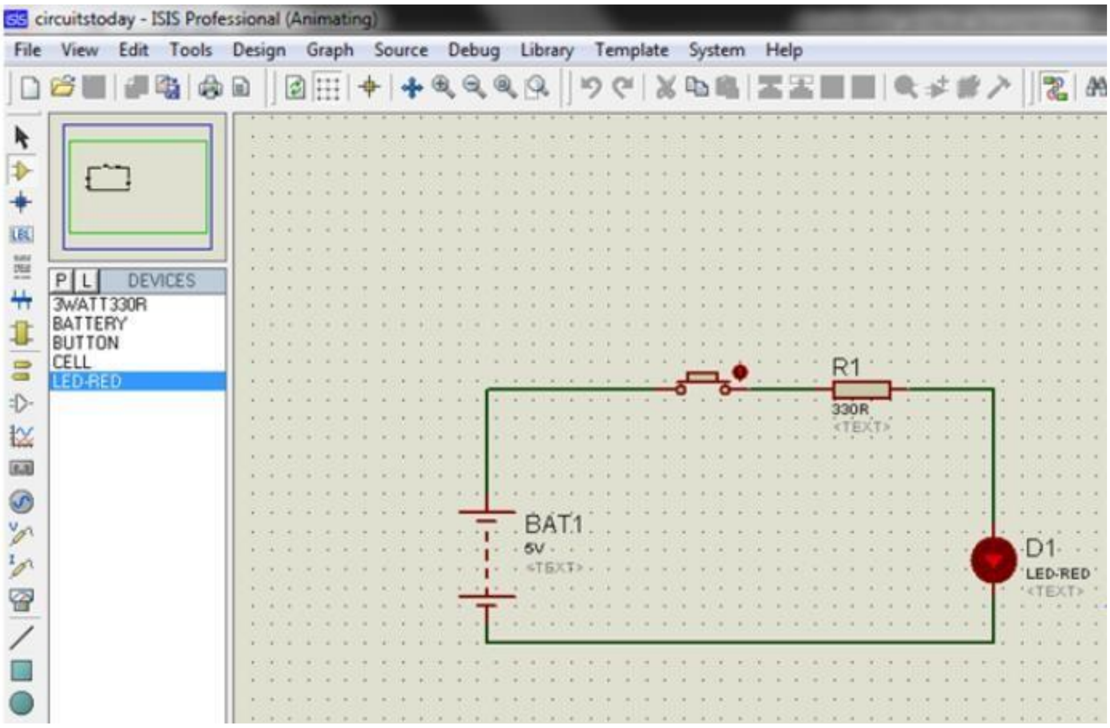


Fig 6.8.3.11 :Simulation Animating

Simulation can be stepped, paused or stopped at any time

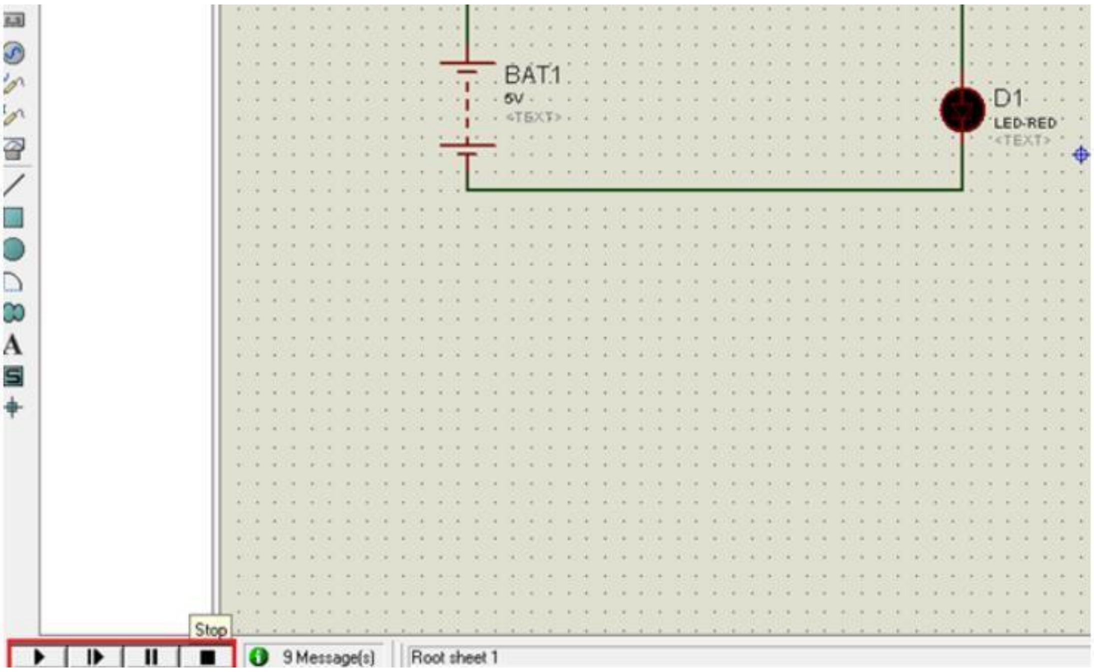


Fig 6.8.3.12: Stimulation Step -pause-stop button

Project Code:

```
#include<LiquidCrystal.h>

#include<SoftwareSerial.h>

LiquidCrystal lcd(13,12,11,10,9,8);

SoftwareSerial mySerial(2,3);//rx , tx  //esp8266


int trigger_pin = A5;

int echo_pin = A4;

int time;

int distance;


void serialFlush(){

  while(Serial.available() > 0) {

  char tt = Serial.read();

  }
```

```
}

void sendwifi(String chr,unsigned int len)
{
  mySerial.print("AT+CIPSEND=0,");
  mySerial.println(len-1);
  delay(2000);
  mySerial.print(chr);
  delay(2000);
}
char res[130];
char buff[130];

void setup()
{
  pinMode(trigger_pin, OUTPUT);
  pinMode(echo_pin, INPUT);
  lcd.begin(16,2);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("IOT Based");
  lcd.setCursor(0,1);
  lcd.print("Liquid Monitor");
  delay(2000);
  Serial.begin(9600);
    mySerial.begin(115200);//esp8266
    mySerial.print("AT\r\n");
    delay(1000);
    mySerial.print("ATE0\r\n");
```

```
    delay(1000);
    mySerial.print("AT+CWMODE=3\r\n");
    delay(1000);
    mySerial.print("AT+CWSAP=\"Project\",\"project1235\",5,0\r\n\r\n");
    delay(1000);
    mySerial.print("AT+CIPMUX=1\r\n");
    delay(1000);
    mySerial.print("AT+CIPSERVER=1,23\r\n");
    delay(1000);
    lcd.clear();lcd.setCursor(0, 0);lcd.print("WATING FOR LINK");
    while(1)
    {
      if(mySerial.available())
      {
        //if(Esp.find("0,LINK"))
        if(mySerial.find("0,CONNECT"))
        {
            lcd.clear();lcd.setCursor(0, 0);lcd.print("LINK FOUND");
          break;
        }

      }
    }
    delay(1000);
    sendwifi("WELCOME \r\n",11);
    delay(1000);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("READY");
  delay(2000);
}
```

```
void loop()
{
  digitalWrite (trigger_pin, HIGH);
  delayMicroseconds (10);
  digitalWrite (trigger_pin, LOW);
  time = pulseIn (echo_pin, HIGH);
  distance = (time * 0.034) / 2;

   lcd.clear();
   lcd.setCursor(0,0);
   lcd.print("Level: ");
   lcd.print(distance);
   lcd.print("%");
   if(distance<=10)
    {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Level: LOW");
      delay(10);
      sprintf(buff,"Level: LOW \r\n",14);
      sendwifi(buff,strlen(buff));
      delay(3000);
    }
    if(distance>=220)
    {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Level: HIGH");
```

```
    delay(10);
    sprintf(buff,"Level: HIGH \r\n",15);
    sendwifi(buff,strlen(buff));
    delay(3000);
  }


 sprintf(buff,"Level: %u\r",distance);
 sendwifi(buff,strlen(buff));
 sprintf(buff,"%%\r\n",5);
 sendwifi(buff,strlen(buff));
 delay(1000);
}
```

CHAPTER 7: RESULTS

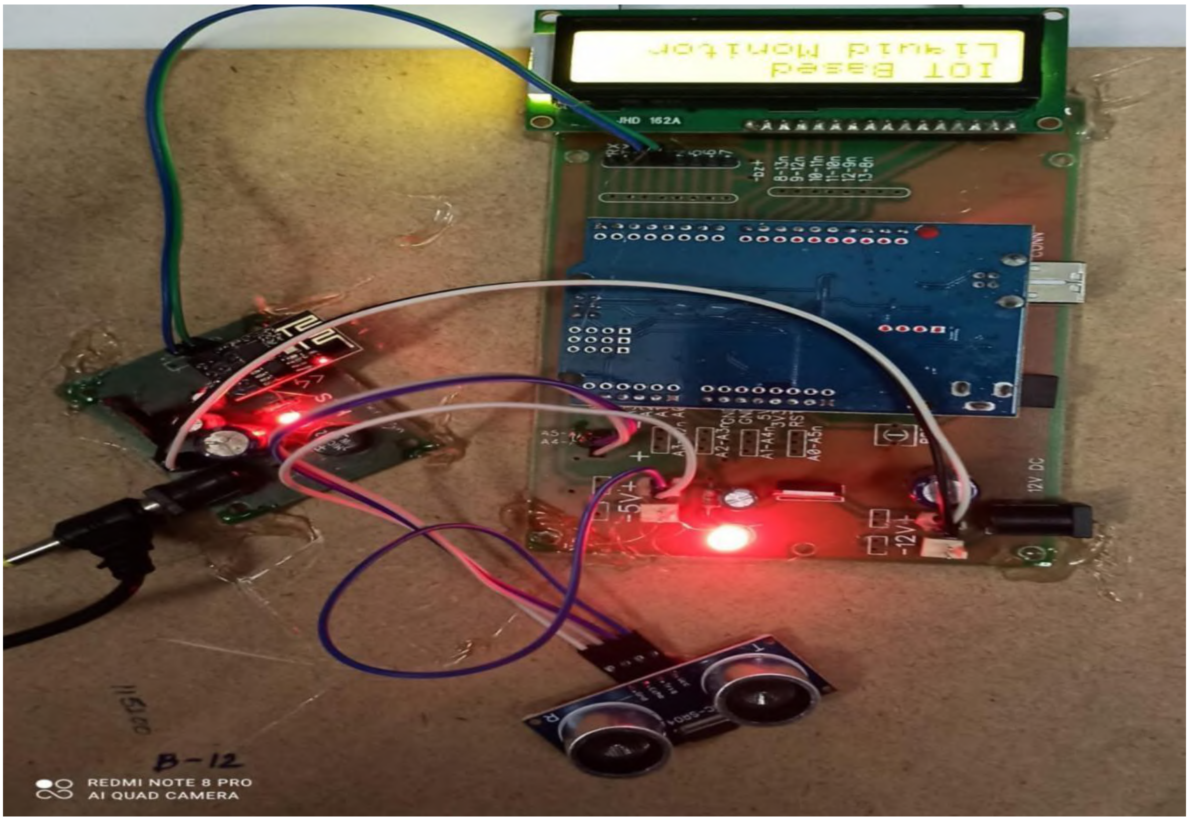


Fig7.1 : The circuit kit of the project

The Arduino UNO is connected to the system with a required code dumped in it.Through the functioning of ultrasonic sensor the level of liquid is analyzed and is sent to Arduino board to process and send the data to the LCD and wifi module. This Data is sent through internet using wifi module and this data can be accesed through telnet app on smart device such computer or smartphone which has a reliable internet connection

Fig7.2: LIQUID LEVEl IN TERMS OF PERCENRAGE

Here the LCD is showing the level of liquid in container in digital format.

FIG 7.3: LIQUID LEVGEL IN TERMS OF LEVEL

The liquid level is shown as low when level of liquid is less than 10% of the container.

CHAPTER 8:

CONCLUSION

Conclusion:

The IOT is one of the most rapidly growing technolgy as it helps us to live a better lives by connecting technologies together and conservation of liquids like water and fossil fuels is very important as they are not abundantly available,Which requires constant monitoring.In order make it possible we have come up with this project.The main motto of this project is to remotely monitor the liquid levels in a container with very less hardware.

Future Inhancements:

- This can be further developed to calculate the water consumption in a time period.
- This system can be implemented for more than one tank or container for example. This can be used with a connected set of tanks as well.
- Thereby especially the industrial firms can manage the water effectively.

REFERENCES

[1] Yogita patil, Ramandeep Singh, “Smart tank Management System for Residential Colonies Using Atmega 128A Microcontroller”, International Journal of Scientific & Engineering Research, Volume 5, Issue 6, June-2014

2] http://arduino info.wikispaces.com/Ultrasonic+Distance+Sensor

[3] Asaad Ahmed Mohammedahmed Eltaieb, Zhang Jian Min, “Automatic Water level control System”, International Journal of Science and Research, volume 4 issue 12 December 2015.

[4] Partik Roy, “Construction of Digital Water Level Indicator and Automatic Pump Control System”, International journal of research, volume-3 issue-13 September 2016.

[5] http://arduino- info.wikispaces.com

[6] https://components101.com/ultrasonic-sensor-working-pinout-datasheet

[7] Beza Negash Getu, Hussian A. Attia, Automatic Water Level Sensor and Controller System, IEEE volume-1 issue-3 2016.

[8] https://circuitdigest.com/microcontroller-projects/arduino-ultrasonic-sensor-based-distance-measurement

[9] J.A Jodice, “Relay Performance Testing”, IEEE,volume- 12 issue- 1, 1997.

[10] https://www.electronics tutorials.ws/resistor/potentiometer.html

[11] Abhishek Saini, Shikhar Rana, Simranjeet Singh, Mohit, Harpreet Kaur Channi, “Designing and Modeling of Water Level Indicator”, International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Volume 2, Issue 6, November-December 2017.

[12] Supriya R. Khaire, Revati M. Wahul, “Water Quality Data Transfer and Monitoring System in IOT Environment: A Survey”, International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Volume 2, Issue 6, November-December 2017.

A

PROJECT REPORT

On

INFORMATION HIDING IN IMAGE ENHANCEMENT

Submitted by

Mr.B.Yashwanth (17K81A0468)

Mr.K.Mohan Rao (17K81A0485)

Ms.V.Bhavana (17K81A04B9)

in partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

IN

Electronics and Communication Engineering

Under The Guidance of

Mr.D.Prasad

(B.Tech., M.Tech)

DEPARTMENT OF ECE

ST.MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the project entitled "**INFORMATION HIDING IN IMAGE ENHANCEMENT**" is being submitted by **1. Mr.B. Yashwanth**(17K81A0468), **2.Mr.K.Mohan Rao**(17K81A0485), **3.Ms. V.Bhavana**(17K81A04B9) in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology** in **Electronics and Communication Engineering** is recorded of bonafide work carried out by them. The results embodied in this report have been verified and found satisfactory.

Signature	Head of the Department
Mr. D.Prasad	**Dr.B. Hari krishna** ,
Department of Electronics and	Department of Electronics and
Communication Engineering.	Communication Engineering.

Internal Examiner	External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **B.YASHWANTH** WITH ROLL NO.**17K81A0468, K.MOHAN** WITH ROLL NO.**17K81A0485**, **V.BHAVANA** WITH ROLL NO.**17K81A04B9**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**INFORMATION HIDING IN IMAGE ENHANCEMENT**
" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of **Bachelor of Technology** in Department of Electronics and Communication Engineering, session: <2017 - 2021>, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work Entitled **"INFORMATION HIDING IN IMAGE ENHANCEMENT"** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

B Yashwanth	17K81A0468
K Mohan Rao	17K81A0485
V Bhavana	17K81A04B9

ACKNOWLEDGEMENTS

While bringing out this project report to its final form, we came across a number of people whose contributions in various ways helped my field of research and they deserve special thanks. It is a pleasure to convey our gratitude to all of them.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our college management** for their kind support and permission to use the facilities available in the institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.Santosh Kumar Patra**, our **beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna, Head & professor,** the **Dept. of Electronics and Communication Engineering** for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our **Dept. Project Coordinator** Dr. **A. Anand, Assistant Professor, Dept.of ECE** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide, Mr.**D.Prasad, Assistant Professor**, Department of ECE and our class project coordinator **Mr.Karthik,Assistant Professor**, Department of ECE for guidelines and encouragement in carrying out this project work.

We are highly grateful and obliged to our college level project coordinators **Dr.Maalydri, Professor in IT & Dr.R.Santhosh Kumar, Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and cooperation that is difficult to express in words.

1.B Yashwanth

2.K Mohan Rao

3.V Bhavana

ABSTRACT

This paper proposes an information hiding method to embed data while executing image enhancement steps. The 2D Median Filter is adapted and re-engineered to demonstrate the feasibility of this concept. In particular, the filtering embedding steps are performed for each pixel in a sliding window manner. Pixels enclosed within the predefined window (neighbourhood) are gathered, linearized and sorted. Then, the linearized pixels are divided into partitions, in which each partition is assigned to represent a certain sequence of bits. The performance of the proposed method is evaluated by using the BSD300 data set for various settings. The embedding capacity, image quality, data extraction error rate are reported and analyzed. Besides, the robustness of the proposed method against brute force attack is also discussed. In the best case scenario, when the window size is 7×7, ~ 0.97 bpp is achieved with acceptable image quality while having $\sim 3.5\%$ data extraction error rate.

TABLE OF CONTENTS

Chapter 1 INTRODUCTION

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

INTRODUCTION TO THE PROJECT

The earth is associate aquatic planet and the maximum amount as eightieth of its surface is roofed by water... Moreover, there is a strong interest in knowing what lies in underwater. Present days, an image of deep waters has a scope to large investigation to explore the underwater for sea floor expedition and navigation. Enthusiasm of underwater imaging includes the inspection of plants, seabed exploration, the search for wrecks up and to the exploration of natural resources. There were several issues faced by the human in the underwater, if he dives deep into the ocean and stay there for a long time to perform experimentation. [1]. Due to the above reasons, unmanned remote vehicles are used to sea floor exploration.

A. Historical Development

Underwater image quality improvement approaches present a path to magnify the object recognition in underwater surrounding. A heap of research started for the upgradation of image visual quality, but a little amount of work has been carried out in this area. In the deep waters, image quality is degrading due to poor illumination conditions and the light properties differ in water compared to air.[2]. There were several parameters which decreases the quality of an image in underground waters. So in order to remove all these effects
there are several techniques has been implemented and practiced.

B. Need for Pre process

Initially processing is necessary for deep water images due to their poor-quality during acquisition. Necessity for pre-processing of deep-water images [1] are discussed below:

(i) Quality of images taken from deep water is deteriorated due to light ray attributes like scattering and absorption of light.

(ii) Specificity of surroundings such as lighting inequalities, water torridness, and blue complexion is more or less influential when vehicles move.

(iii) Video or image captured from deep waters like unknown rigid scene, and the depth of the scene and low light sensitivity due to Marine snow etc.

TRADITIONAL TECHNIQUES FOR IMAGE ENHANCEMENT

There are several techniques which are used very frequently for processing the image to improve the visual quality. Some of them are as follows:

(i) Contrast Stretching (ii) Adaptive Histogram Equalization

A. Contrast stretching

The contrast stretching is a method to transform high intense region of image into brighter and less intense region into darker by using a predefined transformation function T(r) [2]. Generally, the underwater images will have less grey values. There are 256 grey values. '0' indicates black and '255' indicates white. In this method the current grey value of the image is stretched towards 255 i.e., from black to white, pixel by pixel. That means the contrast of the image is stretched, so that the quality of the image is improved for better vision.

For example:

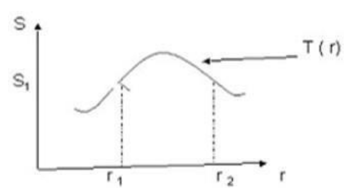


Fig.1.2 Two different gray levels look same

Here two different thresholds are considered for the entire image and the values between them are stretched to the maximum extent, so that the contrast increases. And more over by this method the entire global image contrast is enhanced.

Fig.1.2.1 (a) Raw image **(b) Enhanced image**

But the disadvantage here is that the transformation function is not unique. Depending on the application the suitable transformation function is chosen.

B. Adaptive histogram equalization

Adaptive histogram equalization is a PC based image processing technique which is used to improve the quality of image properties like contrast. It is similar to contrast stretching method but with a slight difference. It computes several intensities of specific gray value, each corresponding to a distinct portion of an image, and with the help of them intensities are rearranged by applying a suitable transformation function. For example, a simple transformation function such as each pixel transformed based on the histogram of a square surrounding the pixel [3]. Existing values will be mapped to new values keeping actual number of intensities in the resulting image equal or less than the original number of intensities. The transformation function applied on the histogram is proportional to the cumulative distributive function (CDF) of pixel values in the neighbourhood. Therefore it suits for enhancing the local details and enhancing the edge information of each region of an image.

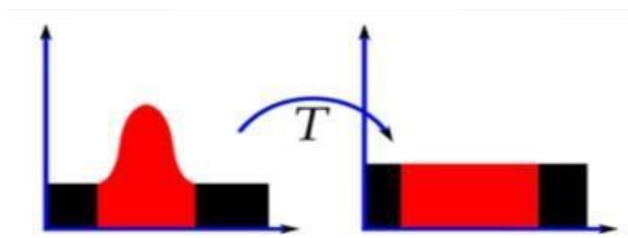


Fig.1.2.2. Histograms of an image before and after histogram equalization

Histogram equalization is a technique for changing the overall pixel intensities based on transformation function and contrast of an image. Histogram equalization is an effective technique which will benefit for the images with extreme contrast values. The limitation of this technique highlights the unwanted noise present in the background of an image and lead to loss in the information signal. It results in undesired effects in the resultant images [4].

Fig.1.2.3. (a) Raw image (b) Enhanced image

Here the noise in relatively homogeneous regions of the image are amplified which results in poor SNR. And also, only the local objects of the image are enhanced and the background is left unenhanced.

Image enhancement, as its name implies, improves the quality of image by means of emphasizing the desired details or removing/suppressing the undesired noisy/irrelevant information. Image enhancement plays a crucial role in digital image processing, and it has become one of the processes that takes place in any smart device after an image is captured. Some commonly deployed image enhancement techniques include contrast enhancement, image sharpening, smoothing, noise removal, to name a few. Each enhancement technique can either act as a standalone process to enhance the appearance of an image, or utilized as a pre-processing step before further analysis is carried out. In fact, image enhancement (commonly known as filter) is often utilized to enhance image quality before it is shared online on social media or communicated for consumption by a wider audience. However, image enhancement is particularly important when the received original/source image (e.g., from field reporter, amateur photographer) contains noise. While the noisy image is enhanced, the processes involved are not published or shared. One can also imagine the need to share specific tone-mapping operators as well as the parameters used when shrinking the dynamic range of a noisy high dynamic range image into the standard dynamic range. To accommodate this additional data, information hiding appears to be one of the commonly implemented solutions.

INTRODUCTION TO IMAGE PROCESSING

1.3.1. IMAGE:

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person. Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Fig 1.3.1(a) General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

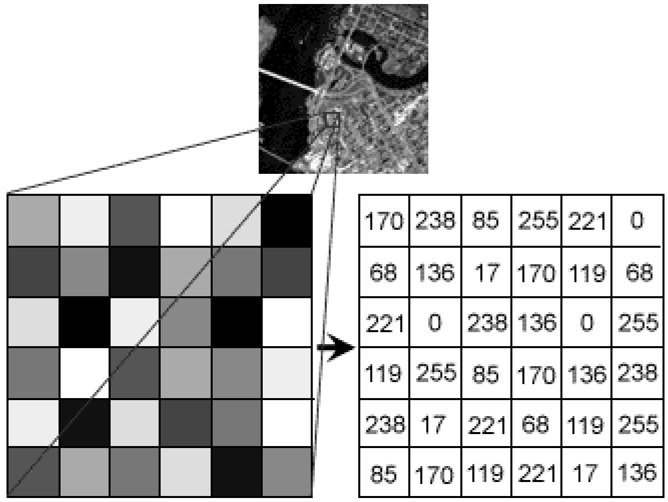


Fig 1.3.1(b) Image pixel

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
Transparency	Red	Green	Blue

Fig 1.3.2 Transparency image

IMAGE FILE SIZES:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

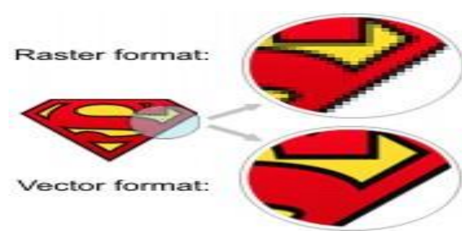


Fig1.3.3 Resolution image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

RASTER FORMATS:

These formats store images as bitmaps (also known as pixmaps)

- **JPEG/JFIF:**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobes PDF files.

- **EXIF:**

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

- **TIFF:**

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

- **PNG:**

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

- **GIF:**

GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

- **BMP:**

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM:**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

- **SVG:**

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all-purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

IMAGE PROCESSING:

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSIN

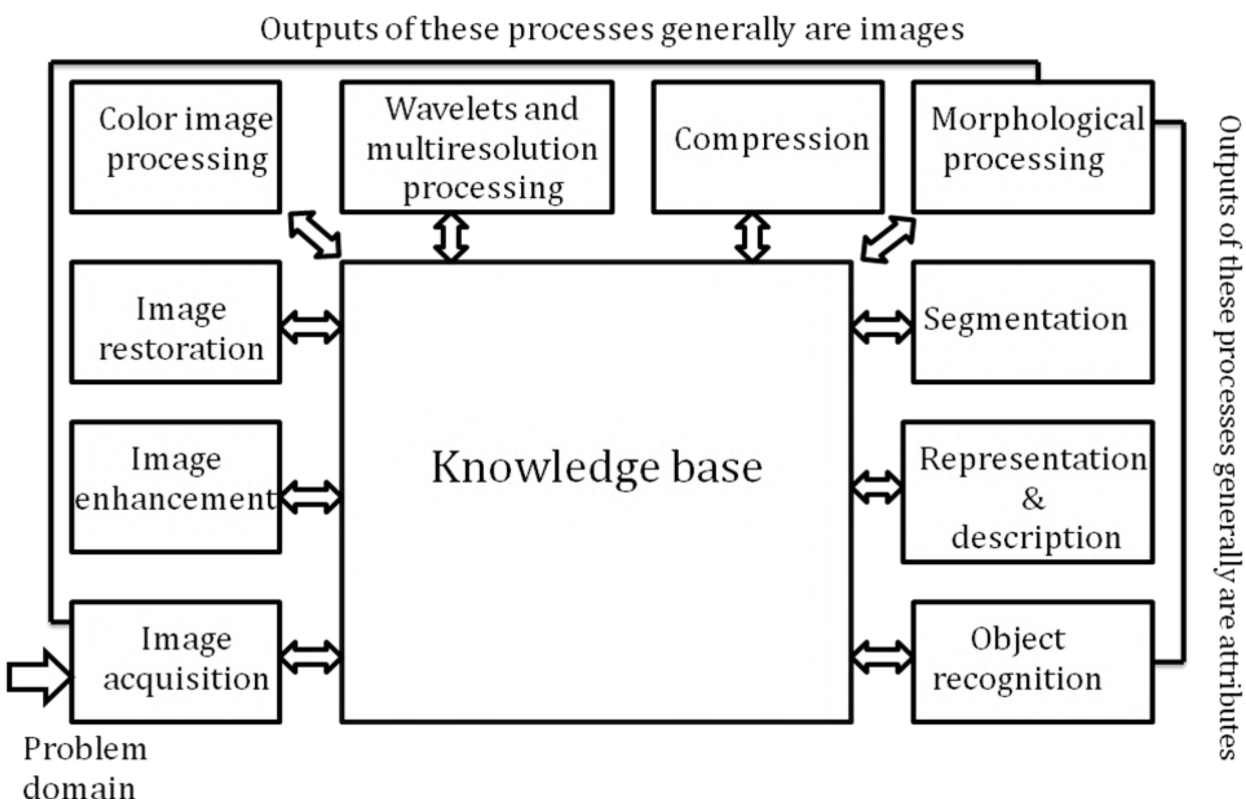


Fig 1.4.1 Image fundamental

Image Acquisition:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Fig:1.4.2 Digital camera image

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Image Enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because "it looks better." It is important to keep in mind that enhancement is a very subjective area of image processing.

Fig 1.4.3 Image enhancement

Image restoration:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Fig 1.4.4 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, whereas removal of image blur by applying a deblurring function is considered a restoration technique.

Color image processing:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Fig 1.4.5 Color & Gray scale image

Wavelets and multiresolution processing:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform-based image processing since the late1950's, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

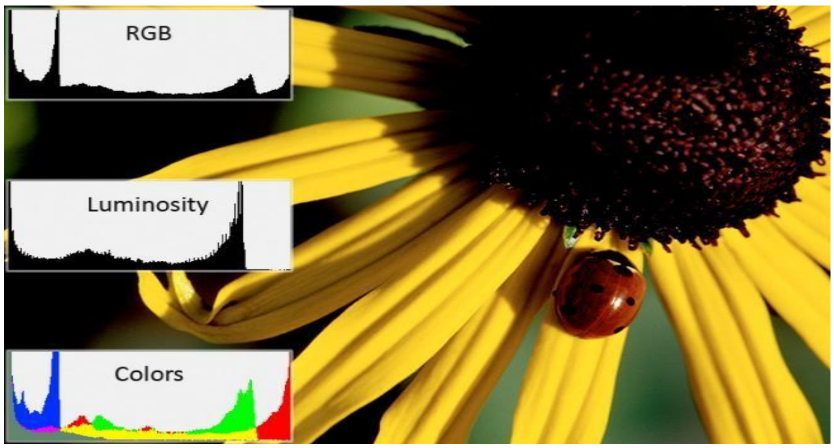


Fig 1.4.6 rgb histogram image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called Multiresolution theory. Multiresolution theory incorporates and unifies techniques from a variety of

disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

Compression:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

Morphological processing:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Fig 1.4.8 blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

Segmentation:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

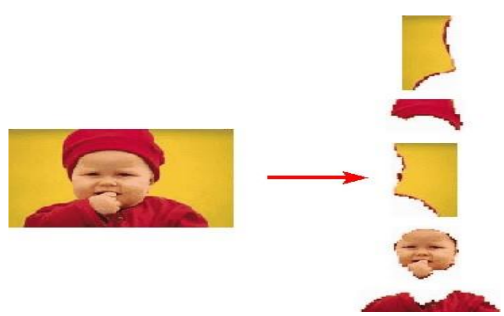


Fig 1.4.9 Image segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

Representation and description:

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

Object recognition:

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

Knowledgebase:

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an inter related to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

COMPONENTS OF AN IMAGE PROCESSING SYSTEM:

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

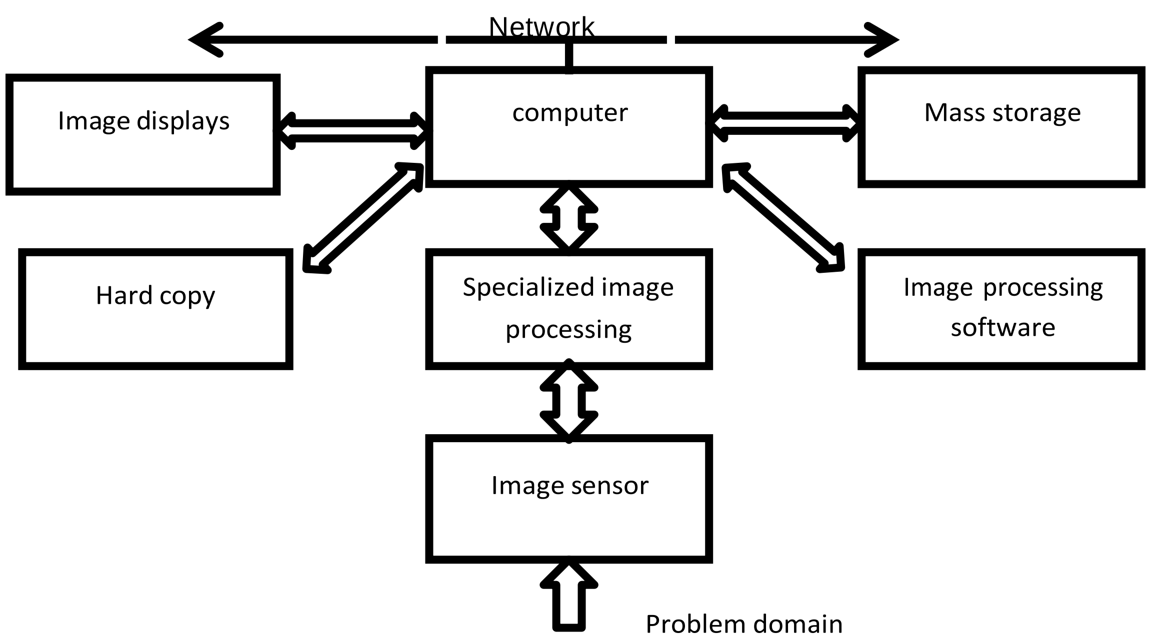


Fig 1.5 Component of image processing

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 1.24 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

- **Image sensors:**

With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

- **Specialized image processing hardware:**

Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

- **Computer:**

The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

- **Image processing software:**

Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

- **Mass storage:**

Mass storage capability is a must in image processing applications. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage for image processing applications falls into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), Gbytes (meaning giga, or one billion, bytes), and Tbytes (meaning tera, or one trillion, bytes).

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig. 1.24. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in "jukeboxes" are the usual media for archival applications.

- **Image displays:**

Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are there requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

- **Hardcopy:**

Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

- **Network:**

Networking is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of optical fiber and other broadband technologies. Color and texture are two low-level features widely used for image classification, indexing and retrieval. Color is usually represented as a histogram, which is a first order statistical measure that captures global distribution of color in an image One of the main drawbacks of the histogram- based approaches is that the spatial distribution and local variations in color are ignored. Local spatial variation of pixel intensity is commonly used to capture texture information in an image. Grayscale Co-occurrence Matrix (GCM) is a well-known method for texture extraction in the spatial domain. A GCM stores the number of pixel neighborhoods in an image that have a particular grayscale combination. Let I be an image and let p and Np respectively denote any arbitrary pixel and its neighbor in a given direction. If GL denotes the total number of quantized gray levels and gl denotes the individual gray levels, where, gl$\in$ {0, . . .,GL _ 1}, then each component of GCM can be written as follows:

$$\mathrm{gcm}(i,j) = \Pr((\mathrm{gl}_{\mathrm{p}}, \mathrm{gl}_{N_{\mathrm{p}}}) = (i,j))$$

gcm (i, j) is the number of times the gray level of a pixel p denoted by glp equals i, and the gray level of its neighbor Np denoted by glNp equals j, as a fraction of the total number of pixels in the image.

Thus, it estimates the probability that the gray level of an arbitrary pixel in an image is i, and that of its neighbor is j. One GCM matrix is generated for each possible neighborhood direction, namely, 0, 45, 90 and 135.Average and range of 14 features like Angular Second Moment, Contrast, Correlation, etc., are generated by combining all the four matrices to get a total of 28 features. In the GCM approach for texture extraction, color information is completely lost since only pixel gray levels are considered.

To incorporate spatial information along with the color of image pixels, a feature called color correlogram has recently been proposed. It is a three-dimensional matrix that represents the probability of finding pixels of any two given colors at a distance 'd' apart Auto correlogram is a variation of correlogram, which represents the probability of finding two pixels with the same color at a distance 'd' apart. This approach can effectively represent color distribution in an image. However, correlogram features do not capture intensity variation Many image databases often contain both color as well as gray scale images. The color correlogram method does not constitute a good descriptor in such database.

Another method called Color Co-occurrence Matrix (CCM) has been proposed to capture color variation in an image. CCM is represented as a three-dimensional matrix, where color pair of the pixels p and Np are captured in the first two dimensions of the matrix and the spatial distance 'd' between these two pixels is captured in the third dimension. This approach is a generalization of the color correlogram and reduces to the pure color correlogram for d = 1. CCM is generated using only the Hue plane of the HSV (Hue, Saturation and Intensity Value) color space.

The Hue axis is quantized into HL number of levels. If individual hue values are denoted by hl, where $\mathrm{hl} \in \{0, \ldots, \mathrm{HL} - 1\}$, then each component of CCM can be written as follows:

$$\mathrm{ccm}(i, j) = \Pr((\mathrm{hl}_{\mathrm{p}}, \mathrm{hl}_{N_{\mathrm{p}}}) = (i, j))$$

Four matrices representing neighbors at angles 0, 90, 180 and 270 are considered. This approach was further extended by separating the diagonal and the non-diagonal components of CCM to generate a Modified Color Co-occurrence Matrix (MCCM). MCCM, thus, may be written as follows: MCCM = (CCMD;CCMND)

Here, CCMD and CCMND correspond to the diagonal and off-diagonal components of CCM. The main drawback of this approach is that, like correlogram, it also captures only color information and intensity information is completely ignored.

An alternative approach is to capture intensity variation as a texture feature from an image and combine it with color features like histograms using suitable weights. One of the challenges of this approach is to determine suitable weights since these are highly application-dependent. In certain applications like Content-based Image Retrieval (CBIR), weights are often estimated from relevance feedback given by users.

While relevance feedback is sometimes effective, it makes the process of image retrieval user-dependent and iterative. There is also no guarantee on the convergence of the weight-learning algorithms. In order to overcome these problems, researchers have tried to combine color and texture features together during extraction.

In the Single-channel Co-occurrence Matrix (SCM), variations for each color channel, namely, L, U and V are considered independently. In the Multi-channel Co-occurrence Matrix (MCM), variations are captured taking two channels at a time – UV, LU and LV. Since the LUV color space separates out chrominance (L and U) from luminance (V), SCM in effect, generates one GCM and two CCMs from each image independently. As a result, correlation between the color channels is lost

However, in MCM, the count of pair wise occurrences of the values of different channels of the color space is captured. Thus, each component of MCM can be written as follows:

mcmUV(i; j) = Pr((up; vane) = (i; j))

mcmLU(i; j) = Pr((lp; uNp) = (i; j))

mcmLV(i; j) = Pr((lp; vNp) = (i; j))

Here, mcmUV (i, j) is the number of times the U chromaticity value of a pixel p denoted by up equals i, and the V chromaticity value of its neighbor Np denoted by vNp equals j, as a fraction of the total number of pixels in the image. Similarly, mcmLU (i, j) and mcmLV (i, j) are defined. One MCM matrix is generated for each of the four neighborhood directions, namely, 0, 45, 90 and 135.

Deng and Manjunath (2001) proposed a two-stage method called JSEG, which combines color and texture after image segmentation. In the first stage, colors are quantized to the required levels for differentiating between various regions of an image. Pixel values of the regions are then replaced by their quantized color levels to form a color map.

Spatial variation of color levels between different regions in the map is viewed as a type of texture composition of the image.

Local Fourier Transformation (LFT) coefficients. Eight templates equivalent to LFT are operated over an image to generate a characteristic map of the image. Each template is a $3 \cdot 3$ filter that considers eight neighbors of the current pixel for LFT calculation. First and second order moments of the characteristic map are then used to generate a set of features.

In this paper, we propose an integrated approach for capturing spatial variation of both color and intensity levels in the neighborhood of each pixel using the HSV color space. In contrast to the other methods, for each pixel and its neighbor, the amount of color and intensity variation between them is estimated using a weight function. Suitable constraints are satisfied while choosing the weight function for effectively relating visual perception of color and the HSV color space properties. The color and intensity variations are represented in a single composite feature known as Integrated Color and Intensity Co-occurrence Matrix (ICICM). While the existing schemes generally treat color and intensity separately, the proposed method provides a composite view to both color and intensity variations in the same feature. The main advantage of using ICICM is that it avoids the use of weights to combine individual color and texture features. We use ICICM feature in an image retrieval application from large image databases.

Early result on this work was reported in (Vadivel et al., 2004a). In the next section, we describe the proposed feature extraction technique after introducing some of the properties of the HSV color space. Choice of quantization levels for color and intensity axes, selection of parameter values and a brief overview of the image retrieval application

1.5.1 Integrated color and intensity co-occurrence matrix:

We propose to capture color and intensity variation around each pixel in a two-dimensional matrix called Integrated Color and Intensity Co-occurrence Matrix (ICICM). This is a generalization of the Grayscale Co-occurrence Matrix and the Color Co-occurrence Matrix techniques. For each pair of neighboring pixels, we consider their contribution to both color perception as well as gray level perception to the human eye. Some of the useful properties of the HSV color space and their relationship to human color perception are utilized for extracting this feature. In the next sub-section, we briefly explain relevant properties of the HSV color space. In the subsequent subsection, we describe how the properties can be effectively used for generating ICICM.

HSV color space:

HSV Color space: Basically, there are three properties or three dimensions of color that being hue, saturation and value HSV means Hue, Saturation and Value. It is important to look at because it describes the color based on three properties. It can create the full spectrum of colors by editing the HSV values. The first dimension is the Hue. Hue is the other name for the color or the complicated variation in the color. The quality of color as determined by its dominant wavelength. This Hue is broadly classified into three categories. They are primary Hue, Secondary Hue and Teritiary Hue.

The first and the foremost is the primary Hue it consists of three colors they are red, yellow and blue. The secondary Hue is formed by the combination of the equal amount of colors of the primary Hue and the colors of the secondary Hue which was formed by the primary Hue are Orange, Green and violet. The remaining one is the teritiary Hue is formed by the combination of the primary Hue and the secondary Hue. The limitless number of colors are produced by mixing the colors of the primary Hue in different amounts. Saturation is the degree or the purity of color. Then the second dimension is the saturation. Saturation just gives the intensity to the colors.

The saturation and intensity drop just by mixing the colors or by adding black to the color. By adding the white to the color in spite of more intense the color becomes lighter. Then finally the third dimension is the Value. The value is the brightness of the color. When the value is zero the color space is totally black with the increase in the color there is also increase in the brightness and shows the various colors. The value describes the contrast of the color. That means it describes the lightness and darkness of the color. As similar to the saturation this value consists of the tints and shades. Tints are the colors with the added white and shades are the colors with the added black.

Properties of the HSV color space:

Sensing of light from an image in the layers of human retina is a complex process with rod cells contributing to scotopic or dim-light vision and cone cells to photopic or bright-light vision (Gonzalez and Woods, 2002). At low levels of illumination, only the rod cells are excited so that only gray shades are perceived. As the illumination level increases, more and more cone cells are excited, resulting in increased color perception.

Various color spaces have been introduced to represent and specify colors in a way suitable for storage, processing or transmission of color information in images. Out of these, HSV is one of the models that separate out the luminance component (Intensity) of a pixel color from its chrominance components (Hue and Saturation). Hue represents pure color, which is perceived when incident light is of sufficient illumination and contains a single wavelength. Saturation gives a measure of the degree by which a pure color is diluted by white light. For light with low illumination, corresponding intensity value in the HSV color space is also low.

The HSV color space can be represented as a Hexa cone, with the central vertical axis denoting the luminance component, I (often denoted by V for Intensity Value). Hue, is a chrominance component defined as an angle in the range [0,2p] relative to the red axis with red at angle 0, green at 2p/3, blue at 4p/3 and red again at 2p. Saturation, S, is the other chrominance component, measured as a radial distance from the central axis of the hexacone with value between 0 at the center to 1 at the outer surface. For zero saturation, as the intensity is increased, we move from black to white through various shades of gray.

On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the purest form of the color represented by its hue. When saturation is near 0, all the pixels in an image look alike even though their hue values are different.

As we increase saturation towards 1, the colors get separated out and are visually perceived as the true colors represented by their hues. Low saturation implies presence of a large number of spectral components in the incident light, causing loss of color information even though the illumination level is sufficiently high. Thus, for low values of saturation or intensity, we can approximate a pixel color by a gray level while for higher saturation and intensity, the pixel color can be approximated by its hue. For low intensities, even for a high saturation, a pixel color is close to its gray value. Similarly, for low saturation even for a high value of intensity, a pixel is perceived as gray. We use these properties to estimate the degree by which a pixel contributes to color perception and gray level perception.

One possible way of capturing color perception of a pixel is to choose suitable thresholds on the intensity and saturation. If the saturation and the intensity are above their respective thresholds, we may consider the pixel to have color dominance; else, it has gray level dominance. However, such a hard thresholding does not properly capture color perception near the threshold values. This is due to the fact that there is no fixed level of illumination above which the cone cells get excited. Instead, there is a gradual transition from scotopic to photopic vision. Similarly, there is no fixed threshold for the saturation of cone cells that leads to loss of chromatic information at higher levels of illumination caused by color dilution. We, therefore, use suitable weights that vary smoothly with saturation and intensity to represent both color and gray scale perception for each pixel

NON-INTERVAL:

Due to the large range for each component by directly calculating the characteristics for the retrieval then the computation will be very difficult to ensure rapid retrieval. It is essential to quantify HSV space component to reduce computation and improve efficiency. At the same time, because the human eye to distinguish colors is limited, do not need to calculate all segments. Unequal interval quantization according the human color perception has been applied on H, S, V components.

Based on the color model of substantial analysis, we divide color into eight parts. Saturation and intensity is divided into three parts separately in accordance with the human eyes to distinguish. In accordance with the different colors and subjective color perception quantification, quantified hue(H), saturation(S) and value(V)

In accordance with the quantization level above, the H, S, V three-dimensional feature vector for different values of with different weights to form one dimensional feature vector and is given by the following equation:

$$G = Qs*Qv*H+Qv*s+V$$

Where Qs is the quantized series of S and Qv is the quantized series of V. And now by setting Qs = Qv = 3, Then G = 9H+3S+V

$$H = \begin{cases} 0 & if \quad h \in [316,20] \\ 1 & if \quad h \in [21,40] \\ 2 & if \quad h \in [41,75] \\ 3 & if \quad h \in [76,155] \\ 4 & if \quad h \in [156,190] \\ 5 & if \quad h \in [191,270] \\ 6 & if \quad h \in [271,295] \\ 7 & if \quad h \in [296,315] \end{cases} \qquad S = \begin{cases} 0 & if \quad s \in [0,0.2) \\ 1 & if \quad s \in [0.2,0.7) \\ 2 & if \quad s \in [0.7,1) \end{cases} \qquad V = \begin{cases} 0 & if \quad v \in [0,0.2) \\ 1 & if \quad v \in [0.2,0.7) \\ 2 & if \quad v \in [0.7,1) \end{cases}$$

In this way three component vector of the HSV from one dimensional vector, which quantize the whole color space for the 72 kinds of the main colors. So we can handle 72 bins of one dimensional histogram.

This qualification is effective in reducing the images by the effect of the light intensity, but also reducing the computational time and complexity.

IMAGE RETRIEVAL:

Image retrieval is nothing but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords or the descriptions to the images so that the retrieval can be performed. Manual image annotation is time consuming, expensive and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc...,

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others. one can today generate volumes of images with content as diverse as family get-togethers and national park visits. Low-cost storage and easy Web hosting has fueled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchable image data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real -world image search system designers. As far as technological advances are concerned, growth in

Content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real world implications, applications, and constraints of the technology. Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future.

An image retrieval system designed to serve a personal collection should focus on features such as personalization, flexibility of browsing, and display methodology. For example, Google's Picasa system [Picasa 2004] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods.

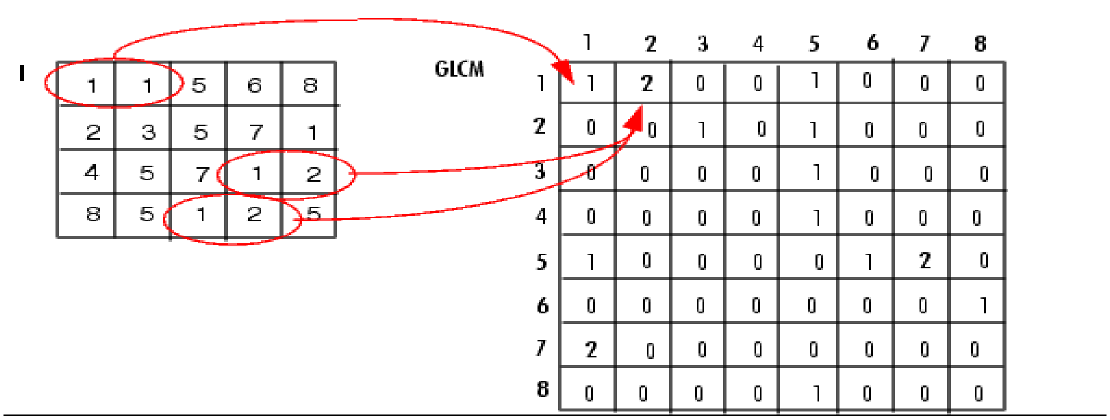
The notable few applications include an FPGA implementation of a color-histogram-based image retrieval system [Kofola's and Andreadis 2003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 2003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 2002].

Discussion. Regardless of the nature of the collection, as the expected user-base grows, factors such as concurrent query support, efficient caching, and parallel and distributed processing of requests become critical. For future real-world image retrieval systems, both software and hardware approaches to address these issues are essential. More realistically, dedicated specialized servers, optimized memory and storage support, and highly parallelizable image search algorithms to exploit cluster computing powers are where the future of large-scale image search hardware support lies.

OVERVIEW OF TEXTURE:

We all know about the term Texture but for defining it is a hard time. One can differentiate the two different Textures by recognizing the similarities and differences. Commonly there are three ways for the usage of the Textures:

Based on the Textures the images can be segmented to differentiate between already segmented regions or to classify them. We can reproduce Textures by producing the descriptions. The texture can be analyzed in three different ways. They are Spectral, Structural and Statistical:



1.6.4 overview of texture

DIGITAL IMAGE PROCESSING

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An important characteristic underlying the design of image processing systems is the significant level of testing & experimentation that normally is required before arriving at an acceptable solution. This characteristic implies that the ability to formulate approaches &quickly prototype candidate solutions generally plays a major role in reducing the cost & time required to arrive at a viable system implementation.

What is DIP

An image may be defined as a two-dimensional function f(x, y), where x & y are spatial coordinates, & the amplitude off at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital image by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensor, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis& computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting & somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision. There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes.

Low-level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low- level process is characterized by the fact that both its inputs & outputs are images.

Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images. Finally higher- level processing involves “Making sense” of an ensemble of recognized objects, as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

What is an image?

An image is represented as a two-dimensional function f (x, y) where x and y are spatial co-ordinates and the amplitude of ‘f’ at any pair of coordinates (x, y) is called the intensity of the image at that point.

1. Gray scale image:

A grayscale image is a function I (xylem) of the two spatial coordinates of the image plane.

I(x, y) is the intensity of the image at the point (x, y) on the image plane.

I (xylem) takes non-negative values assume the image is bounded by a rectangle $[0, a] \times [0, b]$ I: $[0, a] \times [0, b] \rightarrow [0, \text{info})$

2. Color image:

It can be represented by three functions, R (xylem) for red, G (xylem) for green *and* B (xylem) for blue.

An image may be continuous with respect to the x and y coordinates and also in amplitude.

Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate's values is called sampling. Digitizing the amplitude values is called quantization.

3. Coordinate convention:

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image f (x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size M X N. The values of the coordinates (xylem) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates.

In many images processing books, the image origin is defined to be at (xylem)= (0,0). The next coordinate values along the first row of the image are (xylem)= (0,1). It is important to keep in mind that the notation (0,1) is used to signify the second sample along the first row.

It does not mean that these are the actual values of physical coordinates when the image was sampled. Following figure shows the coordinate convention. Note that x ranges from 0 to M-1 and y from 0 to N-1 in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (xylem) the toolbox uses the notation (race) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate topples, (alb), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentation refers to the coordinates. Less frequently the toolbox also employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refers to rows. This is the opposite of our use of variables x and y.

Image as Matrices:

The preceding discussion leads to the following representation for a digitized image function:

f (0,0)   f (0,1)   ………   f (0, N-1)

f (1,0)   f (1,1)   …………   f (1, N-1)

f (xylem)=   .   .   .

.   .   .

f (M-1,0)   f(M-1,1)................f (M-1, N-1)

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements.

A digital image can be represented naturally as a MATLAB matrix:

f (1,1)   f (1,2)..........f (1, N)

f (2,1)   f (2,2).........f (2, N)

.   .   .

f =   .   .   .

f (M,1)   f(M,2).........f (M, N)

Where f (1,1) = f(0,0) (note the use of a monoscope font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation f (p, q) denotes the element located in row p and the column q. For example, f (6,2) is the element in the sixth row and second column of the matrix f. Typically we use the letters M and N respectively to denote the number of rows and columns in a matrix. A 1xN matrix is called a row vector whereas an Mx1 matrix is called a column vector. A 1x1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using mono-scope characters. We use conventional Roman, italic notation such as f (x, y), for mathematical expressions.

Reading Images:

Images are read into the MATLAB environment using function imread whose syntax is

Imread ('filename')

Format name	**Description**	**recognized extension**
TIFF	Tagged Image File Format	.tif, .tiff
JPEG	Joint Photograph Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	. xwd

Table 1.7.4 Reading Images

Here filename is a spring containing the complete of the image file (including any applicable extension). For example, the command line " >> f = imread ('8. jpg')";

Reads the JPEG (above table) image chest Xray into image array f. Note the use of single quotes (') to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output If a semicolon is not included.

Data Classes:

Although we work with integers coordinates the values of pixels themselves are not restricted to be integers in MATLAB. Table above list various data classes supported by MATLAB and IPT are representing pixels values. The first eight entries in the table are refers to as numeric data classes. The ninth entry is the char class and, as shown, the last entry is referred to as logical data class.

Name	**Description**
Double	Double _ precision, floating_ point numbers the Approximate.
Uint8	unsigned 8_bit integer in the range [0,255] (1byteper Element).
Uint16	unsigned 16_bit integers in the range [0, 65535] (2byte per element).
Uint 32	unsigned 32_bit integers in the range [0, 4294967295]
Int 16	signed 16_byte integers in the range [32768, 32767] (2 bytes per element).
Single	_precision floating _point numbers with values
Char	characters (2 bytes per elements).

TABLE 1.7.5 DATA CLASSSES

All numeric computations in MATLAB are done in double quantities, so this is also a frequent data class encounter in image processing applications. Class unit 8 also is encountered frequently, especially when reading data from storages devices, as 8 -bit images are most common representations found in practice. These two data classes, classes logical, and, to a lesser degree, class unit 16 constitute the primary data classes on which we focus. Many ipt functions however support all the data classes listed in table. Data class double requires 8 bytes to represent a number uint8 and int 8 require one byte each, uint16 and int16 requires 2bytes and unit 32.

Int 32 and single required 4 bytes each. The char data class holds characters in Unicode representation. A character string is merely a 1*n array of characters logical array contains only the values 0 to 1, with each element being stored in memory using function logical or by using relational operators.

Image Types:

The toolbox supports four types of images:

1 .Intensity images;

2. Binary images;

3. Indexed images;

4. R G B images.

Most monochrome image processing operations are carried out using binary or intensity images, so our initial focus is on these two image types. Indexed and RGB colour images.

1.Intensity Images:

An intensity image is a data matrix whose values have been scaled to represent intentions. When the elements of an intensity image are of class unit8, or class unit 16, they have integer values in the range [0,255] and [0, 65535], respectively. If the image is of class double, the values are floating point numbers. Values of scaled, double intensity images are in the range [0, 1] by convention.

2. Binary Images:

Binary images have a very specific meaning in MATLAB.A binary image is a logical array 0s and1s.Thus, an array of 0s and 1s whose values are of data class, say unit8, is not considered as a binary image in MATLAB .A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, we create an array B using the statement.

B=logical (A)

If A contains elements other than 0s and 1s. Use of the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.

Using relational and logical operators also creates logical arrays.

To test if an array is logical, we use the I logical function: is logical(c).

If c is a logical array, this function returns a 1.Otherwise returns a 0. Logical array can be converted to numeric arrays using the data class conversion functions

3. Indexed Images:

An indexed image has two components:

A data matrix integer, x

A color map matrix, map

Matrix map is an m*3 arrays of class double containing floating point values in the range [0, 1]. The length m of the map is equal to the number of colors it defines. Each row of map specifies the red, green and blue components of a single color. An indexed images uses “direct mapping” of pixel intensity values color map values. The color of each pixel is determined by using the corresponding value the integer matrix x as a pointer in to map. If x is of class double, then all of its components with values less than or equal to 1 point to the first row in map, all components with value 2 point to the second row and so on. If x is of class units or unit 16, then all components value 0 point to the first row in map, all components with value 1 point to the second and so on.

4.RGB Image:

An RGB color image is an M*N*3 array of color pixels where each color pixel is triplet corresponding to the red, green and blue components of an RGB image, at a specific spatial location. An RGB image may be viewed as “stack” of three gray scale images that when fed in to the red, green and blue inputs of a color monitor

Produce a color image on the screen. Convention the three images forming an RGB color image are referred to as the red, green and blue components images. The data class of the components images determines their range of values. If an RGB image is of class double the range of values is [0, 1].

Similarly, the range of values is [0,255] or [0, 65535]. For RGB images of class units or unit 16 respectively. The number of bits use to represents the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8bit image, the corresponding RGB image is said to be 24 bits deep.

Generally, the number of bits in all component images is the same. In this case the number of possible colors in an RGB image is (2^b) ^3, where b is a number of bits in each component image. For the 8bit case the number is 16,777,216 colors

Chapter 2

LITERATURE SURVEY

Literature Review on research area:

Steganographic methods on spatial domains using the XOR operator. Message insertion is performed with two first XOR operations imposed on the 1st and 8th bits and the second is on the 2nd, 7th bit. The result of the operation is then compared and used as a rule for inserting the message. The cover image used is a 512 * 512 grayscale image and three message sizes, 1024 bits, 2048 bits and 4096 bits. The value of PSNR obtained is about 69 dB with a message length of 4096 bits. proposing steganography on the image using the differencing bits technique on the 5th and 6th bits. If there is a difference in the 5th and 6th bits equal to the bits of the secret information then no change is made. Meanwhile, if there is a difference in value, the value changes to the 5th bit so that the value of the difference corresponds to the bit value in the secret information. The cover image used is grayscale image and colour image with size 512 * 512.

With this method obtained PSNR 51.17977 dB on Lena grayscale image and PSNR 52.3438 on Lena colour image, with a payload capacity of 262144 bits message proposed a combination of steganographic and cryptographic methods, with messages encrypted using the OTP method before being embedded on the LSB. To improve imperceptibility and secure embedding of messages is done on the image edge area. Image edge area detection is done by the Canny method. The cover image used is type JPEG with the size of 11035 bytes, while the message pinned 1024 bytes obtained PSNR 69.1106 dB.

In addition, the stego image quality is also measured by a histogram, where the cover image histogram and stego image are identical also proposes embedding messages in the image edge area. In his research combined steganography techniques using LSB and cryptographic techniques using DES. Before the image message is inserted, the message is encrypted using the DES method. The cover image used is a color image with the size 1024 * 1024, while the message is also a color image with size 64 * 64. The encrypted message is embedded in the image edge area detected by the Canny method.

Based on the results of this research, the average value of PSNR 72.21584 dB, which is obtained from five kinds of imagery.

Review of related literature:

Least Significant Bit (LSB) is a very popular method in the spatial domain of steganographic images. This method is widely used and continues to be developed to date, because of its advantages in steganographic image quality. However, the traditional LSB method is very simple and predictable. It needs a way to improve the security of hidden messages in this way. This research proposes a simple and safe way to hide messages in LSB techniques. Three times the XOR operation is done to encrypt the message before it is embedded on the LSB. To facilitate the process of encryption and decryption of messages, three MSB bits are used as keys in XOR operations. The results of this study prove that this method provides security to messages with very simple operation. The imperceptibility quality of the stego image is also excellent with a PSNR value above 50 db.There are many conventional information hiding methods, including Least Significant Bit (LSB) insertion, Histogram Shifting (HS) and Difference Expansion (DE), to name a few.

These techniques are studied extensively and many of their variants are proposed [1, 2, 3] to achieve a balance trade-off among embedding capacity, image quality, robustness against attacks, etc. However, these techniques are not widely adopted into the usual operations performed by the users, and often they are implemented as an additional step after the image is processed. In most cases, the user has to explicitly install or develop the data embedding algorithm to enable data embedding into the image of interest. Therefore, in this paper, we design an information hiding method as part of the image enhancement process. In other words, data can be inserted into the image while executing the image enhancement steps. As a proof of concept, the proposed method is demonstrated by using the Median Filter.

Conclusion on Reviews:

However, these techniques are not widely adopted into the usual operations performed by the users, and often they are implemented as an additional step after the image is processed.

In most cases, the user has to explicitly install or develop the data embedding algorithm to enable data embedding into the image of interest. Therefore, in this paper, we design an information hiding method as part of the image enhancement process. In other words, data can be inserted into the image while executing the image enhancement steps. As a proof of concept, the proposed method is demonstrated by using the Median Filter.

Chapter 3

Proposed system

Block Diagram:

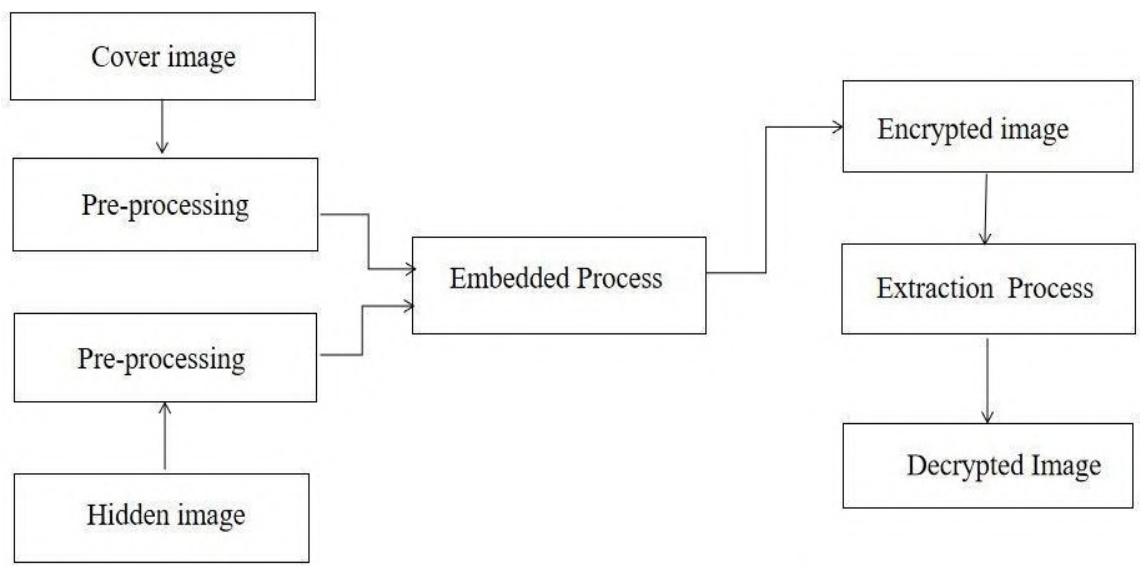


Fig :3.1 Block Diagram of the Proposed System

Median filter is commonly utilized for noise removal in image enhancement. Specifically, median filter focuses on a target pixel X, and considers the w × w-neighborhood pixels surrounding X.

These pixels are sorted, and the middle value is selected to replace the target pixel. The process is repeated for all pixels in the image.

This technique is feasible because noises are abrupt changes in an image and most natural images have more smooth areas in comparison to edges.

Median filter is particularly effective for salt-and-pepper and speckle noises.

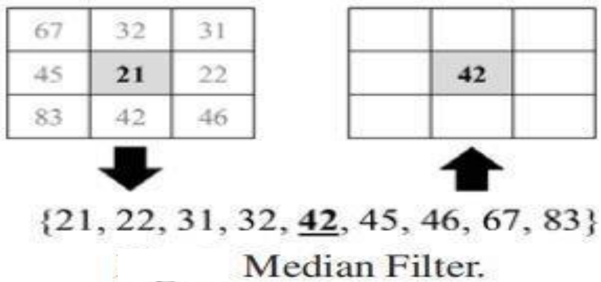
An example of median filter with w = 3, i.e., 3 × 3 neighborhood. Here, the neighborhood

N = {67, 32, 31; 45, 21, 22; 83, 42, 46}.

After the sorting step, the pixels are arranged in ascending order to produce

N = {21, 22, 31, 32, 42, 45, 46, 67, 83}.

Next, the middle value 42 (i.e., 5-th pixel value in this case) is selected to replace the initial target pixel, i.e., 21



Flow Chart:

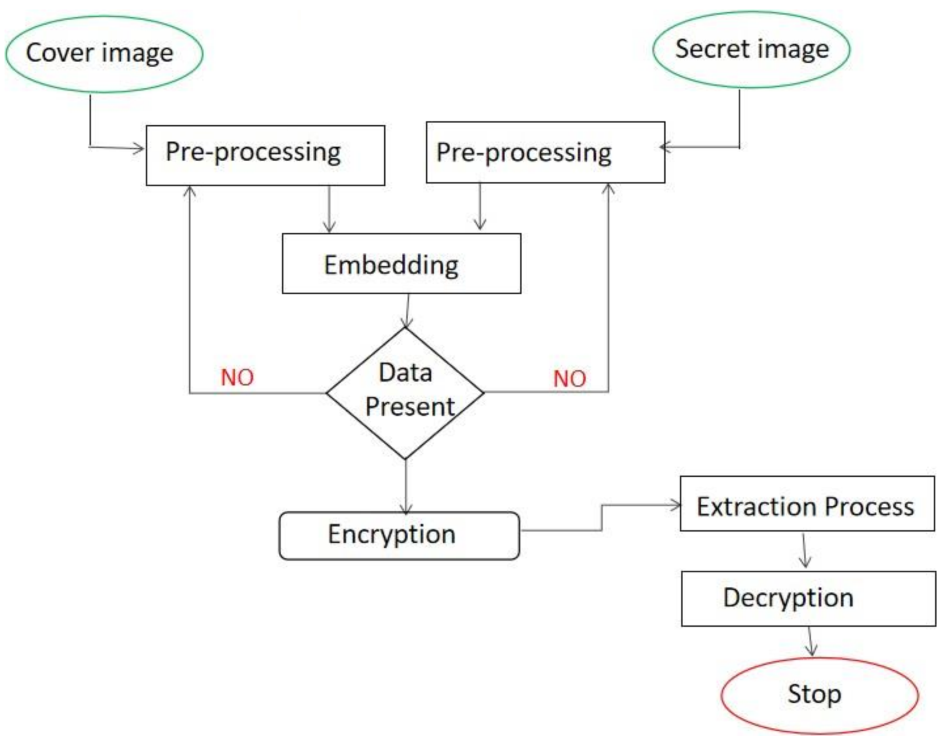


Fig 3.2 Flow Chart of image steganography

Data Embedding:

We re-engineer the conventional Median Filter algorithm to hide data expressed in binary representation. To ease the presentation, let M and N be the number of rows and columns in an 8-bit gray scale image, G. Then, let G (m, n) denote the (m, n)-th pixel in G, for m ∈ [1, M] and n ∈ [1, N]. Without loss of generality, let the to-be-embedded data Φ = {φj} be as sequence of 0's and 1's, where φj refers to the j-th bit of Φ. Next, for the target pixel X, the pixels within the w × w-neighborhood NX are gathered, sorted, stored in a temporary array Γ. Let γi denote the i-th pixel in the sorted array, where γi ≤ γi+1 for is = 1, 2, ··· , w2. Next, Γ is divided into r partitions to represent φj . When r = 2, two partitions are formed, where Γ0 consists of the first pixel to the w2/r- th pixel, and the second partition Γ1 has the rest of the pixels from Γ. Here, z refers to an integer less than or equal to z. Using the same example in Fig. 1, Γ0 = {21, 22, 31, 32} and Γ1 = {42, 45, 46, 67, 83}. If φj = '0, any pixel in Γ0 can be selected to replace the target pixel X. Else when φj = '1, a pixel from Γ1 will be selected. These steps are repeated to hide data φj in every pixel of the image. Finally, the output image with embedded data, G is generated.

Data Extraction:

During data extraction, G is processed one pixel at a time. Specifically, for each target pixel X = G (m, n), the same w × w-neighborhood, i.e., NX is considered. Similar to the embedding process, the pixels in NX are sorted and stored into a temporary array Γ. However, the pixel located in the middle of Γ, namely γ w2/2+1, is used as the threshold for data extraction. Specifically, if the target pixel X < γ w2/2+1, then '0' is extracted. Otherwise, if the target pixel X > γ w2/2+1, then '1' is extracted. The steps are repeated for each pixel in G to extract all the embedded data φj . It is noteworthy that, the pixel values within the neighborhood NX in G may not be the same as those within NX in G although both X and X are the (m, n)-th pixel in the respective images. Therefore, error could occur during decoding.

CHAPTER 4

Software Introduction:

Introduction to MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

The MATLAB system:

The MATLAB system consists of five main parts

- **Development Environment**:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library**:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language**:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

- **Graphics:**

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs.

It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using IMAGE PROCESSING toolbox.

GRAPHICAL USER INTERFACE (GUI):

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.) and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension. fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.

- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type

guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted,

GUIDE opens a new (i.e., blank) window.

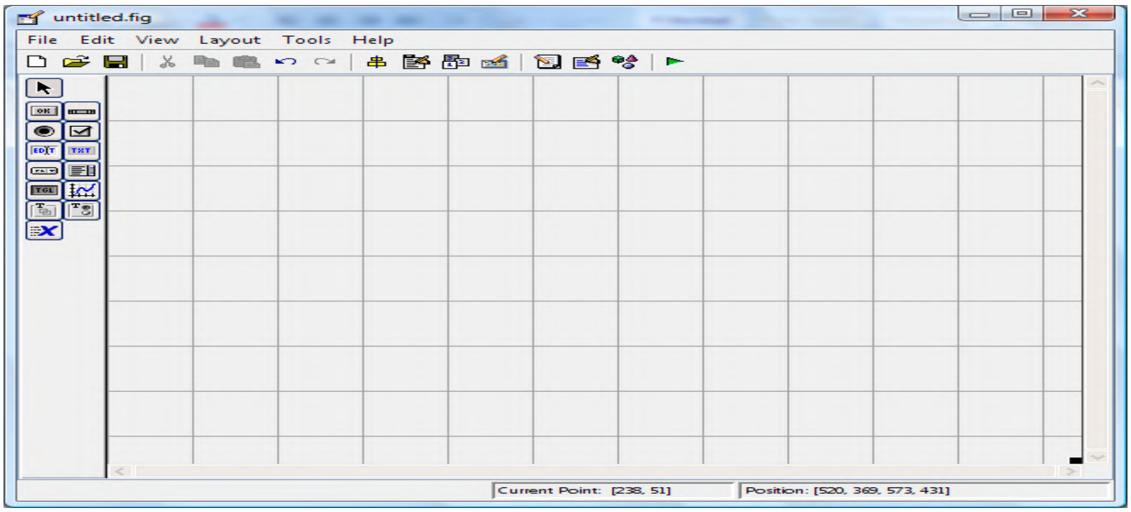


Fig:4.3 Blank Window

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed. GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few

Getting Started

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

Introduction - describes the components of the MATLAB system.

Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

DEVELOPMENT ENVIRONMENT

4.5.1Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.

On a UNIX platform, to start MATLAB, type MATLAB at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB. The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example, emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type help browser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Content's tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorite's tab - View a list of documents you previously designated as favorites.

Display Pane-After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go. Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and who's.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace as from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a. mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to run MATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

MANIPULATING MATRICES

Entering Matrices

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Durer’s matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

MATLAB displays the matrix you just entered. `

```
A =

   16    3    2   13

    5   10   11    8

    9    6    7   12

    4   15   14    1
```

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

Operators

Expressions use familiar arithmetic operators and precedence rule.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

Table: Operators

Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, for a list of more advanced mathematical and matrix functions, type help specfun help elmat.

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.14159265...
I	Imaginary unit, $\sqrt{-1}$
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2- \varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

4.6.4 Table: Functions

GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of uncontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

GUI Development Environment

The process of implementing a GUI involves two basic tasks.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of its

children (uncontrol and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the callbacks, which are defined as subfunctions. This M-file is referred to as the application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uncontrol's; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

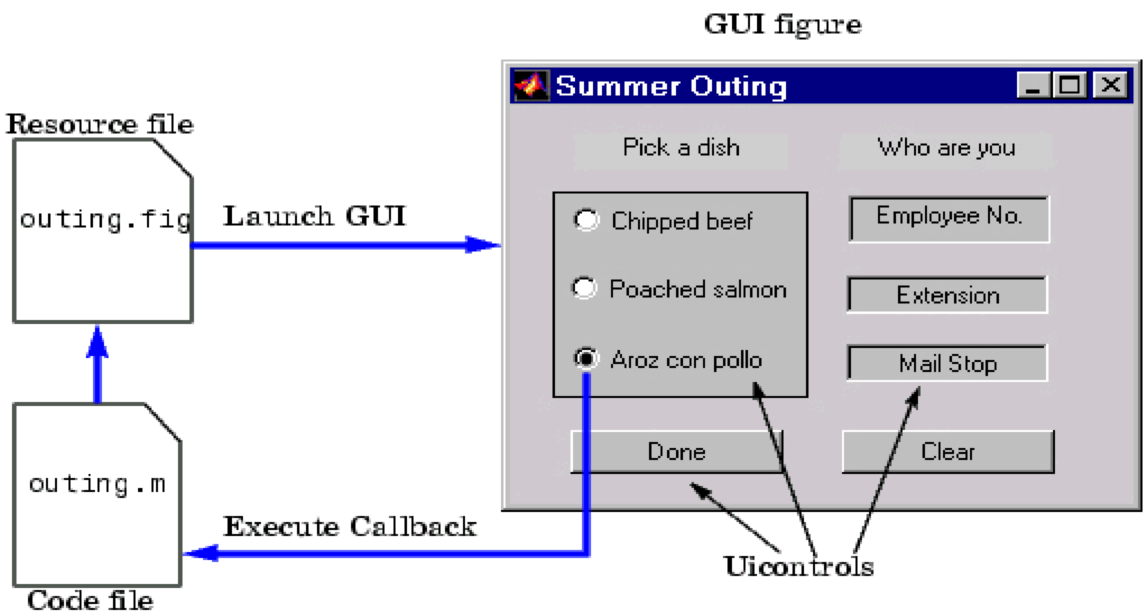


FIG 4.7.2 graphical user blocks

Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features).

The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques

used in the application M-file.

Application Examples - a collection of examples that illustrate techniques

which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as find obj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- List boxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close button) before activating the GUI.

Programming the Callback When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file. function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)

```
button_state = get(h,'Value');

if button_state == get(h,'Max')

   % Toggle button is pressed

elseif button_state == get(h,'Min')

   % toggle button is not pressed

end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

a(:,:,1) = rand (16,128);

a(:,:,2) = rand (16,128);

a(:,:,3) = rand(16,128);

set (h,'CData’, a)

Radio Buttons

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual exclude(off)

set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback (h, event data, handles, varargin)

off = [handles. radiobutton2, handles. radiobutton3, handles. radiobutton4];

mutual exclude(off)

% Continue with callback

 .

 .

 .
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback (h, eventdata, handles, varargin)

if (get(h,'Value') == get(h,'Max'))

   % Then checkbox is checked-take appropriate action

else

   % Checkbox is not checked-take appropriate action

end
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback (h, event data, handles,varargin)

user string = get(h,'string');

% Proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns Nan. You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is Nan, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata, handles,varargin)

user entry = str2double(get(h,'string'));

if isnan(user_entry)

   errordlg ('You must enter a numeric value','Bad Input','modal')

end

% Proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menu bar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menu bar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text

Static text controls display lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items. The String property contains the list of strings displayed in the list box. The first item in the list has an index of The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

List boxes differentiate between single and double clicks on an item and set the figure Selection Type property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure Selection Type property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure, you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu are fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback (h, event data, handles, varargin)

val = get(h,'Value');

switch val

case 1

% The user selected the first item

case 2

% The user selected the second item

% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback (h, event data, handles,varargin)

val = get(h,'Value');

string_list = get(h,'String');

selected_string = string_list{val}; % convert from cell array to string

% etc.
```

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's Button-down callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes Button-down property to define the callback.

Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

axes (handles. axes1)

makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axis. See GUI with Multiple Axes for an example that uses two axes.

Figure

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control.

Chapter 5

RESULT ANALYSIS

5.1 Results:

Experiments conducted in this research using grayscale image as the cover image, and binary image as a message, see Figure as cover images and as a message. The cover image and message image have the same size, 256 * 256 in size.
Here is a cover image used in this study:

Fig 5.1.1 Original Image

The hidden data should be robust against unauthorized extraction / viewing. In the simplest scenario, the attacker has the knowledge about the proposed method as well as the fact that only one fixed pixel pair with 2 partitions is used throughout the image to hide data. The number of possible cases (i.e., correctly guessing the threshold value used during the filter embedding step) to consider can be formulated as:

$\Lambda = [\, w2/r \times (w2/r + 1)] \times 2$

where in these two partitions, each consists of $w2/r$ pixels and $(w2/r + 1)$ pixels, respectively. The value 2 signifies that there is $r = 2$ cases to consider.

Fig:5.1.2 Secret Image

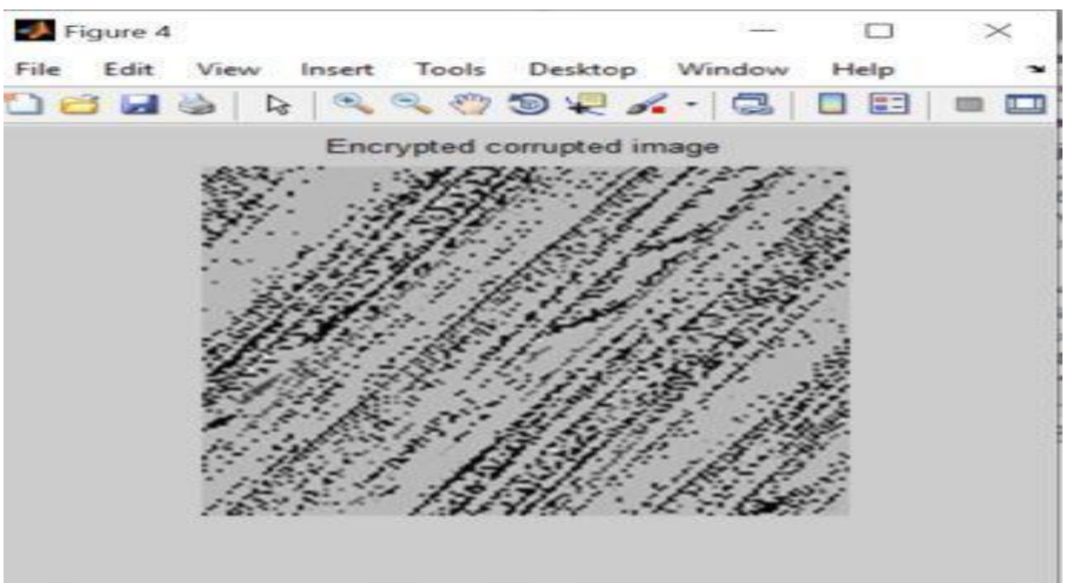


Fig 5.1.3 Embedded Image

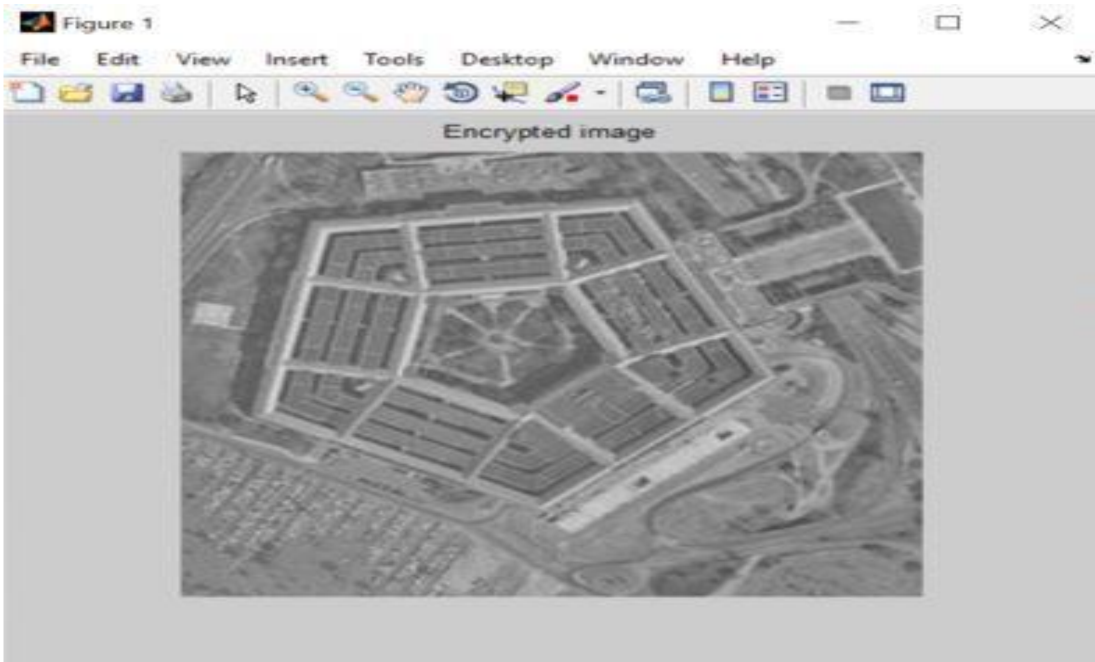


Fig 5.1.4 Encrypted Image

Fig 5.1.5 Decrypted Image

Chapter 6

APPLICATIONS AND ADVANTAGES

APPLICATIONS OF THE PROJECT:

- In Military applications
- In various Business Companies
- Protection of data alteration
- Access control system for digital content distribution
- Media Database system

ADVANATAGES OF THE PROJECT:

- It used in the way of hiding not the information but the password to reach that information.
- Messages do not attract attention to themselves i.e., difficult to detect. only receiver can detect.
- It can be done faster with large amount of software.
- This method featured security, capacity and robustness, the three needed aspects of steganography that makes it useful in hidden exchange of information through text documents and establishing secrets communication.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION:

In this paper, data is embedded while executing the image enhancement steps performed on image. In particular, pixels within the neighbor for a target pixel is collected and sorted in ascending order. This pixel's list is then partitioned and each partition is assigned to represent data in bit(s). Performance of the proposed method is evaluated to identify trends and patterns. Trade-offs among embedding capacity, image quality and data extraction error rate are further discussed. Suggestions are provided to reduce the data extraction error rate and to enhance the robustness of the proposed method against unauthorized viewing. Under various settings, the proposed method can embed 0.99bpp when $w = 3$, and the lowest data extraction error rate is $\sim 3.5\%$ when $w = 7$.

FUTURE SCOPE:

Hence more attention needs to be given to increase the robustness of the embedding algorithm. Implementation of blind steganalysis techniques is difficult compared to targeted steganalysis. An implementation hybrid method which includes encryption and transforms would provide a better performance than the existing data hiding schemes. As future work, we will consider color image and investigate into the correlation among the RGB color channels for deploying the proposed filtering-embedding method.

CODE:

```
clc;
clear all;
close all;
 %%%%%%%% Browse for an Input image %%%%%%%%%%%%%
[FileName,PathName] = uigetfile({'*.jpg';'*.png';'*.gif';'*.bmp'},'Select "Cover Image" to
Hide Message.');

if isequal(FileName,0) || isequal(PathName,0)
   warndlg('Image is not selected');
else
           Img=imread(strcat(PathName,FileName));
    imshow(img);
    title('Original image');
   helpdlg('Image is selected');
end
%img=rgb2gray(img);
save img;

%%%%%%%%%%%%%% Browse for an Logo image %%%%%%%%%%%%%%%

 msg_type=input('Enter1forIMAGE Message:\n');
   if msg_type == 1

     [FileName,PathName]=uigetfile({'*.jpg';'*.png';'*.gif';'*.bmp'},
'Select IMAGE  MESSAGE.');
      msg = imread( strcat(PathName,FileName) );
      figure,imshow(msg);
        title('Secret image');
   else
      error('Invalid Message Type  Selection');
   end
   save msg;

 %%%%%%%%%%%% Embedded with secret data and compressed simultaneously by
SMVQ or image inpainting %%%%%%%%%%%%

load img;
load msg;
```

```
I=imresize(img,[512 512]);
save I
I=double(I);
len=length(I);
block=4;
num=256;

rpool=mat2cell(I,ones(1,len/block)*block,ones(1,len/block)*block);
cbindex=zeros(len/block);
getval=cbindex;
VQ=imresize(msg,[16 256]);
VQ=double(VQ);
mask7=[1 2 3 4 5 9 13];
blockval=zeros(block);
mask=1:16;
tic
for i=1:len/block
   cbindex(1,i)=blockVQ(VQ,rpool{1,i},mask,1);
   cbindex(i,1)=blockVQ(VQ,rpool{i,1},mask,1);
end
getval=cbindex;

for i=2:len/block
   for j=2:len/block
      upblock=VQ(:,getval(i-1,j));
      leftblock=VQ(:,getval(i,j-1));
      blockval(1,:)=upblock([4 8 12 16]);
      blockval(:,1)=leftblock([13 14 15 16]);

      ind=blockVQ(VQ,blockval,mask7,num);

      checkval=rpool{i,j}(:);
      minval=inf;
      for n=1:num
         err=sum(abs(checkval-VQ(:,ind(n))));
         if minval>err
            minval=err;
            cbindex(i,j)=n;
         end
      end
```

```
        getval(i,j)=ind(cbindex(i,j));


    end
end
arnold(getval,2)
load newim
stego=mat2gray(newim);
figure,imshow(stego),title('image encoded with arnold coding');

stego=imresize(stego,[256 256]);
imwrite(stego,'output.jpg');
corruptedimage=imread('output.jpg');

   corruptedimage=gray2rgb(corruptedimage);
figure,imshow(corruptedimage);title('Encrypted corrupted image');

%%%%%%%%%%%% Showing an Encrypted image %%%%%%%%%

 inpaintimage = stego;
     I1 = randperm(numel (inpaintimage));
     inpaintimageNaN = double(inpaintimage);
     inpaintimageNaN(I1(1:round(numel(I1)*0.5))) = NaN;
     %figure, imshow(uint8(st)),
     title('Corrupted image - 50%')
     for k=1:2;
         inpaintimage = inpaintn(inpaintimageNaN);
     end
      inpaintimage=gray2rgb(inpaintimage);
      inpaintimage=imresize(inpaintimage,[1024 1024]);
     figure, imshow((inpaintimage)), title('encyption Embedded image');
     imwrite(inpaintimage,'Encrypted_new.jpg');

    save inpaintimage;

%delete 'output.jpg';

clc
close all
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=imread('Encrypted.jpg');
figure,imshow(a);title('Encrypted image');
inpaintimage=imresize(inpaintimage,[128 128]);
%%%%%%%%%%%%%% Decrypt the image with secret data and compressed
simultaneously by SMVQ or image inpainting %%%%%%%%%%%%%

rpool=cell(len/block);
for i=1:len/block
   rpool{1,i}=inpaintimage(:,cbindex(1,i));
   rpool{i,1}=inpaintimage(:,cbindex(1,i));
end

for i=1:len/block
   for j=1:len/block

       rpool{i,j}=VQ;
   end
end

rpool=rpool(1,1);
img=cell2mat(rpool);

%%%%%%%%%%%%%%%%%%%%% arnold decoding
   arnold(img,2)
   load iminverse
   figure,imshow(iminverse),title('Decoded image using arnold methodology');

%%%%%%%%%%%%%% Decrypted image
%%%%%%%%%%%%%%%

inpaintimage = iminverse;
    I1 = randperm(numel(inpaintimage));
    inpaintimageNaN = double(inpaintimage);
inpaintimageNaN(I1(1:round(numel(I1)*0.5))) = NaN;
            %figure, imshow(uint8(st)), title('Corrupted image - 50%')
for k=1:2;
        inpaintimage2=inpaintn(inpaintimageNaN);
```

```
        end
    figure,imshow(inpaintimage2),title('reconstructed image');
     inpaintimage2=imresize(inpaintimage2,
[1024 1024]);
      imwrite(inpaintimage2,'out_new.jpg');
    %%%%%%%%%%%%%%%%%%

    %imwrite(inpaintimage,'Decrypted.jpg');
    load img

    [m n]=size(inpaintimage2);
    mse=sum(sum(img))/(m*n);
    psnr=10*log10(255*255/mse);
    disp('psnr values is ---')

disp(psnr);

    load inpaintimage
   inpaintimage2=uint8(inpaintimage2);
    size=length(inpaintimage); %%%%%%%%%% length of compressed codes

%   k=imfinfo('Encrypted_new.jpg');
% k1=imfinfo('out_new.jpg');
% ib=k.Width*k.Height*k.BitDepth/8;
% cb=k1.FileSize;
% cr=ib/cb;
% disp('compression ratio --')
% disp(real(cr))

ss1=graycoprops(inpaintimage2);
ss2=graycoprops(inpaintimage2);
ss1=ss1. Homogeneity;
ss2=ss2. Homogeneity;
ss=ss1+ss2;
%disp('SSIM value is -')
%disp(ss);

figure,imshow(msg),title('secret image');

%
```

REFERENCES:

[1] X. Li, W. Zhang, X. Gui, and B. Yang, “Efficient reversible data hiding based on multiple histograms modification,” IEEE Transactions on Information Forensics and Security, vol. 10, no. 9, pp. 2016–2027, Sep. 2015.

[2] B. Ou, X. Li, Y. Zhao, R. Ni, and Y. Shi, “Pairwise prediction-error expansion for efficient reversible data hiding,” IEEE Transactions on Image Processing, vol. 22, no. 12, pp. 5010–5021, Dec 2013.

[3] T. Zhang, X. Li, W. Qi, and Z. Guo, “Location-based pvo and adaptive pairwise modification for efficient reversible data hiding,” IEEE Transactions on Information Forensics and Security, pp. 1–1, 2020.

[4] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam, and C. H. PremChand, “Removal of high-density salt and pepper noise through modified decision based unsymmetric trimmed median filter,” IEEE Signal Processing Letters, vol. 18, no. 5, pp. 287–290, May 2011.

[5] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in Proc. 8th Int’l Conf. Computer Vision, July 2001, vol. 2, pp. 416–423.

[6] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, April 2004.

[7] D. M. Thodi and J. J. Rodriguez, ‘‘Prediction-error based reversible water_x0002_marking,’’ in Proc. Int. Conf. Image Process., Oct. 2004, pp. 1549–1552.

[8] D. M. Thodi and J. J. Rodriguez, ‘‘Expansion embedding techniques for reversible watermarking,’’ IEEE Trans. Image Process., vol. 16, no. 3,pp. 721–730, Mar. 2007.

[9] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, ‘‘Reversible data hiding,’’ IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354–362, Mar. 2006.

[10] G. Coatrieux, W. Pan, N. Cuppens-Boulahia, F. Cuppens, and C. Roux,‘‘Reversible watermarking based on invariant image classification anddynamic histogram shifting,’’

A

PROJECT REPORT

On

IOT ENERGY METER WITH CURRENT, VOLTAGE AND COST MONITORING SYSTEM

Submitted by

Ms.K.Navitha (17K81A0482)

Mr.C.Rahul Reddy(17K81A0470)

Mr.K.Ganesh Kumar(17K81A0488)

in partial fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

IN

Electronics and Communication Engineering

Under The Guidance of

Mr. Chandan Roy

(B.Tech., M.Tech., (Ph.D))

DEPARTMENT OF ECE

ST.MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

BONAFIDE CERTIFICATE

This is to certify that the project entitled "**IOT ENERGY METER WITH CURRENT, VOLTAGE AND COST MONITORING SYSTEM**" is being submitted by **1.Ms. K. Navitha** (17K81A0482), **2.Mr. C. Rahul Reddy** (17K81A0470), **3.Mr. K. Ganesh Kumar** (17K81A0488) in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **Electronics and Communication Engineering** is recorded of bonafide work carried out by them. The results embodied in this report have been verified and found satisfactory.

	Head of the Department
Mr. Chandan Roy	**Dr.B. Hari krishna** ,
Department of Electronics and	Department of Electronics and
Communication Engineering.	Communication Engineering.

Internal Examiner	External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **C. RAHUL REDDY** WITH ROLL NO.**17K81A0470, K. GANESH KUMAR** WITH ROLL NO.**17K81A0488**, **K. NAVITHA REDDY** WITH ROLL NO.**17K81A0482**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**IOT ENERGY METER WITH CURRENT, VOLTAGE AND COST MONITORING SYSTEM**
" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of **Bachelor of Technology** in Department of Electronics and Communication Engineering, session: <2017 - 2021>, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work Entitled **"IOT ENERGY METER WITH CURRENT, VOLTAGE AND COST MONITORING SYSTEM"** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

K Navitha 17K81A0482
C Rahul Reddy 17K81A0470
K Ganesh Kumar 17K81A0488

ACKNOWLEDGEMENTS

While bringing out this project report to its final form, we came across a number of people whose contributions in various ways helped my field of research and they deserve special thanks. It is a pleasure to convey our gratitude to all of them.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our college management** for their kind support and permission to use the facilities available in the institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr. Santosh Kumar Patra**, our **beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna, Head & professor,** the **Dept. of Electronics and Communication Engineering** for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our **Dept. Project Coordinator** Dr. **A. Anand, Assistant Professor, Dept.of ECE** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We are highly grateful and obliged to our college level project coordinators **Dr.Maalydri, Professor in IT & Dr.R.Santhosh Kumar, Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and cooperation that is difficult to express in words.

Finally, I am deeply indebted to my Mother, **Mrs.C.Latha**, my Father, **Mr.C.Rama Krishna Reddy**, my Brother, **Mr.C.Sai Kalyan Reddy** and to my family members for their moral support and continuous encouragement while carrying out this study. I dedicate this thesis to the almighty.

1.C Rahul Reddy

2.K Navitha

3.k Ganesh Kumar

ABSTRACT

Observing and monitoring your power utilization for verification is a not a easy task today since regularly checking the meter room is not an easy task Well, it is very important to know whether you are charged likewise so the need is very sure. Well, we have made a system that allows users to monitor energy meter readings over IOT. Our proposed system utilizes an energy meter with a microcontroller system to monitor energy utilization. The meter is used to monitor units consumed, estimated cost, Line Voltage and current consumed. Simple web application shows the Live Output of these readings over the IOT. This enables users to effortlessly check the units consumed, Line Voltage and current consumed Live from anywhere through the app. In this way the energy meter observing framework enables the client to adequately check the power meter readings.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Introduction

The energy consumption can be monitored by using an electric device called an Energy meter. The cost and the regular usage of Power consumption are informed to the user to overcome high bill usage. The Energy meter shows the amount of units consumed and transfers the data to both the customer and to the electrical board so this helps in reducing man-power. The user can check their Power usage from anywhere and at any time interval. The IoT is used to Turn on/off the household appliances using relay and Arduino interfacing. The objective of this system is to monitor the amount of electricity consumed. The distributor and the consumer both will be benefitted by eventually reducing the total Power consumption.

Project overview

Embedded System

An embedded system is a system which is going to do a predefined specified task and is even defined as a combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems include computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.
The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes. The applications software on such processors is sometimes referred to as firmware.

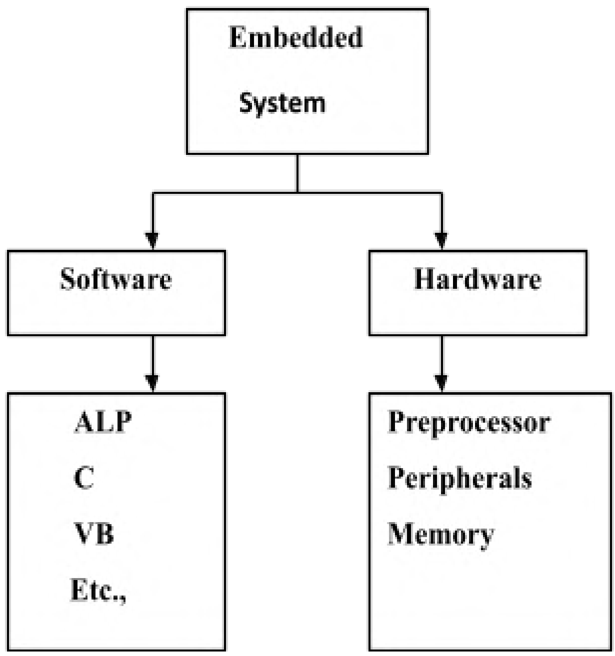


Figure 1: Block diagram of Embedded System

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

Memory: It is used to store data or addresses.

Peripherals: These are the external devices connected.

Processor: It is an IC which is used to perform some task.

Applications of embedded systems:

- Manufacturing and process control.
- Construction industry.
- Transport.
- Buildings and premises.
- Domestic service.
- Communications,
- Office systems and mobile equipment.
- Banking, finance and commercial.
- Medical diagnostics, monitoring and life support.
- Testing, monitoring and diagnostic systems.

Processors are classified into four types like:

- Microprocessor (µp)
- Microcontroller (µc)
- Digital Signal Processor (DSP)

- Application Specific Integrated Circuits (ASIC)

Microprocessor (µp):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Three basic characteristics differentiate microprocessors:

□ **Instruction set :** The set of instructions that the microprocessor can execute.

□ **Bandwidth :** The number of bits processed in a single instruction.

□ **Clock speed :** Given in megahertz (MHz), the clock speed determines how many instructions per second the processor can execute.

In both cases, the higher the value, the more powerful the CPU. For example, a 32-bit microprocessor that runs at 50MHz is more powerful than a 16-bit microprocessor that runs at 25MHz. In addition to bandwidth and clock speed, microprocessors are classified as either RISC (reduced instruction set computer) or CISC (complex instruction set computer).

A microprocessor has three basic elements, as shown above. The ALU performs all arithmetic computations, such as addition, subtraction and logic operations (AND, OR, etc). It is controlled by the Control Unit and receives its data from the Register Array. The Register Array is a set of registers used for storing data. These registers can be accessed by the ALU very quickly. Some registers have specific functions - we will deal with these later. The Control Unit controls the entire process. It provides the timing and a control signal for getting data into and out of the registers and the ALU and it synchronizes the execution of instructions (we will deal with instruction execution at a later date).

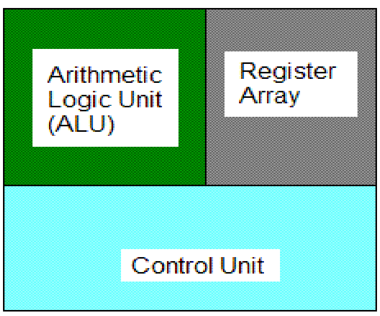


Figure 2: Elements of microprocessor.

Basic Elements of a Microprocessor:

Micro Controller (μc):

A microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.Microcontrollers are designed for embedded applications,

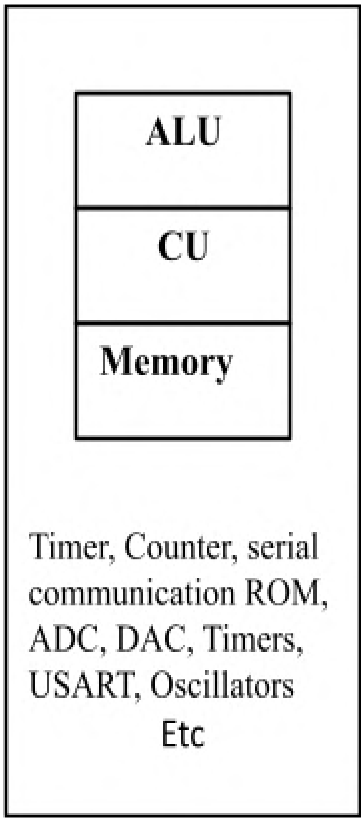


Figure 3: Block Diagram of Microcontroller (μc)

Digital Signal Processors (DSPs):

Digital Signal Processors is one which performs scientific and mathematical operations. Digital Signal Processor chips - specialized microprocessors with architectures designed specifically for the types of operations required in digital signal processing. Like a general-purpose microprocessor, a DSP is a programmable device, with its own native instruction code. DSP chips are capable of carrying out millions of floating point operations per second, and like their better-known general-purpose cousins, faster and more powerful versions are continually being introduced. DSPs can also be embedded within complex "system-on-chip" devices, often containing both analog and digital circuitry.

Application Specific Integrated Circuit (ASIC)

ASIC is a combination of digital and analog circuits packed into an IC to achieve the desired control/computation function.

ASIC typically contains:

- □ CPU cores for computation and control.
- □ Peripherals to control timing critical functions.
- □ Memories to store data and programs.
- □ Analog circuits to provide clocks and interface to the real world which is analog in nature.
- □ I/Os to connect to external components like LEDs, memories, monitors etc.

Computer Instruction Set:

There are two different types of computer instruction set there are:

1. RISC (Reduced Instruction Set Computer) and
2. CISC (Complex Instruction Set computer).

Reduced Instruction Set Computer (RISC)

A RISC (reduced instruction set computer) is a microprocessor that is designed to perform a smaller number of types of computer instruction so that it can operate at a higher speed (perform more million instructions per second, or millions of instructions per second). Since each instruction type that a computer must perform requires additional transistors and circuitry, a larger list or set of computer instructions tends to make the microprocessor more complicated and slower in operation.

Besides performance improvement, some advantages of RISC and related design improvements are:

- □ A new microprocessor can be developed and tested more quickly if one of its aims is to be less complicated.
- □ Operating system and application programmers who use the microprocessor's instructions will find it easier to develop code with a smaller instruction set.

□ The simplicity of RISC allows more freedom to choose how to use the space on a microprocessor.

Higher-level language compilers produce more efficient code than formerly because they have always tended to use the smaller set of instructions to be found in a RISC computer.

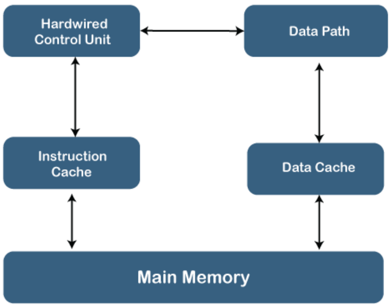


Figure 4(a): RISC Characteristics

RISC characteristics:

- **Simple instruction set:** In a RISC machine, the instruction set contains simple, basic instructions, from which more complex instructions can be composed.
- **Same length instructions:** Each instruction is the same length, so that it may be fetched in a single operation.
- **1machine-cycle instructions:** Most instructions are completed in one machine cycle, which allows the processor to handle several instructions at the same time. This pipelining is a key technique used to speed up RISC machines.

Figure 4(b): RISC-V of India

Complex Instruction Set Computer (CISC)

CISC, which stands for Complex Instruction Set Computer, is a philosophy for designing chips that are easy to program and which make efficient use of memory. Each instruction in a CISC instruction set might perform a series of operations inside the processor. This reduces the number of instructions required to implement a given program, and allows the programmer to learn a small but flexible set of instructions.

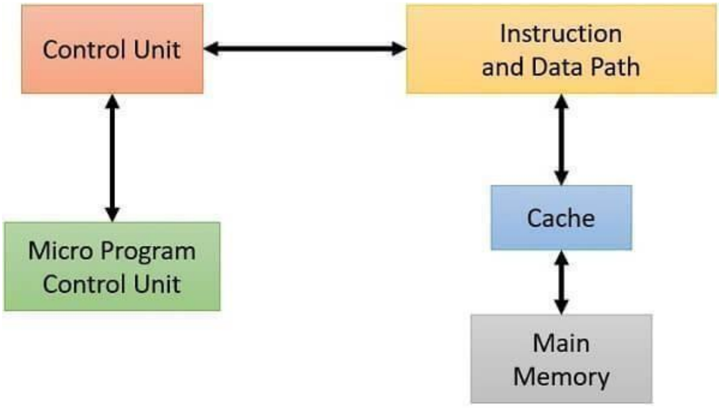


Figure 5(a): CISC architecture

The advantages of CISC

- At the time of their initial development, CISC machines used available technologies to optimize computer performance.
- Microprogramming is as easy as assembly language to implement, and much less expensive than hardwiring a control unit.
- The ease of micro-coding new instructions allowed designers to make CISC machines upwardly compatible: a new computer could run the same programs as earlier computers because the new computer would contain a superset of the instructions of the earlier computers.
- As each instruction became more capable, fewer instructions could be used to implement a given task. This made more efficient use of the relatively slow main memory.
- Because microprogram instruction sets can be written to match the constructs of high-level languages, the compiler does not have to be as complicated.

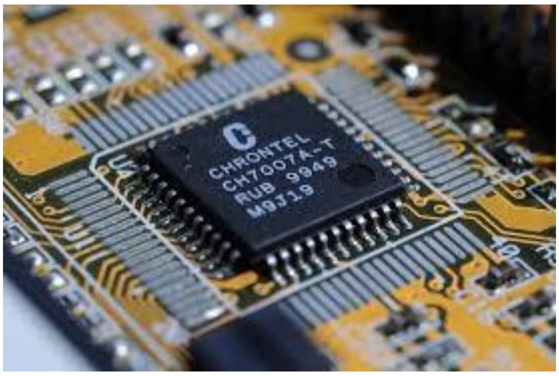


Figure 5(b): CISC chip

The disadvantages of CISC

Still, designers soon realized that the CISC philosophy had its own problems, including:

- Earlier generations of a processor family generally were contained as a subset in every new version, so instruction set & chip hardware became more complex with each generation of computers.
- So that as many instructions as possible could be stored in memory with the least possible wasted space, individual instructions could be of almost any length---this means that different instructions will take different amounts of clock time to execute, slowing down the overall performance of the machine.
- Many specialized instructions aren't used frequently enough to justify their existence; approximately 20% of the available instructions are used in a typical program.
- CISC instructions typically set the condition codes as a side effect of the instruction. Not only does setting the condition codes take time, but programmers have to remember to examine the condition code bits before a subsequent instruction changes them.

Memory Architecture

There two different type's memory architectures there are:

- Harvard Architecture.
- Von-Neumann Architecture.

Harvard Architecture

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

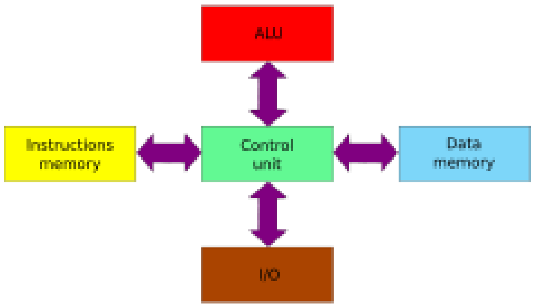


Figure 6: Harvard Architecture

Modern uses of the Harvard architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

- Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behavior must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.

- Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction prefetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Von-Neumann Architecture:

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. The Von-Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A stored-program digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advanced over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patches to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

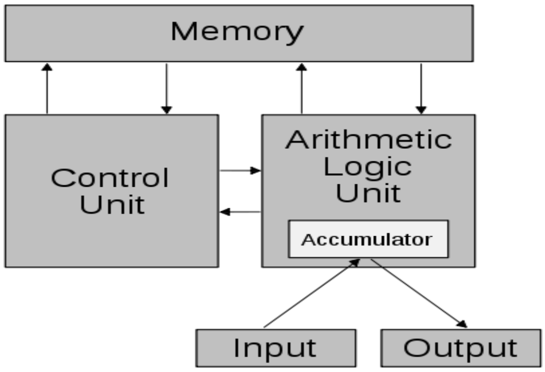


Figure 7: Von-Neumann Architecture.

Basic Difference between Harvard and Von-Neumann Architecture:

- The primary difference between Harvard architecture and the Von Neumann architecture is in the Von Neumann architecture data and programs are stored in the same memory and managed by the same information handling system.
- Whereas the Harvard architecture stores data and programs in separate memory devices and they are handled by different subsystems.
- In a computer using the Von-Neumann architecture without cache; the central processing unit (CPU) can either be reading and instruction or writing/reading data to/from the memory. Both of these operations cannot occur simultaneously as the data and instructions use the same system bus.
- In a computer using the Harvard architecture the CPU can both read an instruction and access data memory at the same time without cache. This means that a computer with Harvard architecture can potentially be faster for a given circuit complexity because data access and instruction fetches do not contend for use of a single memory pathway.
- Today, the vast majority of computers are designed and built using the Von Neumann architecture. Von Neumann architecture may be somewhat slower than the contrasting Harvard Architecture for certain specific tasks, but it is much more flexible and allows for many concepts unavailable to Harvard architecture such as self-programming, word processing and so on.

Power Supply

Block Diagram:

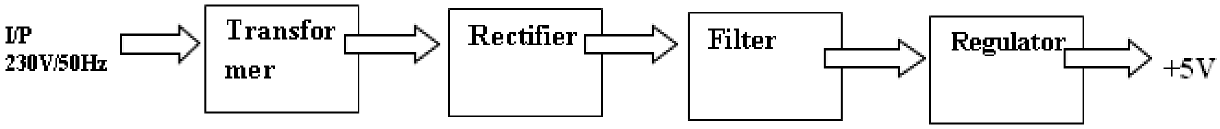


Figure 8: Block diagram of the Power Supply Unit

Description:

1.2.4.2(a) Transformer:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or primary winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the secondary winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

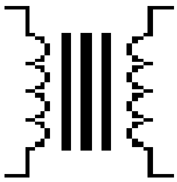


Figure 9: Transformer Symbol

(or)

Transformer is a device that converts one form of energy to another form of energy like a transducer.

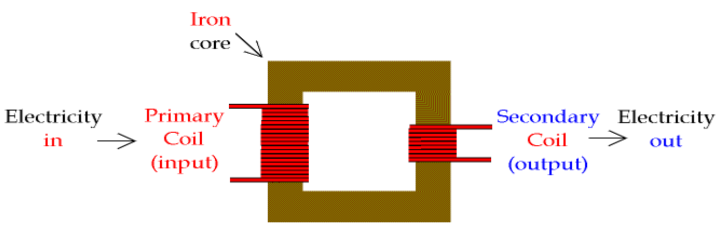


Figure 10: Transformer

Basic Principle:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

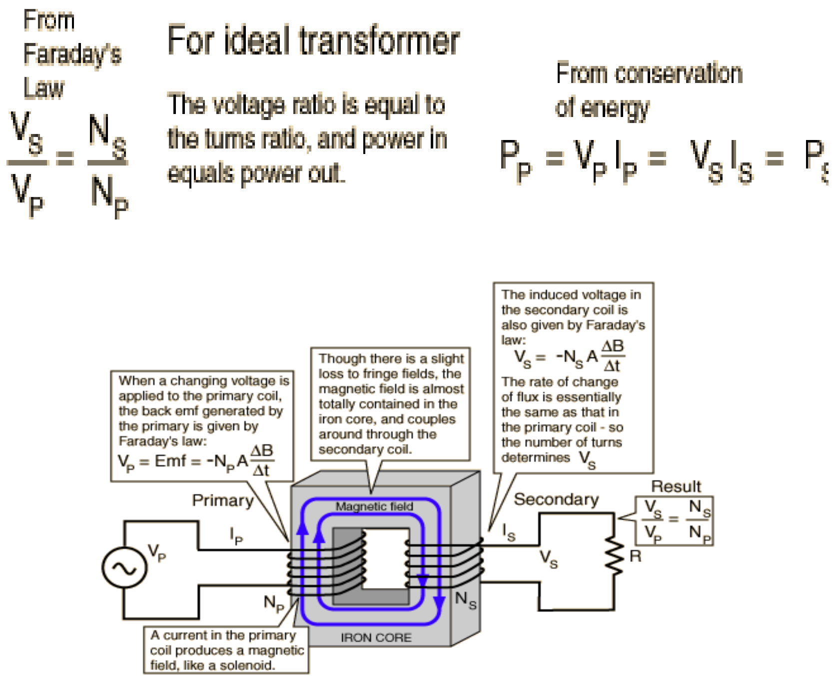


Figure 11: Basic Principle

Transformer Working:

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in the figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

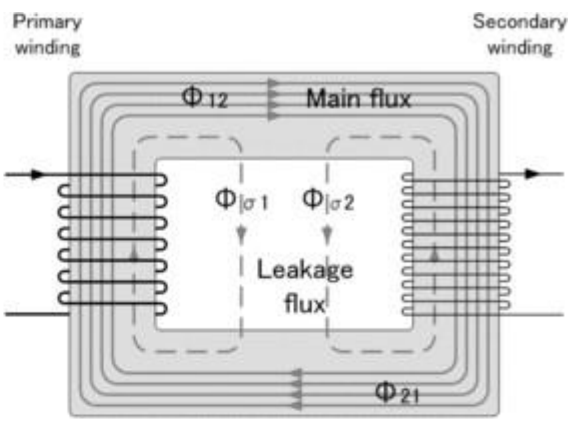


Figure 12: Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow.

The iron core is laminated to prevent 'eddy currents' flowing in the core. These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

Classification of Transformer:

- Step-Up Transformer.
- Step-Down Transformer.

Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. The turn's ratio of the two sets of windings determines the amount of voltage transformation.

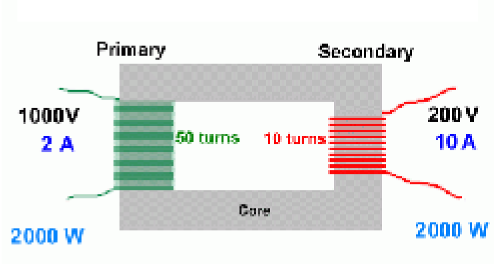


Figure 13: Step-Down Transformer

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phases smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated.

Single-phase step-down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase step up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

Step-Up Transformer:

A step-up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground.

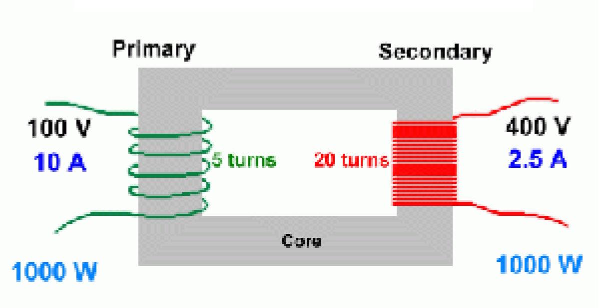


Figure 14: Step-Up Transformer

Applications:

Generally these Step-Up Transformers are used in industry applications only.

Turns Ratio and Voltage:

The ratio of the number of turns on the primary and secondary coils determines the ratio of the voltages.

$$\frac{V_s}{V_p} = \frac{N_s}{N_p}$$

where V_p is the primary (input) voltage, V_s is the secondary (output) voltage, N_p is the number of turns on the primary coil, and N_s is the number of turns on the secondary coil.

1.2.4.2(b) Diodes:

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

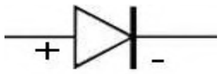


Figure 15: Diode Symbol

A diode is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

1.2.4.2(c) Rectifier:

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as 'half-wave' and 'full-wave' rectifiers. Both use components called diodes to convert AC into DC.

Half-wave Rectifier:

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in the figure .

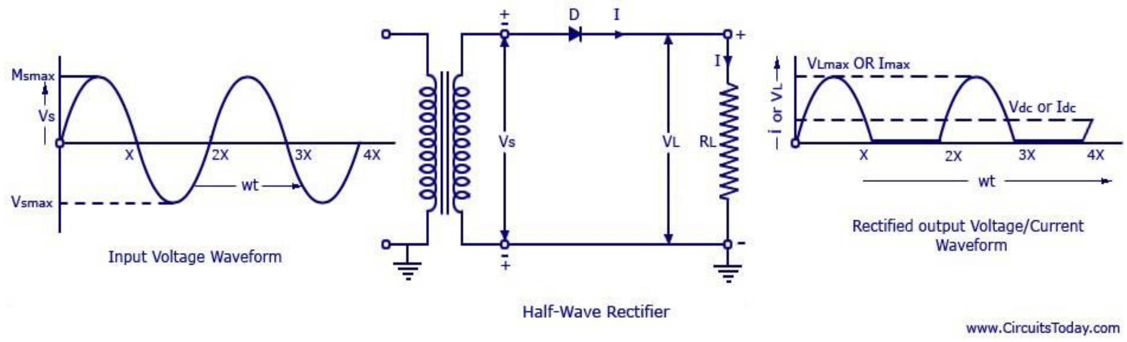


Figure 16: Half Wave Rectifier

Figure 2 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is 0V. Because there is a 0.7V voltage loss across the diode,the peak output voltage will be 0.7V less than Vs.

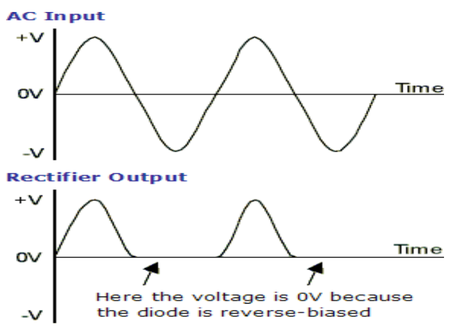


Figure 17: Half-Wave Rectification

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

Full-wave Rectifier:

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC. The resulting waveform is shown in figure 19.

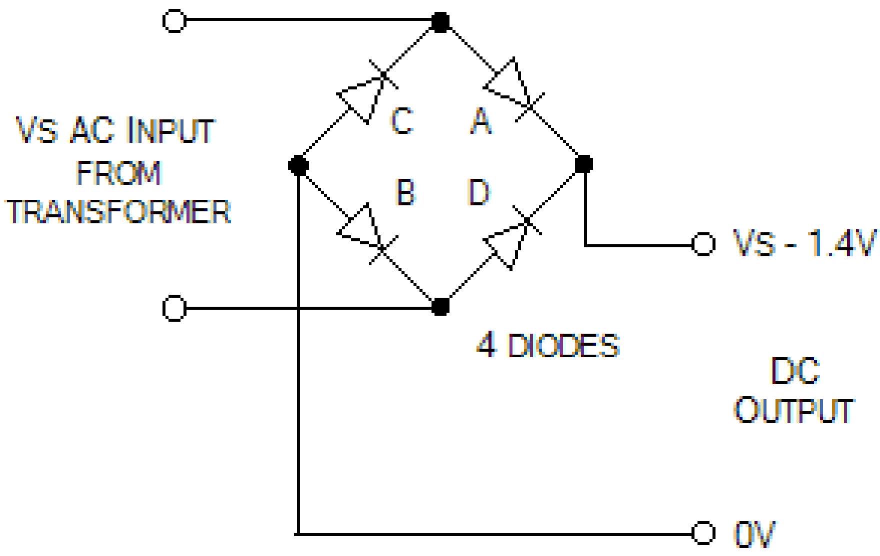


Figure 18: Full-Wave Rectifier

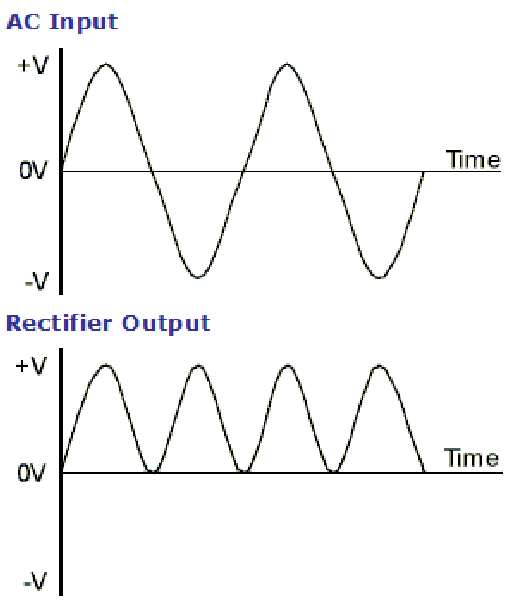


Figure 19: Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased.

While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

Capacitor Filter:

The capacitor-input filter, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

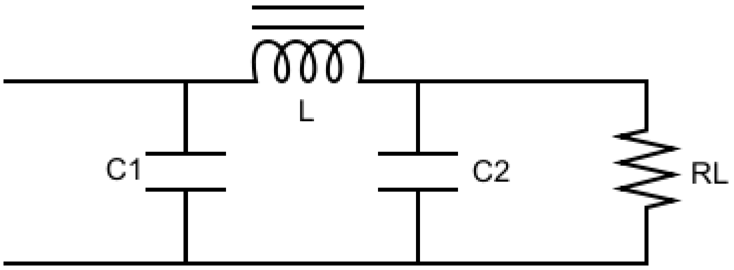


Figure 20: Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result only the DC component appears across the load RL.

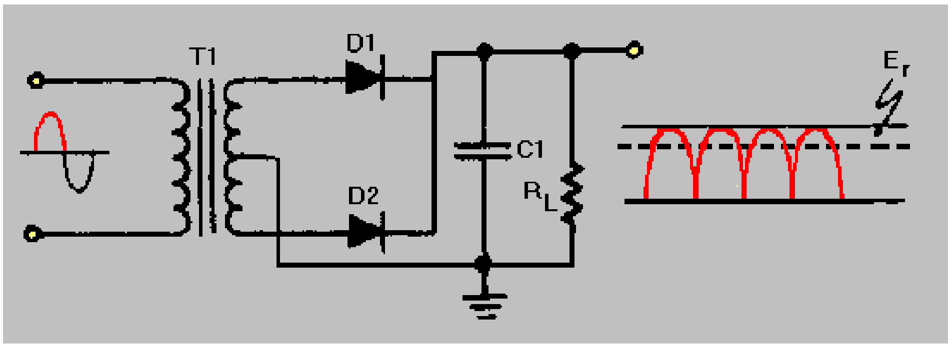


Figure 21: Centered Tapped Full-Wave Rectifier with a Capacitor Filter

1.2.4.2(d) Voltage Regulator:

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulators.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx).

78xx:'78' indicates the positive series and 'xx' indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicates the negative series and 'xx' indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consists the three pins there are

Pin1: It is used as an input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pins. Through this pin we get the output.

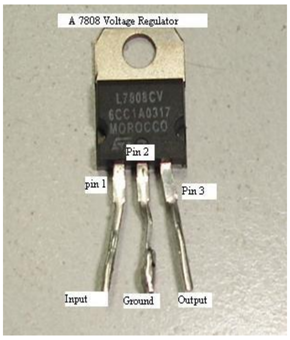


Figure 22: Regulator

Circuit Diagram:

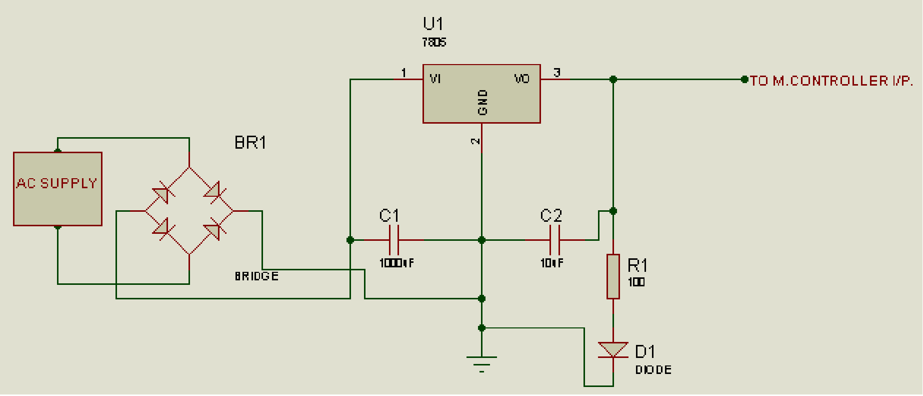


Figure 23: Circuit Diagram of Power supply

CHAPTER 2

LITERATURE SURVEY

Anitha et al., [1] proposed Smart energy meter surveillance using IoT about IoT, internet of things as an emerging field and IoT based devices have created a revolution in electronics and IT. The foremost objective of this project is to create awareness about energy consumption and. efficient use of home appliances for energy savings. Due to manual work, the existing electricity billing system has major drawbacks. This system will give the information on meter reading, power cut when power consumption exceeds beyond the specified limit using IoT. The Arduino esp8266 microcontroller is programmed to perform the objectives with the help of IOT. It is proposed to overcome all the disadvantages in the already existing energy meter. All the details are sent to the consumer's mobile through the IoT and it is also displayed in the LCD. It is a time saving and it helps to eliminate human interference using IoT.

Devadhanishini et al., [2] Smart Power Monitoring Using IoT that Energy Consumption is a very important and challenging issue. Automatic Electrical Energy meter is used in large electric energy distribution systems. The integration of the Arduino WIFI and SMS provides the system as a Smart Power Monitoring system. Smart energy meters provide data for optimization and less the power consumption. This system also includes a motion sensor such that if there is no human in the house or house it will automatically turn off the power supply.

Mohammed Hosseiu et al., [3] presented a paper titled Design and implementation of smart meters using IoT describing the growth of IoT and digital technology. The future energy grid needs to be implemented in a distributed topology that can dynamically absorb different energy sources. IoT can be utilized for various applications of the smart grid consisting of power consumption, smart meter, electric power demand, side management and various areas of energy production. In this paper, the Smart Energy Metering(SEM) is explained as the main purpose of SEM is necessary for collecting information on energy consumption of household appliances and monitoring the environmental parameters and providing the required services to home users.

Himanshu K Patel et al., [4] demonstrated an Arduino based smart energy meter that removes human intervention in meter readings and bill generation thereby reducing the error that usually causes in India. The system consists of the provision of sending an SMS to users for update on energy consumption along with final bill generation along with the freedom of reload via SMS. The disconnection of power supply on demand or due to pending dues was implemented using a relay. The system employs GSM for bidirectional communication.

Bibek Kanti Barman, et al., [5] proposed smart meters using IoT for efficient energy utilization plays a very vital role for the development of smart grid in power systems. Hence proper monitoring and controlling of power consumption is a main priority of the smart grid. The energy meter has many problems associated with it and one of the key problems is there is no full duplex communication to solve this problem, a smart energy meter is proposed based on the Internet of Things. The smart energy meter controls and calculates the consumption of energy using ESP 8266 12E, a Wi-Fi module and sends it to the cloud from where the consumer or customer can observe the reading.Therefore, energy examination by the consumer becomes much easier and controllable. This system also helps in detecting energy loss. Thus, this smart meter helps in home automation using IoT.

Garrab et al., [6] proposed AMR approach for energy saving in Smart Grids using Smart Meter and partial Power Line Communication on the rising demand of energy. Smart meters are one of the proposed solutions for the Smart Grid. In this article, an AMR solution which gives detailed end-to-end application. It is based on an energy meter with low-power microcontrollerMSP430FE423A and the Power Line Communication standards. The microcontroller includes an energy metering module ESP430CEl.

CHAPTER 3

PROPOSED SYSTEM

Existing System

- Now-a-days to measure the electricity consumption, the electrical board hires persons who visit each home and records the reading manually which gives scope for corruption and human error in reading. No over power use alert system.
- The billing details are provided to consumers on a monthly basis.

Proposed System

- The Existing reading system is time consuming and it needs a number of laborers. The proposed project will eliminate the need for labor and it is cost efficient.
- The proposed system gives the information about the energy consumption in real time on an IOT dashboard on mobile application and PC, billing through IOT.
- This smart energy meter protects your home from bad supply, alerting the consumer and utility when the energy consumption exceeds the set limit.
- We can disconnect and reconnect the power through a mobile application when the consumers are out of station to prevent the waste age of energy.

Block Diagram

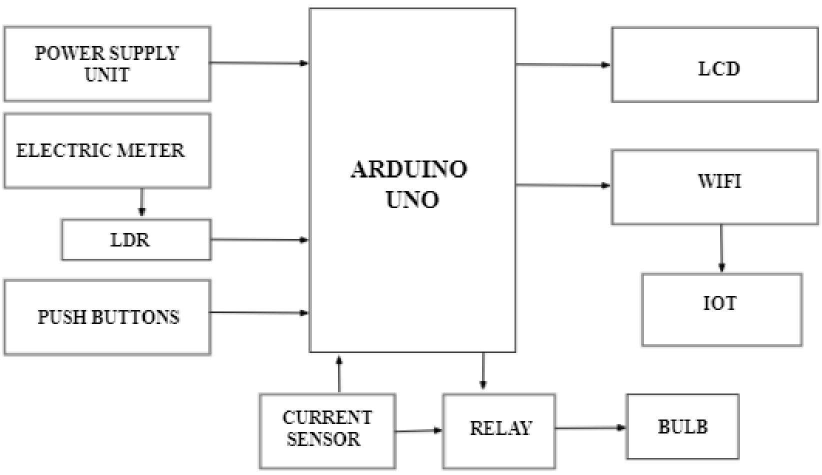


Figure 3.3: Block Diagram of IOT based Energy meter with current, voltage and cost monitoring system.

Schematic Diagram

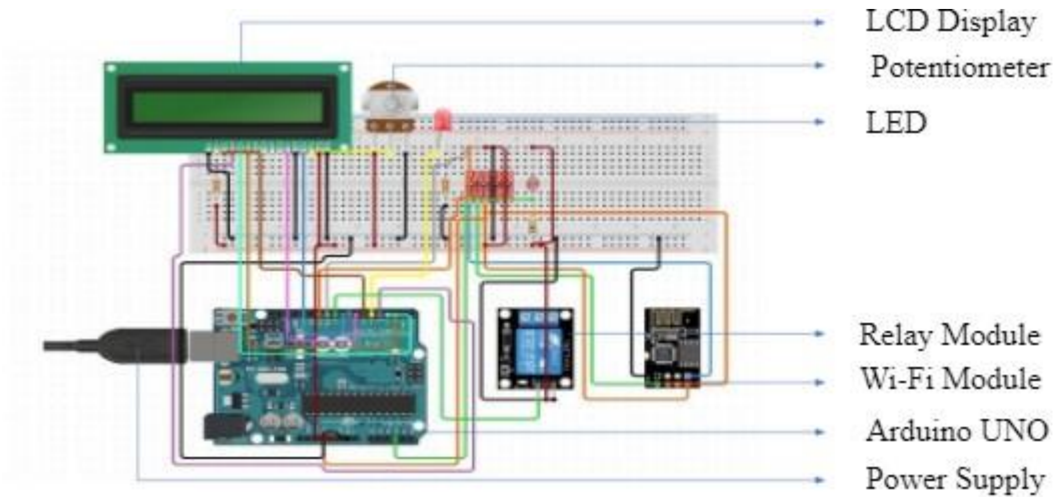


Figure 3.4 Schematic diagram of IOT Energy Meter with current, voltage and cost monitoring system

Explanation

The power supply is given to the arduino uno. All the required components are connected to the arduino.LDR is used to detect the blink of light on the electric meter. The blink of light indicates that there is a power supply. LDR detects the blink, sends the information to the LCD and wifi module, and the arduino which is being programmed acts as a medium. Every Blink indicates that a unit is consumed by the consumer. we get each and every count displayed on the LCD. We can set a limit for our energy consumption.

The consumed energy information is stored in the cloud using a wifi module. We can link the app to the ip address of the wifi module so that we can access the data. The consumer can access the data through an application from anywhere at any time. Within the App we get every information about the energy consumption, and if we set any limit for the energy consumption we get the warning notification as you have consumed maximum units. If it reaches the maximum limit it automatically stops the power supply. If we recharge it or pay the bill we will get the power supply and get the notification as recharge done in our app and it is also displayed on the LCD.

CHAPTER 4

SOFTWARE DESCRIPTION

Introduction to Proteus:

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possesses a 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

About Proteus:

It is a software suite containing schematic simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.
The designer can also develop 2D drawings for the product.

Features:

ISIS has a wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semiconductor relays, switches, real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps microcontrollers, processors, sensors etc.

ARES offers PCB design up to 14 inner layers, with surface mount and through hole packages. It is embedded with the footprints of different categories of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in ISIS can be directly transferred to ARES.

Starting a New Design:

Step 1: Open ISIS software and select New design in File menu

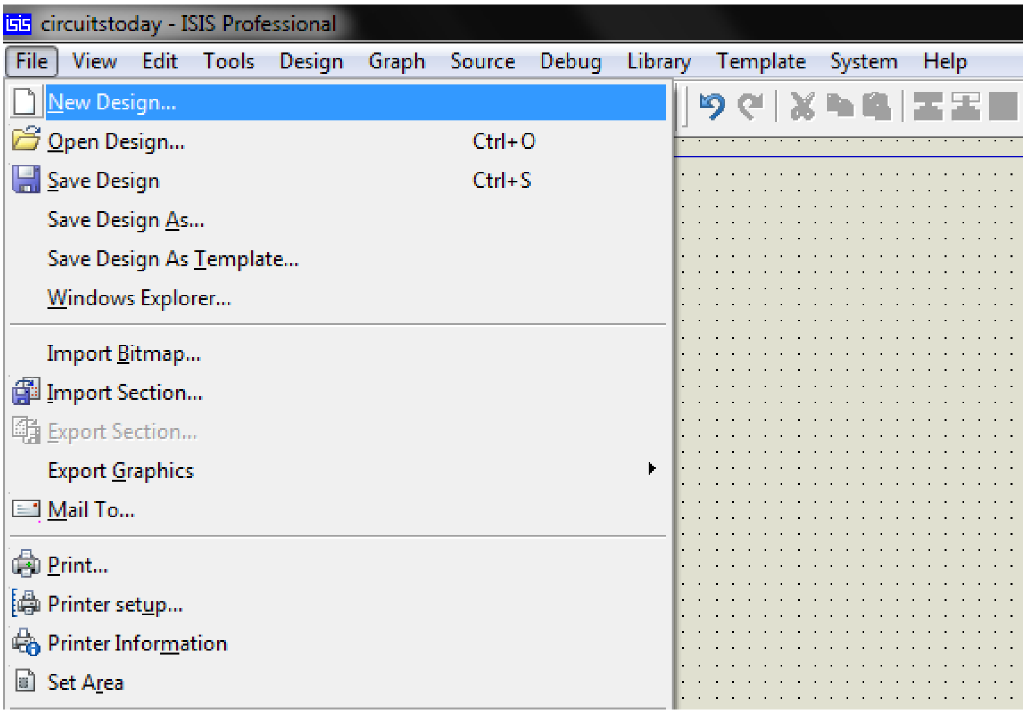


Figure 4.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

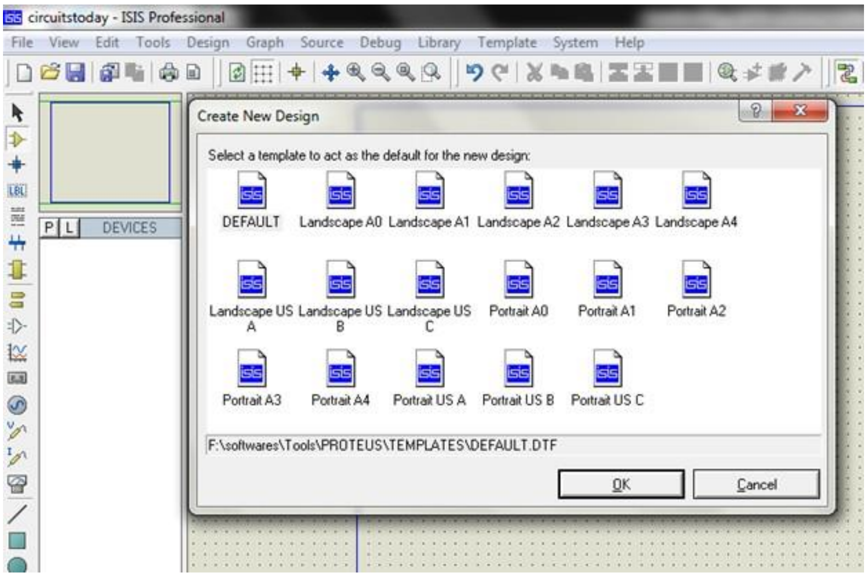


Figure 4.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

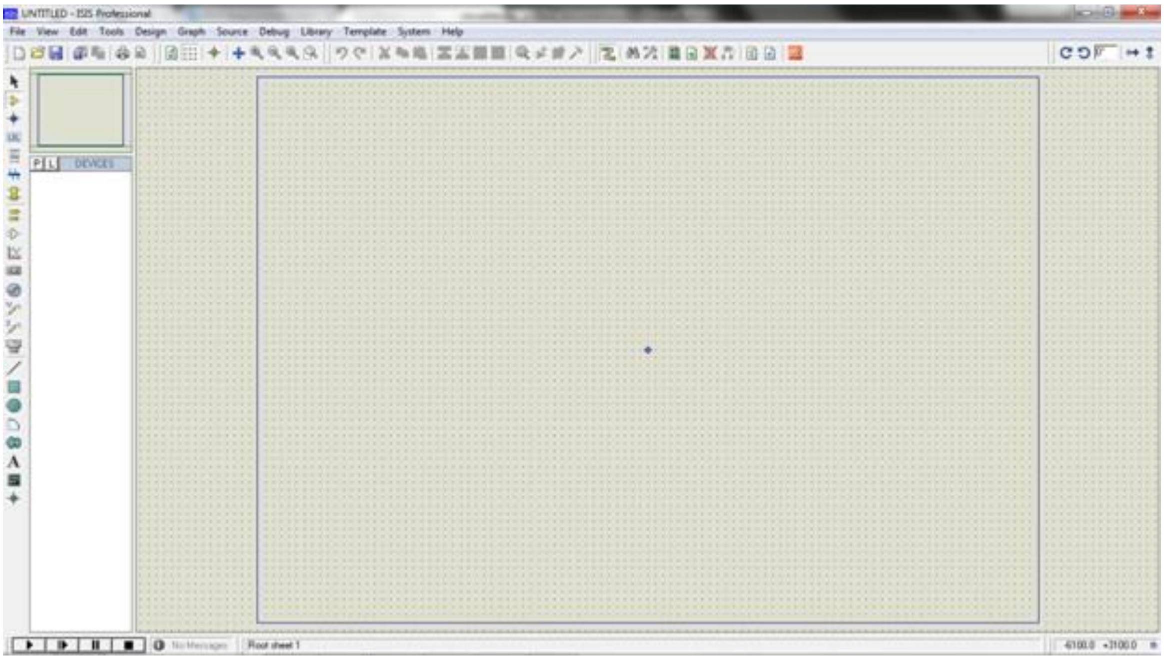


Figure 4.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

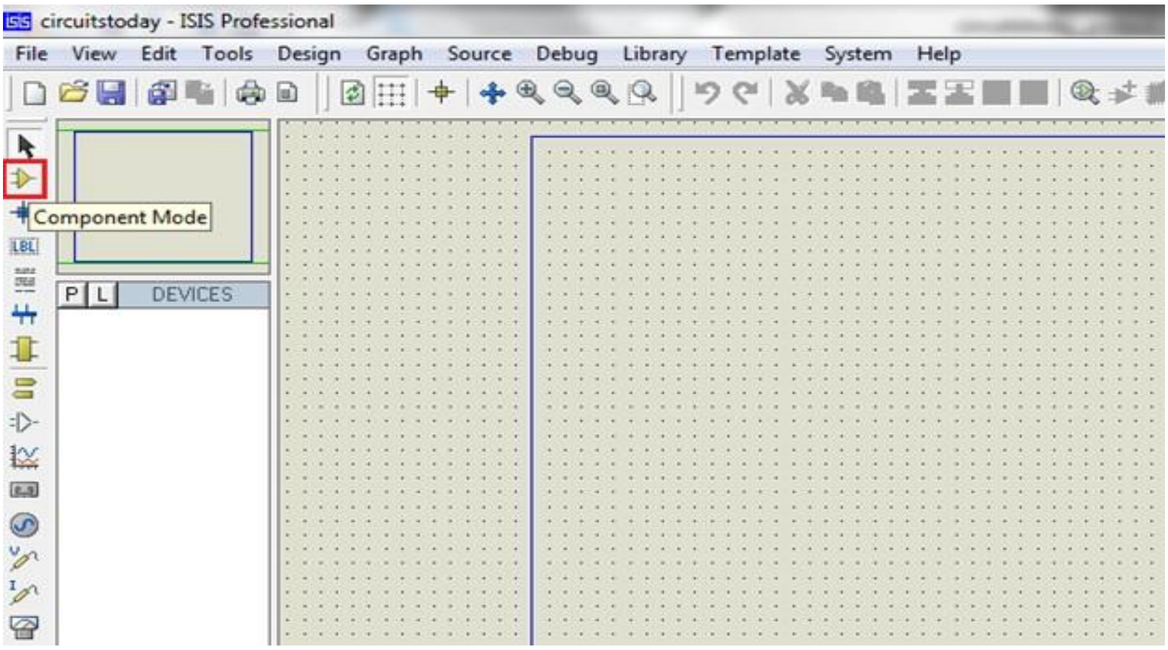


Figure 4.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

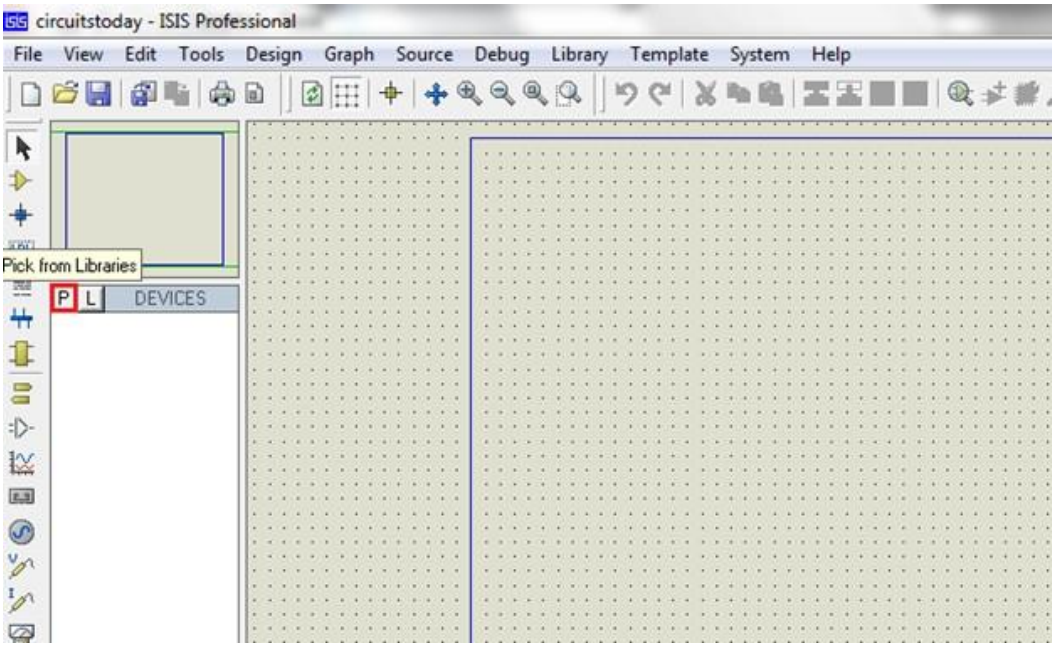


Figure 4.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in the Keywords text box.

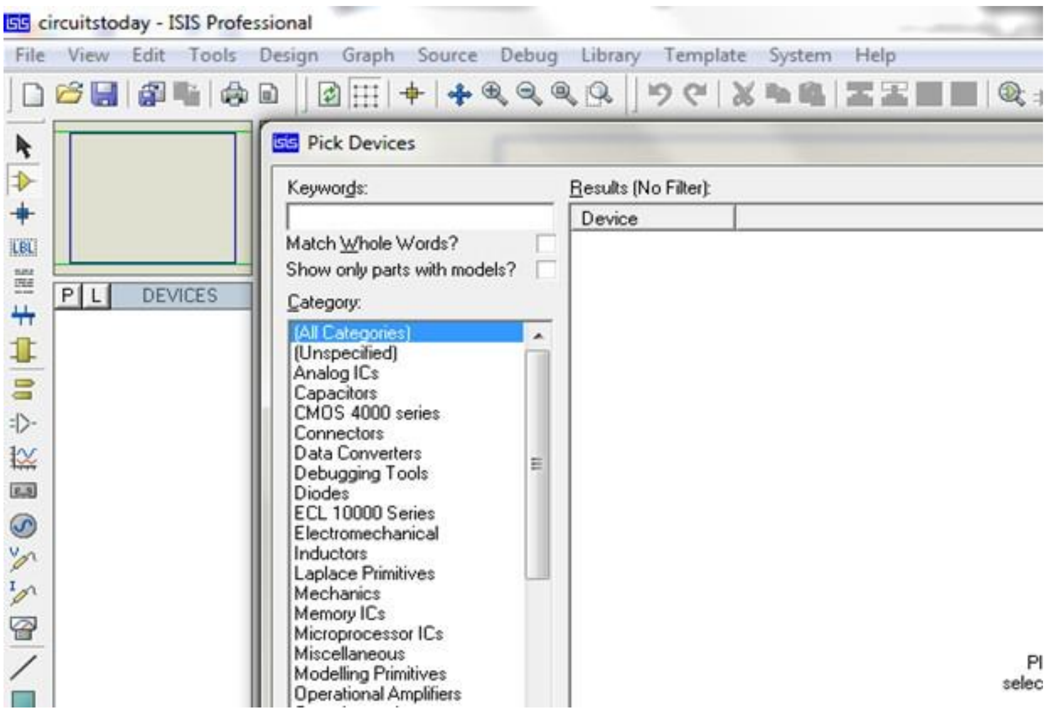


Figure 4.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

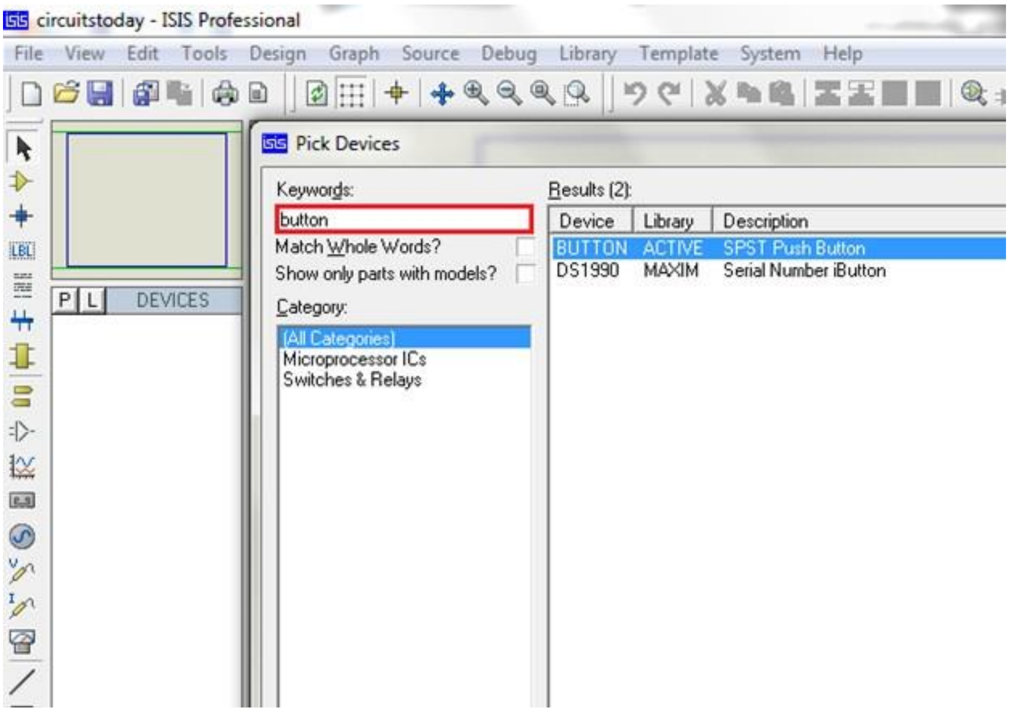


Figure 4.6(a) Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

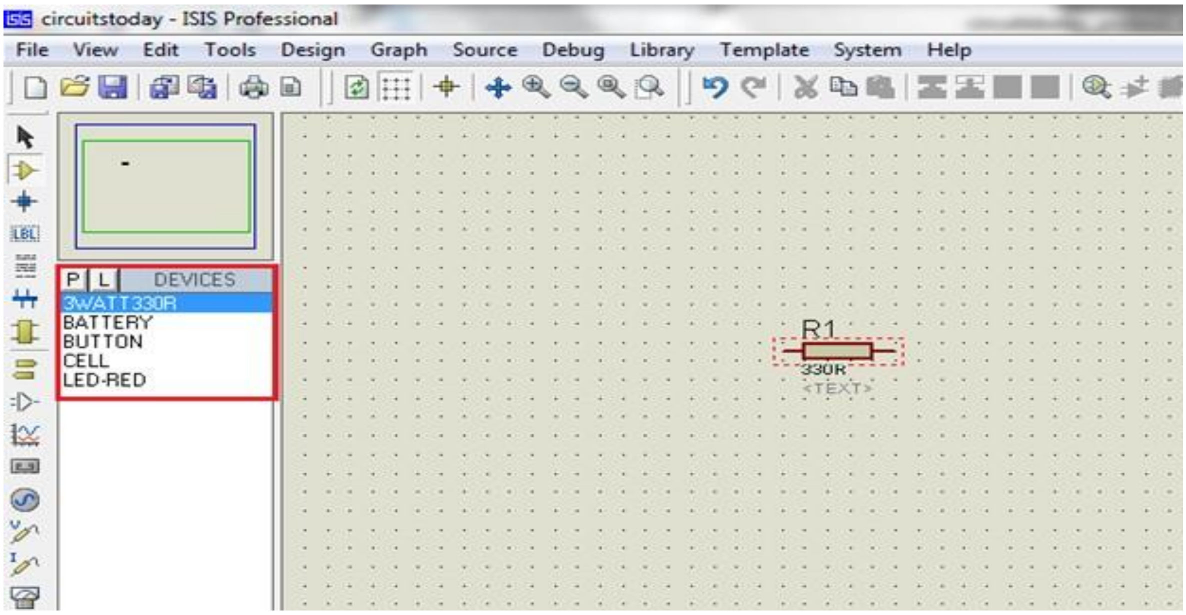


Figure 4.7 Component Selection

Place all the required components and route the wires, make connections.

Either selection mode above the component mode or component mode allows it to connect through wires. Left click from one terminal to another to make a connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

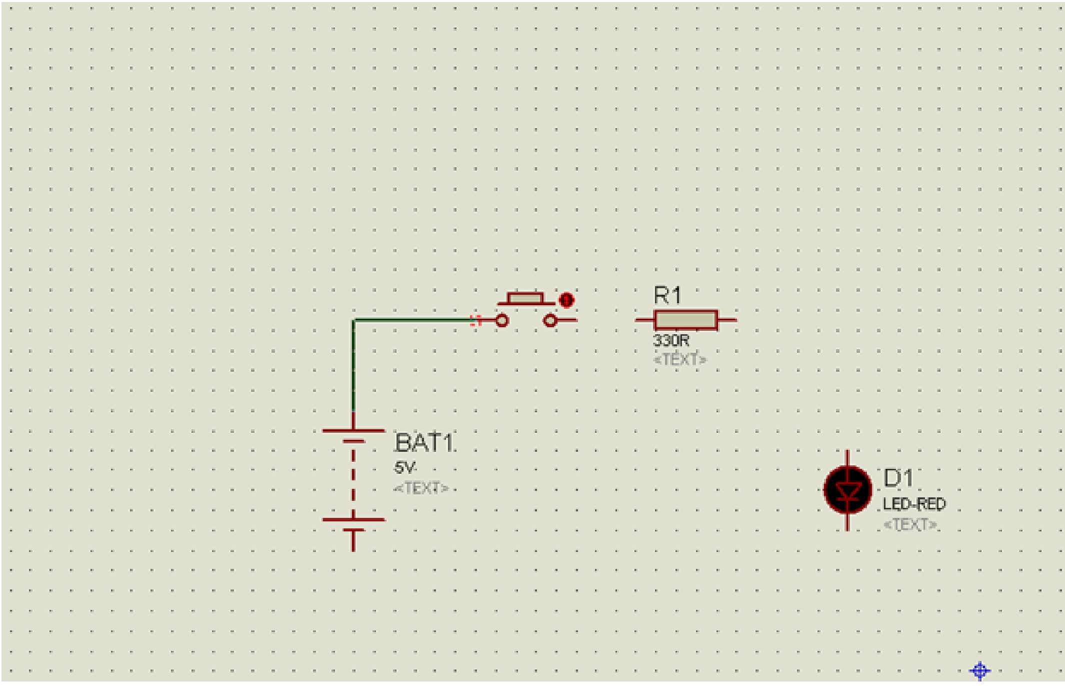


Figure 4.7(a) Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

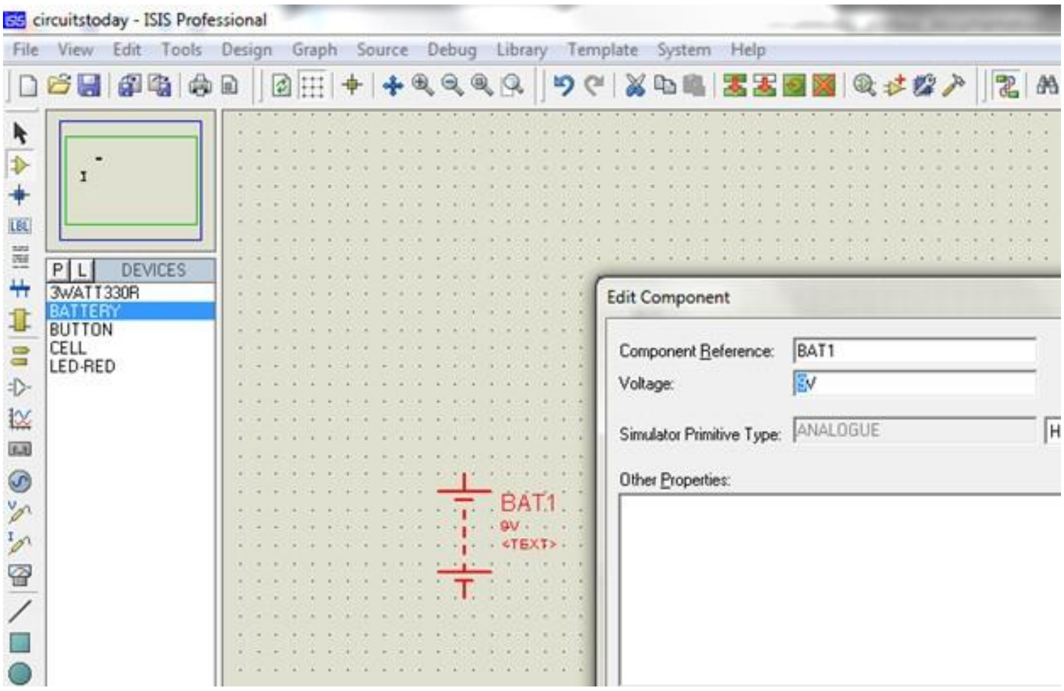


Figure 4.7(b) Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

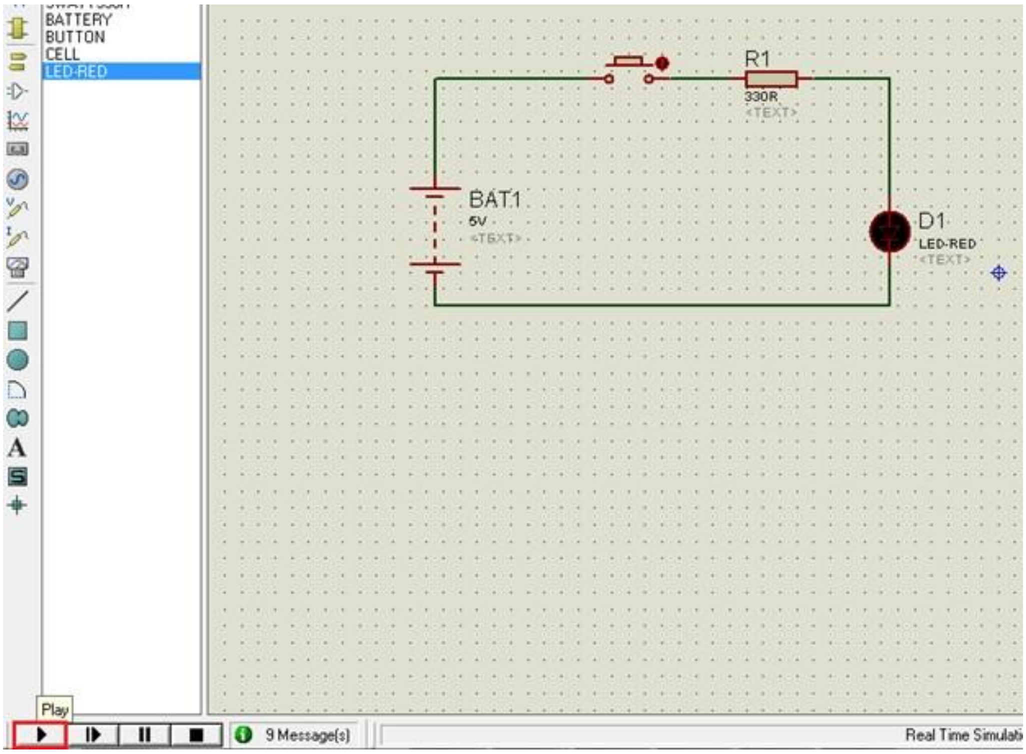


Figure 4.8 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make the LED glow.

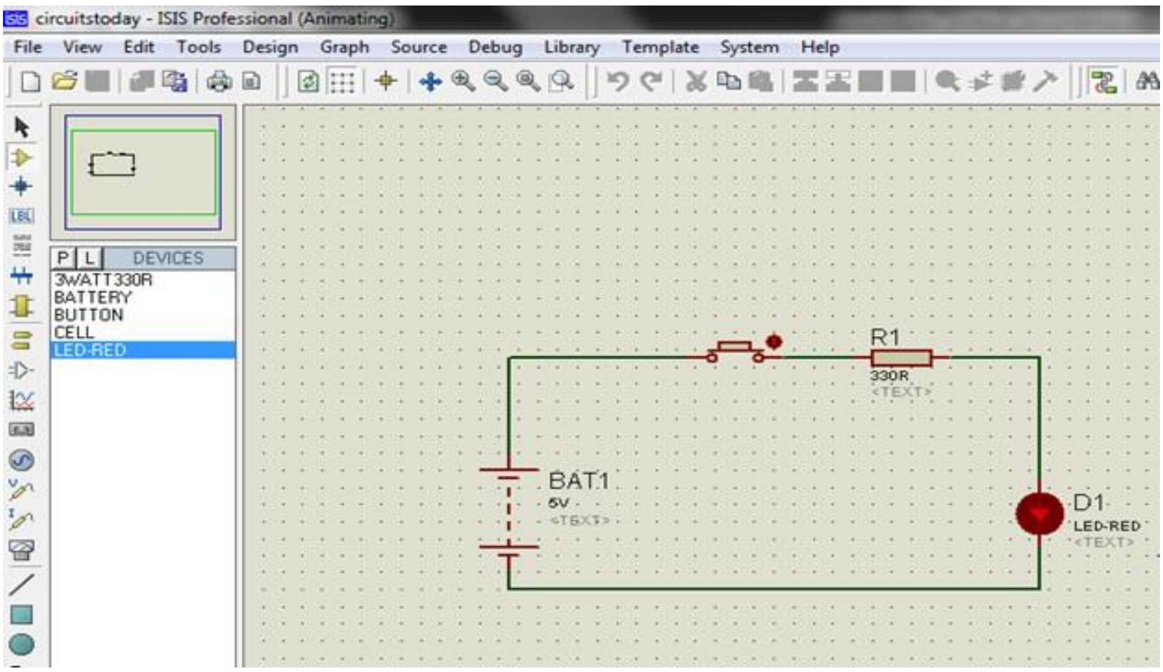


Figure 4.8(a) Simulation Animating

Simulation can be stepped, paused or stopped at any time.

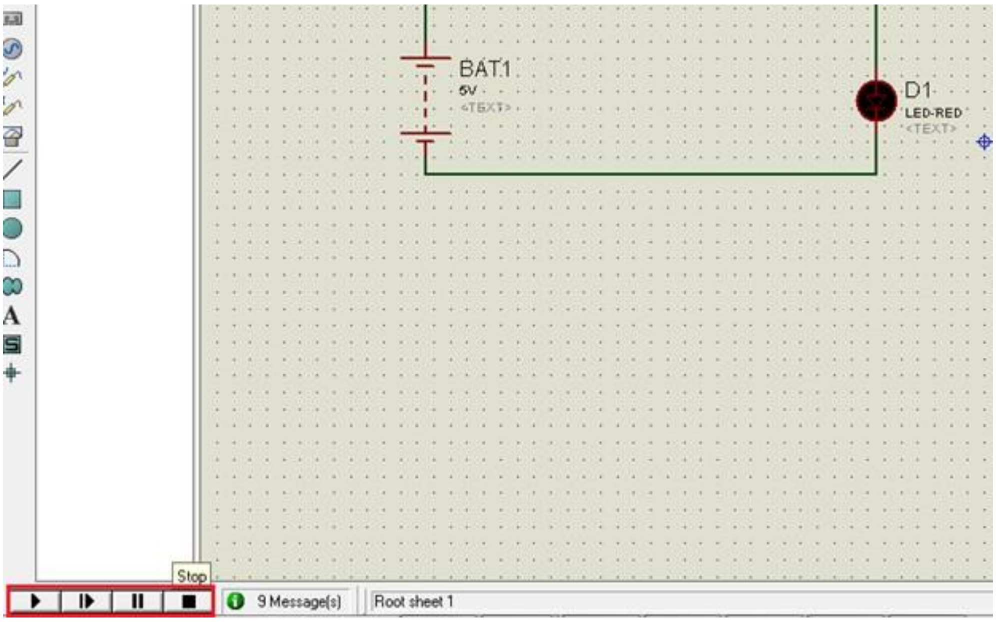


Figure 4.8(b) Simulation Step-Pause-Stop Buttons

Arduino IDE

The Arduino integrated development environment (IDE) is a cross-platform application(for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution.

Arduino Pro IDE	
Developer(s)	Arduino Software
Preview release	v0.0.2 / 28 October 2019; 3 months ago[8]
Repository	github.com/arduino/Arduino
Written in	C, C++
Operating system	Windows, macOS, Linux
Platform	IA-32, x86-64, ARM
Type	Integrated development environment
License	LGPL or GPL license
Website	blog.arduino.cc/2019/10/18/arduino-pro-ide-alpha-preview-with-advanced-features/

In October 2019 the Arduino organization began providing early access to a new Arduino Pro IDE with debugging[9] and other advanced features.

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

Installation:

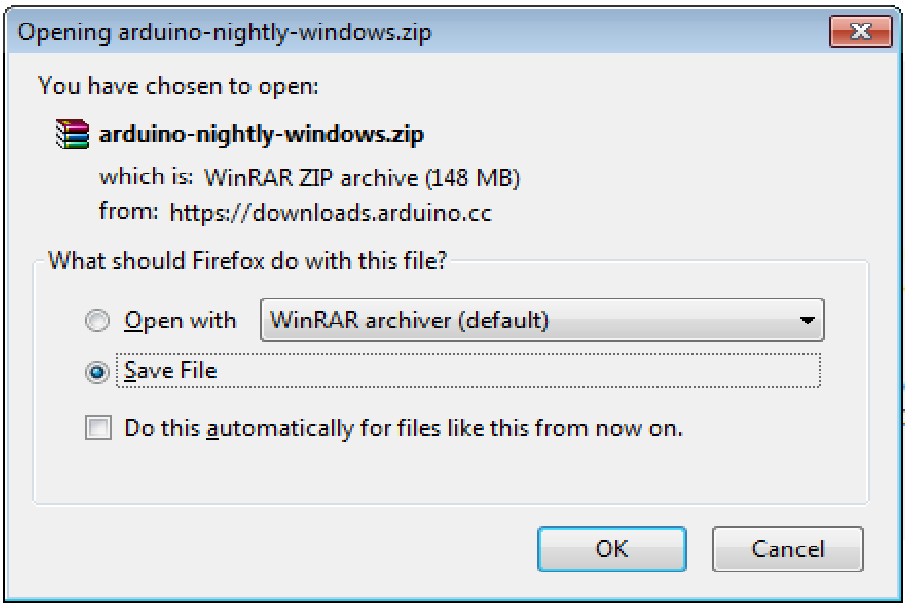
In this section, we will learn in easy steps how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



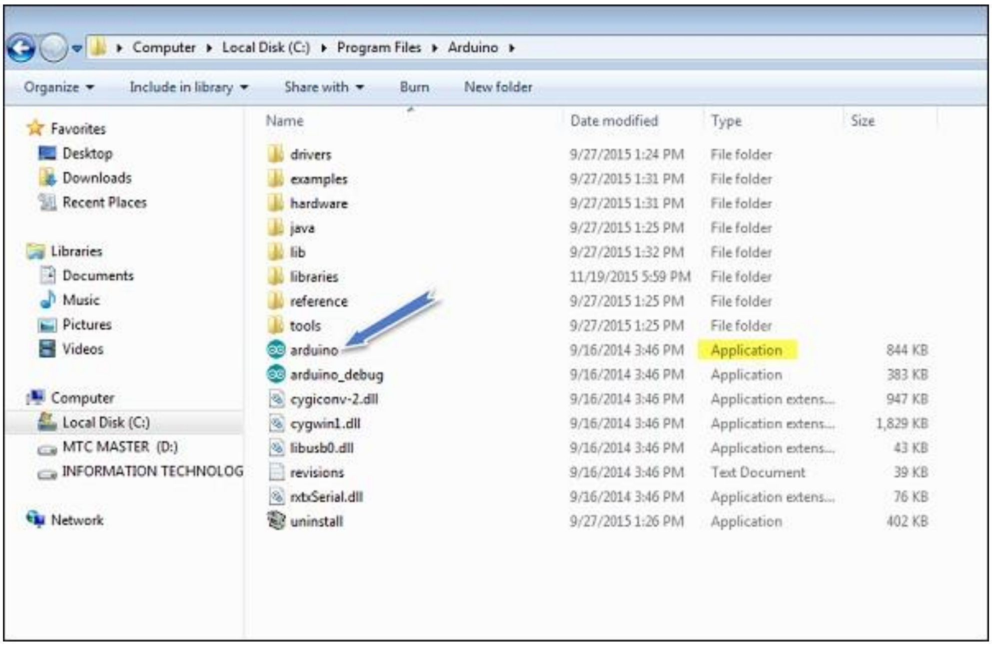
Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

.

Step 4 − Launch Arduino IDE:

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

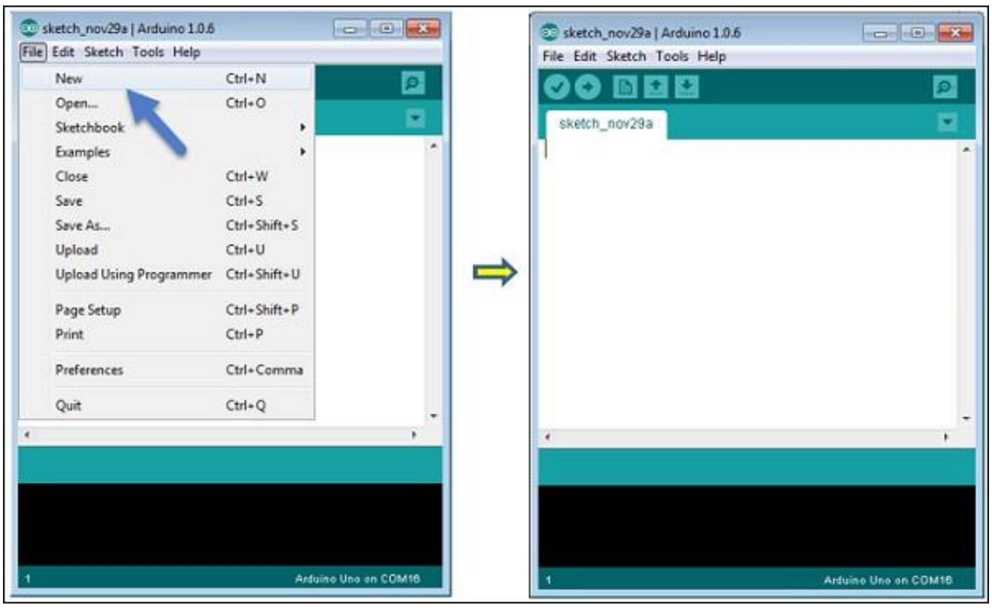


Step 5 − Open your first project.

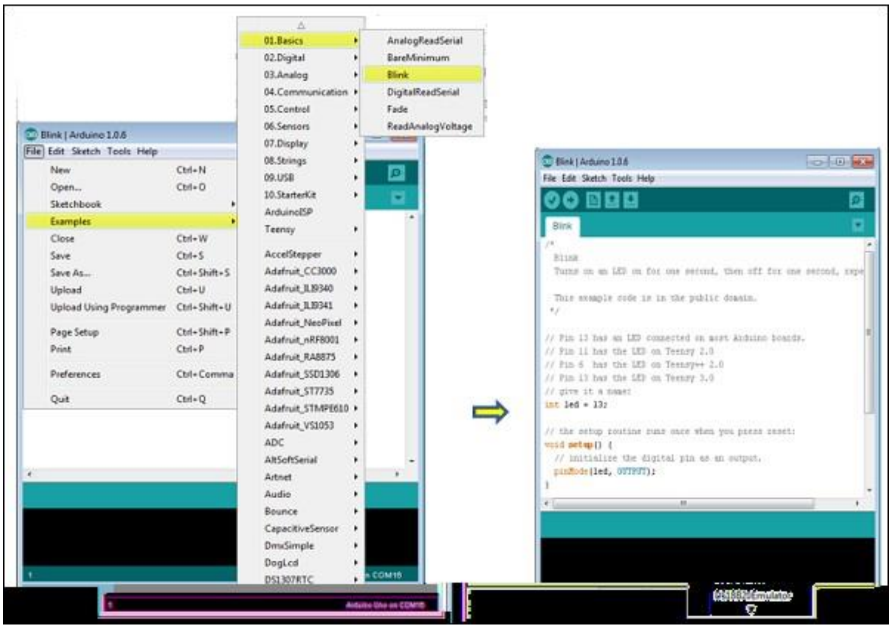
Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

TO create a new prOject, select File → **New**.



TO Open an existing prOject example, select File → Example → Basics → Blink.

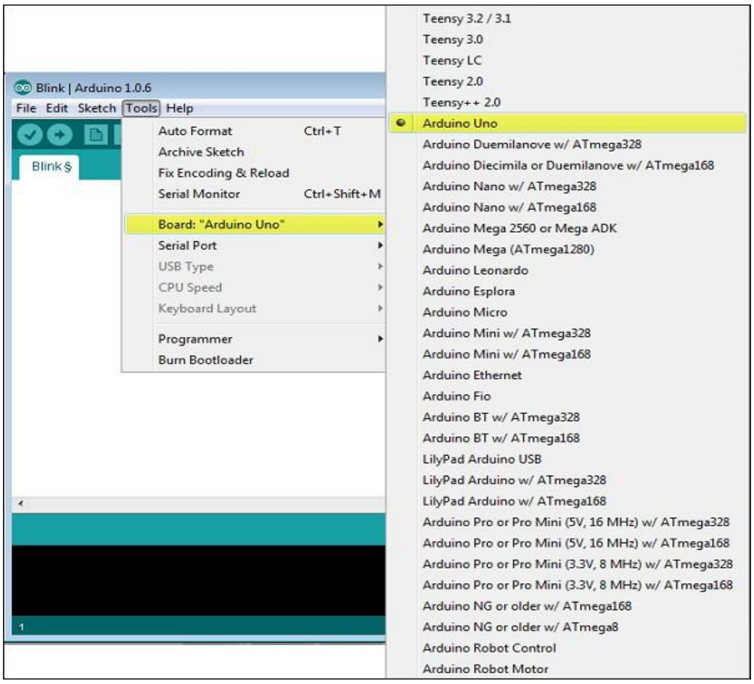


Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

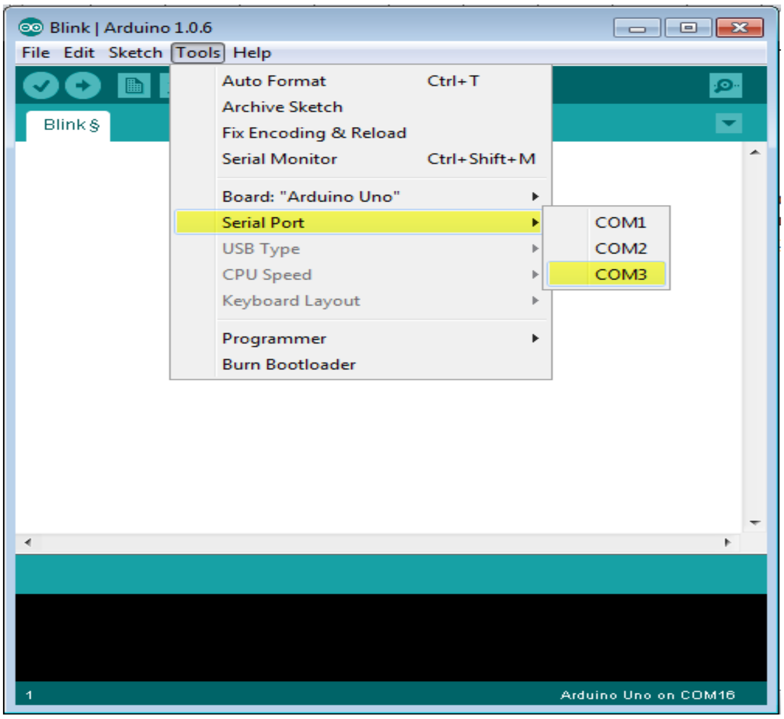
Go to Tools → Board and select your board.



Here, we have selected the Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

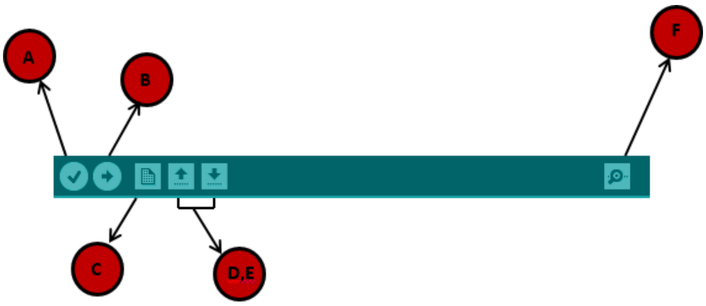
Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to Tools → Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.



Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.



A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketches.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

CHAPTER 5

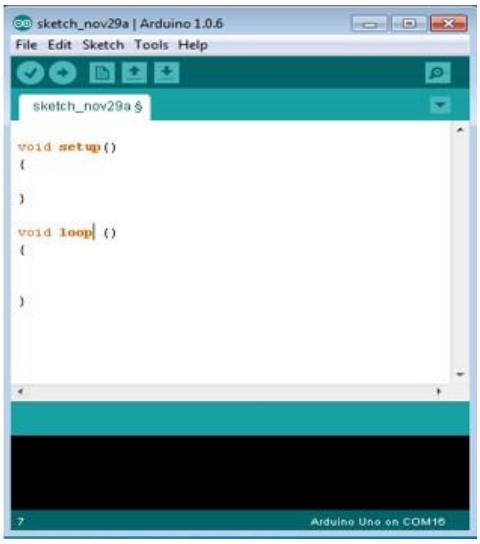
HARDWARE DESCRIPTION

Arduino UNO

Arduino is a prototype platform (open-source) based on easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connecting to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.



Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Table 5.1.1(a) Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header

Arduino Pro mini 5v/16mhz		16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 5.1.1(b) Arduino boards based on ATMEGA32u4 microcontroller

Board Name	**Operating Volt**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 5.1.1(c) Arduino boards based on ATMEGA2560 microcontroller

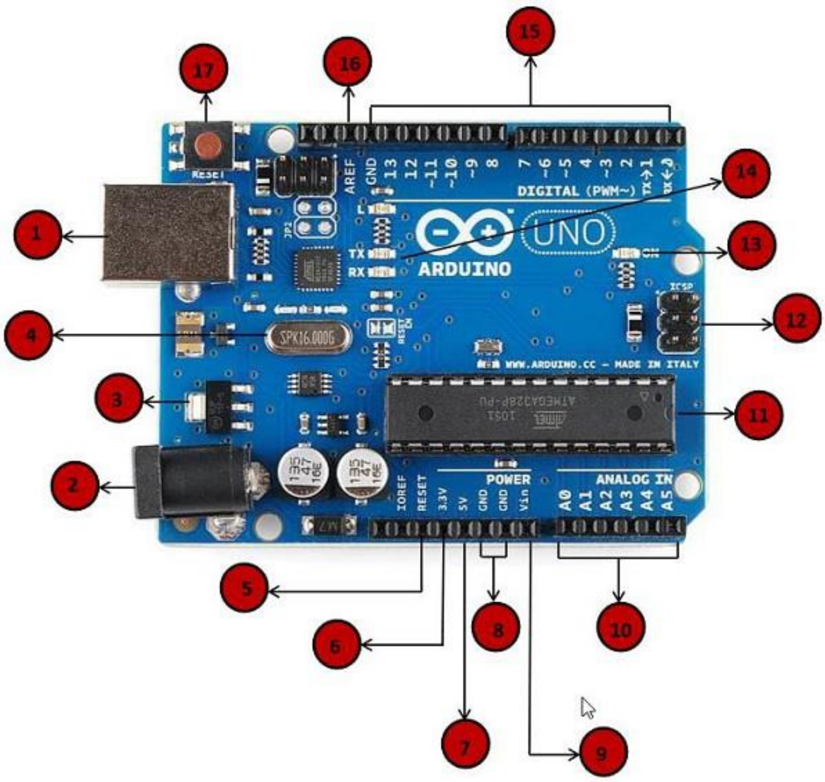
Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2 B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 5.1.1(d) Arduino boards based on AT91SAM3X8E microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have the majority of these components in common.



1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.

5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with the Arduino board work fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often

	referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speeds while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Liquid Crystal Display (LCD):

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to a controller is an LCD display. Some of the most common LCDs connected to the controllers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around the LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus the cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines(RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

Description Of 16x2:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most,It however doesn't show the use of the Status Port as an input.. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

Schematic diagram:

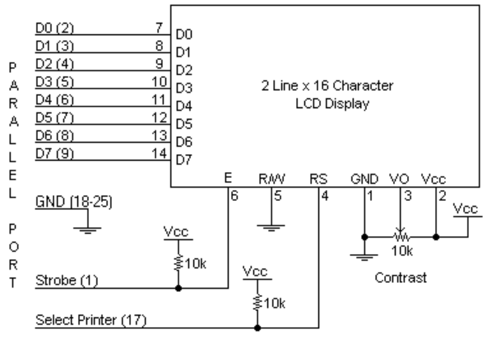


Figure 5.2.2 Schematic Diagram of LCD

- Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.
- We make no effort to place the Data bus into reverse direction. Therefore we hard wire the R/W line of the LCD panel into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.
- The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. You can use a bench power supply set to 5v or use an onboard +5 regulator. Remember a few decoupling capacitors, especially if you have trouble with the circuit working properly.
- Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output.As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction.

16 x 2 Alphanumeric LCD Module Features:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing.
- 61 x 15.8 mm viewing area.
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line.
- Can display 224 different symbols.
- Low power consumption (1 mA typical).
- Powerful command set and user-produced characters.
- TTL and CMOS compatible.
- Connector for standard 0.1-pitch pin headers.

LCD Module Specifications:

Pin	Symbol	Level	Function
1	V_{SS}	-	Power, GND
2	V_{DD}	-	Power, 5V
3	Vo	-	Power, for LCD Drive
4	RS	H/L	Register Select Signal H: Data Input L: Instruction Input
5	R/W	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)
6	E	H,H->L	Enable
7-14	DB0-DB7	H/L	Data Bus; Software selectable 4- or 8-bit mode
15	NC	-	NOT CONNECTED
16	NC	-	NOT CONNECTED

FEATURES:

• 5 x 8 dots with a cursor.

• Built-in controller (KS 0066 or Equivalent)

• + 5V power supply (Also available for + 3V).

• 1/16 duty cycle.

• B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED).

Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are:

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

Figure: Address locations for a 1x16 line LCD

Even limited to character based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8,16,20,24,32 and 40 characters are all standard, in one, two and four line versions.

Several different LC technologies exist. "supertwist" types, for example, offer Improved contrast and viewing angles over the older "twisted nematic" types. Some modules are available with back lighting, so that they can be viewed in dimly-lit conditions. The back lighting may be either "electro-luminescent", requiring a high voltage inverter circuit, or simple LED illumination.

Pin Description

Most LCDs with 1 controller have 14 Pins and LCDs with 2 controllers have 16 Pins (two pins are extra in both for back-light LED connections).

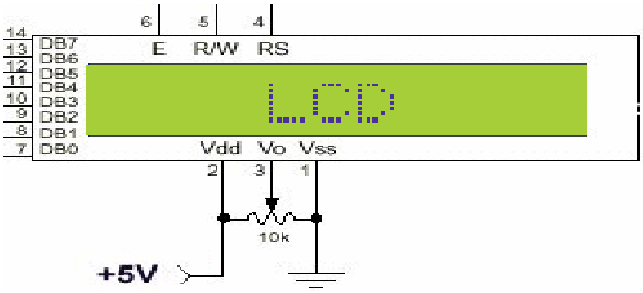


Figure 5.2.4(a) Pin diagram of 1x16 lines LCD

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Fig 5.2.4(b) Pin Specifications

CONTROL LINES:

EN: Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS: Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW: Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are writing commands, so RW will almost always be low.

Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Logic status on control lines:

• E - 0 Access to LCD disabled

- 1 Access to LCD enabled

• R/W - 0 Writing data to LCD

- 1 Reading data from LCD

• RS - 0 Instructions

-1 Character

Writing data to the LCD:

1) Set R/W bit to low
2) Set RS bit to logic 0 or 1 (instruction or character)
3) Set data to data lines (if it is writing)
4) Set E line to high
5) Set E line to low

Read data from data lines (if it is reading) on LCD:

1) Set R/W bit to high
2) Set RS bit to logic 0 or 1 (instruction or character)
3) Set data to data lines (if it is writing)
4) Set E line to high
5) Set E line to low

Entering Text:

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the d.i.l. A switch pack with a re-wiring is necessary.

LCD Commands:

There are some present command instructions in the LCD, which we need to send to the LCD through some microcontroller. Some important command instructions are given below:

Table 5.2.5(a) Commands

Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left
0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

ACS712 Current Sensor

The invention of electricity has led to a revolutionary change in the life of humans. We invented many innovative applications of electricity to make our daily life easier. Today almost all of our equipment runs on electricity. The flow of charge is known as Current. Different devices need a different amount of current based on their functional requirements. Some devices are so sensitive that they get damaged when a high amount of current is delivered to them. So, to save such a situation and monitor the amount of current required or being used in an application, measurement of current is necessary. This is where the Current Sensor comes into play. One such sensor is the ACS712 Current Sensor.

What is the ACS712 Current Sensor?

Current flowing through a conductor causes a voltage drop. The relation between current and voltage is given by Ohm's law. In electronic devices, an increase in the amount of current above its requirement leads to overload and can damage the device.

Measurement of current is necessary for the proper working of devices. Measurement of voltage is Passive task and it can be done without affecting the system. Whereas measurement of current is an Intrusive task which cannot be detected directly as voltage.

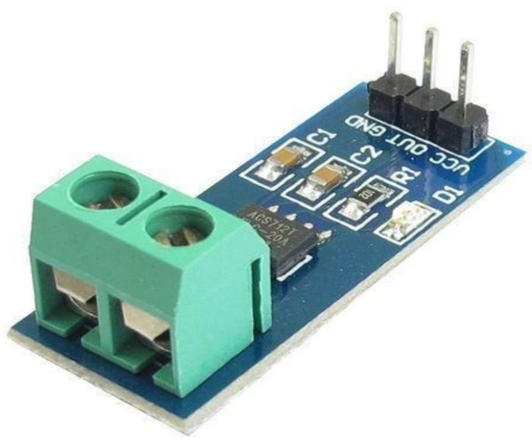


Figure 5.3 ACS712

For measuring current in a circuit, a sensor is required. ACS712 Current Sensor is the sensor that can be used to measure and calculate the amount of current applied to the conductor without affecting the performance of the system.

ACS712 Current Sensor is a fully integrated, Hall-effect based linear sensor IC. This IC has a 2.1kV RMS voltage isolation along with a low resistance current conductor.

Working Principle:

Current Sensor detects the current in a wire or conductor and generates a signal proportional to the detected current either in the form of analog voltage or digital output.

Current Sensing is done in two ways – Direct sensing and Indirect Sensing. In Direct sensing,to detect current, Ohm's law is used to measure the voltage drop occurring in a wire when current flows through it. A current-carrying conductor also gives rise to a magnetic field in its surroundings. In Indirect Sensing, the current is measured by calculating this magnetic field by applying either Faraday's law or Ampere law. Here either a Transformer or Hall effect sensor or fiber optic current sensor are used to sense the magnetic field.

ACS712 Current Sensor uses the Indirect Sensing method to calculate the current. To sense the current liner, a low-offset Hall sensor circuit is used in this IC. This sensor is located at the surface of the IC on a copper conduction path. When current flows through this copper conduction path it generates a magnetic field which is sensed by the Hall effect sensor. A voltage proportional to the sensed magnetic field is generated by the Hall sensor, which is used to measure current.

The proximity of the magnetic signal to the Hall sensor decides the accuracy of the device. Nearer the magnetic signal higher the accuracy. ACS712 Current Sensor is available as a small, surface mount SOIC8 package. In this IC current flows from Pin-1 and Pin-2 to Pin-3 and Pin-4. This forms the conduction path where the current is sensed. Implementation of this IC is very easy.

ACS712 can be used in applications requiring electrical isolation as the terminals of the conduction path are electrically isolated from the IC leads. Thus, this IC doesn't require any other isolation techniques. This IC requires a supply voltage of 5V. Its output voltage is proportional to AC or DC current. ACS712 has a nearly zero magnetic hysteresis.

Where Pin-1 to Pin-4 forms the conduction path, Pin-5 is the signal ground pin. Pin-6 is the FILTER pin that is used by an external capacitor to set the bandwidth. Pin-7 is the analog output pin. Pin-8 is the power supply pin.

Applications of ACS712 Current Sensor:

This IC can detect both AC and DC current so it has a wide range of applications. ACS712 is used in Peak detection circuits, circuits to increase gain, rectification application for AtoD converters, Overcurrent fault latch, etc…The filter pin provided by this IC is used to eliminate the attenuation effect in resistor divider circuits.

ACS712 is used in many industrial, commercial and communication applications. This IC is applicable for Automobile applications. Some of the typical applications of this IC can be found in motor control circuits, for load detection and management, SMPS, overcurrent fault protection circuits.

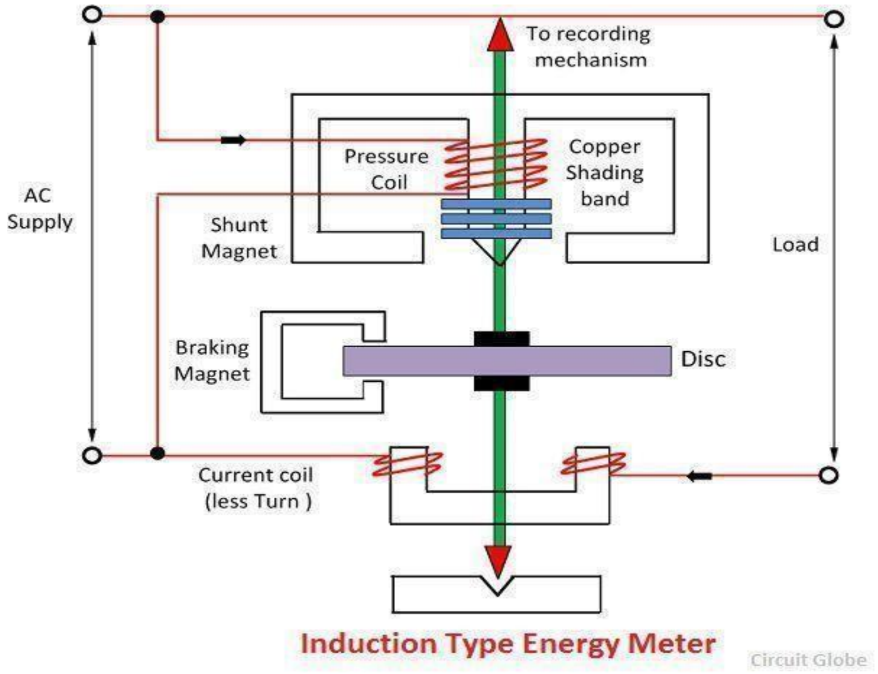
This IC can measure current for high voltage loads operating at 230V AC mains. To read the values it can be easily interfaced with the ADC of a microcontroller. What would be the value of output voltage provided by ACS712 when a DC load current is applied to it?

Energy Meter:

Definition: The meter which is used for measuring the energy utilised by the electric load is known as the energy meter. The energy is the total power consumed and utilised by the load at a particular interval of time. It is used in domestic and industrial AC circuits for measuring the power consumption.

Construction of Energy Meter:

The construction of the single phase energy meter is shown in the figure below.



The energy meter has four main parts. They are the

1. Driving System
2. Moving System
3. Braking System
4. Registering System

The detailed explanation of their parts is written below.

1. Driving System – The electromagnet is the main component of the driving system. It is the temporary magnet which is excited by the current flow through their coil. The core of the electromagnet is made up of silicon steel lamination. The driving system has two electromagnets. The upper one is called the shunt electromagnet, and the lower one is called the series electromagnet.

The series electromagnet is excited by the load current flow through the current coil. The coil of the shunt electromagnet is directly connected with the supply and hence carries the current proportional to the shunt voltage. This coil is called the pressure coil.The upper one is called the shunt electromagnet, and the lower one is called the series electromagnet.

The centre limb of the magnet has the copper band. These bands are adjustable. The main function of the copper band is to align the flux produced by the shunt magnet in such a way that it is exactly perpendicular to the supplied voltage.

2. Moving System – The moving system is the aluminium disc mounted on the shaft of the alloy. The disc is placed in the air gap of the two electromagnets. The eddy current is induced in the disc because of the change of the magnetic field. This eddy current is cut by the magnetic flux. The interaction of the flux and the disc induces the deflecting torque.

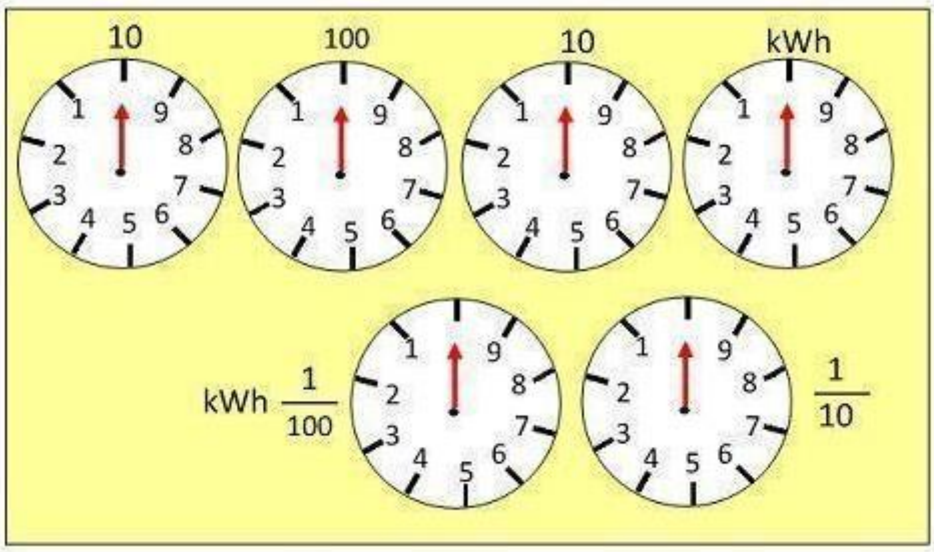
When the devices consume power, the aluminium disc starts rotating, and after some number of rotations, the disc displays the unit used by the load. The number of rotations of the disc is counted at a particular interval of time. The disc measured the power consumption in kilowatt hours.

3. Braking system – The permanent magnet is used for reducing the rotation of the aluminium disc. The aluminium disc induces the eddy current because of their rotation. The eddy current cuts the magnetic flux of the permanent magnet and hence produces the braking torque.

This braking torque opposes the movement of the disc, thus reducing their speed. The permanent magnet is adjustable due to which the braking torque is also adjusted by shifting the magnet to the other radial position.

4. Registration (Counting Mechanism) – The main function of the registration or counting mechanism is to record the number of rotations of the aluminium disc. Their rotation is directly proportional to the energy consumed by the loads in the kilowatt hour.

The rotation of the disc is transmitted to the pointers of the different dial for recording the different readings. The reading in kWh is obtained by multiplying the number of rotations of the disc with the meter constant. The figure of the dial is shown below.



Pointer Type of Register

Working of the Energy Meter:

The energy meter has the aluminium disc whose rotation determines the power consumption of the load. The disc is placed between the air gap of the series and shunt electromagnet. The shunt magnet has the pressure coil, and the series magnet has the current coil.The pressure coil creates the magnetic field because of the supply voltage, and the current coil produces it because of the current.

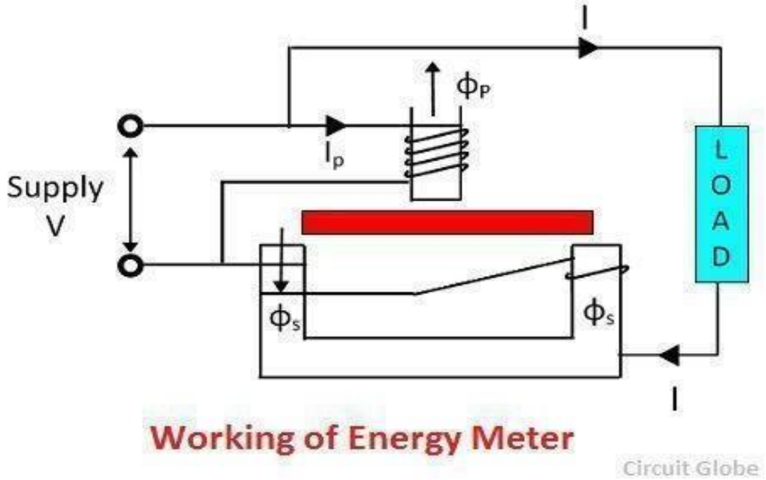
The field induced by the voltage coil is lagging by 90º on the magnetic field of the current coil because of which eddy current is induced in the disc. The interaction of the eddy current and the magnetic field causes torque, which exerts a force on the disc. Thus, the disc starts rotating.

The force on the disc is proportional to the current and voltage of the coil. The permanent magnet controls Their rotation. The permanent magnet opposes the movement of the disc and equalises it on the power consumption. The cyclometer counts the rotation of the disc.

The energy meter has the aluminium disc whose rotation determines the power consumption of the load. The disc is placed between the air gap of the series and shunt electromagnet.

Theory of Energy Meter:

The pressure coil has the number of turns which makes it more inductive. The reluctance path of their magnetic circuit is very less because of the small length air gap. The current I_p flows through the pressure coil because of the supply voltage, and it lags by 90º.



Working of Energy Meter

The I_p produces the two Φ_p which is again divided into Φ_{p1} and Φ_{p2}. The major portion of the flux Φ_{p1} passes through the side gap because of low reluctance. The flux Φ_{p2} goes through the disc and induces the driving torque which rotates the aluminium disc.

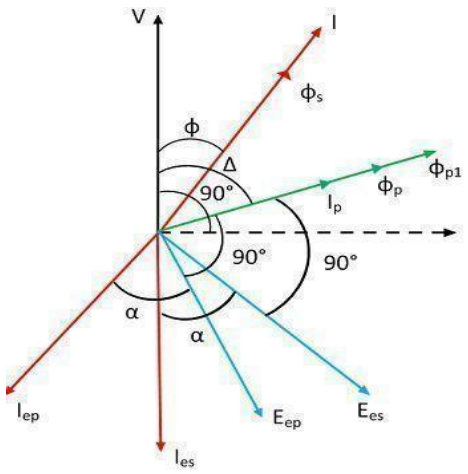
The flux Φ_p is proportional to the applied voltage, and it is lagged by an angle of 90º. The flux is alternating and hence induces an eddy current I_{ep} in the disc.

The load current passing through the current coil induces the flux Φ_s. This flux causes the eddy current I_{es} on the disc. The eddy current I_{es} interacts with the flux Φ_p, and the eddy current I_{ep} interacts with Φ_s to produce another torque. These torques are opposite in direction, and the net torque is the difference between these two.

The flux Φp2 goes through the disc and induces the driving torque which rotates the aluminium disc.

The flux Φp is proportional to the applied voltage, and it is lagged by an angle of 90º. The flux is alternating and hence induces an eddy current Iep in the disc.

The phasor diagram of the energy meter is shown in the figure below.



Phasor Diagram of Energy Meter

Circuit Globe

Let

V – applied voltage

I – load current

Ø – the phase angle of load current

I_p – pressure angle of load

Δ – the phase angle between supply voltage and pressure coil flux

f – frequency

Z – impedance of eddy current

a – the phase angle of eddy current paths

E_{ep} – eddy current induced by flux

I_{ep} – eddy current due to flux

E_{ev} – eddy current due to flux

I_{es} – eddy current due to flux

The net driving torque of the dis is expressed as

$$T_d \propto \varnothing_1 \varnothing_2 \frac{f}{Z} sin\beta cos\alpha = K_1 \varnothing_1 \varnothing_2 \frac{f}{Z} sin\beta cos\alpha$$

where K_1 – constant

Φ_1 and Φ_2 are the phase angles between the fluxes. For energy meters, we take Φ_p and Φ_s.

β – phase angle between fluxes Φ_p and Φ_p = $(\Delta - \Phi)$, therefore

$$Driving\ Torque, T_d = K_1 \emptyset_1 \emptyset_2 \frac{f}{Z} \sin(\Delta - \emptyset)\ cos\alpha$$

$$But\ \emptyset_p \propto V, and\ \emptyset_p \propto I$$

$$T_d \propto K_2 VI \frac{f}{Z} \mathrm{Sin}(\Delta - \emptyset) cos\alpha$$

If f, Z and α are constants,

$$T_d = K_3 VIsin(\Delta - \emptyset)$$

If N is steady speed, braking torque

$$T_B = K_4 N$$

At a steady state, the speed of the driving torque is equal to the braking torque.

$$K_4 N = K_3 VI(\Delta - \emptyset)$$

$$N = KVIsin(\Delta - \emptyset)$$

If $\Delta = 90^o$,

$$N = KVIsin(90° - \emptyset) = KVIcos\emptyset$$

Speed,

$$= K \times power$$

The speed of the rotation is directly proportional to the power.

$$Total\, number\, of\, revolution = \int Ndt = K\int VIsin(\Delta - \emptyset)$$

If $\Delta = 90^{o}$, total number of revolutions

$$= K\int VIcos\emptyset dt$$

$$= K\int power\ dt = K\ X\ energy$$

The three phase energy meter is used for measuring the large power consumption.

WIFI ESP8266:

Working:

The ESP8266 WiFi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, and is designed to occupy a minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth coexistence interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

Note: The ESP8266 Module is not capable of 5-3V logic shifting and will require an external Logic Level Converter. Please do not power it directly from your 5V dev board. This ESP8266 WiFi Module Adaptor comes with an on-board logic level converter and 3.3V LDO regulator for easy interfacing to Arduino boards.

Features:

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- Default baud rate: 115200

LDR

Working

Photo resistors, also known as light dependent resistors (LDR), are light sensitive devices most often used to indicate the presence or absence of light, or to measure the light intensity. LDRs have a sensitivity that varies with the wavelength of the light applied and are nonlinear devices.

LDR is a component that has a (variable) resistance that changes with the light intensity that falls upon it. This allows them to be used in light sensing circuits.

When an LDR is exposed to a light of high intensity, the resistance value will decrease. It could drop from 1 MΩ to 2 kΩ. Circuit of a day/night switch npn transistor is the control device, and an LED is the output device. While the capacitor is charging, the LED1 will be ON.

The light energy produces more free electrons which increases the current for a certain voltage across the LDR which means a drop in resistance. Very little current will flow through it when it is cold. This means that its resistance increases as the temperature falls.

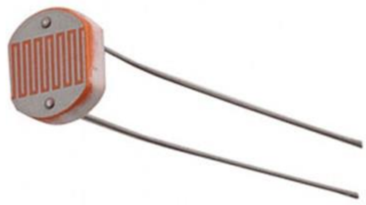


Figure 5.6 LDR

Specifications:

Electrical Characteristics

Parameter	Conditions	Min	Typ	Max	Unit
Cell resistance	1000 LUX 10 LUX	- -	400 9	- -	Ohm K Ohm
Dark Resistance	-	-	1	-	M Ohm
Dark Capacitance	-	-	3.5	-	pF
Rise Time	1000 LUX 10 LUX	- -	2.8 18	- -	ms ms
Fall Time	1000 LUX 10 LUX	- -	48 120	- -	ms ms
Voltage AC/DC Peak		-	-	320	V max
Current		-	-	75	mA max
Power Dissipation				100	mW max
Operating Temperature		-60	-	+75	Deg. C

Figure 5.6.2 LDR Specifications

Advantages:

LDR's are cheap and are readily available in many sizes and shapes. Practical LDRs are available in a variety of sizes and package styles, the most popular size having a face diameter of roughly 10 mm. They need very small power and voltage for its operation.

Disadvantages:

Highly inaccurate with a response time of about tens or hundreds of milliseconds.

CHAPTER 6

RESULT ANALYSIS

RESULT:

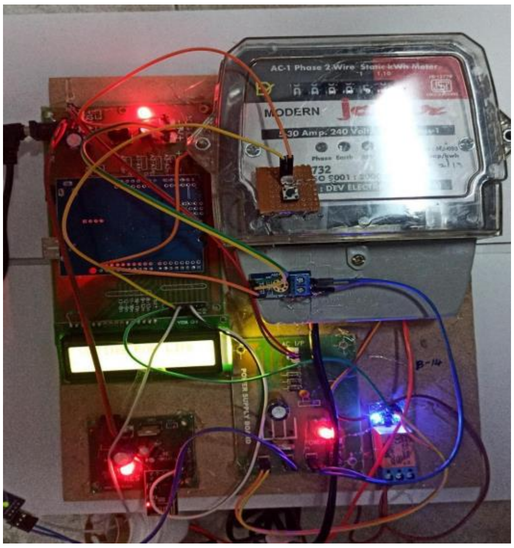


Figure.6 : IOT ENERGY METER WITH CURRENT, VOLTAGE AND COST MONITORING SYSTEM

CASE 1:

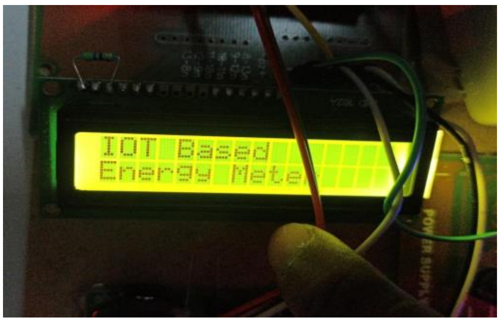


Figure.6.1: Welcome message

- This figure represents the welcome message instead of that we are displaying IOT Based Energy Meter.

CASE 2:

figure.6.2: LCD Displaying recharge update.

- This figure represents that the recharge is done and the unit's consumption has come to zero.
- Now the reading will start and it is displayed on the LCD screen.

CASE 3:

Figure 6.3 : LCD Displaying maximum units consumed.

- This figure represents that we have consumed maximum units so the power supply will be turned off.

- Once the recharge is done we will get the power supply again. So we get a warning message as maximum units are consumed on the LCD screen.

CASE 4:

Figure 6.4 : Displaying the notifications on a mobile application

- Connecting the mobile application with an IP address. Once it is connected we will receive the data through a wifi module which is connected to the system.
- For every reading we can see the notification within the app.
- Once it comes to a set limit we get warning notifications as max units consumed till we recharge it or pay the bill.
- We will receive the notification of recharge done as shown in figure.

CHAPTER 7

LIMITATIONS & ADVANTAGES

LIMITATIONS:

- We can't track or monitor the usage of our energy consumption at residency and industries.
- System has to be maintained manually.
- Managing and storing the vast quantities of metering data.
- Ensuring security of metering data.

ADVANTAGES:

- It provides accuracy in meter readings.
- This system helps in effectively controlling energy consumption and also avoids energy wastage.
- meter reading can be accessed from anywhere on the globe at any time.
- This system eliminates human involvement in energy management.
- It reduces the human interference to collect the monthly reading.

CHAPTER 8

CONCLUSION & FUTURE ENHANCEMENT

CONCLUSION:

- This system helps in controlling energy consumption and avoiding energy wastage.
- This IOT based energy meter plays a vital role in continuously monitoring the energy consumption and transferring the data to the server. This data can be accessed from anywhere on the globe at any time.

FUTURE ENHANCEMENT:

- It can be introduced in the department of power and energy and can be sanctioned for the collection of details of the consumption units for each and every customer.
- Further it can be linked with any payment gateway so that the electricity bill payment can also be done online with it.

CODE

```
#include<LiquidCrystal.h>
#include<SoftwareSerial.h>
LiquidCrystal lcd(13,12,11,10,9,8);
SoftwareSerial mySerial(2,3);//rx , tx //esp8266

int ldr=4;
int sw=5;
int load=6;

const int voltageSensor = A4;
float vOUT = 0.0;
float vIN = 0.0;
float R1 = 30000.0;
float R2 = 7500.0;
int value = 0;
unsigned int count=0;

void serialFlush(){
  while(Serial.available() > 0) {
  char tt = Serial.read();
  }
}
void sendwifi(String char,unsigned int len)
```

```
{
  mySerial.print("AT+CIPSEND=0,");
  mySerial.println(len-1);
  delay(2000);
  mySerial.print(chr);
  delay(2000);
}
char res[130];
char buff[130];

void setup()
{
  lcd.begin(16,2);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("IOT Based");
  lcd.setCursor(0,1);
  lcd.print("Energy Meter");
  pinMode(ldr,INPUT);
  pinMode(sw,INPUT_PULLUP);
  pinMode(load,OUTPUT);
  digitalWrite(load,HIGH);
  delay(3000);
  Serial.begin(9600);
  mySerial.begin(115200);//esp8266
```

```
mySerial.print("AT\r\n");
 delay(1000);
 mySerial.print("ATE0\r\n");
 delay(1000);
 mySerial.print("AT+CWMODE=3\r\n");
 delay(1000);
 mySerial.print("AT+CWSAP=\"Project\",\"project1235\",5,0\r\n\r\n");
 delay(1000);
 mySerial.print("AT+CIPMUX=1\r\n");
 delay(1000);
 mySerial.print("AT+CIPSERVER=1,23\r\n");
 delay(1000);
 lcd.clear();lcd.setCursor(0, 0);lcd.print("WAITING FOR LINK");
 while(1)
 {
  if(mySerial.available())
  {
    //if(Esp.find("0,LINK"))
   if(mySerial.find("0,CONNECT"))
   {
      lcd.clear();lcd.setCursor(0, 0);lcd.print("LINK FOUND");
    break;
   }
  }
 }
```

```
    delay(1000);
    sendwifi("WELCOME \r\n",11);
    delay(1000);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("READY");
  delay(2000);
}
void loop()
{
  value = analogRead(voltageSensor);
  vOUT = (value * 5.0) / 1024.0;
  vIN = vOUT / (R2/(R1+R2));
  lcd.setCursor(0,0);
  lcd.print("Input = ");
  lcd.print(vIN);
  delay(1000);
  if(digitalRead(sw)==LOW)
  {
    count=0;
    lcd.clear();lcd.setCursor(0,0);lcd.print("Units: ");lcd.print(count);
    lcd.setCursor(0,1);lcd.print("Recharge Done");
    sprintf(buff,"Recharge Done..%u\r\n",0);
    sendwifi(buff,strlen(buff));
    digitalWrite(load,HIGH);
    delay(3000);
  }
```

```
 if(digitalRead(ldr)==LOW)
 {
   count++;
   lcd.clear();lcd.setCursor(0,0);lcd.print("Units: ");lcd.print(count);
   sprintf(buff,"Units: %u\r\n",count);
   sendwifi(buff,strlen(buff));
   delay(1000);
 }
 if(count>10)
 {
   lcd.clear();lcd.setCursor(0,0);lcd.print("Units: ");lcd.print(count);
   lcd.setCursor(0,1);lcd.print("Max Units Consumed");
   sprintf(buff,"Max Units Consumed: %u\r\n",count);
   sendwifi(buff,strlen(buff));
   digitalWrite(load,LOW);
   delay(2000);
 }
}
```

REFERENCES

1. Anitha.k ,prathik, Smart Energy Meter surveillance Using IoT ,Institute of Electrical and Electronics Engineers(IEEE), 2019.

2. Devadhanishini, et.al Smart Power Monitoring Using IoT5th International Conference on Advanced Computing & Communication Systems (ICACCS) 2019.

3. Mohammad Hossein Yaghmaee Design and Implementation of an Internet of Things Based Smart Energy Metering 6th IEEE International Conference on Smart Energy Grid Engineering 2018.

4. Himanshu kpatel arduino based smart energy meter 2nd Int'l Conf. on Electrical Engineering and Information & Communication Technology (ICEEICT) 2018.

5. Bibek Kanti Barman, et.al proposed a paper smart meter using IoT department of international electronics and electrical engineering (IEEE) 2017.

6. Garrab.A, Bouallegue.A, Ben Abdullah, A new AMR approach for energy savings in Smart Grids using Smart meter and partial power line communication, IEEE First International Conference on ICICS,vol 3, pp. March 2012.

7. Landi,c.: Dipt. Dining.dellIng, SecondaUniv di Napoli,Aversa,Italy; Merola p.ARM-based energy management system using smart meter and Web server,IEEE instrumentation and measurement technology conference beijing, pp.1-5 may 2011.

8. B. S. Koay, S. S. Cheah, Y. H. Sng, P. H. Chong, P. Shum, Y. C. Tong, X. Y. Wang, Y. X. Zuo and H. W. Kuek, "Design and implementation of Bluetooth energy meter", IEEE Proceedings of the 4th International Joint Conference of the ICICS, vol. 3, pp. 1474-1477, Dec,2003.

A

MAJOR PROJECT REPORT

On

ALCOHOL DETECTION AND HEALTH MONITORING SYSTEM USING IOT

Submitted By

1) G. SUSMITHA -17K81A04B3
2) B.JYESHNA -18K85A0412
3) P.Kiran Kumar-18K85A0422

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Mrs.Ch.Swathi, M.Tech

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

LASYA INFOTECH

St.MARTIN'S ENGINEERING COLLEGE

AN AUTONOMOUS INSTITUTE

(AffiliatedtoJawaharlalNehruTechnological University,Hyderabad)

Dhulapally, Secunderabad 500100

NBA&NAACA+Accredited

2020-2021

Department Of Electronics &Communication Engineering

CERTIFICATE

This is to certify that the project entitled "ALCOHOL DETECTION AND HEALTH MONITORING SYSTEM USING IOT",is being submitted by 1.**Ms.J.Susmitha(17K81A04B3),2.Ms.B.Jyeshna(18K85A0412),3.Mr.P.KiranKumar(18K85A0422)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Internal Guide	**Head of the Department**
Mrs. CH.Swathi **Assistant Professor**	**Dr.B. Hari Krishna** **Professor**

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT G.SUSMITHA WITH ROLLNO.17K81A04B3, B.JYESHNA WITH ROLLNO.18K85A0412,P.KIRAN KUMAR WITH ROLLNO.18K85A0422, OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "ALCOHOL DETECTION AND HEALTH MONITORING SYSTEM USING IOT" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email :contact@lasyainfotech.com,ov@lasyainfotech.com
Website :www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of *Bachelor of Technology* ***in Department of ELECTRONICS AND COMMUNICATION ENGINEERING, session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled "ALCOHOL DETECTION AND HEALTH MONITORING SYSTEM USING IOT" is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.***

G. SUSMITHA (17K81A04B3)

B. JYESHNA (18K85A0412)

P.KIRAN KUMAR(18K85A0422)

ACKNOWLEDGEMETS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research andproviding us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr.B.HARI KRISHNA**, **Head& Professor**, Dept. of Electronics and Communication Engineering for his advice, supervision and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in our memory for years to come.

We would like to express our gratitude to our **dept. project coordinator Dr.A.Anand, Professor** for his involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time to come.

We sincerely thank our project internal guide **Mrs.Ch.Swathi, Assistant Professor**, Department of ECE and our class project coordinator **Mr.K.Karthik, Assistant Professor** Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level Project Coordinator **Dr.Maalayadri,Professor in IT** and **Dr.R.Santhosh Kumar ,Associate Professor in CSE,** for his advice, supervision, and the vital contribution as and when required during this research work and for their support and cooperation that is difficult to express in words.

1. B.SUSMITHA
2. B.JYESHNA
3. P .KIRAN KUMAR

ABSTRACT

Factories, Offices, Hospitals, Military and other such industries need to monitor their staff/personnel follow all work ethics that include, not coming to premises under the influence of alcohol or under bad health conditions. This ensures proper work ethics are followed. So our proposed system allows for alcohol & health monitoring plus reporting system that monitors this and reports it to concerned personnel remotely over internet. Our system consists of an IOT based circuit system that uses a microcontroller based circuit system. The system has alcohol as well as blood pressure monitoring sensors to check for alcohol consumption as well as inappropriate blood pressure monitoring. This ensures no occurrences of accidents due to alcohol influence or bad health conditions

TABLEOFCONTENTS

LISTOFFIGURES

CHAPTER1

INTRODUCTION

1.1 OVERVIEWTOTHEPROJECT

Air and sound pollution is a growing issue these days. It is necessary to monitor air quality andkeep it under control for a better future and healthy living for all. Here we propose an air qualityas well as sound pollution monitoring system that allows us to monitor and check live air qualityas well as sound pollution in a particular areas through IOT. System uses air sensors to sensepresenceofharmfulgases/compoundsintheairandconstantlytransmitthisdatatomicrocontroller. Also system keeps measuring sound level and reports it to the online server overIOT. The sensors interact with microcontroller which processes this data and transmits it overinternet. This allows authorities to monitor air pollution in different areas and take action againstit.Alsoauthoritiescankeepawatchonthenoisepollutionnearschools,hospitalsandnohonkingareas, and if system detects air quality and noise issues it alerts authorities so they can takemeasuresto control the issue.

Ininfrastructureandindustrialplantstherapidgrowthcreatingenvironmentalissueslikepollution(Air, Water,Noise),climatechange,malfunctioningandhasgreatlyconsequencefortherequirement of an, operationally adaptable, efficient, cheap and smart monitoring systems. In thiscontextwherecombinationofmanychallengesofcomputerscience,wirelesscommunicationandelectronics; the Smart Sensor Networks are an emerging field of research. In this paper a solutionto monitor the air and noise pollution levels in industrial environment or by using wirelessembeddedcomputingsystemaparticularareaofinterestisproposed.ThetechnologylikeInternetof Things (IoT) is included in the form of solution which is outcome of merged field of computerscience and electronics. For monitoring the fluctuation of parameters like noise and air pollutionlevels from their normal levels in this case the sensing devices are connected to the embeddedcomputingsystem.Fortherequirementofcontinuousmonitoring,controllingandbehavioranalysisthismodelisadaptableanddistributiveforanyinfrastructuralenvironment.Fortwoor three parameters like noise, CO and radiation levels the implementation is tested with respect tothe normal behavior levels or given specifications which provide a monitoring over the pollutioncontrolto makethe environment smart and Eco-friendly.

1.2 MATERIALREQUIREMENT:

1.2.1 HARDWARE REQUIREMENT:

1. ArduinoUno
2. MQ3Alcohol sensor
3. Blood pressure sensor
4. LCDDisplay
5. ESP8266 Wi-Fi Module
6. Powersupply unit

1.2.2 SOFTWAREREQUIREMENT:

- ArduinoIDE
- Proteussoftware

1.3 PROCUMENTOFEQUIPMENT:

We brought all the required hardware components in a electronic manufacturing shop and beforeassemblingandinterfacinghardwarecomponentstoArduinoUNOmicro-controller,weconnected individual sensors to Arduino in Proteus Software and understood working of everyindividual sensor in Proteus Software. Now, we assembled all sensors and interfaced to ArduinoUNOmicro-controllerusingconnectingwiresthroughbreadboardandwewroteprogramtoworkall the hardware sensors in Arduino IDE, we understood the working of all sensors in a practicalway. We detected the alcohol level and monitored the blood pressure of the person and all theinformationis sent to theonlineserver.

1.4 INTRODUCTIONTO EMBEDDEDSYSTEM

An embedded system is a combination of computer hardware and software, either fixed incapability or programmable, designed for a specific function or functions within a larger system.Industrial machines, agricultural and process industry devices, automobiles, medical equipment,cameras, household appliances, airplanes, vending machines and toys, as well as mobile devices,are 2 possible locations for an embedded system. An embedded system is a dedicated computersystemdesignedforoneortwospecificfunctions.Thissystemisembeddedasapartofacomplet edevice system that includes, such as electrical and mechanical components [1]. The embeddedsystem is unlike the general-purpose computer, which is engineered to manage a wide range ofprocessingtasks.Becauseanembeddedsystemisengineeredtoperformcertaintasksonly,designengi neers may optimize size, cost, power consumption, reliability and performance. Embeddedsystems are typically produced on broad scales and share across a variety of environments andapplications.Anembeddedsystemisonethathas computer-hardwarewith softwareembeddedinit as one of its most important components. It is a dedicated computer-based system for anapplication or product. An embedded system has three main components. 1. It has hardware 2. Ithasmainapplicationsoftware.3.Ithasarealtimeoperatingsystem(RTOS)Anembeddedsystemhas software designed to keep in view three constraints: (i) Available system memory. (ii)Availableprocessorspeed.(iii)Theneedtolimitpowerdissipationwhenrunningthesystem.Anembe dded system is a system which is going to do a predefined specified task is the embeddedsystem and is even defined as combination of both software and hardware. A general-purposedefinition of embedded systems is that they are devices used to control, monitor or assist theoperationofequipment,machineryorplant."Embedded"reflectsthefactthattheyareanintegralpart of the system. At the other extreme a general-purpose computer may be used to control theoperation of a large complex processing plant, and its presence will be obvious. Some of thesecomputers are however very simple systems as compared with a personal computer [2]. The verysimplestembeddedsystemsarecapableofperformingonlyasinglefunctionorsetoffunctionstomee t a single predetermined purpose. In more complex systems an application program thatenablestheembeddedsystemtobeusedforaparticularpurposeinaspecificapplicationdetermines the functioning of the embedded system. The ability to have programs means that thesameembeddedsystemcanbeusedfora3varietyofdifferentpurposes[2].Insomecases,a microprocessor may be designed in such a way that application software for a particular purposecanbeaddedtothebasicsoftwareinasecondprocess,afterwhichitisnotpossibletomakefurtherc hanges. The applications software on such processors is sometimes referred to as firmware. Thesimplest devices consist of a single microprocessor (often called a "chip"), which may itself bepackaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC).Its input comes from a detector or sensor and its output goes to a switch or activator which (forexample) may start or stop the operation of a machine or, by operating a valve, may control theflow of fuel to an engine[3]. As the embedded system is the combination of both software andhardware.

1.4.1 HISTORYOFEMBEDDEDSYSTEM

Embedded systems date back to the 1960s. Charles Stark Draper developed an integrated circuit(IC) in 1961 to reduce the size and weight of the Apollo Guidance Computer, the digital systeminstalled on the Apollo Command Module and Lunar Module. The first computer to use ICs, ithelped astronauts collect realtime flight data. In 1965, Auto now a part of Boeing, developed theD-17B,the computer used in theMinuteman Imissileguidancesystem.

1.4.2 BLOCKDIAGRAMOF EMBEDDEDSYSTEM

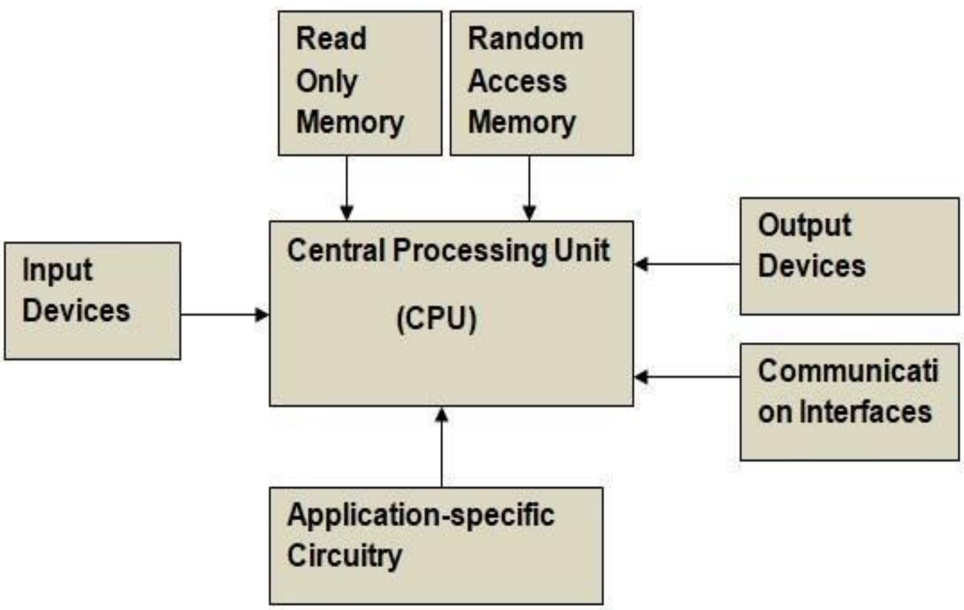


Fig.1.2.1:Blockdiagram ofembeddedsystem

SoftwaredealswiththelanguageslikeALP,C,andVBetc.,andHardwaredealswithProcessors,Peripherals,and Memory.

1.4.3 EMBEDDEDSYSTEMHARDWARE:

Embedded system hardware can be microprocessor- or microcontroller-based. In either case, anintegrated circuit is at the heart of the product that is generally designed to carry out computationforreal-timeoperations[4].Microprocessorsarevisuallyindistinguishablefrommicrocontrollers,butwhilethe microprocessoronlyimplementsacentralprocessingunit(CPU)and,thus,requiresthe addition of other components such as memory chips, microcontrollers are designed as self-containedsystems.

1.4.4 EMBEDDEDSYSTEMSOFTWARE:

A typical industrial microcontroller is unsophisticated compared to the typical enterprise desktopcomputer and generally depends on a simpler, less-memory intensive program environment. Thesimplest devices run on bare metal and are programmed directly using the chip CPU's machinecode language. Often, embedded systems use operating systems or language platforms tailored toembedded use, particularly where real-time operating environments must be served. At higherlevels of chip capability, such as those found in SoCs designers have increasingly decided thesystems are generally fast enough and the tasks tolerant of slight variations in reaction time thatnear-real-timeapproaches aresuitable.

1.4.5 EXAMPLESOFEMBEDDEDSYSTEM

Someexamplesofembeddedsystemsarebelow

- ATM
- DigitalCameras
- Microwaveovens
- Factorycontrollers
- Washingmachine5

- Calculator

• TVremote

• Trafficlights

• Digitalwatches

• Mp3player

• Videogamesconsoles

• Printers

• GPSreceivers

• Dishwashers

• Thermostats

• Anti-lockbankingsystem

• Medicalimaging

1.4.6 TYPESOFEMBEDDEDSYSTEM

1. RealTimeSystems

Real-time systems are those which give a quick response to critical situations. They are used inmilitary, medical and industrial applications. Engineers working in these systems have highdemand is current days. To develop the real-time embedded system we require timing analysis,multitasking design, debugging, cross-platform testing and architecture design. In these systems,quickresponse is very important.

2. StandaloneEmbeddedSystem

This type of embedded system works for itself as a device without needing any interconnectedcomputer. It can take data in the form of Analog or digital signals. This system first process dataandthen outputs data bydisplaying on thescreen.

3. NetworkedEmbeddedSystem

Networked embedded systems are those systems which are connected to the network to giveoutputtotheattachedresources[5].Thedevicesinthenetworkedembeddedsystemareconnectedto the network with network interfaces. The network can be either a local area network (LAN) orawideareanetwork(WAN).

4. MobileEmbeddedSystem

Mobile embedded systems are limited in resources including memory. Examples of mobileembeddedsystems include

- Personaldigitalassistants(PDA)
- Cellularphones
- Mp3players
- Digitalcameras

5. Small-ScaleEmbeddedSystem

Small-scaleembeddedsystemsconsistof8-16bitmicrocontroller.Thissystemcanperformtasksatasmalllevel.Theyhaveon-chipROMandRAM.Smallscalesystemscanbeevenactivatedbythebattery.Thetoolsusedtodevelopsmall-scaleembeddedsystemsareaneditor,crossassembler,assembler and integrated development environment (IDE). The purpose of this system is notcomputation but to control as a computer embedded inside it. It behaves as a component of acomputer and its function is not to compute. The small-scale system is dedicated to some specifictask.

6. MediumScaleEmbeddedSystem

This embedded system has 16-32 bit microprocessor or microcontroller with external RAM andROM They can perform medium to complex level works. The integration between hardware andsoftwareiscomplexintheseembeddedsystems.Programminglanguagesusedtodevelopmediumscale embedded systems include Java, C, Visual C++, debugger, C++, RTOS, simulator, sourcecodeengineering tool and IDE.

7. SophisticatedEmbeddedSystem

The embedded system which can do large-scale works with multiple 32-64 bit chips is known assophisticated embedded systems[6]. They can perform distributed work on a large scale. Thecomplexity of hardware and software is very high in these systems. In sophisticated embeddedsystems, hardware and software are assembled together on large scale and designing of hardwareproductsis also includedin thesesystems.

1.4.7 APPLICATIONSOFEMBEDDEDSYSTEM

- Manufacturingandprocesscontrol
- Constructionindustry
- Transport
- Buildingsandpremises
- Domesticservice
- Communications
- Officesystemsandmobileequipment
- Banking,financeandcommercial
- Medicaldiagnostics,monitoringandlifesupport
- Testing,monitoringanddiagnosticsystems

1.5 MICROCONTROLLER

A microcontroller is a small computer on a single metal oxide semiconductor (MOS) integratedcircuit(IC)chip.AmicrocontrollercontainsoneormoreCPUsalongwithmemoryandprogrammableinput/outputperipherals.ProgrammemoryintheformofferroelectricRAM,NORflashorOTP ROMisalsooftenincludedonchip,aswellasasmallamountofRAM.Microcontrollers are designed for embedded applications, in contrast to the microprocessors usedinpersonalcomputersorothergeneralpurposeapplicationsconsistingofvariousdiscretechips[7].InmodernterminologyamicrocontrollerIs the similar placeholder chip (SoC). SoC may include a microcontroller as one of its components, but usually integrates itwith advanced peripherals like graphics processing unit (GPU), Wi-Fi module, or one or morecoprocessors.

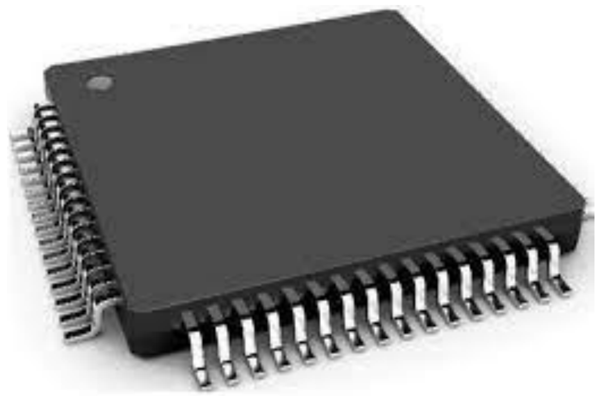


Fig.1.3.1:Microcontroller

Microcontrollers are used in automatically controlled products and devices, such as automobileenginecontrolsystems,implantablemedicaldevices,remotecontrols,officemachines,appliances,powertools,toysandotherembeddedsystems.Byreducingthesizeandcostcomparedtoadesignthatusesaseparatemicroprocessor,memory,andinput/outputdevices,microcontrollers make it economical to digitally control even more devices and processes. Mixedsignalmicrocontrollersarecommon,integratinganalogcomponentsneededtocontrolnon-digitalelectronicsystems.Inthecontextoftheinternetofthings,microcontrollersareaneconomicalandpopularmeansofdatacollection,sensingandactuatingthephysicalworldasedgedevices.Somemicrocontrollersmayusefour-bitwordsandoperateatfrequenciesaslowas4kHzforlowpowerconsumption (single-digit milliwatts or microwatts). They generally have the ability to retainfunctionalitywhilewaitingforaneventsuchasabuttonpressorotherinterrupt;powerconsumptionwhilesleeping(CPUclockandmostperipheralsoff)maybejustnanowatts,makingmanyofthemwellsuitedforlonglastingbatteryapplications[8].Othermicrocontrollersmayserveperformance-criticalroles,wheretheymayneedtoactmorelikeadigitalsignalprocessor(DSP),withhigher clock speeds and powerconsumption.

1.5.1 HISTORY

Theoriginsofboththemicroprocessorandthemicrocontrollercanbetracedbacktotheinventionof the MOSFET (metal-oxide-semiconductor field-effect transistor), also known as the MOStransistor. It was invented by Mohamed M. Atalla and DawonKahng at Bell Labs in 1959, andfirst demonstrated in 1960.9 The same year, Atalla proposed the concept of the MOS integratedcircuit,whichwasanintegratedcircuitchipfabricatedfromMOSFETs.By1964,MOSchipshadreachedhighertransistordensityandlowermanufacturingcoststhanbipolarchips[2].MOSchipsfurtherincreasedincomplexityataratepredictedbyMoore'slaw,leadingtolarge-scaleintegration (LSI) with hundreds of transistors on a single MOS chip by the late 1960s. Theapplication of MOS LSI chips to computing was the basis for the first microprocessors, asengineers began recognizing that a complete computer processor could be contained on a singleMOS LSIchip.

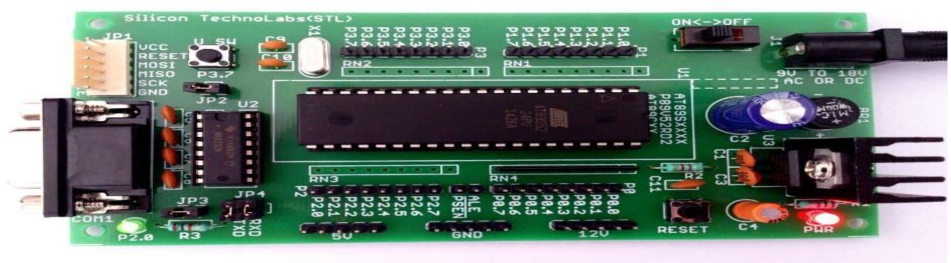


Fig.1.3.2:8051MicrocontrollerBoard

Most microcontrollers at this time had concurrent variants. One had EPROM program memory,with a transparent quartz window in the lid of the package to allow it to be erased by exposure toultravioletlight.Theseerasablechipswereoftenusedforprototyping.Theothervariantwaseitheramaskprogrammed ROMoraPROMvariantwhichwasonlyprogrammableonce.Forthelatter,sometimes the designation OTP was used, standing for "one-time programmable". In an OTPmicrocontroller,thePROMwasusuallyofidenticaltypeastheEPROM,butthechippackagehadno quartz window; because there was no way to expose the EPROM to ultraviolet light, it couldnotbeerased.Becausetheerasableversionsrequiredceramicpackageswithquartzwindows,they were significantly more expensive than the OTP versions, which could be made in lower-costopaqueplasticpackages.Fortheerasablevariants,quartzwasrequired,insteadoflessexpensive glass,foritstransparencytoultravioletlight—towhichglassislargelyopaque—butthemaincostdifferentiatorwas theceramicpackageitself.

In 1993, the introduction of EEPROM memory allowed microcontrollers (beginning with theMicrochip PIC16C84) to be electrically erased quickly without an expensive package as requiredforEPROM,allowingbothrapidprototyping,andin-systemprogramming.(EEPROMtechnologyhad been available prior to this time, but the earlier EEPROM was more expensive and lessdurable,makingitunsuitableforlow-costmass-producedmicrocontrollers.)Thesameyear,Atmelintroduced the first microcontroller using Flash memory, a special type of EEPROM. Othercompanies rapidly followed suit, with both memory types. Nowadays microcontrollers are cheapandreadilyavailable for hobbyists, withlargeonlinecommunitiesaroundcertainprocessors.

1.6 INTRODUCTIONTOINTERNETOFTHINGS

The Internet of things (IoT) describes the network of physical objects—a.k.a. "things"—that areembedded with sensors, software, and other technologies for the purpose of connecting andexchanging data with other devices and systems over the Internet.Things have evolved due to theconvergenceofmultipletechnologies,real-timeanalytics,machinelearning,ubiquitouscomputing,commoditysensors,andembeddedsystems.[1]Traditionalfieldsofembeddedsystems, wireless sensor networks, control systems, automation (including home and buildingautomation), and others all contribute to enabling the Internet of things. In the consumer market,IoT technology is most synonymous with products pertaining to the concept of the "smart home",including devices and appliances (such as lighting fixtures, thermostats, home security systemsand cameras, and other home appliances) that support one or more common ecosystems, and canbecontrolledviadevicesassociatedwiththatecosystem,suchassmartphonesandsmartspeakers.TheIoT can also be usedin healthcaresystems.

Internet of Things (IOT) is a network of devices that connect directly with each other to captureandsharedatathroughasecureservicelayerthatconnectstoacentralcommandandcontrolserverin the cloud. The closure\look suggest that the way people collect, record and analyze data—notjust in health care but in every industry today. The idea of devices connecting directly with eachotheris basically calledInternet of Things.

The Internet of Thingsalso a called the Internet of Objects, refers to a wireless network betweenobjects,usuallythenetworkwillbewirelessandself-configuring.InternetofThings(IOT)isoneof the major component advances in present time that links the internet with everyday sensor andworking devices. Smart objects play an important role in the Internet of Things vision, sinceembedded communication and information technology would have the potential to change theutility of these objects. Using sensors they are able to recognize their condition, and via built-inconnectingpower they would be ableto interact with each other.

1.6.1 BenefitsofIOT

Ubiquitousnetworks -:PersonalWi-Fionyour smartphonesandonmanyoftheotherdevices.Everyone(and everything) wants as well asneedsto be connected.

Connectedcomputing-:Wewantallofthedevices,smartphones,televisions(coloredorblackandwhite), dvd players, vehicles etc. to keep record of what we are doing, seeing, reading, and/orlistening to as we sway through the day, from one place to another – the handoffs from device todeviceis happeningalready.

Analytics-as-a-Service-: The API and App economies are already wide and growing whichenables to “do something interesting” as long as it can be connected to an API or can invoke anAppthatcarriesoutanetwork-basedservice.Thethingisadatageneratoraswellascollectorthatlearnsfrom,makesforecast,andmaybeeventakesdata-drivenactionsinresponsetothedatathatarecollected too.

Marketingautomation-: Smartphonecustomerengagement,geological-location,Apple’siBeacon etc. are all developing a network of knowledge and information regarding customers’locations, intentions, preferences, as well as buying patterns. Obviously, the degree of geologicallocation-basedknowledgeneedstomaintaintherightbalancebetweenuserprivacyaswellasthetimelydelivery ofimportantand significant products andservices to theparticularuser.

Supply Chain Analytics-: Delivering the just-in-time products at the time of need (inclusive ofthe use of RFID-based tracking). Significantly, everything is a customer (inclusive of machines,automobiles, manufacturing plants, ATM machines, etc.), as well as the IOT is monitoring,watching,as well as waiting forthe product needsto arise.

1.7 MOTIVATIONOFTHEPROJECT

In this era of modernization, technologies are advancing rapidly. Every day we realize some newtechnology coming in market to simplify our lives more than ever. Here this system is created tomakeonemoresysteminthefulfillmentofthegoalofsmartcity.Herethissystemusedto detect alcohol andmonitorthe blood pressure of the personandhelpsintakingmeasurestocontrolstheaccidents occurred to a person due to alcohol consumption and bad health condition.

CHAPTER2

LITERATURESURVEYANDEXISTINGMODELS

2.1 LITERATUREREVIEWONRESEARCHAREA:

1. Internet of Things (IoT) and cloud computing plays a vital role in today's Tele-monitoring health system. This system keeps track of patient's physiological parameters through collection of body sensors' data using Arduino board. The patient's health card are developed by the doctors and displayed on a webpage where doctors and patients can access and communicate each other without physical presence. Using cloud computing, the data can be stored, updated and accessed from anywhere in the world. It is very suitable for rural areas where medical facilities are not available.

2. Owing to costlier healthcare and long waiting time in hospitals, the concept of in-home patient monitoring system have been emerging in the recent years. This system collects data of various body parameters through biosensors, wearable devices and smart textiles and it transmits the data to central node server securely through Cipher text Policy Attribute Based Encryption (CP-ABE) method. In turn, the server shares the collected data to the hospitals for further treatment. It is very beneficial for elders and chronic patients who require continuous monitoring.

3. The specialized healthcare monitoring system for elderly people is a growing need in the aging population world. This system performs the alcohol detection and monitoring blood pressure by measuring the body parameters regularly and report the data to the higher officials. The result data are then displayed as statements in a web application where the persons can interact with each other. The main challenge is to make elders equipped with for growing new technology and become familiar towards smartphone and computers.

4. IoT based Smart healthcare with the help of smart devices and objects improves the healthcare monitoring system effectively, thus by reducing the inefficiencies of existing healthcare system. Smart devices with new and upgraded technologies enhances the data

accuracy to be collected, real-time accessibility of person's condition, intelligent integration of data collected, maintaining the integrated data smartly through cloud service, etc. IoT along with smart devices reduce complexity and complications in the healthcare system. The penetration of mobile technologies and smart devices over healthcare system cause huge impact on the world. The full-fledge utilization of M-health and E-health applications in today's world is made aware to the people for improving and maintaining the good quality of life. Apart from regular monitoring of persons condition through M-health system, the main objective is to educate them through recommendations of healthy eating habits and effective workout routines for improving their quality of healthy life.

5.In remote mobile health monitoring system, the person health parameters are recorded by a smart phone by eliminating an additional hardware and transmit data through a web interface. It facilitates end to end monitoring screen through three steps. Firstly, the real time health parameters are measured through wearable sensors and transmitted to a smart phone which shows the patient health status in graphical interface. Secondly, this system provides a data to family member and doctor through web interface for further monitoring.

2.2 REVIEWOFRELATEDLITERATURE:

The purpose of this project is to identify the alcohol consumption and health monitoring of a person by using alcohol sensor and blood pressure sensor and by observing the output through wifi module over internet, we as a society have to ensure that every person is maintaining minimum etiquette,Intoday'sworld,healthcareisthebiggestconcern,healthcare has a lot of effects on the wellbeing of the people. Our project is a small step to this biginitiative to ensure that accidents related to health stays under control. This project helps in detecting the alcohol consumption and blood pressure of a person. Our project will be a boon tothe society as our project will be making sure that every individual will be able to keep a track ofthe health from the website . It is the need of the hour to monitor health and keep it undercontrolforabetter futureand healthy living forall.

Themotiveofmakingasmartcitycanbefulfilledbyusingtechnology,thusmakingthelifebetterand also enhancing the quality of services, therefore meeting every individual's needs. Withmodern technology in fields of information and communication, it has become easy to interactwith the authorized people of city to tell the where abouts of the area or city, how well the city isdeveloping and how to make it possible to achieve a better life quality. In this system, anapplication was created to make one more step in the fulfillment of the goal. The components of alcohol and blood pressuretheiramountsarecalculatedandchecked.Iftheamountishigherthannormalthenthe officials are reported about it. After that the people are made to clear the area and taken to a safeplace.

2.3 CONCLUSIONONREVIEWS:

This projectfocuses on a real-time alcohol detection and healthcare monitoring system using IoT and cloud computing service which are more beneficial for elders and chronic diseases' patients. The current methods available for realization of Healthcare services are surveyed and the challenges that are part of realization are also highlighted. This project proposed an intelligent real-time patient health monitoring system that monitors the subject's vital parameters such as pressure, alcohol detection. Appropriate medications are suggested based on the diagnosis of the provided set of symptoms. The system sends an alert message to the caretakers in case of any abnormality through cloud. The system enables the caretaker to optimize the usage of available medical resources and minimize the costs in monitoring the persons. In the future, we will focus on improving wearing sensor experience by using softer materials and enabling controlled sharing of information among the doctors, the patient, and the patients' family through social networking paradigm.

CHAPTER 3PROJECTDESCRIPTION

3.1 BLOCKDIAGRAM:

The system is used for detecting alcohol consumtion and blood pressure.The system consistsof alcohol sensor to measure the concentration of alcohol and blood pressure sensorfor measuringbloo.These sensed information is transmitted to the Arduino and itprocess the information and continuously displays these values on the lcd display which isconnected to it, simultaneously this information is transmitted to the online server by the wi-fimodule.Byusingthewebsitethingspeakwecanmonitorthelevelofairandnoisepollutionintheenvironment. This allows authorities to monitor air pollution in different areas and take actionagainstit.

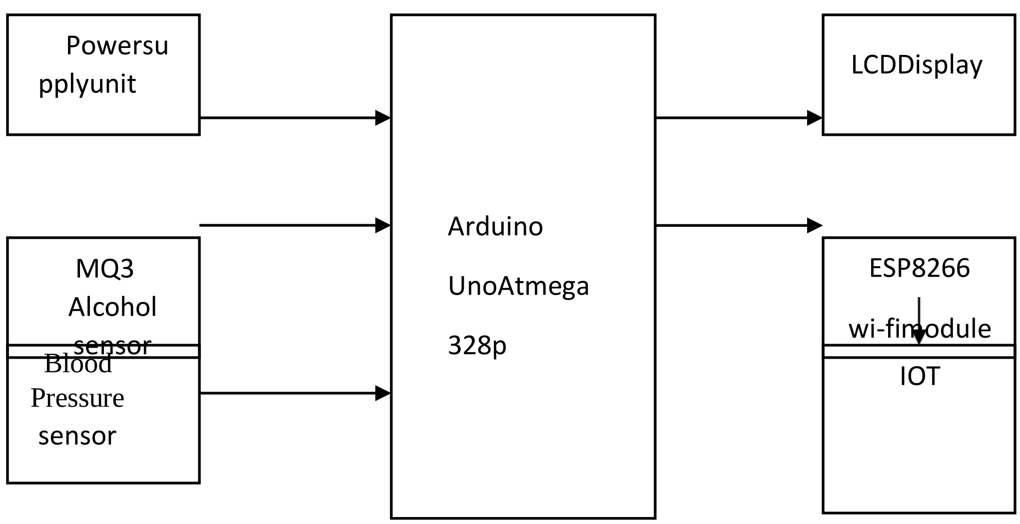


Fig.3.1.1:BlockDiagram

Theblocksofthecircuitare:1.A

rduinouno

2.MQ3alcoholsensor

3.Blood pressure sensor

4.LCDDisplay

5.ESP8266 Wi-Fi Module

6.Power supply

unit7.Onlineserver

1. ArduinoUNO

Arduinoisopensourcesoftware.ArduinoUnoisamicrocontroller,ATmega328Pcontaininit.Ithas14digitali/opinsand6analoginputpin,a16MHzquartzcrystal,aUSBconnection,apowerjackandaresetbutton.SimplyconnecttoacomputerwithaUSBcableorpoweritwithaAC-to-DCadapterorbatterytoget started.Embedded Cprogramis used for coding.

2. MQ3alcohol sensor

3. Blood pressuresensor

4. LCD

TheLCDisusedtodisplaywheneverthereisagasleakageorfiredetectedinthearea.Wheneverthere is a gas leakage the Arduino that is connected to the sensor and LCD sends a message toLCD to display ‘Gas leakage’. When there is fire outbreak the Arduino again sends a message toLCDtodisplay‘FireDetected’notifyingusifthereisagasleakageorfiredetectedinthevicinity.

5.ESP8266Wi-FiModule

The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack thatcan give any microcontroller access to your WiFi network. The ESP8266 is capable of eitherhosting an application or offloading all WiFi networking functions from another applicationprocessor. Each ESP8266 module comes pre-programmed with an AT command set firmware,meaning,youcansimplyhookthisuptoyourArduinodeviceandgetaboutasmuchWiFi-abilityas a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely costeffectiveboardwith a huge, and ever growing,community.

6. PowerSupplyUnit

Every electrical and electronic device that we use in our day-to-day life will require a powersupply.Ingeneral, weusean ACsupplyof 230V50Hz, butthispower hastobechangedintotherequired form with required values or voltage range for providing power supply to different typesof devices. There are various types of power electronic converters such as step-down converter,step-up converter, voltage 31 stabilizer, AC to DC converter, DC 25 to DC converter, DC to ACconverter, and so on. For example, consider the microcontrollers that are used frequently fordeveloping many embedded systems based projects and kits used in real-time applications. Thesemicrocontrollersrequirea5V DC supply.

3.2 SchematicDiagram

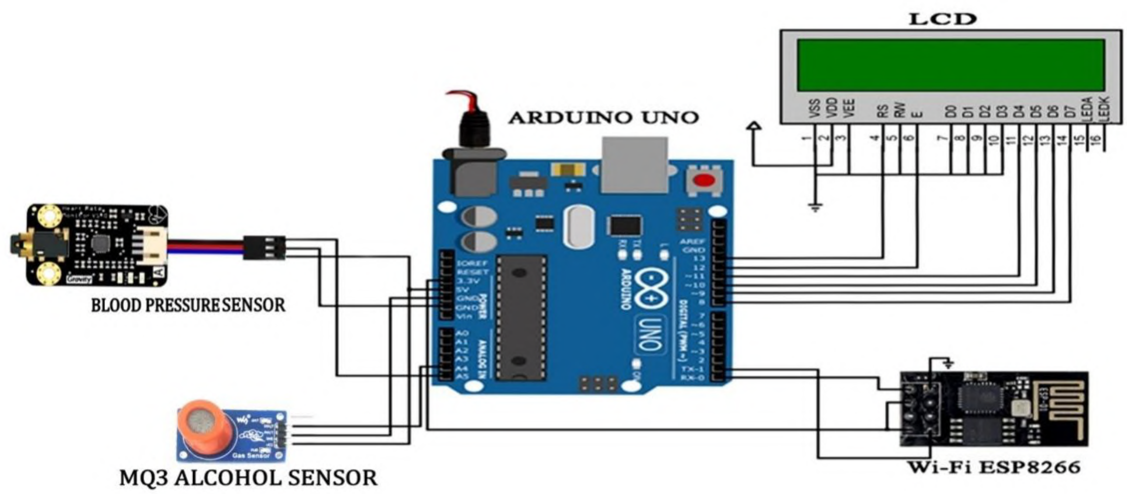


Fig.3.2.1:SchematicDiagram

3.2.1 SchematicExplanation

Arduino UNO has 6 analog pins from A0 to A6 which are generally input pins. It also has 14digitalpinsfrom0-13whicharetheoutputpins.Theanalog pinA4isconnectedtotheoutputpinof the mq2 gas sensor and analog pin A5 is connected to the output pin of the sound sensor. Thedigitalpins8-13is connectedtothelcdtodisplaytheoutputvalues.Herepin0isthereceiverpinand pin 1 is the transmitter pin. These pins are used for the esp8266 wi-fi module to transmit theinformationto the online server.

CHAPTER4

HARDWAREANDSOFTWAREDESCRIPTION

4.1 HARDWARECOMPONENTS

The hardware components used for the implementation of ALCOHOL DETECTION AND HEALTH MONITORING SYSTEM USINGIOTare

1. ArduinoUno
2. MQ3alcoholsensor
3. Blood pressuresensor
4. LCD
5. ESP8266Wi-Fimodule
6. Powersupply

4.1.1 ARDUINOUNO

Arduino is an open-source electronics platform based on easy-to-use hardware and software.Arduinoboardsareabletoreadinputs-lightonasensor,afingeronabutton,oraTwittermessageand turn it into an output activating a motor, turning on an led, publishing as it reached a widercommunity, the Arduino board started something online. You can tell your board what to do bysending a set of instructions to the microcontroller on the board. To do so you use the Arduinoprogramminglanguage(basedWiring),theArduinoSoftware(IDE),basedonProcessing.ArduinowasbornattheareaInteractionDesignInstituteasaneasytoolforfastprototyping,aimedat students without a background in electronics and programming wearable, 3dprinting, andembedded environments. All Arduino boards are completely open-source, empowering fromeverydayobjectstocomplexscientificinstruments.Professionalgatheredaroundopen-sourceplatform, the software too is opensource, it is growing through the contributions of users worldwide,theArduino board is shown in below figure.

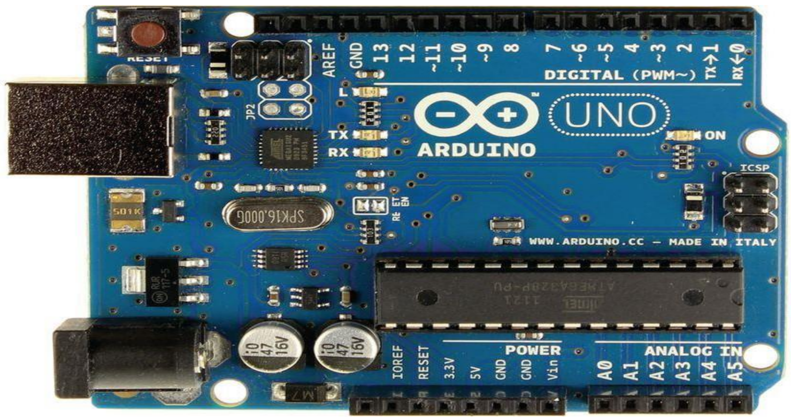


Fig.4.1.1:Arduinounoboard

4.1.1.1 WhyArduino?

Arduino Uno is inexpensive. Arduino is cross platform, its IDE can run on multiple OperatingSystemssuchasWindows,Macintosh,Linuxandmanymore.Ithassimpleandclearprogrammingenvironment.Itssoftwareandhardwareareopen-sourceandextensible,thatis,Arduinoprogramming can be expanded through C++ libraries and can also leap to AVR ‘C’ languageprogramming. It is simple and accessible to user experience, Arduino has been used in thousandsof different projects and applications. The Arduino software is easy-to-use for beginners, yetflexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and studentsuse it to build low cost scientific instruments, to prove the chemistry and physics principles or togetstartedwithprogrammingandrobotics.Designersandthearchitectsbuildinteractiveprototypes,musicians,artistsuseitforinstallationsandtoexperimentwithnewmusicalinstruments. Makers of course use it to build many of the projects exhibited at the Maker Faire,for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists,programmers can start tinkering just following the step by step I constructions of a kit or sharingideasonlinewithothermembersoftheArduinocommunity.Therearemanyothermicrocontrollers platforms available for physical computing. Parallax Basic Stamp, Netmedia'sBX-24, Phidgets, MIT's Handy board, and many others offer similar functionality. All of thesetools take the messy details of microcontroller programming and wrap it up in an easy-tousepackage. Arduino also simplifies the process of working with microcontrollers, but it offers someadvantageforteachers,students,andinterestedamateursoverothersystems.•Inexpensive

Arduino boards are relatively inexpensive compared to other microcontroller platforms. The leastexpensive version of the Arduino module can be assembled by hand, and even the pre-assembledArduino modules cost less than $50 • Cross-platform The Arduino Software (IDE) runs onWindows, Macintosh OSX, and Linux operating systems. Most microcontroller systems arelimited to Windows. • Simple, clear programming environment The Arduino Software (IDE) iseasy-to-useforbeginners,yetflexibleenoughforadvanceduserstotakeadvantageofaswell.Forteachers,convenientlybasedontheProcessingprogrammingenvironment,sostudentslearningtoprograminthatenvironmentwillbefamiliarwithhowtheArduinoIDEworks.•OpensourceandextensiblesoftwareTheArduinosoftwareispublishedasopensourcetools,availableforextensionby experienced programmers. The language can be expanded through C++ libraries, and peoplewanting to understand the technical details can make the leap from Arduino to the AVR Cprogramminglanguageonwhichit'sbased.Similarly,youcanaddAVR-CcodedirectlyintoyourArduinoprogramsifyouwantto.•OpensourceandextensiblehardwareTheplansoftheArduinoboards are published under a Creative Commons license, so experienced circuit designers canmaketheirownversionofthemodule,extendingitandimprovingit.Evenrelativelyinexperienceduserscanbuildthebreadboardversionofthemoduleinordertounderstandhowitworksandsavemoneyand the parts ofArduino board is shown below

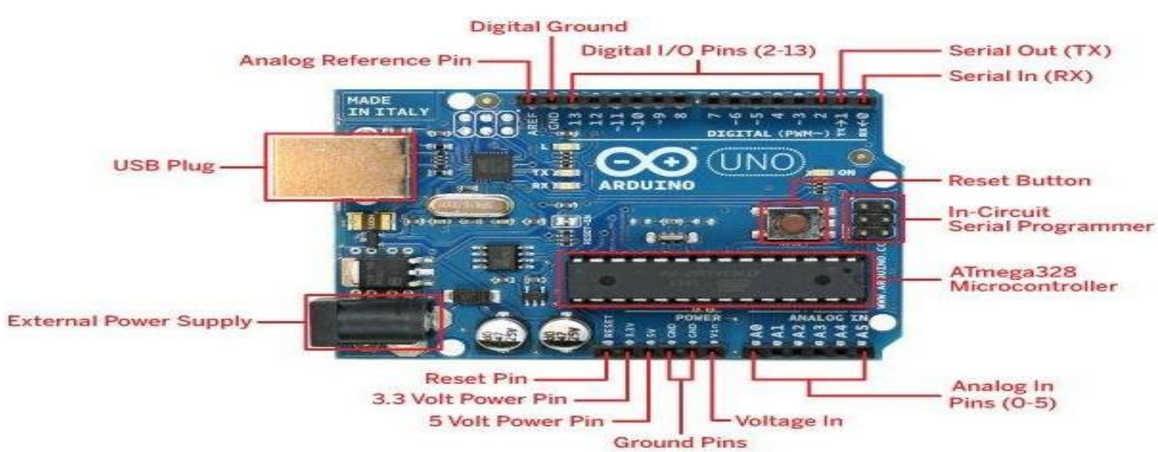


Fig.4.1.2:Arduinoboard description

4.1.1.2 Featuresof Arduino

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage	7-9V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) (0.5 KB used by bootloader)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Table4.1.1:Featuresof Arduinouno

4.1.1.3 Programming

TheArduinoUnocanbeprogrammedwiththeArduinoSoftware(IDE).SelectArduinoUnofromtheToolsBoardAfteryourcodeiscompiledusingArduinoIDE,itshouldbeuploadedtothemainmicrocontroller of the Arduino UNO using a USB connection. Because the main microcontrollerdoesn't have a USB transceiver, you need a bridge to convert signals between the serial interface(UART interface) of the microcontroller and the host USB signals. The ATmega328 on theArduino Uno comes pre-programmed with a boot loader that allows you to upload new code to itwithouttheuseofanexternalhardwareprogrammer.ItcommunicatesusingtheoriginalSTK500protocol. You can also bypass the bootloader and program the microcontroller through the ICSP(In Circuit Serial Programming) header using Arduino are similar. The ATmega16U2 (or 8U2 intherev1&rev2boards)firmwaresourcecodeisavailableintheArduinorepository.TheATmega16U2 or8U2isloadedwithaDFUbootloader,whichcanbeactivatedbyOnRev1boards connecting the solder jumper on the back of the board and then resetting the 8U2. You can thenuse Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load anewfirmware.OryoucanusetheISPheaderwith anexternalprogrammer(overwritingtheDFUbootloader). Arduino is an opensource hardware platform that is being used by people around theglobeforbuildingelectronicsprojects.Itisanintegratedplatformwhichcontainsboththephysicaland programmable circuit otherwise known as microcontroller and a software (or IDE) that youcan run on your computer to write and upload the code onto the physical board. Arduino Board isquite popular among many people who want to get started with electronics, and unlike otherembedded system boards Arduino does not require any additional hardware to upload the code(generally known as programmer). The Arduino Program can be written and uploaded using theArduinoIDEthatneedsjustanUSBcabletoconnect.Sincetheinterfaceissimpleandcomplicacies are less Arduino is preferred by most of the aspiring engineers. Here we will try tounderstand about the Arduino Architecture and its functionalities. The processor of the ArduinoBoard uses Harvard Architecture for which the program code and program data have separatememory. The memory of it is divided into two namely program memory and data memory. Thedata will be stored in the data memory whereas the program code will be stored in the flashprogram memory. For ex: The Atmega328 microcontroller has 32kb of flash memory, 2kb ofSRAM,1kbofEPROMandoperatesat16MHzclockspeed.SomeoftheotherbasicfunctionsofArduin oare:

• Digitalwritepinisusedtowritethedigitalvalueof thegivenpin.

• Pinmodepinisusedtosetthepinto I/Omode.

• Analogreadpinreads andreturnsthevalue.

• Analogwritepinwritesthevalueofthepin

• Serial.Beginspinsetsthebeginningofserialcommunicationbysettingtherate ofbit

• Digitalreadpinreadsthedigitalvalueofgiven pin.

4.1.1.4 Advantages

Inexpensive-
Arduinoboardsarerelativelyinexpensivecomparedtoothermicrocontrollerplatforms.Theleastexpen siveversionoftheArduinomodulecanbeassembledbyhand,andeventhepre-assembled Arduino modules cost less than $50.

1. Cross-platform - The Arduino software runs on Windows, Macintosh OSX, and the Linuxoperatingsystems. Most microcontroller systemsarelimited to Windows.

2. Simpleprogrammingenvironment-HeretheArduinoprogrammingenvironmentiseasy-to-usefor beginners, yet flexible enough for advanced users to take advantage of as well. For teachers,it's conveniently based on the Processing programming environment, so students learning toprogramin that environmentwill befamiliar withthe look and feelof Arduino.

3. Opensourceandextensiblesoftware-
TheArduinosoftwareispublishedasopensourcetools,availableforextensionbyexperiencedprogram mers.ThelanguagecanbeexpandedthroughC++libraries, and people wanting to understand the technical details can make the leap from Arduinoto the AVR C programming language on which it's based. Similarly, you can add AVR-C codedirectlyinto yourArduino programs if youwant to

4. Open source and extensible hardware - The Arduino is based on Atmel's ATmega8 andATmega168microcontrollers.TheplansforthemodulesarepublishedunderaCreativeCommonsli cense, so experienced circuit designers can make their own version of the module, extending itand improving it. Even 19 relatively inexperienced users that can build the breadboard version ofthemodule in order to understand how it works and savemoney.

4.1.1.5 Applications

• ItisaneasyUSBinterface.Thisallowsinterface withUSBasthisislikeaserialdevice.

• The chip on the board plugs straight into your USB port and supports on your computer as avirtual serial port. The benefit of this setup is that serial communication is an extremely easyprotocol which is time-tested and USB makes connection with modern computers and makes itcomfortable.

• It is easy-to-find the microcontroller brain which is the ATmega328 chip. It has more numberof hardware features like timers, external and internal interrupts, PWM pins and multiple sleepmodes.

• It is an open source design and there is an advantage of being open source is that it has a largecommunityofpeopleusingandtroubleshootingit.Thismakeseasytohelpindebuggingprojects

. • It is 16MHz clock which is fast enough for most applications and does not speeds up themicrocontroller.

• Itisveryconvenienttomanagepowerinsideitandithadafeatureofbuilt-involtageregulation.This also be powered directly off a USB port without any external power. You can connect anexternalpower sourceofup to 12vand this regulates it to both5v and 3.3v.

• 13 digital pins and 6 analog pins. This sort of pins allows you to connect hardware to yourArduinoUnoboardexternally.Thesepinsareusedasakeyforextendingthecomputingcapabilityof the Arduino Uno into the real world. This has an ICSP connector for bypassing the USB portand interfacing the Arduino directly as a serial device. This port is necessary to re-boot load yourchipif it corrupts and can no longerused to yourcomputer.

• Ithas a32KBofflashmemoryforstoringyour code.

• Anon-boardledisattachedtodigitalpin13tomakefastthedebuggingofcodeandtomakethedebugprocesseasy finally it has the button toreset.

4.1.1.6 Power

TheArduinoUnocanbepoweredviathroughUSBconnectionorwithanexternalpowersupply.The power source is selected automatically. External (non-USB) power can come either from anAC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mmcentre-positive plug into the board's power jack. Leads from a battery can be inserted in the Gndand Vin pin headers of the POWER connector. The board can operate on an external supply of

6to20volts.Ifsuppliedwithlessthan7V,however,the5Vpinmaysupplylessthanfivevoltsand theboardmaybeunstable.Ifusingmorethan12V,thevoltageregulatormayoverheatanddamagetheboard. Therecommended rangeis 7to 12 volts. the power pinsareas follows.

• VIN. The input voltage to the Arduino board when it's using an external power source (asopposed to 5 volts from the USB connection or other regulated power source). You can supplyvoltage through this pin, or, if supplying voltage via the power jack, access it through this pin. •5V. The regulated power supply used to power the microcontroller and other components on theboard.Thiscancomeeitherfrom VINviaanon-boardregulator,orbesuppliedbyUSBoranotherregulated5V supply.

• 3.3V.A3.3voltsupplygenerated bytheon-boardregulator.Maximum currentdrawis50mA.

• GND.Groundpins.

4.1.1.7 IntroductiontoAtmega328

ATmega328isaneight(8)bitMicrocontroller.Itcanhandlethedatasizedofuptoeight(8)bits.It is an AVR based micro-controller. Its built-in internal memory is around 32KB. It operatesranging from 3.3V to 5V. It has an ability to store the data even when the electrical supply isremoved from its biasing terminals. it is the excellent features include the cost efficiency and lowpower dissipation, programming lock for security purposes, real timer counter with the separateoscillatorandItisgenerallyornormallyusedinEmbeddedSystemsapplications.ATmega-328isshownin thefiguregivenbelow.

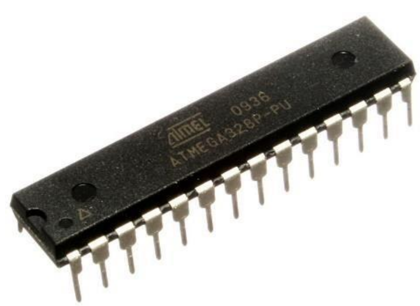


Fig.4.1.3:ATMega328 Microcontroller

4.1.1.8 ArduinoUNOAtmega328

The Arduino UNO is a micro controller board based on the ATmega. UNP means one in ItalianandisnamedtomarktheupcomingreleaseofthegivenArduino1.0ithas14digitalinput/outputpins , 6 analog inputs, 16MHz ceramic resonator. A USB connection. A power jack, An ICSPheaderandaresetbutton.itcontainsinformationeverythingneededtosupportthemicrocontroller.simplyconnectittoacomputerwithaUSB cableorpoweritwith aac-to-dcadapterorbatterytoget started. The UNO differs from all preceding boards in that it does not use the FTDI USBtoserial driver chip. Instead it features the ATMega 16U2ATMega 8U2 up to version instead R2)programmedasaUSB to-serial converter.

4.1.1.9 ArchitectureandWorkingof ArduinoUno

TheArduinoUNOcanbealsopoweredviatheUSBconnectionorwithanexternalpowersupply.The power source is selected automatically. External (non-USB) power can come either from anAC-to-DCadapterorbattery.Theadaptercanconnectedbyplugginga2.1mmcentrepositiveplugintothe board's power jack. Leads from a battery can be inserted in the Gnd and vin pin headersof the power connector. The board can operate on an external supply of 6to 20 volts. If suppliedwith less than 7v however, the 5v pin may supply less than five volts and the board may beunstable. If using more than 12v.the voltage regulator may overheat and damage the board. Therecommended range is 7 to 12 volts. The processor of the Arduino board uses the Harvardarchitecture where the program code and program data have separate memory. It consists of twomemoriessuchasprogrammemoryanddatamemory.Whereinthedataisstoredindatamemoryand the code is stored in the flash program memory. The Atmega328 microcontroller has 32kb offlashmemory,2kbofSRAM1kbofEPROMandoperateswitha16MHzclockspeed.Architectureof Atmega328 is shown below.

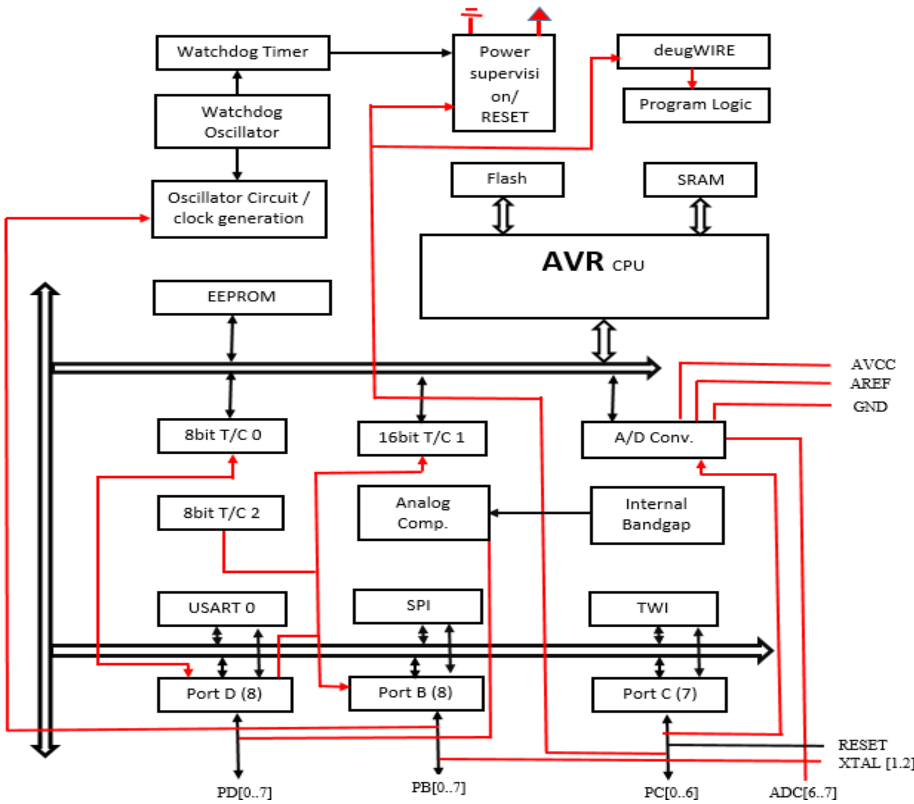


Fig.4.1.4:ATmega328PArchitecture

4.1.1.10 Atmega328PinsDescription

Functionsassociatedwiththepinsmustbeknowninordertousethedeviceappropriately.ATmega-328pins aredivided intodifferent portswhich aregiven in detailbelow.

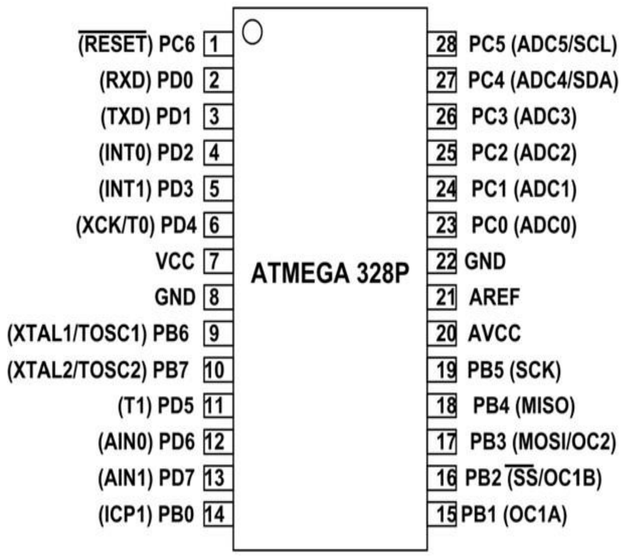


Fig.4.1.5:ATmega328P Ports

AVCCis asupply voltagepinforanalog todigital converter.

VCCis adigital voltage supply GNDdenotes Groundand it hasa0V.

PortAconsistsofthepinsfromPA0toPA7.Thesepinsserveasanaloginputtoanalogtodigitalconverters. If analog to digital converter is not used, port A acts as an eight (8) bit bidirectionalinput/output port.

Port B consists of the pins from PB0 to PB7. This port is an 8 bit bidirectional port having aninternalpull-up resistor.

PortCconsists ofthepins from PC0to PC7. Theoutputbuffers ofport C

PortDconsistsofthepinsfromPD0toPD7.Itisalsoan8bitinput/outputporthavinganinternalpull-upresistor. All of theAVR ports areshown in figuregiven below.

4.1.2 ALCOHOLSENSOR

- MQ3 is one of the most commonly used sensors in the MQ sensor series.
- It is a Metal Oxide Semiconductor (MOS) type of sensor. Metal oxide sensors are also known as Chemiresistors, because sensing is based on the change of resistance of the sensing material when exposed to alcohol.

- So by placing it in a simple voltage divider network, alcohol concentrations can be detected. MQ3 alcohol sensor works on 5V DC and draws around 800mW. It can detect Alcohol concentrations anywhere from 25 to 500 ppm.

Fig4.1.6:MQ3 AlcoholSensor

MQ3 is a heater-driven sensor. That’s why it is enclosed in two layers of fine stainless steel mesh called an Anti-explosion network. It ensures that heater element inside the sensor will not cause an explosion, as we are sensing flammable gas (alcohol). It also provides protection for the sensor and filters out suspended particles so that only gaseous elements are able to pass inside the chamber. This is what the sensor looks like when outer mesh is removed. The starshaped structure is formed by the sensing element and six connecting legs that extend beyond the Bakelite base. Out of six, two leads (H) are responsible for heating the sensing element and are connected via a NickelChromium coil (awell known conductive alloy). The remaining four leads (A & B) responsible for output signals are connected using Platinum Wires. These wires are connected to the body of the sensing element and convey small changes in the current that passes through the sensing element. The tubular sensing element is made up of Aluminum Oxide (AL2O3) based ceramic and has a coating of Tin Dioxide (SnO2). The Tin Dioxide is the most important material being sensitive towards alcohol. However, the ceramic substrate only increases the heating efficiency and ensures that the sensor area is continuously heated to the working temperature. So, to summarize, the Nickel-Chromium coil and Aluminum Oxide based ceramic forms a Heating System; while Platinum wires and coating of Tin Dioxide forms a Sensing System.

4.1.3 PowerSupply

Every electrical and electronic device that we use in our day-to-day life will require a powersupply.Ingeneral,weuseanACsupplyof230V50Hz,butthispowerhastobechangedintotherequiredformwithrequiredvaluesorvoltagerangeforprovidingpowersupplytodifferenttypesof devices. There are various types of power electronic converters such as step-down converter,step-up converter, voltage stabilizer, AC to DC converter, DC 25 to DC converter, DC to ACconverter, and so on. For example, consider the microcontrollers that are used frequently fordevelopingmanyembeddedsystemsbasedprojectsandkitsusedinreal-timeapplications.Thesemicrocontrollersrequire a5V DC supply.

4.1.4 LCD(Liquid CrystalDisplay)

A liquid crystal display (LCD) is a thin, flat display device made up of any number of colour ormonochromepixelsarrayedinfrontofalightsourceorreflector.Eachpixelconsistsofacolumn of liquid crystal molecules suspended between two transparent electrodes, and two polarizingfilters, the axes of polarity of which are perpendicular to each other. Without the liquid crystalsbetween them, light passing through one would be blocked by the other. The liquid crystal twiststhe polarization of light entering one filter to allow it to pass through the other. A program mustinteract with the outside world using input and output devices that communicate directly with ahumanbeing.OneofthemostcommondevicesattachedtoancontrollerisanLCDdisplay.Someof the most common LCDs connected to the controllers are 16X1, 16x2 and 20x2 displays. Thismeans 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per lineby2 lines, respectively.

Many microcontroller devices use 'smart LCD'displaysto output visual information. LCDdisplays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is evenpossibletoproduceareadoutusingthe5X7dotspluscursorofthedisplay.TheyhaveastandardASCIIsetofcharactersandmathematicalsymbols.Foran8-bitdatabus,thedisplayrequiresa +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0) For a 4-bit data bus it onlyrequiresthesupplylinesplus6extralines(RSRWD7D6D5D4).WhentheLCDdisplayisnotenabled,datalinesaretri-stateandtheydonotinterferewiththeoperationofthemicrocontroller.

Pin No	Symbol	Level	Description
1	Vss	0 V	Ground
2	V_{DD}	5.0V	Supply voltage for logic
3	V o	variable	Operating voltagefor LCD
4	RS	H/L	H data/L instruction code
5	R/W	H/L	H/Read (MPU-module) L/write (MPU-module)
6	E	H, H-L	Chip enable signal
7	DB0	H/L	Data bit 0
8	DB1	H/L	Data bit 1
9	DB2	H/L	Data bit 2
10	DB3	H/L	Data bit 3
11	DB4	H/L	Data bit 4
12	DB5	H/L	Data bit 5
13	DB6	H/L	Data bit 6
14	DB 7	H/L	Data bit 7
15	A		Power supply for LED backlight (+)
16	K		Power supply for LED backlight (-)

Table4.1.2:Pindescription

DescriptionOf16x2:

This is the first interfacing example for the Parallel Port. We will start with something simple.Thisexampledoesn'tusetheBi-directionalfeaturefoundonnewerports,thusitshouldworkwithmost, if no all Parallel Ports. It however doesn't show the use of the Status Port as an input. Sowhat are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCDModulesareverycommonthesedays,andarequitesimpletoworkwith,asallthelogicrequiredtorun them is on board.

SchematicDiagram

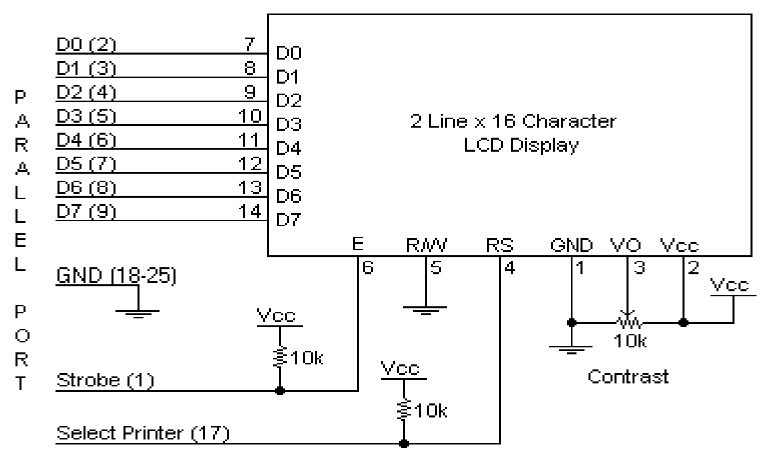


Fig.4.1.11:SchematicDiagramofLCD

• Aboveisthequitesimpleschematic.TheLCDpanel'sEnableandregisterselectisconnectedtothe Control Port. The Control Port is an open collector / open drain output. While most ParallelPorts have internal pullup resistors, there are a few which don't. Therefore by incorporating thetwo 10K external pull up resistors, the circuit is more portable for a wider range of computers,someofwhich may haveno internal pull up resistors.

• WemakenoefforttoplacetheDatabusintoreversedirection.ThereforewehardwiretheR/Wline of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As aresultwecannotreadbacktheLCD'sinternalBusyFlagwhichtellsusiftheLCDhasaccepted andfinishedprocessingthelastinstruction.This problemisovercomebyinsertingknowndelaysintoourprogram.

• The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with alltheexamples, I'veleftthepowersupplyout.You canuseabenchpowersupplysetto5voruseaonboard +5 regulator. Remember a few de-coupling capacitors, especially if you have troublewiththecircuit workingproperly.

HexCode	CommandtoLCDInstructionRegister
0F	LCDON,cursor ON
01	Cleardisplayscreen
02	Returnhome
04	Decrementcursor(shiftcursorto left)
06	Incrementcursor (shiftcursortoright)
05	Shiftdisplay right
07	Shiftdisplay left
0E	DisplayON, cursorblinking
80	Forcecursorto beginningoffirstline
C0	Forcecursortobeginningofsecond line
38	2linesand5×7matrix
83	Cursorline1 position 3
3C	Activatesecondline
08	DisplayOFF,cursorOFF
C1	Jumptosecond line,position 1
OC	DisplayON,cursorOFF
C1	Jumpto secondline, position1
C2	Jumptosecond line,position 2

Table4.1.3:LCDcommands

4.1.5 BLOOD PRESSURESENSOR

- Blood Pressure can be measured both by invasive and non-invasive methods.
- In the non-invasive method, no piercing is required and is easy to use.
- Blood Pressure Sensor is used to measure the blood pressure using the non-invasive method.
- It is similar to sphygmomanometer but instead of the mercury column, a pressure sensor is used to detect the blood pressure.

Fig4.1.12 Blood pressureSensor

Applications

- This sensor is very important for High Blood Pressure patients, as it is also available as 'at-home' solid-state Blood Pressure Monitor.
- This system is portable.
- It is easy to carry and operate and highly useful in remote areas where medical facilities are not available.
- The main sensing element of this system is the pressure sensor present in the chuff. For an accurate and reliable measurement, this pressure sensor should be carefully selected.
- Honeywell's 26 PC SMT pressure sensor is one of the examples of pressure sensors used in this system.
- This sensor is small, low-cost and can measure higher values of pressure. This sensor is used directly with the printed circuit board and can measure pressure faster and more accurately.

As the sensor provides true surface mount capability, true installation cost of this sensor is very low

ADVANTAGE:

- Being non-invasive, this Sensor is safe to use. It is easier to use and can be monitored by any individual. Instead of watching the mercury levels and calculating pressure, this sensor makes the task easier by giving results automatically.

4.1.6 WIFIESP8266 MODULE:

- TheESP8266WiFiModuleisaselfcontainedSOCwithintegratedTCP/IPprotocolstackthatcan giveanymicrocontrolleraccesstoyourWiFinetwork.TheESP8266iscapableofeither hosting an application or offloading all Wi-Fi networking functions from anotherapplicationprocessor.EachESP8266modulecomespre-programmedwithanATcommandsetfirmware,meaning,youcansimplyhookthisuptoyourArduinodeviceandgetaboutasmuchWiFi-abilityasaWiFiShieldoffers(andthat'sjustoutofthebox)!TheESP8266 module is an extremely cost effective board with a huge, and ever growing,community.
- Thismodulehasapowerfulenoughon-boardprocessingandstoragecapabilitythatallowsittobeintegratedwiththesensorsandotherapplicationspecificdevicesthroughitsGPIOswith minimal development up-front and minimal loading during runtime. Its high degreeofon-chipintegrationallowsforminimalexternalcircuitry,includingthefront-endmodule,isdesignedtooccupyminimalPCBarea.TheESP8266supportsAPSDforVoIPapplicationsandBluetoothco-existanceinterfaces,itcontainsaself-calibratedRFallowingitto workunder all operatingconditions,and requiresnoexternalRFparts.

- ThereisanalmostlimitlessfountainofinformationavailablefortheESP8266,allofwhichhasbeenprovidedbyamazingcommunitysupport.IntheDocumentssectionbelowyou willfindmanyresourcestoaidyouinusingtheESP8266,eveninstructionsonhowtotransformin gthis module intoan IoT(Internet of Things) solution!

- Note: The ESP8266 Module is not capable of 5-3V logic shifting and will require anexternal Logic Level Converter. Please do not power it directly from your 5V dev board.ThisESP8266WiFiModuleAdaptorcomeswithon-boardlogiclevelconverterand3.3VLDOregulator foreasy interfacing toArduino boards.

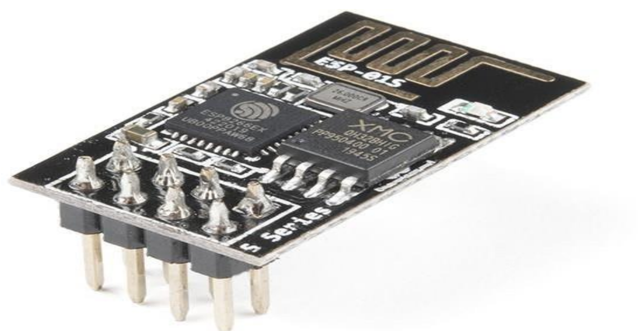


Fig4.1.13ESP8266Wi-FiModule

Features:

- 802.11 b/g/n
- Wi-FiDirect(P2P),soft-AP
- IntegratedTCP/IPprotocolstack
- IntegratedTRswitch,balun,LNA,power amplifierandmatchingnetwork
- IntegratedPLLs, regulators,DCXOandpowermanagementunits
- +19.5dBmoutput power in802.11b mode
- Powerdownleakagecurrentof<10uA
- Integratedlowpower32-bitCPUcouldbeusedasapplicationprocessor
- SDIO1.1/2.0, SPI,UART
- STBC,1×1MIMO,2×1MIMO
- A-MPDU&A-MSDUaggregation&0.4msguardinterval

- Wakeupandtransmit packetsin<2ms
- Standbypower consumption of<1.0mW (DTIM3)
- Defaultbaudrate:115200

4.2 SoftwareRequirement

- Arduinosoftware
- Programminglanguage

4.2.1 Arduinosoftware:

Arduinoisanopen-sourceprototypingplatformbasedon easy-to-usehardwareandsoftware.Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twittermessage - and turn it into an output - activating a motor, turning on an LED, publishingsomething online. You can tell your board what to do by sending a set of instructions to themicrocontroller on the board. To do so you use the Arduino programming language (based onWiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino hasbeenthebrainofthousandsofprojects,fromeverydayobjectstocomplexscientificinstruments. A worldwide community of makers - students, hobbyists, artists, programmers,and professionals has gathered around this open-source platform, their contributions haveaddeduptoanincredibleamountofaccessibleknowledgethatcanbeofgreathelp tonovicesandexpertsalike.ArduinowasbornattheIvreaInteractionDesignInstituteasaneasytoolforfast prototyping, aimed at students without a background in electronics and programming. Assoon as it reached a wider community, the Arduino board started changing to adapt to newneeds and challenges, differentiating its offer from simple 8-bit boards to products for IoTapplications, wearable, 3D printing, and embedded environments. All Arduino boards arecompletelyopen-source,empoweringuserstobuildthemindependentlyandeventuallyadaptthemtotheirparticularneeds.Thesoftware,too,isopen-source,anditisgrowingthroughthecontributionsof users worldwide.

4.2.1.1 WhyArduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands ofdifferent projects and applications. The Arduino 44 software is easy-to-use for beginners, yetflexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and studentsuse it to build low-cost scientific instruments, to prove chemistry and physics principles, or to getstarted with programming and robotics. Designers and architects build interactive prototypes,musicians and artists use it for installations and to experiment with new musical instruments.Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example.Arduinoisakeytooltolearnnewthings.Anyone-children,hobbyists,artists,programmers-canstart tinkering just following the step-by-step instructions of a kit, or sharing ideas online withothermembersoftheArduinocommunity.Therearemanyothermicrocontrollersandmicrocontroller platforms available for physical computing. Parallax Basic Stamp, Net media'sBX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of thesetools take the messy details of microcontroller programming and wrap it up in an easy-to-usepackage. Arduino also simplifies the process of working with microcontrollers, but it offers someadvantageforteachers, students, and interested amateurs over other systems:

• Inexpensive - Arduino boards are relatively inexpensive compared to other microcontrollerplatforms.TheleastexpensiveversionoftheArduinomodulecanbeassembledbyhand, andeventhepre-assembled Arduino modules cost less than$50

• Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linuxoperatingsystems. Most microcontroller systemsarelimitedto Windows.

• Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use forbeginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it'sconvenientlybasedontheProcessingprogrammingenvironment,sostudentslearningtoprogramint hat environment willbefamiliar with howtheArduinoIDEworks.

• Open source and extensible software - The Arduino software is published as opensource tools,availableforextensionbyexperiencedprogrammers.ThelanguagecanbeexpandedthroughC++li braries,andpeoplewantingtounderstandthetechnicaldetailscanmaketheleapfromArduino totheAVRCprogramminglanguageonwhichit'sbased.Similarly,youcanaddAVR-Ccodedirectlyinto yourArduino programs if youwant to.

• Opensourceandextensiblehardware-
TheplansoftheArduinoboardsarepublishedunderaCreativeCommonslicense,so experiencedcircuitdesignerscanmaketheir ownversion

• Of the module, extending it and improving it. Even relatively inexperienced users can build thebreadboardversion of themodulein order tounderstand how it worksandsavemoney

• GettingStartedwithArduinoandGenuinoproducts.

• InstalltheArduinoSoftware(IDE)onWindowsPCs-

• Thisdocumentexplains howto installtheArduinoSoftware(IDE)onWindows machines.

• DownloadtheArduino Software(IDE)

• Proceedwithboardspecificinstructions.

4.2.1.2 HowtoDownload theArduinoSoftware(IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) andtheZippackages.Wesuggestyouusethefirstonethatinstallsdirectlyeverythingyouneedtousethe Arduino Software (IDE), including the drivers. With the Zip package you need to install thedrivers manually. When the download finishes, proceed with the installation and please allow thedriverinstallation process when youget a warning from theoperating system.

4.2.1.3 Installation

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer andpreparethe board to receivethe program viaUSB cable.

Step1−FirstyoumusthaveyourArduinoboard(youcanchooseyourfavoriteboard)andaUSBcable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, orDiecimila,youwillneedastandardUSBcable(A plug toBplug),thekindyouwouldconnect toaUSB printeras shown in the following image.

Fig.4.2.1:USB CableAplug toB plug

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in thefollowingimage.

Fig.4.2.2:USB Cable Aplug to Mini-B plug

Step2−DownloadArduinoIDESoftware.

You can get different versions of Arduino IDE from the Download page on the Arduino Officialwebsite. You must select your software, 47 which is compatible with your operating system(Windows,IOS, or Linux). Afteryour filedownload is complete,unzip thefile.

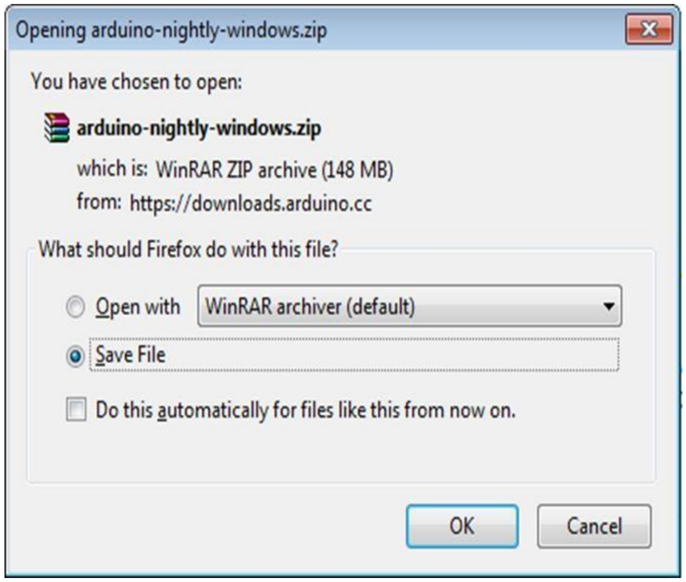


Fig.4.2.3:UnzipofArduinoIDEsoftware

Step3−Powerupyourboard.

TheArduinoUno,Mega,DuemilanoveandArduinoNanoautomaticallydrawpowerfromeither, the USB connection to the computer or an external power supply. If you are using anArduino Diecimila, you have to make sure that the board is configured to draw power from theUSB connection. The power source is selected with a jumper, a small piece of plastic that fitsonto two of the three pins between the USB and power jacks. Check that it is on the two pinsclosestto the USB port.

ConnecttheArduinoboardtoyourcomputerusingtheUSBcable.ThegreenpowerLED(labeledPWR) shouldglow.

Step4−LaunchArduinoIDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside thefolder,youcanfindtheapplicationiconwithaninfinitylabel(application.exe).Double-clicktheiconto start theIDE.

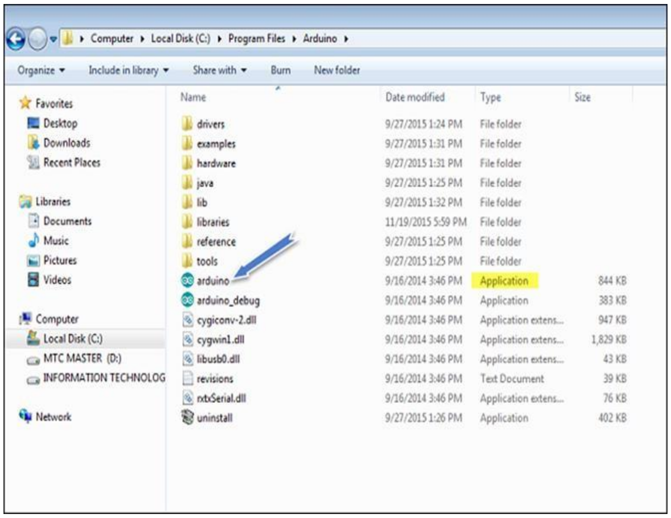


Fig.4.2.4:Open IDEsoftware

Step5−Openyourfirstproject.

Oncethe softwarestarts,you havetwo options:

• Createanewproject

.•Openanexistingprojectexample.

Tocreateanewproject,selectFile→New.

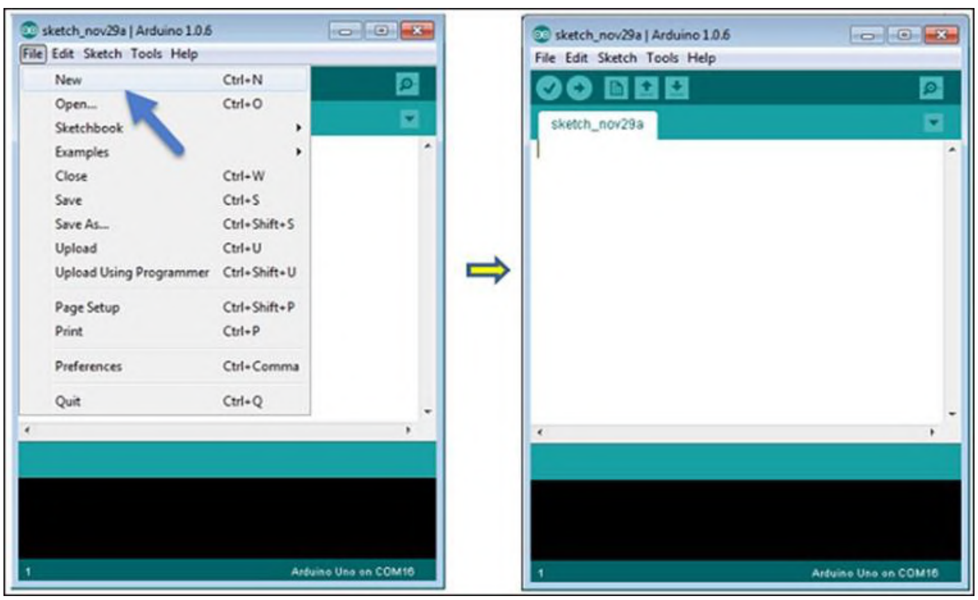


Fig.4.2.5:OpenFirstProject

Toopenanexistingprojectexample,selectFile→ Example→Basics→Blink.

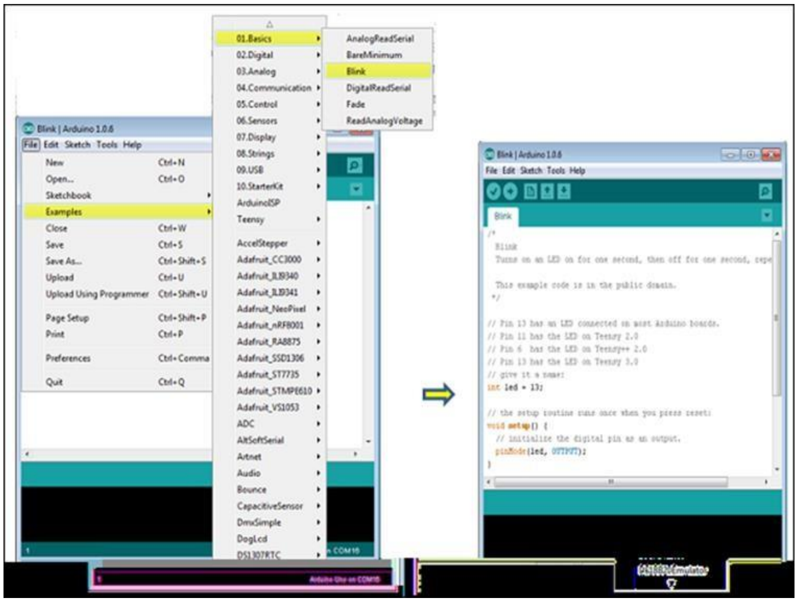


Fig.4.2.6:Openexisting Project

Here, we are selecting just one of the examples with the name Blink. It turns the LED on and offwithsome timedelay. You can selectany otherexamplefrom thelist.

Step 6 − Select your Arduino board. To avoid any error while uploading your program to theboard, you must select the correct Arduino board name, which matches with the board connectedtoyour computer. Go toTools→ Board and select your board.

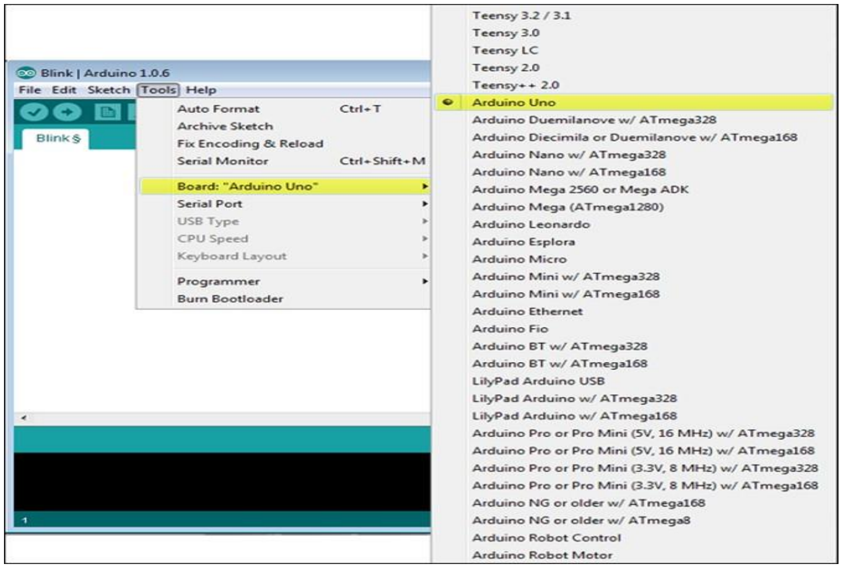


Fig.4.2.7:SelectionofArduinoboard

Here,wehaveselectedArduinoUnoboardaccordingtoourtutorial,butyoumustselectthenamematchingtheboard that youareusing.

Step 7 − Select your serial port. Select the serial device of the Arduino board. Go to Tools →SerialPortmenu.ThisislikelytobeCOM3orhigher(COM1andCOM2areusuallyreservedforhardwareserialports).Tofindout,youcandisconnectyourArduinoboardandre-openthemenu,theentrythatdisappearsshouldbeoftheArduinoboard.Reconnecttheboardandselectthatserialport.

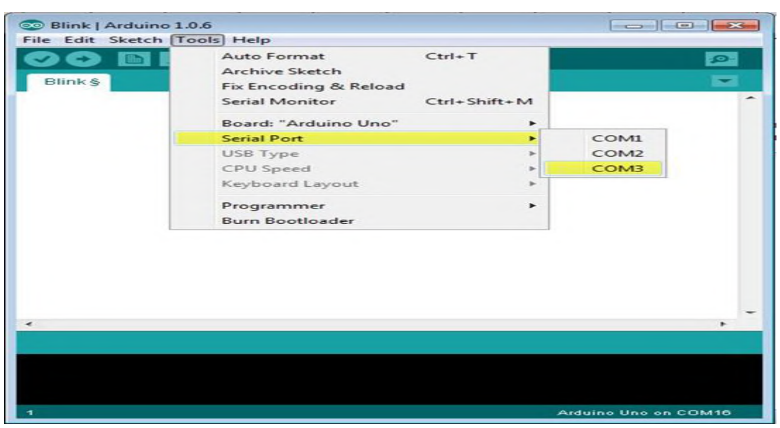


Fig.4.2.8:SelectionofSerialPort

Step8−Uploadtheprogram toyourboard.

Beforeexplaininghowwecanuploadourprogramtotheboard,wemustdemonstratethefunctionofeach symbol appearing in theArduinoIDEtoolbar.

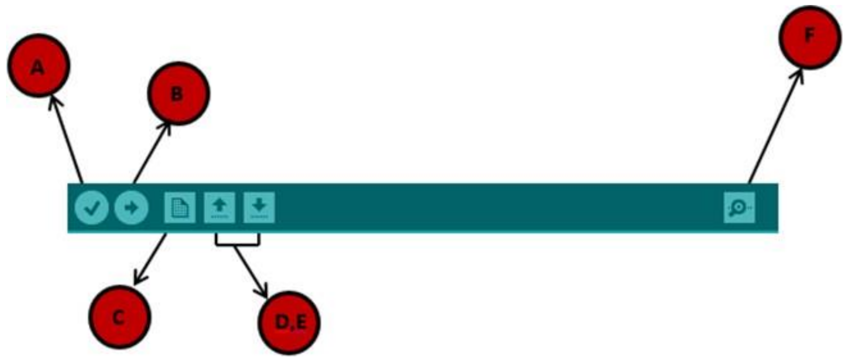


Fig.4.2.9: Uploading the program to boardA−Used to check if thereis any compilation error.

B − Used to upload a program to the Arduino board.C−Shortcut usedto create anewsketch.

D − Used to directly open one of the example sketch.E−Used to saveyoursketch.

F−Serialmonitorusedtoreceiveserialdatafromtheboardandsendtheserialdatatotheboard.Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see theRXandTXLEDsontheboard,flashing.Iftheuploadissuccessful,themessage"Doneuploading"will appear in thestatusbar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset buttonphysicallyontheboard, immediatelybeforeclickingthe uploadbuttonon theArduinoSoftware.

4.2.2 Proteus

4.2.2.1 Proteus

ProteusisasimulationanddesignsoftwaretooldevelopedbyLabcentreElectronicsforElectricaland Electronic circuit design. It also has 2D CAD drawing feature. It deserves to bear the tagline“Fromconcept to completion”.

4.2.2.2 AboutProteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is thesoftware used to draw schematics and simulate the circuits in real time. The simulation allowshuman access during run time, thus providing real time simulation. ARES is used for PCBdesigning. It has the feature of viewing output in 3D view of the designed PCB along withcomponents.Thedesigner can also develop 2D drawings forthe product.

4.2.2.3 Features

ISIS has wide range of components in its library. It has sources, signal generators, measurementand analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring oftheparametersofthecircuit,switches,displays,loadslikemotorsandlamps,discretecomponentslike resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductorswitches,relays,microcontrollers,processors,sensorsetc.ARESoffersPCBdesigningupto14innerlayers,withsurfacemountandthroughholepackages. Itisembeddedwiththefootprints of different category of components like ICs, transistors, headers, connectors and otherdiscretecomponents.ItoffersAutoroutingandmanualroutingoptionstothePCBDesigner.Theschematicdrawn in theISIS can bedirectly transferredARES.

4.2.2.4 StartingNewDesign

Step1:OpenISISsoftwareandselect NewdesigninFilemenu

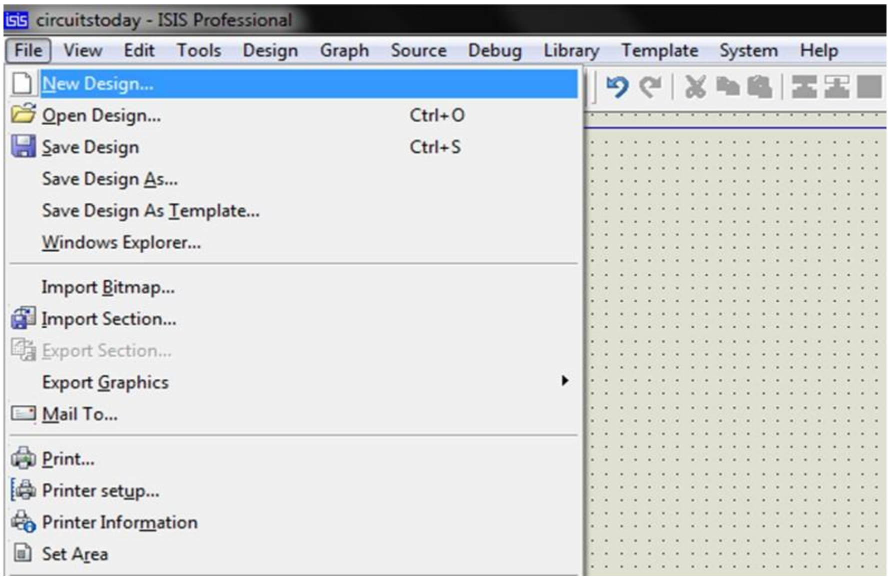


Fig.4.2.12:Proteusfilemenu

Step2: Adialogueboxappearstosavethecurrentdesign.However,wearecreatinganewdesignfile so you can click Yes or No 57 depending on the content of the present file. Then a Pop-Upappears asking to select the template. It is similar to selecting the paper size while printing. Fornow,select default or according to thelayoutsizeof thecircuit.

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better tocreateanewfolderforeverylayoutasitgeneratesotherfilessupportingyourdesign.However,itisnot mandatory.

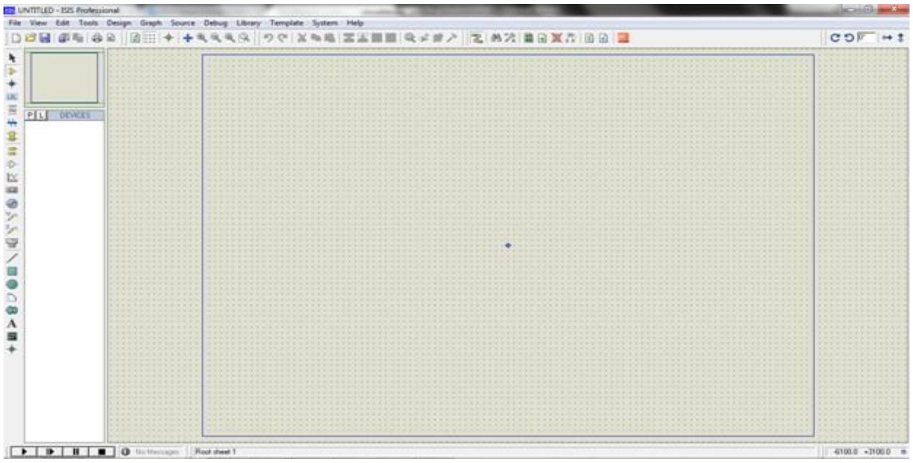


Fig.4.2.13:ProteusDesignSheet

Step4: To Selectcomponents, Clickon thecomponent modebutton.

Step5: ClickonPickfromLibraries.Itshowsthecategoriesofcomponentsavailableandasearchoptionto enter thepart name.

Step6: Selectthecomponentsfromcategories ortypethe partnamein Keywordstextbox.

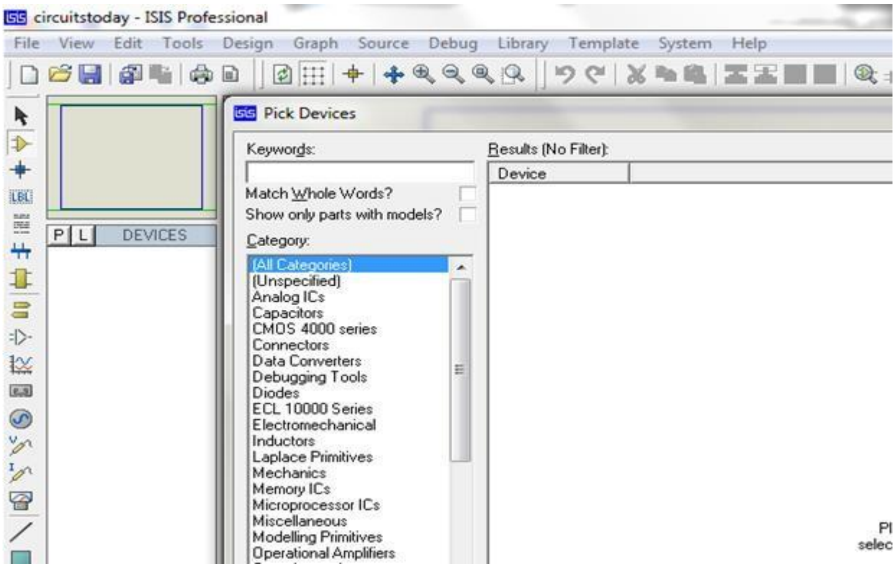


Fig.4.2.14:KeyTextbox

Step 7: The selected components will appear in the devices list. Select the component and placeitinthedesignsheetbyleft-click.Placealltherequiredcomponentsandroutethewiresi.e.,makeconnections. Either selection mode above the component mode or component mode allows toconnect through wires. Left click from one terminal to other to make connection. Double right-clickontheconnectedwireorthecomponenttoremoveconnectionorthecomponentrespectively.

Step8: After connectingthecircuit, clickonthe playbuttonto runthesimulation.

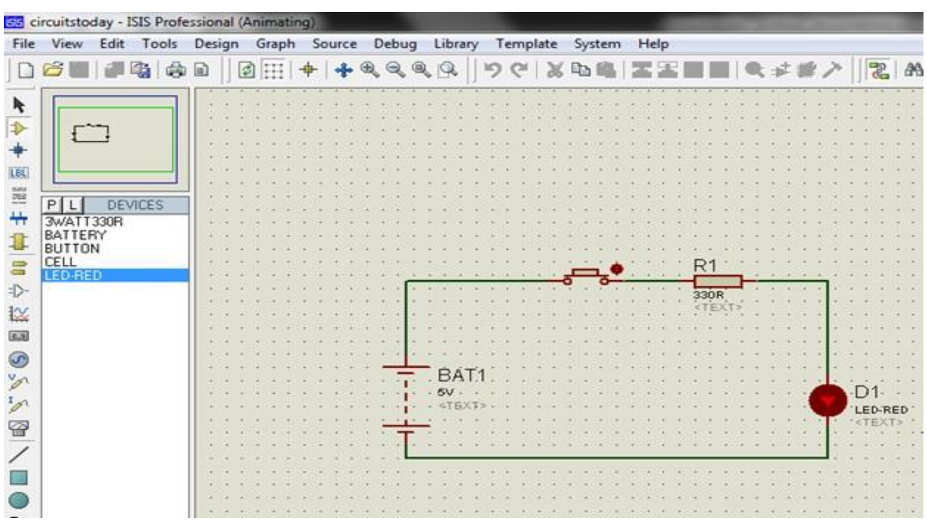


Fig.4.2.15:StimulationSoftware

CHAPTER5

SOFTWARETESTINGANDCODEIMPLEMENTATION

5.1 SOFTWARETESTING

SoftwareTestingisaprocessofexecutingtheapplicationwithanintenttofindanysoftwarebugs.It is used to check whether the application met its expectations and all the functionalities of theapplicationisworking.Thefinalgoaloftestingistocheckwhethertheapplicationisbehavinginthewa yitissupposedtounderspecifiedconditions.Allaspectsofthecodeareexaminedtocheckthe quality of application. The primary purpose of testing is to detect software failures so thatdefects may be uncovered and corrected. The test cases are designed in such way that scope offinding thebugs is maximum.

5.2 TESTINGLEVELS

Therearevarioustestinglevelsbasedonthespecificityoftest.

- Unit testing: Unit testing refers to tests conducted on a section of code in order to verify thefunctionalityof that piece ofcode.This isdoneatthefunction level.
- Integration Testing: Integration testing is any type of software testing that seeks to verify theinterfaces between components against a software design. Its primary purpose is to expose thedefectsassociated with theinterfacing of modules.
- System Testing: System testing tests a completely integrated system to verify that the systemmeetsits requirements.
- Acceptancetesting:Acceptancetestingteststhereadinessofapplication,satisfyingallrequirements.

• Performancetesting:Performancetestingistheprocessofdeterminingthespeedoreffectiveness of a computer, network, software program or device such as response time ormillionsof instructions per secondetc.

5.3 SYSTEMTESTCASES

Atestcaseisasetoftestdata,preconditions,expectedresultsandpostconditions,developedfora test scenario to verify compliance against a specific requirement. We have designed andexecuted afew test casestocheck if theproject meetsthefunctional requirements.

5.4 CODEIMPLEMENTATION

5.4.1 PROJECT CODE

```
#include <SoftwareSerial.h>
SoftwareSerialmySerial(2,3);

#include <LiquidCrystal.h>
LiquidCrystallcd(13,12,11,10,9,8);



String readvoice;
char buff[200],k=0;
char c=0,c1=0;
unsigned char rec = 0,buf[120],buf1[120],buf3[120];
unsignedint count=0,ik=0,ct=0,value=0;
void upload1(unsigned char *chr,unsigned char *chr1);
char res[130];
char check(char* ex,int timeout)
{
```

```
inti=0;
int j = 0,k=0;
while (1)
  {
sl:
if(mySerial.available() > 0)
    {
res[i] = mySerial.read();
if(res[i] == 0x0a || res[i]=='>' || i == 100)
      {
i++;
res[i] = 0;break;
      }
i++;
    }
j++;
if(j == 30000)
    {
k++;
     //Serial.Sprintln("kk");
     j = 0;
    }
if(k > timeout)
    {
    // Serial.println("timeout");
```

```
return 1;
     }
  }//while 1
if(!strncmp(ex,res,strlen(ex)))
  {
    //Serial.println("ok..");
return 0;
   }
else
  {
   // Serial.print("Wrong  ");
   // Serial.println(res);
i=0;
gotosl;
   }
}

voidserialFlush(){
while(Serial.available() > 0) {
char t = Serial.read();
  }
}


intalc=A4,buz=7;
```

```
const char* ssid = "project";
const char* password = "project123";
void setup() {
char ret;
pinMode(buz, OUTPUT);

digitalWrite(buz,HIGH);
lcd.begin(16,2);
lcd.clear();lcd.setCursor(0, 0);lcd.print("IOT ALCOHOL & HEALTH MONITORING SYSTEM");
delay(2000);
Serial.begin(9600);
mySerial.begin(115200);//esp

st:
mySerial.println("ATE0");
ret  = check((char*)"OK",50);
mySerial.println("AT");
ret  = check((char*)"OK",50);
if(ret != 0)
  {
delay(1000);
gotost;
  }
```

```
lcd.clear();lcd.setCursor(0, 0);lcd.print("CONNECTING");
mySerial.println("AT+CWMODE=1");
ret  = check((char*)"OK",50);
cagain:

serialFlush();
mySerial.print("AT+CWJAP=\"");
mySerial.print(ssid);
mySerial.print("\",\"");
mySerial.print(password);
mySerial.println("\"");
if(check((char*)"OK",300))gotocagain;
mySerial.println("AT+CIPMUX=1");
delay(1000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("CONNECTED");delay(1000);



  //lcd.clear();

}

void loop()
{
int ad = digitalRead(alc);
```

```
  /*  while (Serial.available()){

delay(10);

char c = Serial.read();

readvoice += c;

          }



   // lcd.clear();lcd.setCursor(0, 0);lcd.print(readvoice); delay(500);

if(readvoice.length() > 0) {

   // Serial.println(readvoice);

lcd.clear(); lcd.setCursor(0,1);lcd.print(readvoice);delay(500);



if(readvoice == "*+00000#" || readvoice == "*-00000#")

  {

     // strcpy(buf,"120_80");

lcd.clear();lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("120_80");

     // upload1(ad,buf);

delay(5000);

  }

if(ad == LOW)

   {

lcd.clear();lcd.setCursor(0, 0);lcd.print("PERSON DRUNK");



upload1(ad,buf);delay(2000);

   }

readvoice="";
```

```
  }*/

  //Serial.println(ad);
  //lcd.clear();lcd.setCursor(0, 0);lcd.print("A:");lcd.print(ad);
          //  lcd.setCursor(0, 8);lcd.print("BP:");lcd.print(buf);
delay(2000);
   //while(!Serial.available());
if(ad == LOW)
        {
lcd.clear();lcd.setCursor(0, 0);lcd.print("PERSON DRUNK");digitalWrite(buz,LOW);
        //upload1("STUDENT_1_PRESENT");delay(2000);
upload1(ad,buf);delay(2000);digitalWrite(buz,HIGH);
        }
if (Serial.available()> 0)
   {
delay(10);
while(!Serial.available());
       c = Serial.read();
lcd.clear();lcd.setCursor(0, 0);lcd.print("A:");lcd.print(ad);
lcd.setCursor(0, 1);lcd.print("c:");lcd.print(c);
if(c == '*')
          {        while(!Serial.available());
                 c1    =  Serial.read();
lcd.setCursor(5, 1);lcd.print("c1:");lcd.print(c1);
```

```
if(c1 == '+'|| c1 == '-' )
                {                       while(!Serial.available());
buf[0] =        Serial.read();
while(!Serial.available());
buf[1] =        Serial.read();
while(!Serial.available());
buf[2] =        Serial.read();
while(!Serial.available());
buf[3] =        Serial.read();
while(!Serial.available());
buf[4] =        Serial.read();
buf[5] = 0;
                        value = ((buf[0] - '0')*10000)+((buf[1] - '0')*1000)+((buf[2] - '0')*100)+((buf[3] -
      '0')*10)+((buf[4] - '0'));
lcd.setCursor(10, 1);lcd.print("v:");lcd.print(value);
if(value > 120)
                        {
if(value%10 == 0)
                                    {
strcpy(buf,"120_80");
lcd.clear(); lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("120_80");
upload1(ad,buf);
delay(2000);
                                    }
if(value%10 == 1)
```

```
{
strcpy(buf,"125_70");
lcd.clear(); lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("125_70");
upload1(ad,buf);
delay(2000);
}
if(value%10 == 2)
{
strcpy(buf,"110_80");
lcd.clear(); lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("110_80");
upload1(ad,buf);
delay(2000);
}
if(value%10 == 3)
{
strcpy(buf,"115_82");
lcd.clear();lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("115_82");
upload1(ad,buf);
delay(2000);
}
if(value%10 == 4)
{
strcpy(buf,"130_90");
lcd.clear(); lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("130_90");
upload1(ad,buf);
```

```
delay(2000);
        }
if(value%10 == 5)
        {
strcpy(buf,"125_92");
lcd.clear(); lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("125_92");
upload1(ad,buf);
delay(2000);
        }
if(value%10 == 6)
        {
strcpy(buf,"135_82");
lcd.clear();lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("135_82");
upload1(ad,buf);
delay(2000);
        }
if(value%10 == 7)
        {
strcpy(buf,"127_87");
lcd.clear();lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("127_87");
upload1(ad,buf);
delay(2000);
        }
if(value%10 == 8)
        {
```

```
strcpy(buf,"132_89");

lcd.clear(); lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("132_89");

upload1(ad,buf);

delay(2000);

                                }

if(value%10 == 9)

                                {

strcpy(buf,"121_72");

lcd.clear(); lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("121_72");

upload1(ad,buf);

delay(2000);

                                }

if(ad == LOW)

                                  {


lcd.clear();lcd.setCursor(0, 0);lcd.print("PERSON DRUNK");

                                   //upload1("STUDENT_1_PRESENT");delay(2000);

upload1(ad,buf);delay(2000);

                                   }

                                 }

                                }


                              }


                              /* if(ad == LOW)
```

```
                                    {

lcd.clear();lcd.setCursor(0, 0);lcd.print("PERSON DRUNK");
upload1(ad,buf);delay(2000);
                                    }  */

/*
if(ad == LOW)
{

lcd.clear();lcd.setCursor(0, 0);lcd.print("PERSON DRUNK");
  //upload1("STUDENT_1_PRESENT");delay(2000);
upload1(ad,ad);delay(2000);*/
 }

}

char bf2[50];
void upload1(unsigned char *chr,unsigned char *chr1)
{
```

```
lcd.clear();lcd.setCursor(0, 1);lcd.print("UPLOADING");
delay(2000);
serialFlush();
mySerial.println("AT+CIPSTART=4,\"TCP\",\"embeddedspot.top\",80");

delay(8000);
sprintf(buff,"GET
    http://embeddedspot.top/iot/storedata.php?name=iot305&s1=%u&s2=%s\r\n\r\n",chr,chr1);
serialFlush();
sprintf(bf2,"AT+CIPSEND=4,%u",strlen(buff));
mySerial.println(bf2);

delay(5000);



serialFlush();
mySerial.print(buff);

delay(2000);

mySerial.println("AT+CIPCLOSE");
lcd.setCursor(0, 1);lcd.print("UPLOADDED");

}
```

CHAPTER

6RESULTSANALYSI

S

6.1 RESULT:

Thishardwaresetupgivesabriefideaastohowtheproject“Alcohol detection and health monitoring system
usingIOT”works.Thebelowfigureshowsthehardwaresetupoftheproposedsystemandconstitutesofalc
oholsensorandblood pressuresensorforthedetectionofconcentration of alcohol level and monitors the blood pressure respectively.It alsoconstitutes of lcd display for display the information i.e., the values detected by the two sensorsandaesp8266 wi-fimodulefortransmitting thisvalues continuously totheonlineserver.

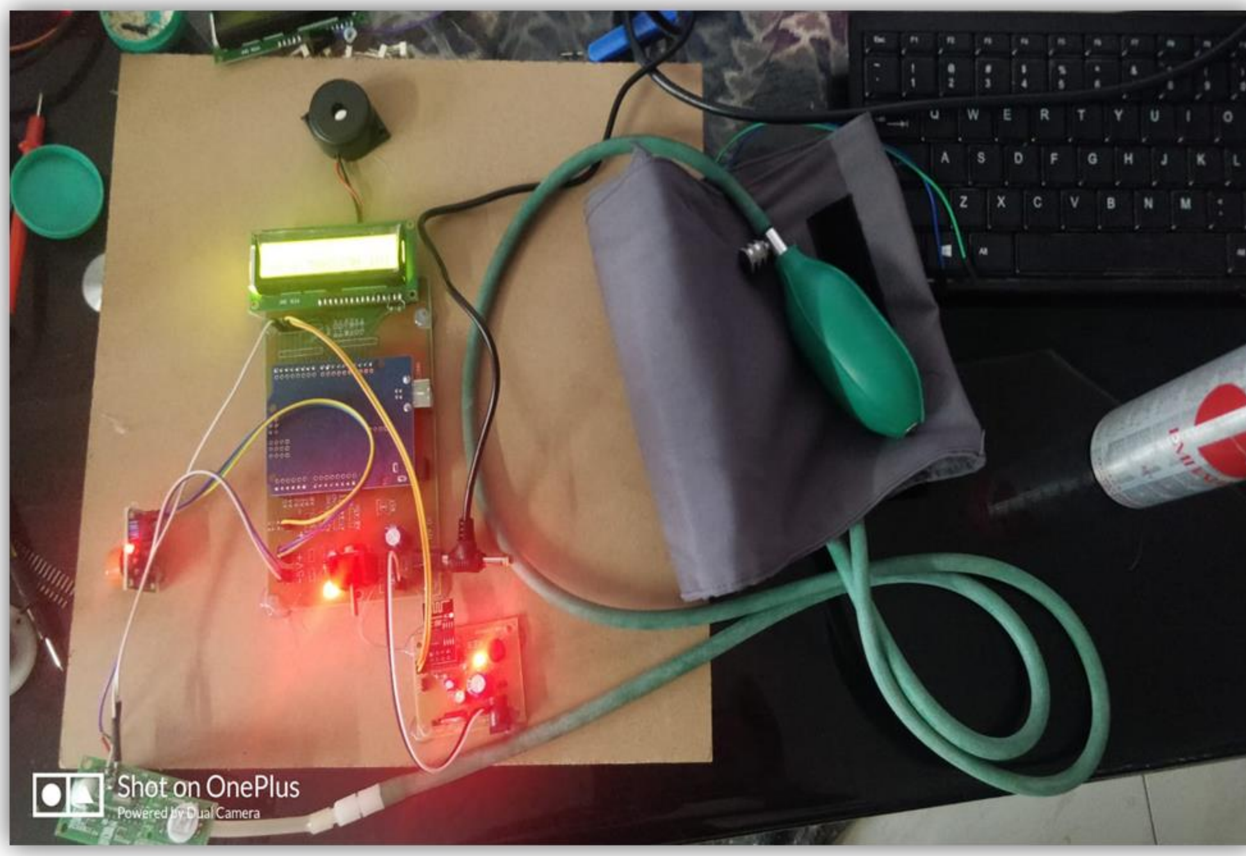


Fig 6.1:TheHardwaresetupoftheproposed system

6.1.1 Displayingnormalreadingvalues onthelcd

Gas sensor and sound sensor sense the concentration of harmful gases and level of intensity ofsound in the environment continuously and sends that information continuously to the Arduinouno. Now the Arduino uno which is brain of this project takes the information from the sensorsand transmits that values to the lcd for displaying them. The below figure represents lcd displaythat displaying normal readings of the sensors and this values are normal range of concentrationofharmful gasesand intensityofsoundthatdoesn't cause any harm tothelivingbeings.

Fig6.2:LCDdisplaythat displayingnormalrangevalues

6.1.2 Displayingofwarningmessages

Herethetwosensorsi.e.,gassensorandsoundsensorconsistsofpotentiometerusingthiswecansetthethreshold.Whenthereadingsfromthegassensorandsoundsensorexceedsthesetthresholditwilldisplaywarningmessagesonthelcd.Iftheconcentrationofharmfulgasesexceedsthe set thresholditwilldisplay“Dangergas”aswarningmessageonthelcdandifthelevelofsoundintensityexceedsthesetthresholditwilldisplay“Dangersound”warningmessageonthelcd.

Fig6.3:Displayingofwarningmessage“Danger gas”

Fig6.4:Displayingofwarningmessage“Dangersound”

6.1.3 Monitoringreadingsofthesensorin theonlineserver

Here the readings from the two sensors are continuously transmitted to the online server by theesp8266wi-fimoduleandwecanmonitorthose readingsbyusinganopenIOTplatformembeddedspot.topwebsite. Here inthiswebsiteallthereadings canbeviewed.

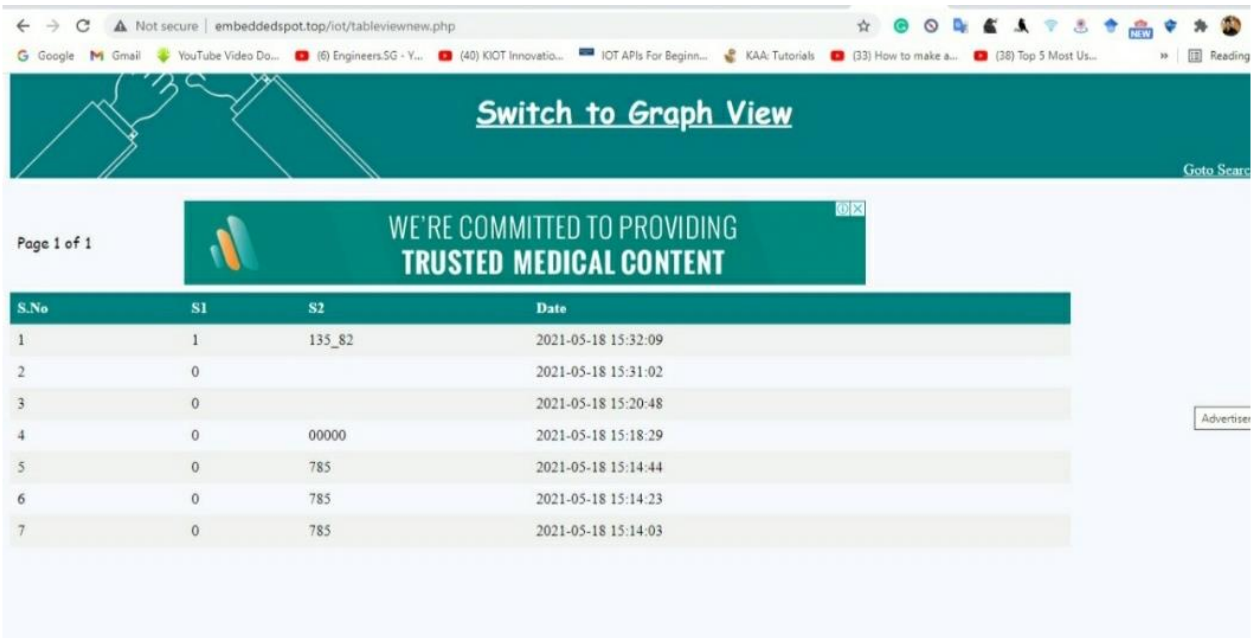


Fig6.5:Embeddedspot.topwebsite

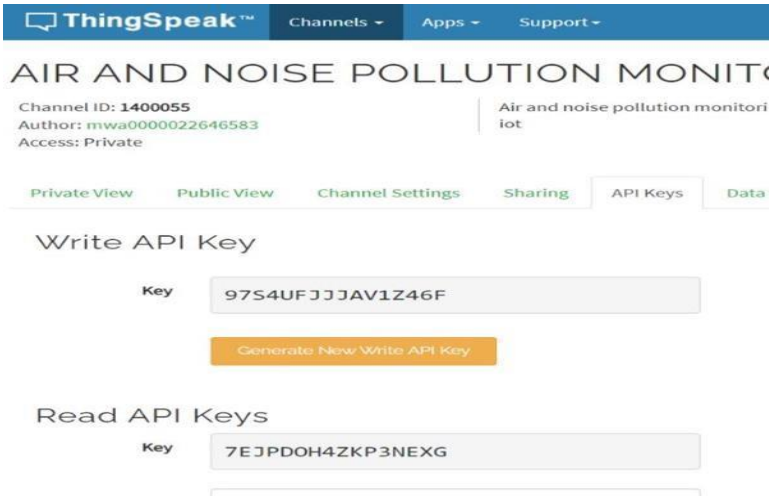


Fig6.6: APIkey

CHAPTER7

APPLICATIONSANDADVANTAGES

7.1 APPLICATIONSOFTHEPROJECT:

- IOT Healthcare is the most demanding field in the medical area.
- This project is for elderly people in our home. Also for the senior citizen living alone or living with 1 or 2 members.
- This project really proves helpful when family members need to go out for some emergency work.
- Disable patients can use this project. Disable patients who find it really difficult to go to doctors on a daily basis or for those patients who need continuous monitoring from the doctor.

7.2 ADVANTAGESOFTHEPROJECT:

- IOT Monitoring proves really helpful when we need to monitor & record and keep track of changes in the health parameters of a person over a period of time.
- So with the IOT health monitoring, we can have the database of these changes in the health parameters. Doctors can take the reference of these changes or the history of the patient while suggesting the treatment or the medicines to the patient.
- Hospital stays are minimized due to Remote Patient Monitoring.
- Hospital visits for normal routine checkups are Minimized.

CHAPTER-8

CONCLUSIONANDFUTUREENHANCEMENT

8.1 CONCLUSION:

The system developed patient monitoring based on Internet of things, is an alternative that can be used to help patients with chronic diseases. Likewise with this set of solutions the aim is to improve the quality of life of patients, not just monitoring them, but also to enable direct them to improve their eating habits and workout routines. The context model developed for the system proved to be efficient when making inferences related to the context, such as recommendations for taking measures through sensors, as well as recommendations and workout routines tips to improve the eating habits of patients.

8.2 FUTURESCOPE:

We can add a GPS module in IOT patient monitoring using the Arduino Uno and the Wi-Fi module project. This GPS module will find out the position or the location of the patient using the longitude and latitude received. Then it will send this location to the cloud that is the IOT using the Wi-Fi module. Then doctors can find out the position of the patient in case they have to take some preventive action.

- OMS, Overview - Preventing chronic diseases: a vital investment, http://www.who.int/chp/chronic_disease_report visited, November 2015.2. Who, J., & Consultation, F. E. Diet, nutrition and the prevention of chrepSer, 916(i- Swan, M. Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified self 2.0 Sensor and Actuator Networks, 1(3), 217-253, 2012

REFERENCE

- Strollo, S. E., Caserotti, P., Ward, R. E., Glynn, N. W., Goodpaster, B. H., &Strotm S. A review of the relationship between leg power and selected chronic dis older adults. The journal of nutrition, health & aging, 19(2), 240-248, 2015.
- Gómez, J., Huete, J. F., Hoyos, O., Perez, L., & Grigori, D. Interaction Systen on Internet of Things as Support for Education. Procedia Computer Science, 2 139, 2013
- National Intelligence Council. Disruptive Technologies Global Trends 20 Technologies with Potential Impacts on US Interests Out to 2025. 2008. A online: http://www.fas.org/irp/nic/disruptive.pdf (accessed on 19 November
- Feller G. Understanding the Three Basic Layers of the Internet of T BankinterFoundatin of Innovation. accessed Septembe http://www.fundacionbankinter.org/system/documents/8193/original/Chapt Understanding_the_three_basic_layers.pdf,2011.

A

PROJECT REPORT

On

EXTENSIVE CAPACITY SIMULATIONS OF MASSIVE MIMO CHANNELS FOR 5G MOBILE COMMUNICATION SYSTEM

Submitted by

1) Ms.S Ramya (17K81A04A7)

2)Mr.SVS Manoj Kumar (17K81A04B1)

3)Mr.S Aravind (17K81A04B2)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATIONS ENGINEERING

Under The Guidance of

Mr.G Upender

Associate Professor, M.Tech (Ph.D)

DEPARTMENT OF

ELECTRONICS AND COMMUNICATIONS ENGINEERING

ST.MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

St.MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the project entitled **EXTENSIVE CAPACITY SIMULATIONS OF MASSIVE MIMO CHANNELS FOR 5G MOBILE COMMUNICATION SYSTEM**, is being submitted by **1. Ms.S Ramya (17K81A04A7) 2. Mr.SVS Manoj Kumar (17K81A04B1) 3. Mr.S Aravind (17K81A04B2)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATIONS ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

	Head of the Department
Mr.G.Upender	Dr.B.Hari Krishna
Department of ECE	Department of ECE

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **S.ARAVIND** WITH ROLL NO.**17K81A04B2, S.RAMYA** WITH ROLL NO.**17K81A04A7**, **S.V.S.MANOJ KUMAR** WITH ROLL NO.**17K81A04B1**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**EXTENSIVE CAPACITY SIMULATIONS OF MASSIVE MIMO CHANNELS FOR 5G MOBILE COMMUNICATION SYSTEM**

" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. ★Kompally-500014.★

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS LASYA IT
SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of **Bachelor of Technology** in Department of Electronics and Communication Engineering, session 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled **EXTENSIVE CAPACITY SIMULATIONS OF MASSIVE MIMO CHANNELS FOR 5G MOBILE COMMUNICATION SYSTEM** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

1) Ms.S Ramya (17K81A04A7)
2) Mr.SVS Manoj Kumar (17K81A04B1)
3) Mr.S Aravind (17K81A04B2)

// ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would b incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr. B.Hari krishna**, **Head &Professor**, the Dept. of Electronics & Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our Dept .project coordinator Mr.G.Upender** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. S Ramya
2. SVS Manoj Kumar
3 S Aravind

Abstract

In this work, an analytic extension of massive MIMO capacity for 5G mobile communication systems is firstly developed. Then, the capacity limit and multiplexing gain are investigated in simulations for different massive MIMO configurations. Furthermore, the two calculated metrics are performed as a function of SNR taking into account the transceiver impairments. On the one hand, an analytic expression of the capacity under uncorrelated Rayleigh fading channel is developed as a function of SNR and OFDM subcarriers number. On the other hand, simulation results are carried out based on the deterministic and uncorrelated fading Rayleigh channel showing that the capacity limit is up bounded by [50-57] bit/s/Hz for a multiplexing gain equal to 256. Finally, three metrics are used to characterize the massive MIMO for the uncorrelated fading channel with a multiplexing gain of (4,4), (16,16), (64,64), (128,128) and (256,256), respectively. Likewise, a trade-off is observed between the capacity limit and the tolerable SNR value for massive MIMO transmission while increasing the multiplexing gain.

Keywords—Massive MIMO, 5G, signal detection, bit error rate, computational complexity.

INDEX

CHAPTER	LIST OF CONTENTS	PAGE NO.

CHAPTER	LIST OF CONTENTS	PAGE NO.

LIST OF FIGURES

LIST OF GRAPHS

LIST OF TABLES

CHAPTER-1

INTRODUCTION

Laptops, mobile data traffic has observed an exponential growth during the past few years, and this growth is expected in next few years as well. With the increase in the number of mobile users, not only the mobile traffic has increased but every user wants higher data rate with more accuracy and reliability. This considerable amount of mobile data traffic is challenging to manage with current technologies. Future generation network called fifth generation (5G) must accommodate this huge traffic and address the current limitation of data rates reliability, and efficiency. Currently, many technologies are trying to answer all these problems in mobile communication, but none have found an optimal solution. On the top of that, these technologies must take complexity, energy efficiency, reliability into account while designing the new system. Recently, a technology called massive MIMO has been proposed, which uses hundreds or even thousands of antennas at the base station, and it can serve tens of user simultaneously [1]. These thousands of antennas focus the transmission and reception of signal onto smaller region and help the system to achieve high diversity and multiplexing gains to improve reliability and increase data rate [2]. In massive MIMO systems, a user sends pilot towards the base station and based on these pilot signals base station estimates the channel between it and the user. The base station should have knowledge of channel during both uplink and downlink as massive MIMO is a technology that is dependent upon spatial

multiplexing [3]. There are several advantages of using these systems such as high spectral efficiency, antenna array gain, high reliability, robustness to internal jamming and interference and energy efficiency. Along with these advantages, Massive MIMO comes up with certain challenges as well, and one of the major challenges that massive MIMO is facing is high computational complexity and poor bit error rate (BER) performance during received signal detection at the base station which due to the higher number of antennas at the base station and more number of users. In Massive MIMO, all the signal transmitted by the user terminals superimpose at the base station and thus interfere with each other. There are several algorithms or methods for Massive MIMO detection and non- linear detectors like sphere detector, and successive interference cancellation detector are computationally very complex. Therefore, these methods are not recommended. Linear detectors are computationally less complex than non-linear detector, but the performance is much degraded. All the conventional detection methods like Maximum Likelihood detection (ML), Minimum mean square error detection

(MMSE), Zero-forcing detection (ZF) are not very efficient in terms of performance and complexity. During detection, these linear methods include inversion of high dimensional matrices which drastically increases the complexity of the system and this complexity increases exponentially with more number of antennas. [4]. In this paper, we present a low complex and efficient algorithm for detection of Massive MIMO systems which is based upon modified Approximate Message Passing(AMP) algorithm.

Wireless data traffic had been dramatically increasing over the past few ears. None the less, the existing techniques are not satisfying with the users' needs in terms of the emergence of applications for daily routines (e.g., proximity aware services). Therefore, there is a wave of popular interest to seek for new paradigms to deal with this problem. In the coming fifth generation (5G) cellular networks, emerging technologies will lead to both disruptive architectural and component design changes.For instance ,in 5G wireless communication systems, diverse researchers study different aspects of millimeter wave transmission, which are plentiful because spectrums have become scarce at microwave frequencies. Massive multiple-input multiple-output, which could increase the system throughput is proposed to utilize a very high number of antennas. We know that 2G–3G–4G cellular networks were built under the design premise of having complete control from the infrastructure side.However, this assumption should be dropped in the 5G systems. The base-station-centric architecture of cellular systems may change, and intelligence at the device side, within different layers of protocol stack, should be exploited, for example, by allowing device-todevice (D2D) connectivity.

Device-to-device communication defined as a direct communication between two mobile users without traversing the base station or the core network is considered to be a promising technique, which also offloads the increasing data traffic into user equipments (UEs). In a traditional cellular network, it is implicitly implied that two parties willing to establish the same call will not be in close proximity to each other. Therefore, all communications must go through the base station. However, in the age of data, mobile users in today's cellular networks are potentially in range for direct communications using high data rate services. Thus, D2D communication, which can decrease latency and increase resource utilization had been proposed as a means of taking advantage of the physical proximity of communicating devices. Figure 1 shows a simple example of D2D communication. The majority of the literatures in D2D communication proposed to use the cellular spectrum for both D2D and cellular communication. Most of these previous studies have focused on issues such as resource allocation and interference mitigation. Although, few existing studies have investigated the D2D access procedure. Here we review the literatures related to device discovery and access procedure. In TR 22.803, the D2D discovery is categorized into several types, which are summarized in. In addition, the D2D discovery procedure and long-term evolution (LTE)-based design are also discussed in. Yanget al. proposed a distributed peer discovery protocol for LTE-A networks.In, they provided an overview of the new agreements in third generation partners hipproject LTE radioaccess networks related to evaluation methodology and channel modeling for D2D discovery and communications. Hong et al. proposed a D2D discovery and link setup procedure and analyzed its performance in terms of energy consumption and delay by utilizing the measurement results of real LTE smartphones. However, all of the existing works lack the overall performance analysis based on the Markov process model. In this work, we will provide the system model based on the Markov process and present the performance analysis. Moreover,we give our proposal on theVienna Matlab platform, which is a system level Matlab simulator developed by Vienna University of Technology and obtain the simulation results.

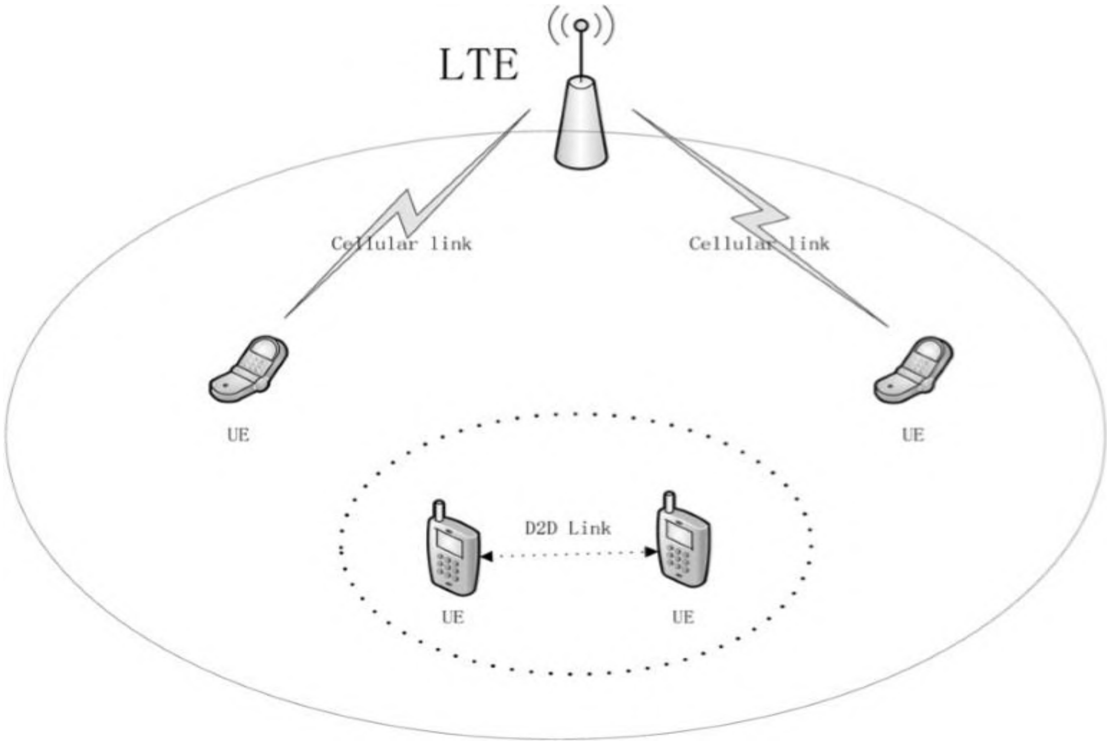


Figure 1.1. An example of device-to-device communication in a cellular network. LTE, long-term evolution.

CHAPTER-2

5G CELLUAR NETWORK

5G (from "5th Generation") is the latest generation of cellular mobile communications. It succeeds the 4G (LTE-A, WiMax), 3G (UMTS, LTE) and 2G (GSM) systems. 5G performance targets high data rate, reduced latency, energy saving, cost reduction, higher system capacity, and massive device connectivity. The first phase of 5G specifications in Release-15 will be completed by April 2019 to accommodate the early commercial deployment. The second phase in Release-16 is due to be completed by April 2020 for submission to the International Telecommunication Union (ITU) as a candidate of IMT-2020 technology.

The ITU IMT-2020 specification demands speeds up to 20 Gbit/s, achievable with wide channel bandwidths and massive MIMO. 3rd Generation Partnership Project (3GPP) is going to submit 5G NR (New Radio) as its 5G communication standard proposal. 5G NR can include lower frequencies (FR1), below 6 GHz, and higher frequencies (FR2), above 24 GHz and into the millimeter waves range. However, the speed and latency in early deployments, using 5G NR software on 4G hardware (non-standalone), are only slightly better than new 4G systems, estimated at 15% to 50% better. Simulation of standalone eMBB deployments showed improved throughput between 2.5×, in the FR1 range, and nearly 20×, in the FR2 range.

Overview

Like the earlier generation 2G, 3G, and 4G mobile networks, 5G networks are digital cellular networks, in which the service area covered by providers is divided into a mosaic of small geographical areas called *cells*. Analog signals representing sounds and images are digitized in the phone, converted by an analog to digital converter and transmitted as a stream of bits. All the 5G wireless devices in a cell communicate by radio waves with a local antenna array and low power automated transceiver (transmitter and receiver) in the cell, over frequency channels assigned by the transceiver from a common pool of frequencies, which are reused in geographically separated cells. The local antennas are connected with the telephone network and the Internet by a high bandwidth optical fiber or wireless backhaul connection. Like existing cellphones, when a user crosses from one cell to another, their mobile device is automatically "handed off" seamlessly to the antenna in the new cell.

Their major advantage is that 5G networks achieve much higher data rates than previous cellular networks, up to 10 Gbit/s; which is faster than current cable internet, and 100 times faster than the previous cellular technology, 4G LTE. Another advantage is lower network latency (faster response time), below 1 ms (millisecond), compared with 30 - 70 ms for 4G. Because of the higher data rates, 5G networks will serve not just cellphones but are also envisioned as a general home and office networking provider, competing with wired internet providers like cable. Previous cellular networks provided low data rate internet access suitable for cellphones, but a cell tower could not economically provide enough bandwidth to serve as a general internet provider for home computers.

5G networks achieve these higher data rates by using higher frequency radio waves, in or near the millimeter wave band from 30 to 300 GHz, whereas previous cellular networks used frequencies in the microwave band between 700 MHz and 3 GHz. A second lower frequency range in the microwave band, below 6 GHz, will be used by some 5G providers, but this will not have the high speeds of the new frequencies. Because of the more plentiful bandwidth at millimeter wave frequencies, 5G networks will use wider frequency channels to communicate with the wireless device, up to 400 MHz compared with 20 MHz in 4G LTE, which can transmit more data (bits) per second. OFDM (orthogonal frequency division multiplexing) modulation is used, in which multiple carrier waves are transmitted in the frequency channel, so multiple bits of information are being transferred simultaneously, in parallel.

Millimeter waves are absorbed by gases in the atmosphere and have shorter range than microwaves, therefore the cells are limited to smaller size; 5G cells will be the size of a city block, as opposed to the cells in previous cellular networks which could be many kilometers across. The waves also have trouble passing through building walls, requiring multiple antennas to cover a cell. Millimeter wave antennas are smaller than the large antennas used in previous cellular networks, only a few inches long, so instead of a cell tower 5G cells will be covered by many antennas mounted on telephone poles and buildings. Another technique used for increasing the data rate is massive MIMO (multiple-input multiple-output). Each cell will have multiple antennas communicating with the wireless device, each over a separate frequency channel, received by multiple antennas in the device, thus multiple bitstreams of data will be transmitted simultaneously, in parallel. In a technique called *beamforming* the base station computer will continuously calculate the best route for radio waves to reach each wireless device, and will organise multiple antennas to work together as phased arrays to create beams of millimeter waves to reach the device. The smaller, more numerous cells makes 5G network infrastructure more expensive to build per square kilometer of coverage than previous cellular networks. Deployment is currently limited to cities, where there will be enough users per cell to provide an adequate investment return, and there are doubts about whether this technology will ever reach rural areas.

The new 5G wireless devices also have 4G LTE capability, as the new networks use 4G for initially establishing the connection with the cell, as well as in locations where 5G access is not available.

The high data rate and low latency of 5G are envisioned as opening up new applications in the near future. One is practical virtual reality and augmented reality. Another is fast machine-to-machine interaction in the Internet of Things. For example, computers in vehicles on a road could continuously communicate with each other, and with the road, by 5G.

Performance targets

5G systems in line with IMT-2020 specifications are expected to provide enhanced device and network-level capabilities, tightly coupled with intended applications. The following eight parameters are key capabilities for IMT-2020 5G:

Capability	Description	5G target	Usage scenario
Peak data rate	Maximum achievable data rate	20 Gbit/s	eMBB
User experienced data rate	Achievable data rate across the coverage area (hotspot cases)	1 Gbit/s	eMBB
User experienced data rate	Achievable data rate across the coverage area	100 Mbit/s	eMBB
Latency	Radio network contribution to packet travel time	1 ms	URLLC
Mobility	Maximum speed for handoff and QoS requirements	500 km/h	eMBB/URLLC
Connection density	Total number of devices per unit area	10^6/km^2	MMTC
Energy efficiency	Data sent/received per unit energy consumption (by device or network)	Equal to 4G	eMBB
Spectrum efficiency	Throughput per unit wireless bandwidth and per network cell	3–4x 4G	eMBB
Area traffic capacity	Total traffic across coverage area	1000 (Mbit/s)/m^2	eMBB

Table 2.2.1 Performance Parameters of 5G network

Note that, for 5G NR, according to 3GPP specification when using spectrum below 6 GHz, the performance would be closer to 4G.

Usage scenario

ITU-R have defined three main types of usage scenario that the capability of 5G is expected to enable. They are Enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC), and Massive Machine Type Communications (mMTC).

Enhanced Mobile Broadband (eMBB)

Enhanced Mobile Broadband (eMBB) refers to the use case of using 5G as an evolution to 4G LTE mobile broadband services with faster connection with higher throughput and more capacity. 5G would need to deliver higher capacity, enhance connectivity, and higher user mobility to match these demands, which would require capabilities in the above table with eMBB mark to deliver.

Ultra Reliable Low Latency Communications (URLLC*)*

Ultra-Reliable Low-Latency Communications (URLLC) refers to the use case of using 5G in mission-critical applications such as factory automation, where uninterrupted and robust exchange of data is of the utmost importance.

Massive Machine Type Communications (mMTC)

Massive Machine-Type Communications (mMTC) refers to the wide area IoT use cases consisting of large numbers of low-cost devices with high requirements on scalability and increased battery lifetime.

Advantages

2.7.1.Speed

5G promises superior speeds in most conditions to the 4G network. Qualcomm presented a simulation at Mobile World Congress that predicts 490 Mbit/s median speeds for 3.5 GHz 5G Massive MIMO and 1.4 Gbit/s median speed for 28 GHz mmWave. 5G NR speed in sub-6 GHz bands can be slightly higher than the 4G with a similar amount of spectrum and antennas, though some 3GPP 5G networks will be slower than some advanced 4G networks, such as T-Mobile's LTE/LAA network, which achieves 500+ Mbit/s in Manhattan.

The 5G specification allows LAA (License Assisted Access) as well but it has not yet been demonstrated. Adding LAA to an existing 4G configuration can add hundreds of megabits per second to the speed, but this is an extension of 4G, not a new part of the 5G standard.

Low communication latency

Network latency is the time it takes to pass a message from sender to receiver. 5G will have much lower latency than previous cellular networks; below 1 millisecond, compared with 30 - 70 ms for 4G.

New use cases

Features of 5G network, including extreme high bandwidth, ultra low latency, and high density connections, are expected to enable many new use cases that are impossible to be done via older network standards. (See Usage scenario) 5G can also increase the effectiveness of ecommerce vendors' activities.

CHAPTER-3

Device-to-Device (D2D) communication

Device-to-Device (D2D) communication in cellular networks is defined as direct communication between two mobile users without traversing the Base Station (BS) or core network. D2D communication is generally non-transparent to the cellular network and it can occur on the cellular frequencies (i.e., inband) or unlicensed spectrum (i.e., outband).

In a traditional cellular network, all communications must go through the BS even if communicating parties are in range for proximity-based D2D communication. Communication through BS suits conventional low data rate mobile services such as voice call and text messaging in which users are seldom close enough for direct communication. However, mobile users in today's cellular networks use high data rate services (e.g., video sharing, gaming, proximity-aware social networking) in which they could potentially be in range for direct communications (i.e., D2D). Hence, D2D communications in such scenarios can greatly increase the spectral efficiency of the network. The advantages of D2D communications go beyond spectral efficiency; they can potentially improve throughput, energy efficiency, delay, and fairness.

Data delivery in non-cooperative D2D communication

Existing data delivery protocols in D2D communications mainly assume that mobile nodes willingly participate in data delivery, share their resources with each other, and follow the rules of underlying networking protocols. Nevertheless, rational nodes in real-world scenarios have strategic interactions and may act selfishly for various reasons (such as resource limitations, the lack of interest in data, or social preferences).

For example, if a node has limited battery resources or the cost of the network bandwidth delivered by mobile network operators is high, it would not willingly relay data for others until appropriate incentives are provided. Meanwhile, malicious nodes may attack the network in different ways to disturb the normal operation of the data transmission process. An adversary, for example, may drop received messages but produce forged routing metrics or false information with the aim of attracting more messages or decreasing its detection probability. This issue becomes more challenging when colluding attackers boost their metrics to deceive the attack detection systems. Dealing with non-cooperative mobile nodes is very challenging because of the distributed network model and intermittent access of nodes to central authorities.

D2D applications

D2D Communications is used for

1. Local Services: In local service, user data is directly transmitted directly between the terminals and doesn't involves network side, e.g. social media apps, which are based on proximity service.

2. Emergency communications: In case of natural disasters like hurricanes, earthquakes etc., traditional communication network may not work due to the damage caused. Ad-hoc network can be established via D2D which could be used for such communication in such situations.

3. IoT Enhancement: By combining D2D with Internet of things (IoT) , a truly interconnected wireless network will be created. Example of D2D-based IoT enhancement is vehicle-to-vehicle (V2V) communication in the Internet of Vehicles (IoV). When running at high speeds, a vehicle can warn nearby vehicles in D2D mode before it changes lanes or slows down.

CHAPTER-4

Device discovery

The players are adverting their network services and presence information through two protocols:

- Bonjour
- SSDP / UPnP (added in firmware 4.1.0 and Elementi 2018)

Elementi is using the information advertised by the players through any of these two protocols to list the players under Devices tab in Browse panel. If device discovery doesn't work for any reason, the players can be manually added into Elementi.

Both services are using multicast packets, which are restricted to the local area network where the players are located (thus the PC running Elementi must be in the same LAN), might get discarded on WiFi network (especially when crowded) and don't work over VPN connection.

Notes:

- On HMP350, HMP300 and DiVA players, both services can be disabled from Control Center > Network page, should the security policy requires that.
- Discovery across subnetworks is normally prohibited - to activate it, either enable multicast routing of SSDP on the site (applies to SpinetiX players with firmware 4.1.0 or later) or install an mDNS reflector (Avahi has this capability) on the gateway between the subnetworks.

SSDP / UPnP

- Added in firmware 4.1.0 and Elementi 2018.
- Universal Plug and Play (UPnP) is a set of networking protocols that permits networked devices, such as SpinetiX players, personal computers, printers, Internet gateways, Wi-Fi access points and mobile devices to seamlessly discover each other's presence on the network and establish functional network services for data sharing and communications. UPnP discovery protocol, known as Simple Service Discovery Protocol (SSDP), accomplishes the advertisement and discovery of network services and presence information without assistance of server-based configuration mechanisms, such as DHCP or DNS, and without special static configuration of a network host.
- SSDP / UPnP removes the dependency on Bonjour name resolution as IP addresses are used to contact players discovered via UPnP. Furthermore, HMP350, HMP300 and DiVA players show up in the Windows Explorer's Network view in Windows computers when Network Discovery is enabled.

- UPnP uses UDP port 1900 and all used TCP ports are derived from the SSDP alive and response messages.

  Bonjour

- Bonjour is Apple Inc's implementation of zero-configuration networking (zeroconf), a group of technologies that includes service discovery, address assignment, and hostname resolution. Bonjour locates devices such as SpinetiX players, printers, other computers, and the services that those devices offer on a local area network using multicast Domain Name System (mDNS) service records and DNS Service Discovery (DNS-SD).

Information advertised

- The HMP advertises the following services using Bonjour (via mDNS):

http._tcp on port 80 with name "HMP - _device name_";

- This is the standard service web browsers look for.

TXT records:

- path=/ (standard way to signify URL path)
- webdav._tcp on port 81 with name "HMP - _device name_"

This is the standard service WebDAV clients look for.

TXT records:

- path=/ (standard way to signify URL path)
- spx-hmp._tcp on port 80 with name "_device name_"

TXT records:

- txtvers=1 (standard way to support future changes)
- cport=81 (this is the content server port)
- mode= one of

normal when running normally

safe when started in safe mode

recovery when in recovery console

- serial=_serial number_ (e.g., 001d5000001a)
- firmware=_firmware version_ (e.g., 2.1.0-0.1.7844)
- model=_model_ (e.g. HMP100, HMP130 or HMP200)

spx-im._tcp on the port used by the network API (usually 1234)

This is only advertised if the network API is enabled

The device name is the name of the device as configured in Control Center.

Web browser

Discovery from a web browser is simple if a Bonjour finder is installed (native with Safari). For instance with Internet Explorer open the Bonjour tab, all the HMP devices in the local network will be listed as "HMP - _device name_".

If the Bonjour finder is not available then one can just type spx-hmp-_serial_number_.local. (e.g., spx-hmp-001d5000001a.local.) in the URL bar of the browser, this should open the device's Control Center.

Command line

If the Bonjour service is installed one can use the command line dns-sd tool to browse the local network.

To discover all HMP devices type (the device names will be listed under the Instance Name column)

```
dns-sd -B _spx-hmp._tcp
```

To look up a particular HMP device type (replace instance name by the name returned using the -B option). It will return the device address as spx-hmp-_serial_number_.local

```
dns-sd -L "instance name" _spx-hmp._tcp
```

To know the IP address of a device you can use ping like the following (it will show the IP address)

```
ping spx-hmp-_serial_number_.local.
```

CHAPTER-5

MARKOV PROCESS

A **Markov chain** is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event.

In probability theory and related fields, a **Markov process**, named after the Russian mathematician Andrey Markov, is a stochastic process that satisfies the Markov property (sometimes characterized as "memorylessness"). Roughly speaking, a process satisfies the Markov property if one can make predictions for the future of the process based solely on its present state just as well as one could knowing the process's full history, hence independently from such history; i.e., conditional on the present state of the system, its future and past states are independent.

A Markov chain is a type of Markov process that has either a discrete state space or a discrete index set (often representing time), but the precise definition of a Markov chain varies. For example, it is common to define a Markov chain as a Markov process in either discrete or continuous time with a countable state space (thus regardless of the nature of time), but it is also common to define a Markov chain as having discrete time in either countable or continuous state space (thus regardless of the state space).

Markov studied Markov processes in the early 20th century, publishing his first paper on the topic in 1906. Random walks based on integers and the gambler's ruinproblem are examples of Markov processes. Some variations of these processes were studied hundreds of years earlier in the context of independent variables. Two important examples of Markov processes are the Wiener process, also known as the Brownian motion process, and the Poisson process, which are considered the most important and central stochastic processes in the theory of stochastic processes, and were discovered repeatedly and independently, both before and after 1906, in various settings. These two processes are Markov processes in continuous time, while random walks on the integers and the gambler's ruin problem are examples of Markov processes in discrete time.

Markov chains have many applications as statistical models of real-world processes, such as studying cruise control systems in motor vehicles, queues or lines of customers arriving at an airport, exchange rates of currencies, storage systems such as dams, and population growths of certain animal species. The algorithm known as PageRank, which was originally proposed for the internet search engine Google, is based on a Markov process.

Markov processes are the basis for general stochastic simulation methods known as Markov chain Monte Carlo, which are used for simulating sampling from complex probability distributions, and have found extensive application in Bayesian statistics.

The adjective **Markovian** is used to describe something that is related to a Markov process.

A Markov chain is a stochastic process with the Markov property. The term "Markov chain" refers to the sequence of random variables such a process moves through, with the Markov property defining serial dependence only between adjacent periods (as in a "chain"). It can thus be used for describing systems that follow a chain of linked events, where what happens next depends only on the current state of the system.

The system's state space and time parameter index need to be specified. The following table gives an overview of the different instances of Markov processes for different levels of state space generality and for discrete time v. continuous time:

	Countable state space	Continuous or general state space
Discrete-time	(discrete-time) Markov chain on a countable or finite state space	Harris chain (Markov chain on a general state space)
Continuous-time	Continuous-time Markov process or Markov jump process	Any continuous stochastic process with the Markov property, e.g., the Wiener process

Table 5.1 Markov processes for different levels

Note that there is no definitive agreement in the literature on the use of some of the terms that signify special cases of Markov processes. Usually the term "Markov chain" is reserved for a process with a discrete set of times, i.e., a discrete-time Markov chain (DTMC), but a few authors use the term "Markov process" to refer to a continuous-time Markov chain (CTMC) without explicit mention. In addition, there are other extensions of Markov processes that are referred to as such but do not necessarily fall within any of these four categories (see Markov model). Moreover, the time index need not necessarily be real-valued; like with the state space, there are conceivable processes that move through index sets with other mathematical constructs. Notice that the general state space continuous-time Markov chain is general to such a degree that it has no designated term.

While the time parameter is usually discrete, the state space of a Markov chain does not have any generally agreed-on restrictions: the term may refer to a process on an arbitrary state space. However, many applications of Markov chains employ finite or countably infinite state spaces, which have a more straightforward statistical analysis. Besides time-index and state-space parameters, there are many other variations, extensions and generalizations (see Variations). For simplicity, most of this article concentrates on the discrete-time, discrete state-space case, unless mentioned otherwise.

The changes of state of the system are called transitions. The probabilities associated with various state changes are called transition probabilities. The process is characterized by a state space, a transition matrix describing the probabilities of particular transitions, and an initial state (or initial distribution) across the state space. By convention, we assume all possible states and transitions have been included in the definition of the process, so there is always a next state, and the process does not terminate.

A discrete-time random process involves a system which is in a certain state at each step, with the state changing randomly between steps. The steps are often thought of as moments in time, but they can equally well refer to physical distance or any other discrete measurement. Formally, the steps are the integers or natural numbers, and the random process is a mapping of these to states. The Markov property states that the conditional probability distribution for the system at the next step (and in fact at all future steps) depends only on the current state of the system, and not additionally on the state of the system at previous steps.

Since the system changes randomly, it is generally impossible to predict with certainty the state of a Markov chain at a given point in the future. However, the statistical properties of the system's future can be predicted. In many applications, it is these statistical properties that are important.

A famous Markov chain is the so-called "drunkard's walk", a random walk on the number line where, at each step, the position may change by +1 or −1 with equal probability. From any position there are two possible transitions, to the next or previous integer. The transition probabilities depend only on the current position, not on the manner in which the position was reached. For example, the transition probabilities from 5 to 4 and 5 to 6 are both 0.5, and all other transition probabilities from 5 are 0. These probabilities are independent of whether the system was previously in 4 or 6.

Another example is the dietary habits of a creature who eats only grapes, cheese, or lettuce, and whose dietary habits conform to the following rules:

- It eats exactly once a day.
- If it ate cheese today, tomorrow it will eat lettuce or grapes with equal probability.
- If it ate grapes today, tomorrow it will eat grapes with probability 1/10, cheese with probability 4/10 and lettuce with probability 5/10.
- If it ate lettuce today, tomorrow it will eat grapes with probability 4/10 or cheese with probability 6/10. It will not eat lettuce again tomorrow.

This creature's eating habits can be modeled with a Markov chain since its choice tomorrow depends solely on what it ate today, not what it ate yesterday or any other time in the past. One statistical property that could be calculated is the expected percentage, over a long period, of the days on which the creature will eat grapes.

A series of independent events (for example, a series of coin flips) satisfies the formal definition of a Markov chain. However, the theory is usually applied only when the probability distribution of the next step depends non-trivially on the current state.

Markov property

The Markov property refers to the memoryless property of a stochastic process.

A stochastic process has the Markov property if the conditional probability distribution of future states of the process depends only upon the present state, not on the sequence of events that preceded it.

FORMAL DEFINITION

Discrete-time Markov chain

A discrete-time Markov chain is a sequence of random variables X_1, X_2, X_3, ... with the Markov property, namely that the probability of moving to the next state depends only on the present state and not on the previous states

$$\Pr(X_{n+1} = x \mid X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = \Pr(X_{n+1} = x \mid X_n = x_n)$$
$$\Pr(X_1 = x_1, \ldots, X_n = x_n) > 0\,.$$

The possible values of X_i form a countable set S called the **state space** of the chain.

Markov chains are often described by a sequence of directed graphs, where the edges of graph n are labeled by the probabilities of going from one state at time n to the other states at time $n + 1$, $\Pr(X_{n+1} = x \mid X_n = x_n)$. The same information is represented by the transition matrix from time n to time $n + 1$. However, Markov chains are frequently assumed to be time-homogeneous (see variations below), in which case the graph and matrix are independent of n and are thus not presented as sequences.

Continuous-time Markov chain

A continuous-time Markov chain $(X_t)_{t \geq 0}$ is defined by a finite or countable state space S, a transition rate matrix Q with dimensions equal to that of the state space and initial probability distribution defined on the state space. For $i \neq j$, the elements q_{ij} are non-negative and describe the rate of the process transitions from state i to state j. The elements $q_{ii}$are chosen such that each row of the transition rate matrix sums to zero, while the row-sums of a probability transition matrix in a (discrete) Markov chain are all equal to one.

CHAPTER-6

ACCESS PROCEDURE

5G NR Initial Access Procedure between UE and gNB including beam management. All the messages exchanged between UE and gNB in 5G NR Initial Access procedure i.e. Random Access Procedure are described. It is also known as initial cell search procedure.

5G NR (New Radio) is the latest cellular wireless technology developed to deliver 10 times fast data rate compare to LTE (i.e. 4G) technology. It follows 3GPP specifications release 15 and above. In this page we will understand initial access procedure i.e. random access procedure. It helps to get the initial uplink grant for UE and helps in performing synchronization with the gNB (i.e. network). It covers Random Access procedure initialization, Random Access Resource selection, Random Access Preamble transmission, Random Access Response reception, Contention Resolution and Completion of the Random Access procedure.

6.1 Beam Management in 5G NR

Beam management procedure is used in 5G NR in order to acquire and maintain a set of TRxP (s) and/or UE beams which can be used for DL and UL transmission/reception. TRxP stands for Transmission Reception Point. Reference signals are used for beam management. In IDLE mode, 5G NR uses PSS, SSS and PBCH DMRS where as in CONNECTED mode it uses CSI-RS (in the downlink) and SRS (in the uplink).

Following operations are performed in 5G NR beam management procedure. They are applied to both the modes viz. SA (StandAlone) and NSA (Non StandAlone).

- Beam Sweeping: It refers to the covering a spatial area with a set of beams transmitted and received according to pre-specified intervals and directions.
- Beam measurement: It refers to evaluation of the quality of the received signal at the gNB or at the UE. Different metrics could be used such as RSRP, RSRQ and SINR or SNR for this purpose.
- Beam determination: It refers to the selection of the suitable beam or beams either at the gNB or at the UE, according to the measurements obtained with the beam measurement procedure.

• Beam reporting: It refers to the procedure used by the UE to send beam quality and beam decision information to the Radio Access Network (RAN).

Random Access Procedures are of two types viz. contention based and contention free. The messages exchanged in these procedures are depicted in the following figure-2.

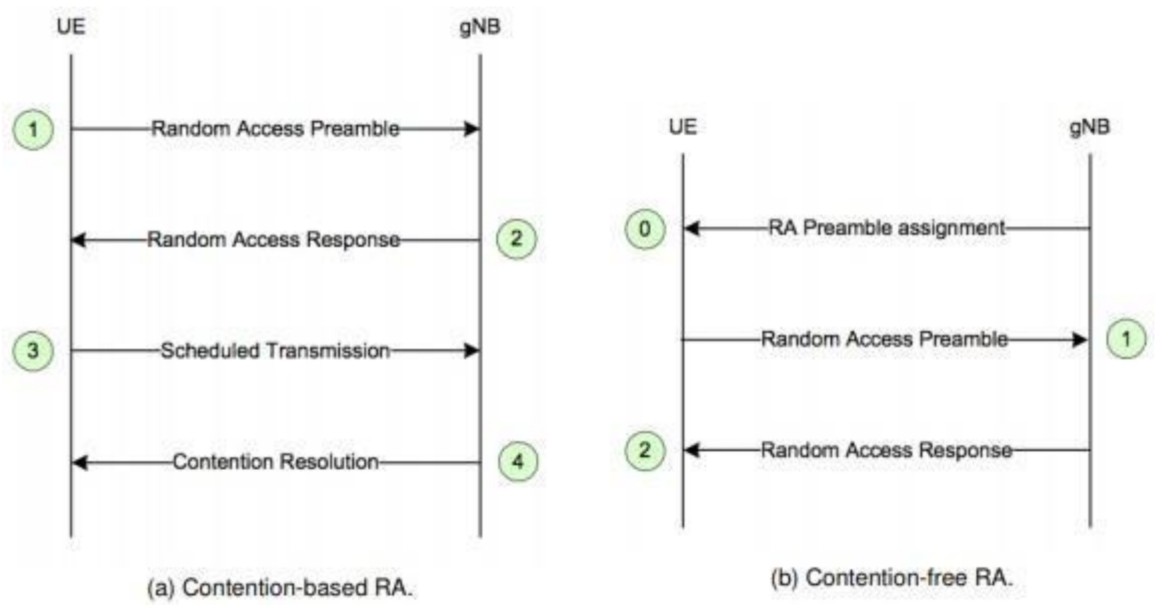


Figure 6.1.1 Random Access Procedures

Contention based RA is applied when UE is not yet synchronized or lost its synchronizaton. Contention free is appled when UE was perviously synchronized to another gNB. Both the precudures rely on transmission of random access preamble from UE to gNB. The preamble is transmitted on specific time/frequency resources which are indicated by gNB to UEs on control channels.

5G NR Initial Access Procedure between UE and gNB

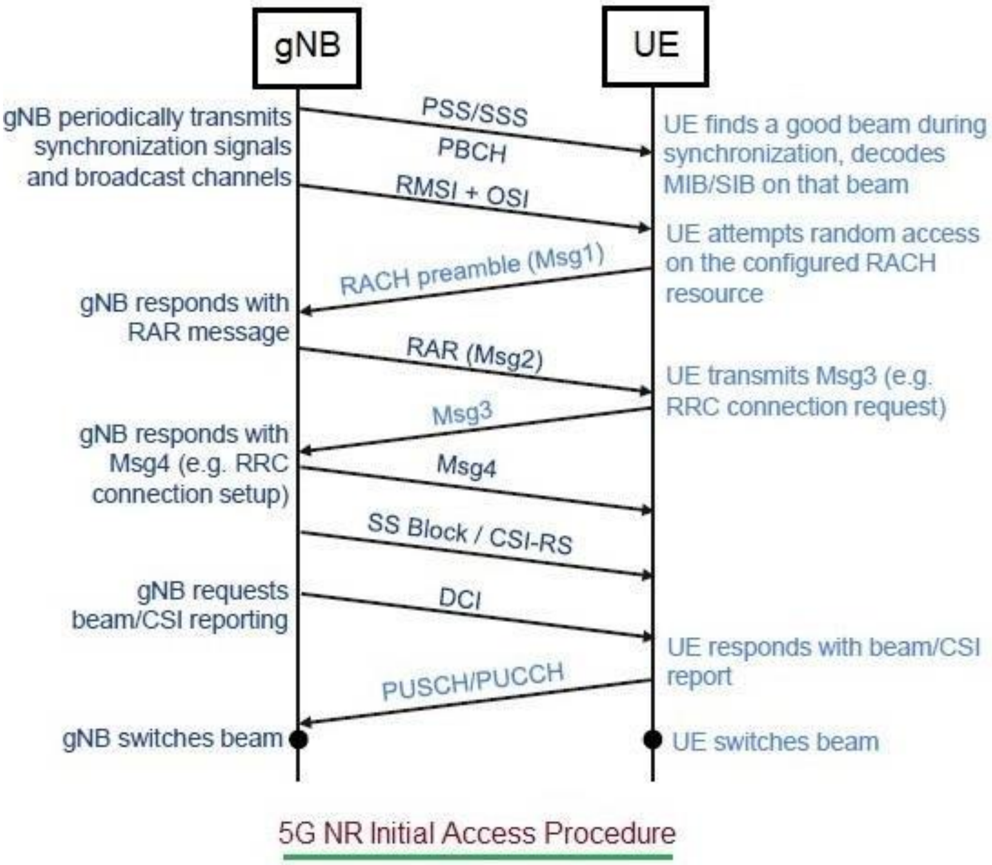


Figure 6.1.2 5G NR Initial Access Procedure

The following steps describe the messages exchanged between the UE and gNB during initial access procedure or random access procedure.

➤As shown in the figure-3, gNB (i.e. 5G NR Base Station) periodically transmits SS blocks carrying synchronization signals (PSS, SSS) and broadcast channels (PBCH) using beam sweeping. One SS block contains 1 symbol PSS, 1 symbol SS and 2 symbols PBCH. SS burst carry one or multiple SS blocks. Both PSS and SSS combination help to identify about 1008 physical-cell identities.

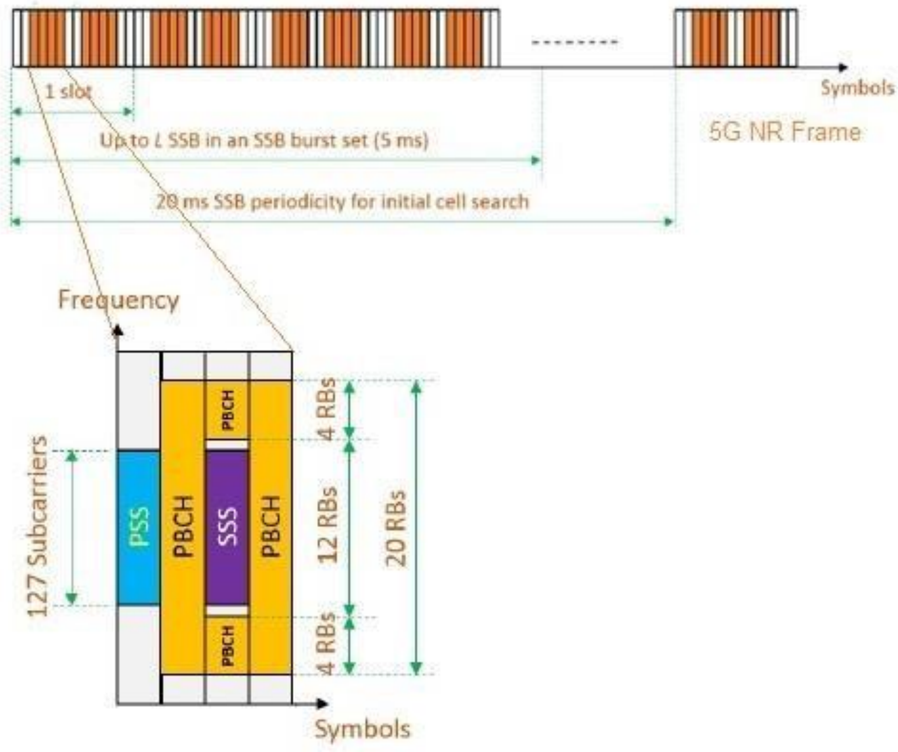


Figure 6.1.3 5G NR Frame

➤UE does beam measurements and determines best beam during sychronization. Consecutively UE decodes 5G NR system informations viz. MIB/SIB on that beam. Minimum SI (System Informations) is carried onto PBCH channel. The rest of the Remaining Minimum System Information (RMSI) is carried on PDSCH. The numerology used for RMSI is indicated in PBCH payload. CORESET is dedicated for RMSI scheduling. CORESET is not confined within PBCH Bandwidth. There is an RMSI PDCCH monitoring window associated with SS/PBCH block, which recurs periodically. OSI (Other System Information) contains on-demand the system information delivery. OSI is carried on PDSCH using same numerology as used for RMSI.

➤UE uses same beam and attempts random access by transmitting RACH preamble (i.e. Message#1) on the configured RACH resource. The gNB responds with RAR ("RA Response") message.

This is Message#2.

➤ The UE transmits Message#3 (i.e. RRC Connection Request). The gNB responds with Message#4(i.e.RRC-Connection-Setup).

➤As shown UE receives SS block and CSI-RS which it uses for generation of beam/CSI report.

➤The gNB requests beam/CSI reporting with the help of DCI (Downlink Control Information). UE responds with beam/CSI report as requested by gNB on PUSCH/PUCCH.

➤ Once the random access procedure is completed, dedicated connection is established between UE and gNB with dedicated connection ID.

➤Either gNB or UE switches beam based on SNR (Signal to Noise Ratio).

CHAPTER-7

BEAM-MANAGEMENT

Even though 3GPP would not preclude the use of Sub 6 GHz deployment of 5G(NR), at least based on the current status it seems that most of the deployment would be in very high frequency (milli meter wave) and this high frequency deployment would be one of the most important characteristics of 5G (NR).

Why we need Beam ?

Mostly by Nature of the wave (by Physics), when we use low and mid range of frequency, we can transmit a signal in all direction (as in (A)) or relatively wide angles (as in (B)). However, when we use very high frequency, we would not have much choice except using a huge antenna array. As a result of using this kind of huge antenna array, the resulting radiation would be a beam as in (C). Refer to Why Massive MIMO page for the details of this background.

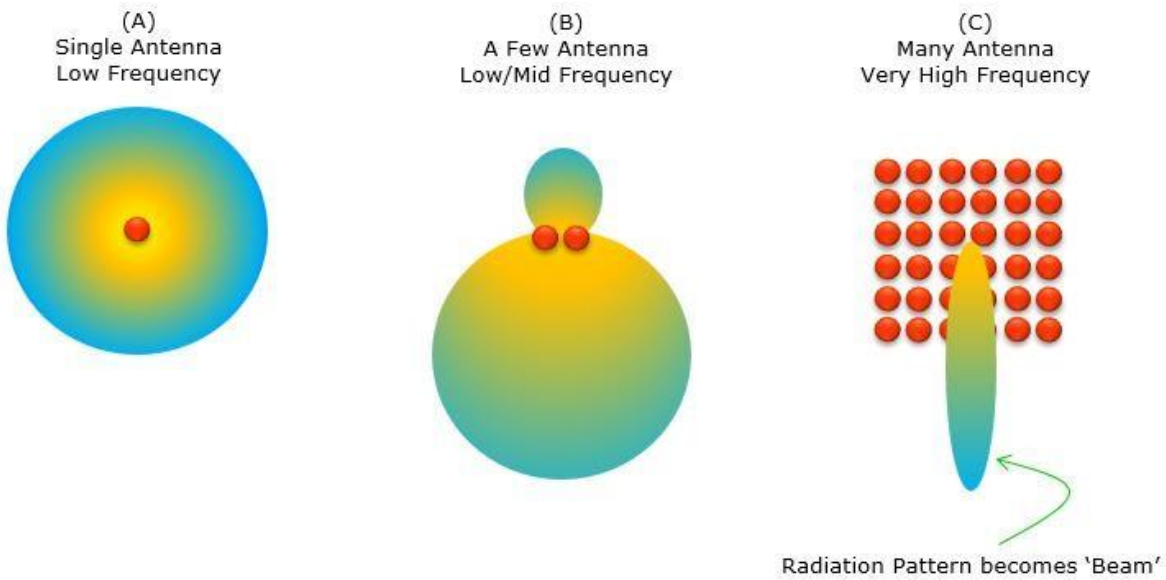


Figure 7.1 Radiation pattern of frequencies

Why Beam Management / Beam Control ?

I don't think transmitting signal in Beam in high frequency deployment would be the matter of choice. It is a kind of 'MUST' implementation. In case of low / mid frequency region without using massive antenna array (as in (A) / (B)), a single transmission would cover a lot of UEs simultaneously. However, when the radiation become beam-shaped as (C), it is very difficult to cover multiple UEs in single transmission unless those multiple UEs are located in very close proximity. To handle this problem, we need a very sophisticated idea of managing/controlling the beam to cover the multiple devices scattered in all directions and the management/control mechanism should be different depending on the situations . All of these collection of idea would fall into the title of "Beam Management" in the specification.

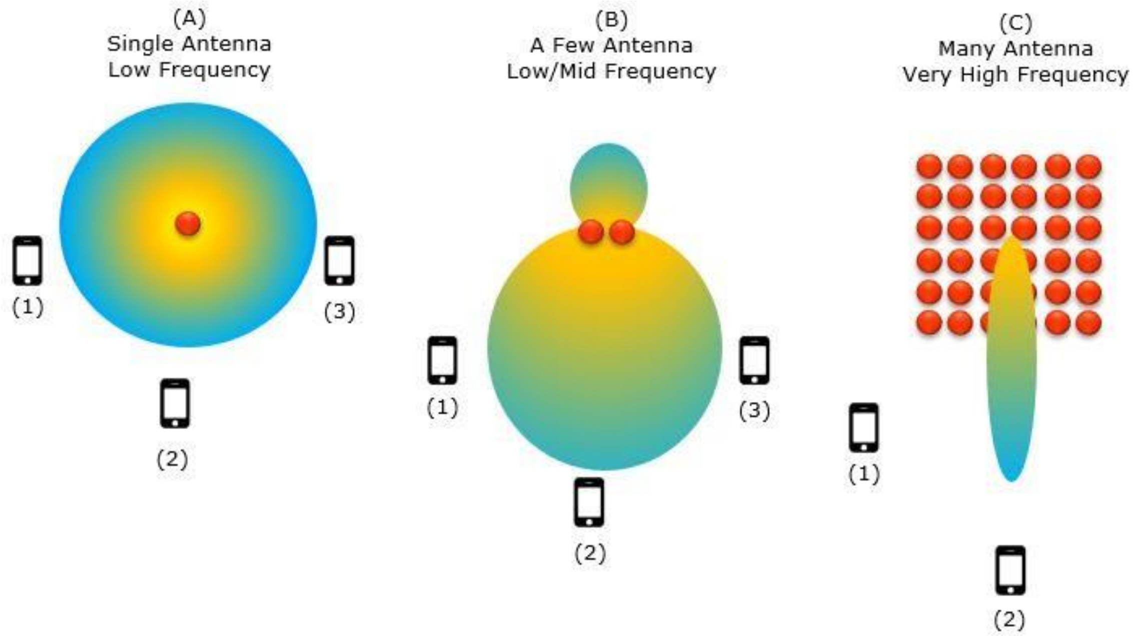


Figure 7.1.2 Beam Management/Control

Beam Management/Control for each specific situation will be described in separate pages with relavent situation (like Beam Management during Synchronization, Beam Management during Initial Attach, Beam Management in connected status etc). In thise page, I would describe on general idea.

Beam Management/Control when a transmitter has no information on the location of the reciever

Now let's look into a more specific cases where the Beam Management/Control become crucial. As an example, let's think of following case. There is a Base Station with Massive MIMO operating at the very high frequency. There is a UE around the Base Station and you are just about to turning on the UE. Once the UE is turned on, it would start Synchronization process. For this step, the Base Station would transmit the special signal called Synchronization

Signal and the signal should be able to reach to every UEs around the base station. However, here comes a serious problem with the base station sending signal in Beam. It is the fact that the signal beam can point to a very narrow area and it cannot cover a very wide area at the same time. Simply put, now you have the following question.

Figure 7.1.3 Base Station and Signal reach

What would be the answer for this ? If everything works as you draw in power point, you may draw a solution as follows. You may want to transmit a lot of beams in all direction simultaneously. Looks good ? Looks like a flower :).

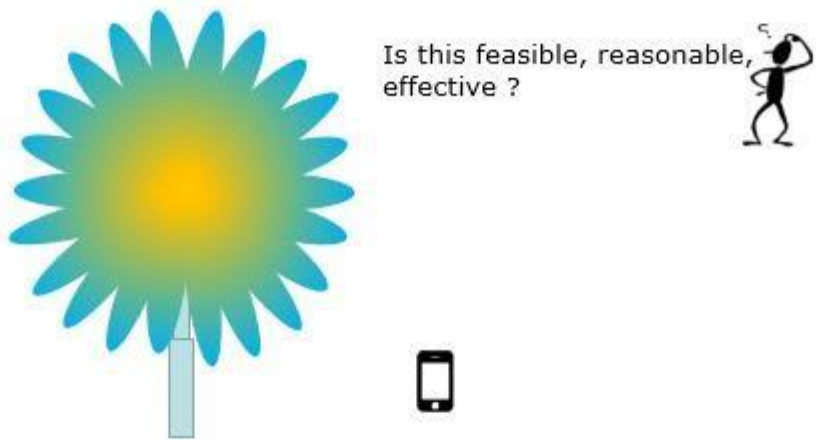


Figure 7.1.4 Beam Capability

Would the solution above be feasible, reasonable and effective ? The simple answer is NO (I would not explain why. You may easily guess why).

Then what can be another idea (possible soultion) for the problem ? There can be multiple ideas and proposals, but the most popular proposal as of now seems to be that the base station transmit the beam to a specific direction at a specific time and then change the direction a little bit in a next time frame and so on until it can scan all the area it should cover.

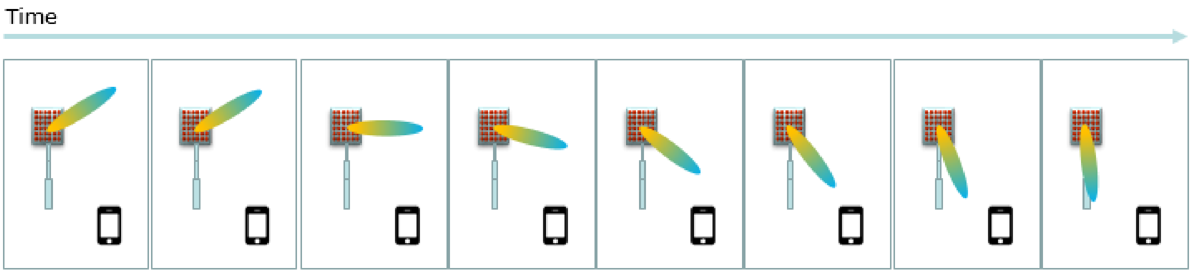


Figure 7.1.5 Beam Directions at different times

Then, the next question would be "how to reflect / implement this concept in the radio frame design ?". I would not go too much detail on this until this is explicitely determined in 3GPP TS (Technical Specification) document, but you can get the general ideas on various options /proposals from TDocs listed in Reference Section.

// Now that 3GPP Technical Specification on Beam Management has been released and I could write on this mechanism based on the formal specification. You may jump to NR Beam Management in a Nutshell section and read from there if you are interested in the formal specification.

Beam Management/Control when the connection is already established

Now let's talk about more serious case of Beam Management. In terms of 3GPP TDocs, Beam Management handles mostly with this topic (Beam Management during the connected states) and the one mentioned in previous section are described as a part of the topic Cell Search / Initial Access.

Once UE gets into a connection states with a Network, at least one beam (or multiple beam) is properly in connection between UE and the network. Theretically there can be so many different ways in which UE and Network beam is connected, but we can reduce it down to roughly four differences case as shown below.

In case 1, UE and Network is connected through a single TRP (Tx/Rx Point) and a single beam.
In case 2, UE and Network is connected through multiple TRP (Tx/Rx Point) and a single beam for each TRP.
In case 3, UE and Network is connected through a single TRP (Tx/Rx Point) and multiple beams
In case 4, UE and Network is connected through multipe TRP (Tx/Rx Point) and multiple beams for each TRP

You may think of many other cases and ask "How about this case ? How about that case ?". Whatever you think of and whatever you are asking, I think all of those would be valid thinking and valid question until 3GPP reach a explicit conclusion. So keep asking and try to find your own answers until you see the explicit 3GPP specification.

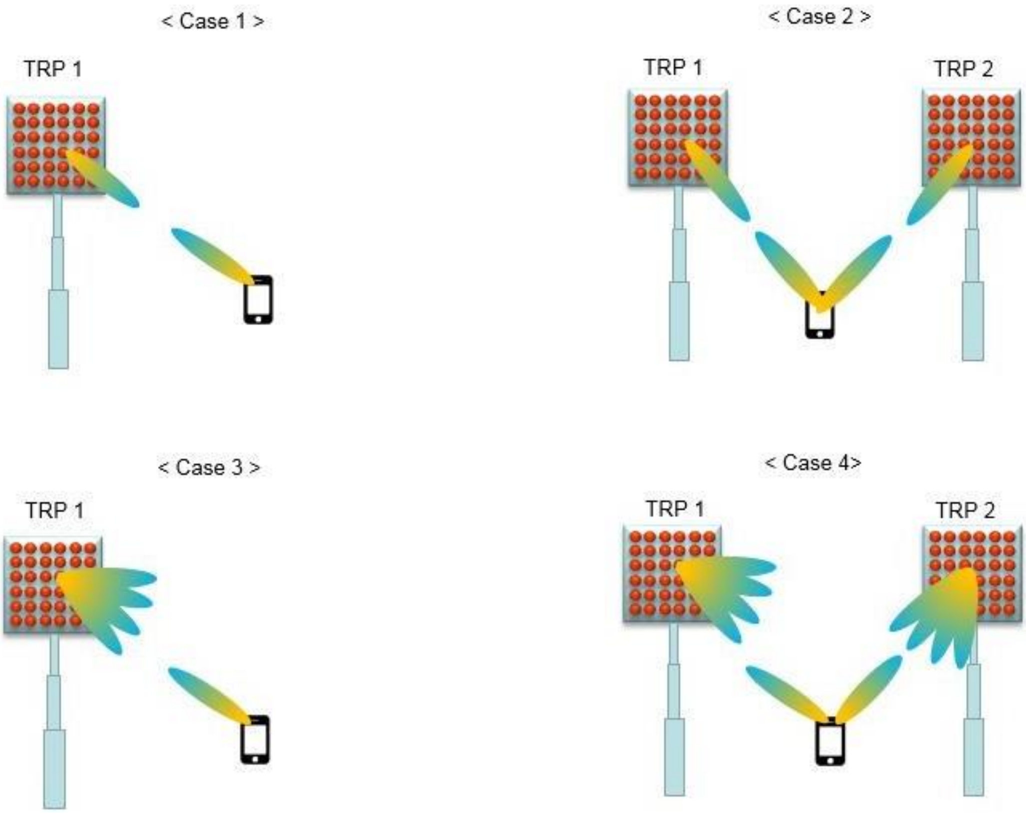


Figure 7.1.6 Beam network in different cases

Now the important and tough task is to maintain the connection. For this, I would not write much details until 3GPP specifies it in detail. Sorry for skipping too much with the execuse of

3GPP specification unavailability :). But I don't want to write many things now and rewrite too much after 3GPP Technical Specification come out. For now, my purpose is to give you very broad and general idea, and let you know what you may need to study in further details if you are really interested in technical details. For this purpose, I will list up most of TDocs about each topics in Reference section, so that you can get more detailed idea proposed by many companies / organizations in the industry.

The general idea of the beam management during the connected states would be

i) Network transmit a specific reference signal for beam management

ii) UE detect the signal and perform some measurement and send feedback to Network

As you may notice, the general idea would be very similar to CSI report mechanism that are currently used in current LTE. However, a lot of details are yet to be determined. For example,

i) Baseband Signal (Symbol) generation formula

ii) Resource Allocation mapping (How to allocate these reference symbols to which specific resource element)

iii) How often UE need to perform these measurement

iv) How UE report the measurement result ? (via RRC messages ? or via MAC / PHY layer transactions ?)

NOTE : Now that the technical specification on all of the questions listed above has been released and I wrote a few separate pages regarding this topic. Refer to CSI RS signal page and CSI Report page for further details based on 3GPP technical specification.

NR Beam Management in a Nutshell

If you ask me to explain about NR Beam Management in a few seconds, I would summarize the whole process in an illustration as shown below. As you see here, Beam Management plays important role in two period - During RACH procedure and After the call connection.

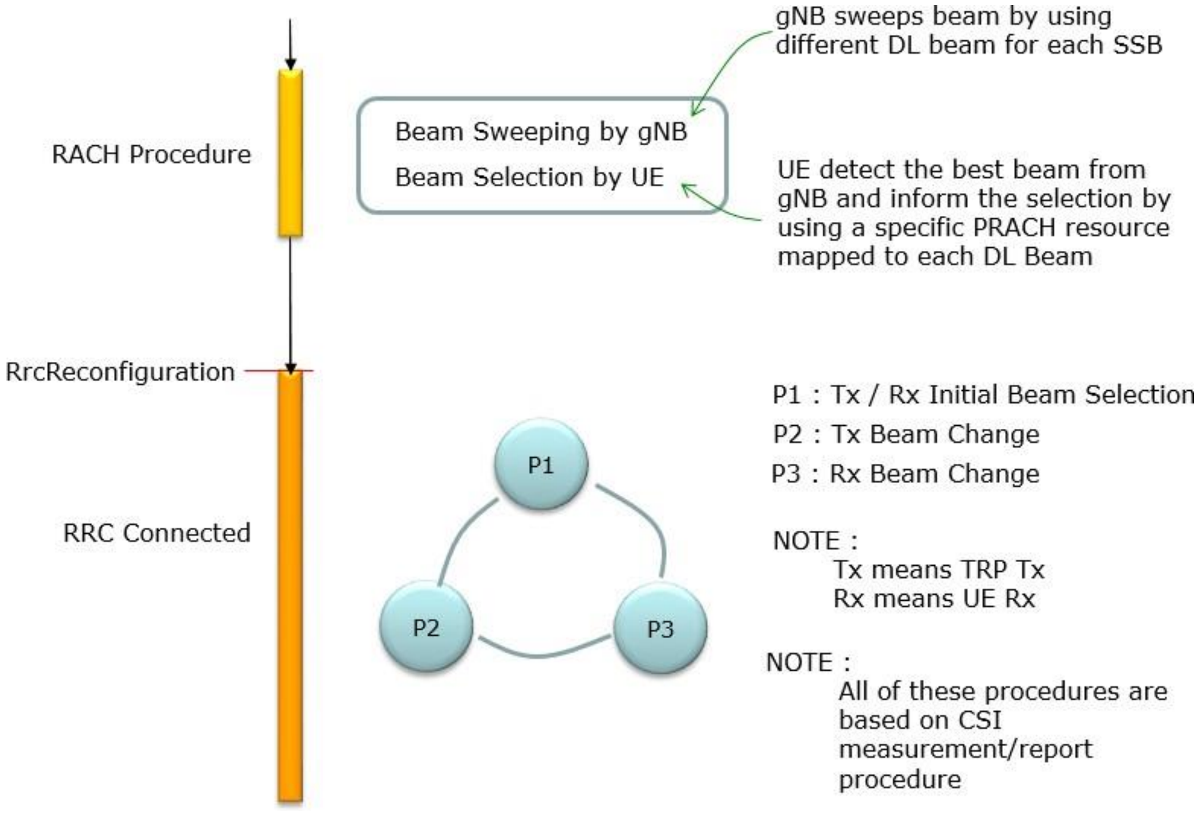


Figure 7.1.7 Beam Configuration

Where to point my beam ?

We can think of this question in terms of two different cases : transmitting case and receiving case.

When transmitting,

Now let's think of which direction a gNB or UE has to point its beam when they try to transmit the signal ?
The answer is simple. They (gNB or UE) has to transmit the signal in the direction that can reach the reciever with the best signal quality'.

Then you would have another question. How can they(gNB or UE) figure out which direction is the one that can reach the reciever with the best signal quality ?
Now the answer would be a little bit trickier, but the big picture is as follows.

- When gNB is transmitting, gNB figure out this direction by evaluating the quality of a specific reference signal of multiple beam from UE. gNB evaluate the quality of the reference signal from each of the multiple beam and chose the best one. The reference signal from UE is called SRS.
- When UE is transmitting, UE figure out this direction by evaluating the quality of a specific reference signal of multiple beam from gNB. UE evaluate the quality of the reference signal from each of the multiple beam and chose the best one. The reference signal from gNB in this case can vary depending on situation. Sometimes it can be SSB and sometimes it can be CSI-RS. (NOTE : CSI-RS play many different roles in addition to beam management and very complex topic. Refer to CSI-RS signal generation and CSI report page for further details).

  NOTE : This kind of estimation of reference signal quality should be done sometime before they transmit signal.

3GPP TR 38.802 (V14.2.0)-6.1.6.1 describes on this situation as follows :

- TRP is able to determine a TRP Tx beam for the downlink transmission based on TRP’s uplink measurement on TRP’s one or more Rx beams
- UE is able to determine a UE Tx beam for the uplink transmission based on UE’s downlink measurement on UE’s one or more Rx beams

  NOTE : TRP is a transmission point of a gNB.

When recieving,

Now let's think of which direction a gNB or UE has to point its beam when they try to recieve signal ? (In this case, the word 'beam' may be a little misleading because the reciever does not form any real beam. So it would be better to change the phrase 'to point its beam' to 'to tune its reciever to a certain direction').

The answer is simple. They (gNB or UE) has to tune their reciever in the direction in which they can receive the signal from the transmitter with best quality.

Then you would have another question. How can they(gNB or UE) figure out which direction is the one in which they can receive the signal from the transmitter with best quality ?

Overall logic is as follows :

- When gNB receiving signal from UE, (before doing this) gNB is supposed to get the information of the best direction from UE in the form of CSI report.
- When UE receiving signal from gNB, (before doing this) UE is supposed to get the information of the best direction from gNB (gNB has detected the best direction based on the measurement of SRS signal quality of multiple beams from UE and indicates the UE of the best direction).

3GPP TR 38.802 (V14.2.0)-6.1.6.1 describes on this situation as follows :

- TRP is able to determine a TRP Rx beam for the uplink reception based on UE’s downlink measurement on TRP’s one or more Tx beams.
- UE is able to determine a UE Rx beam for the downlink reception based on TRP’s indication based on uplink measurement on UE’s one or more Tx beams.

P1, P2, P3 - What is It ?

As illustrated in Beam Management in a Nutshell, P1/P2/P3 are a set of processes that are designed for beam management while in connected state.

I will talk about a high level view on these processes in this section. For further details, you would need to understand the very details on CSI report for Beam Management which is another huge topic and will be described in a separate page here.

According to 38.802-6.1.6.1, P1/P2/P3 are stated as follows.

P-1*: is used to enable UE measurement on different TRP Tx beams to support selection of TRP Tx beams/UE Rx beam(s). For beamforming at TRP, it typically includes a intra/inter-TRP Tx beam sweep from a set of different beams. For beamforming at UE, it typically includes a UE Rx beam sweep from a set of different beams.*

Does this make sense to you ? It may take time to get clear understanding on this. Following illustration is my understanding/interpretation of this statement.

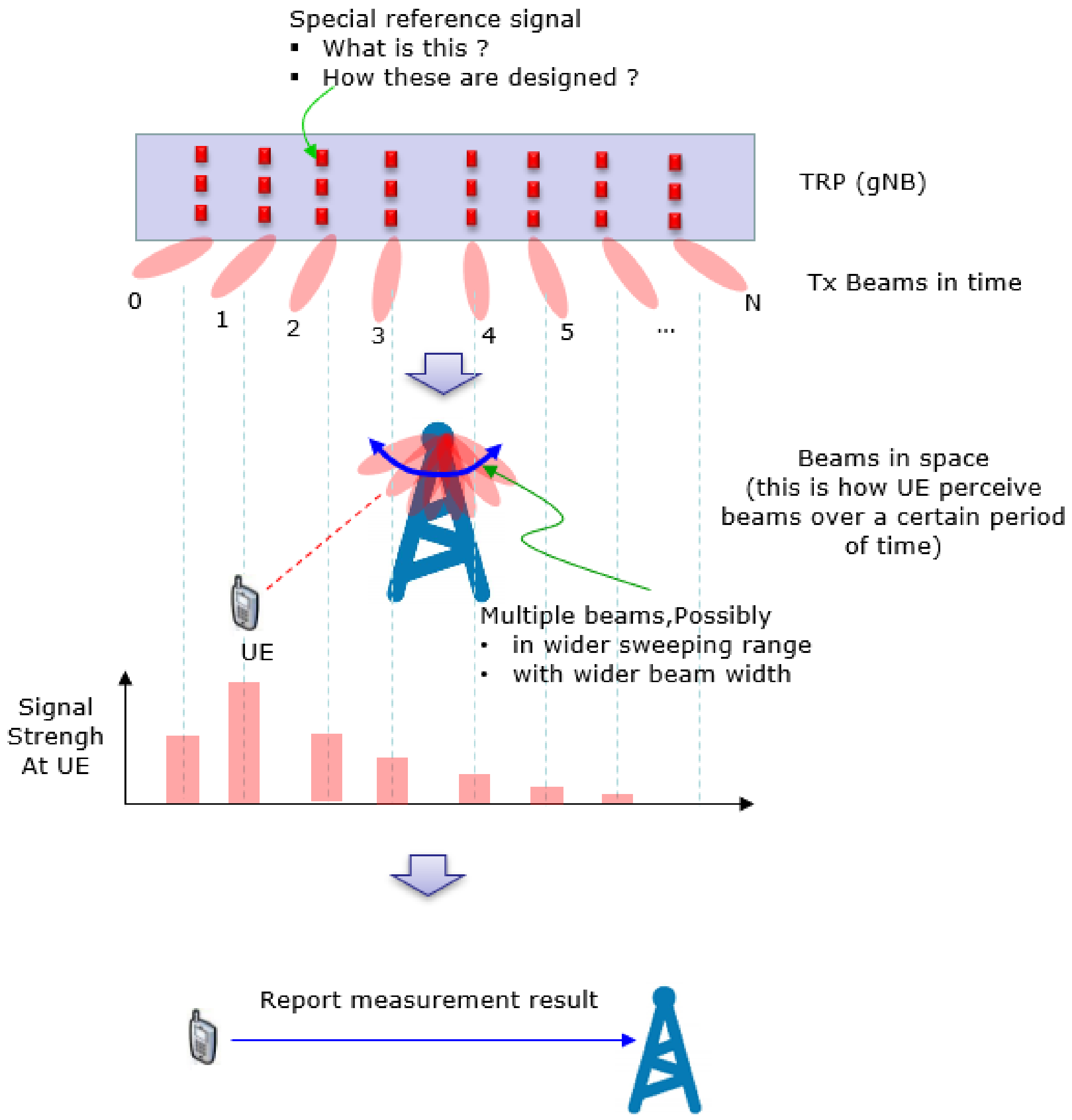


Figure 7.1.8 Beam Process 1

P-2: *is used to enable UE measurement on different TRP Tx beams to possibly change inter/intra-TRP Tx beam(s). From a possibly smaller set of beams for beam refinement than in P-1. Note that P-2 can be a special case of P-1.*

Following is my understanding on this process in illustration.

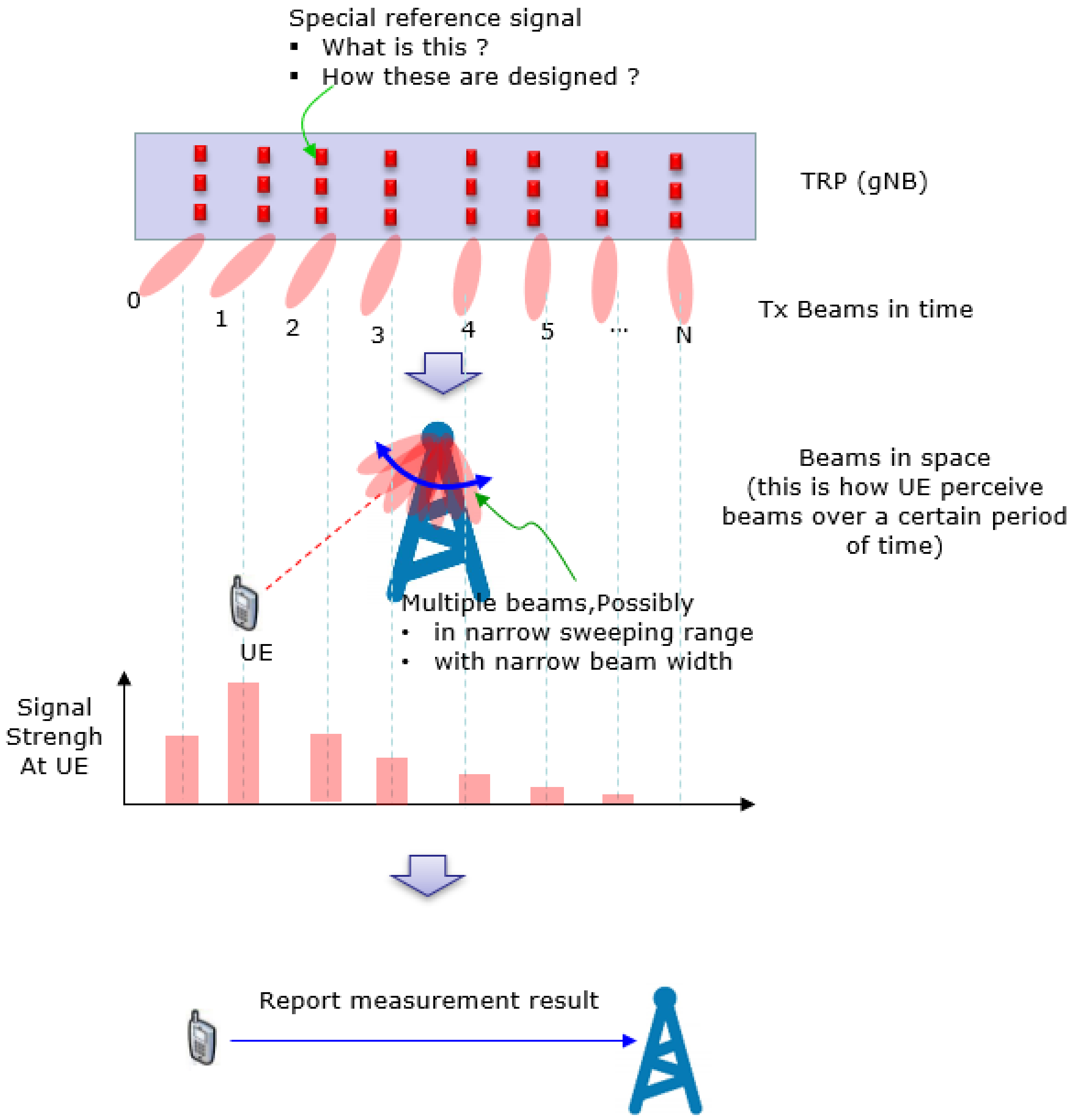


Figure 7.1.9 Beam Process 2

P-3*: is used to enable UE measurement on the same TRP Tx beam to change UE Rx beam in the case UE uses beamforming*

Following is my understanding on this process in illustration.

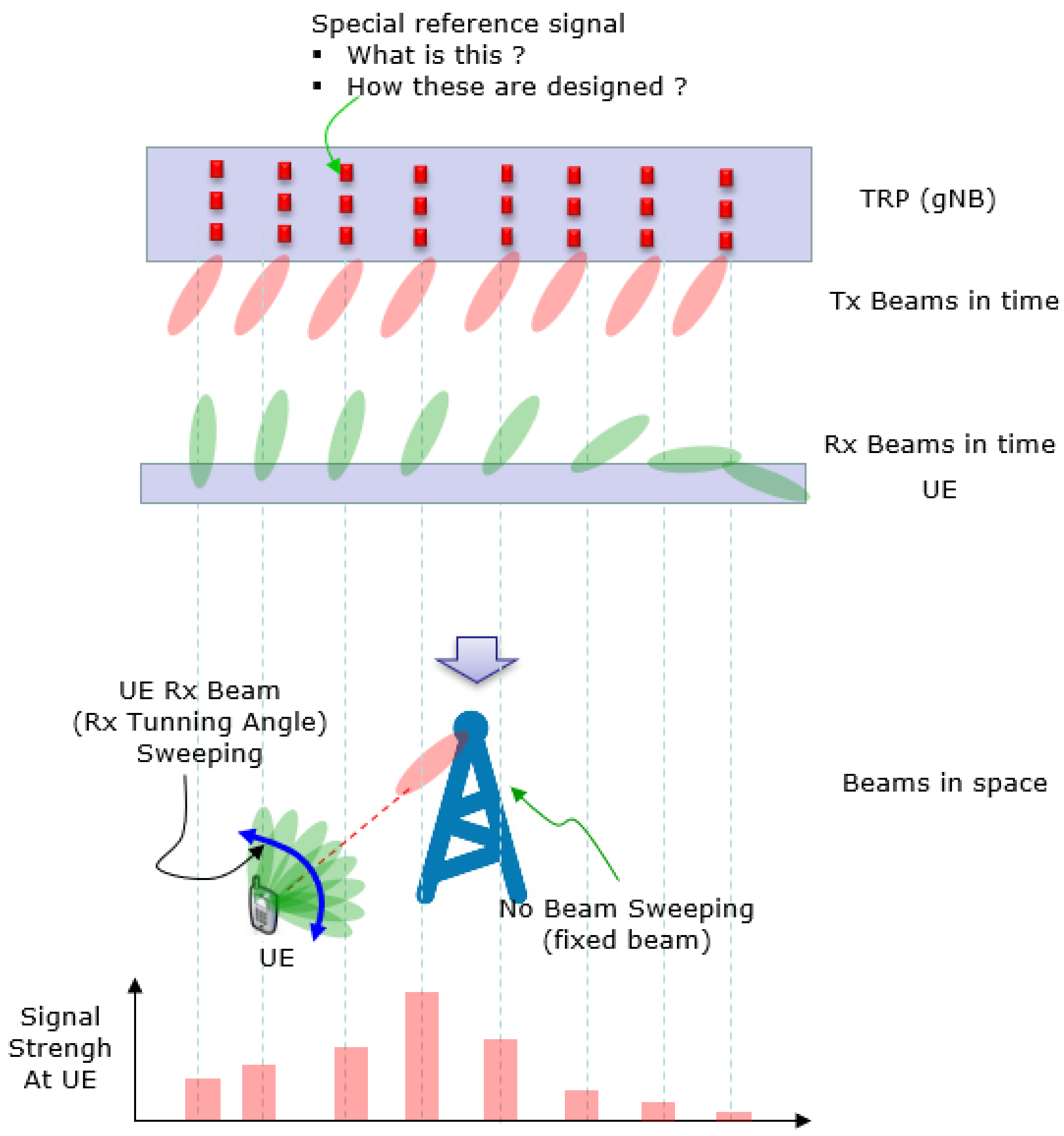


Figure 7.2.1 Beam Process 1

CHAPTER-8

SOFTWARE USED

Introduction to MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.

- Typical uses include
- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN. The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

The MATLAB system:

The MATLAB system consists of five main parts

- **Development Environment:**

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library:**

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language:**

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both “programming in the small” to rapidly create quick and dirty throw-away programs, and “programming in the large” to create large and complex application programs.

Graphics:

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

The MATLAB Application Program Interface (API):

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files. Various toolboxes are there in MATLAB for computing recognition techniques, but we are using IMAGE PROCESSING toolbox.

GRAPHICAL USER INTERFACE (GUI):

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based Mfunction is executed.

- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed. To launch GUIDE from the MATLAB command window, type guide filename Where filename is the name of an existing FIG-file on the current path. If filename is omitted, GUIDE opens a new (i.e., blank) window.

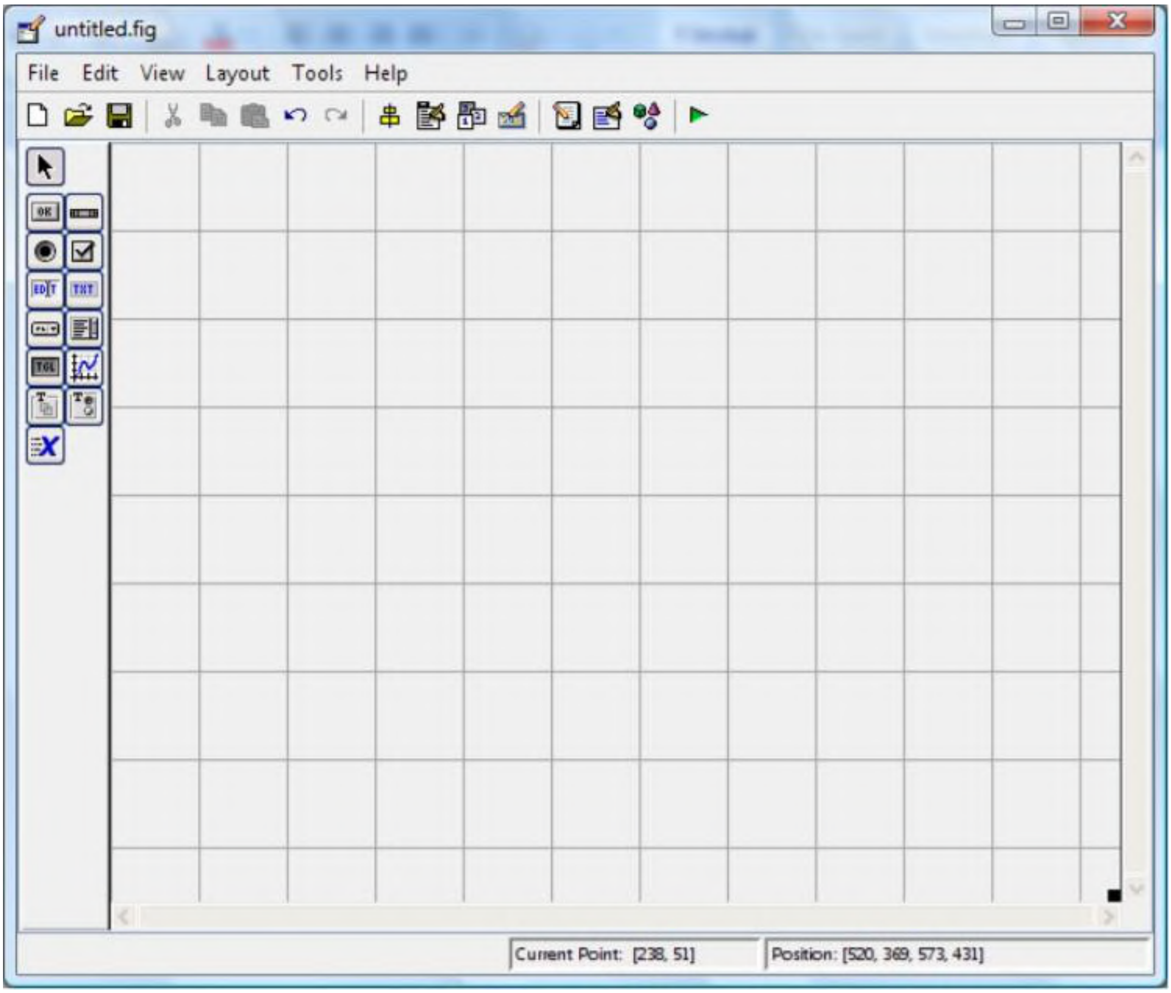


Figure 8.3.1 matlab procedure

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed. GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

Getting Started:

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos. At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

2.4.1 Introduction - describes the components of the MATLAB system.

Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

DEVELOPMENT ENVIRONMENT

8.5.1Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop. On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop. You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools.

The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacsmagik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents. To open the Help browser, click the help button in the toolbar, or type help browser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go. Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find Mfiles and other MATLAB-related files, which are organized in directories on your file system.

Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos. To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function. The workspace is not maintained after you end the MATLAB session.

To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to runMATLAB functions. The Editor/Debugger provides a graphical user interface for basic textediting, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint

CHAPTER-9

SYSTEM MODEL

The massive MIMO architecture is composed of a transmitter, a matrix channel and a receiver as shown in Fig.1. On the one hand, the transmitter design includes a pre-coder with closed-loop MIMO system, cooperative beamforming schema (i.e., in low SNR case, the best strategy is to use the largest eigen-mode only) and time-reversal for frequency channel.

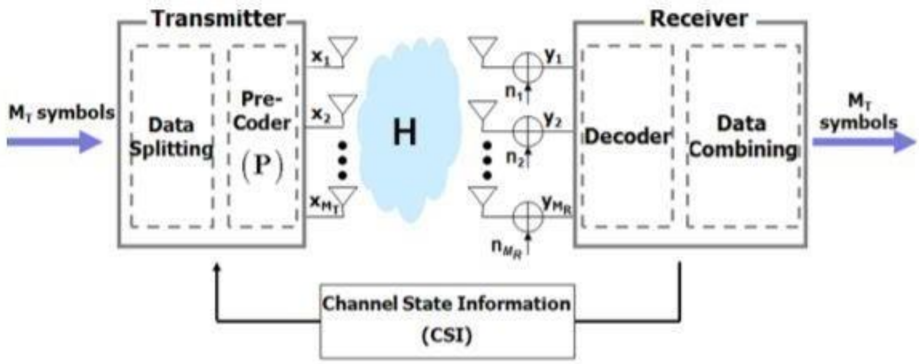


Figure 9.1.1. Massive MIMO architecture.

On the other hand, receiver design involves a decoder and a data combining sub-component. In addition, channel state information (CSI) knowledge can enhance the MIMO system spectral efficiency and reduce potentially the receiver complexity [20]. In the state-of-the-art, there are three different kind of situations concerning the quantity and quality of the CSI available at the transmitter side can be identified: i) No CSI: the transmitter does not have any knowledge of any parameter concerning the channel or the interferences at the receiver. In this case, a reasonable transmitter strategy consists in utilizing space-time codes; ii) Perfect CSI: the transmitter has full knowledge of the instantaneous channel realization and, possibly, of the interferences statistics at the receiver. In this case, since complete information is available, there are many possible strategies and optimization criteria to carry out the design depending on the detection method at the receiver or on the performance metric; iii) Imperfect CSI: the transmitter has inaccurate knowledge about the parameters describing the channel. For example, the transmitter may be informed of an erroneous channel ( ). For the case of imperfect CSI, two main strategies can be considered, either the transmitter is designed to attain the maximum performance level for the worst possible situation of the channel among the ones that are compatible with the CSI (i.e., maximin or worst-case approach) or the transmitter is designed to have the best mean performance averaged over the unknown parameters of the CSI (statistical or Bayesian approach) . In this paper, we consider the case of an imperfect CSI.

As depicted in Fig. 2, the generalized MIMO channel can be expressed by

$$y = \sqrt{SNR}\, H(x - \eta_t) + n \qquad (1)$$

where SNR, H, x,ηt and n are the signal-to-noise-ratio, MIMO matrix channel, the mismatch between the intended signal x and the signal radiated by the transmitter and circularsymmetric complex Gaussian noise, respectively.

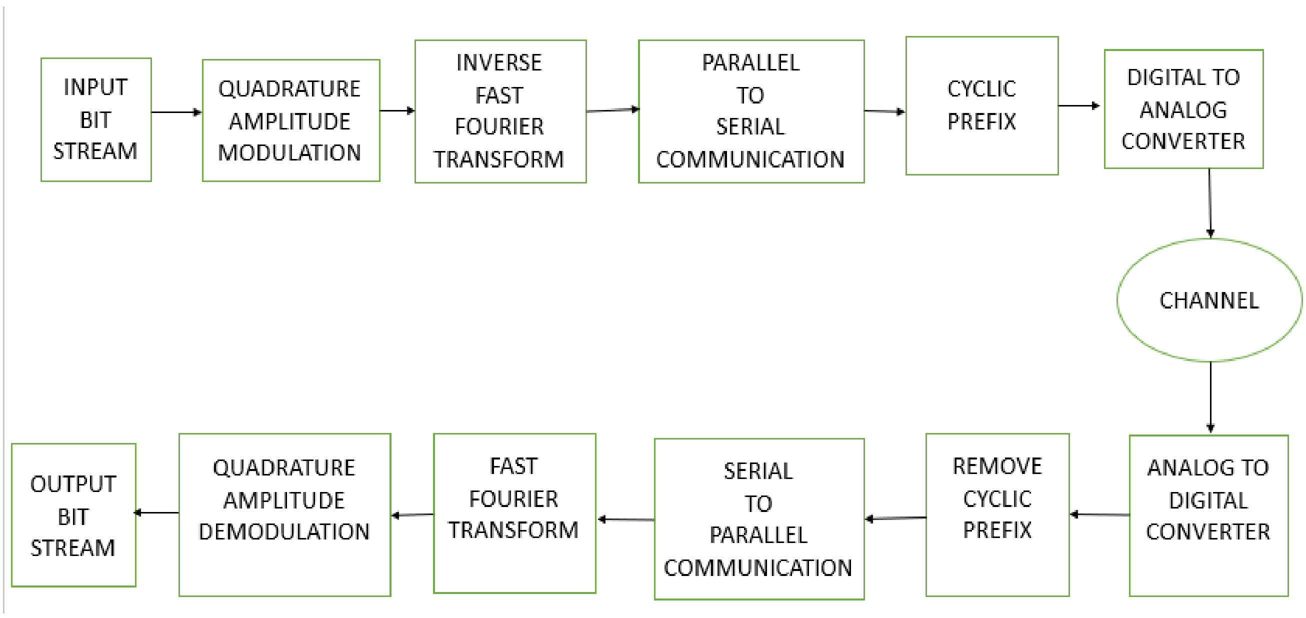


Figure 9.1.2 Block Diagram for Massive MIMO channel.

CHAPTER 10

ACCESS PROCEDURE MODELING AND PERFORMANCE ANALYSIS

Device-to-device communication is a promising technology that reuses the cellular spectrum effectively and improves network throughput significantly; thus, the users whom intend to initiate D2D communication would have higher priority than cellular users. In cellular networks, the core networks set the fixed maximum back-off time for the cellular users. However, in our proposal, we utilized the binary exponential back-off algorithm for the D2D users and defined the minimum contending window .(CW_{min})as 1. In order to ensure the priority of the D2D UE, we assumed that the allowable maximum collision times is smaller than that defined in the LTE-Advance specification. Moreover, the selection of the binary back-off algorithm simplified the design of the device because it just made few modifications based on the 801.11 model. In this paper, we denote p as the collision probability for the D2D users' access, m as the maximum number of collisions, and W_0 as the equivalent representation of CW_{min}. The binary back-off procedure can be considered as the two dimensional discrete time Markov process {s(t),b(t)} s(t) is the collision times counter and s(t) ϵ(0,m),m ϵ Z^+_0.b(t) is the back-off times for the UE, b(t) ϵ[0,Wi-1],Wi =2^i, and i ϵ [0,m]. The state transition diagram is illustrated in Figure 4.

As Figure 4 indicates, the one step of transition probability could be written as

$$P\{i, k-1 | i, k\} = 1, k \in [0, W_i - 1], i \in [0, m] \quad (2)$$

$$P\{0, k | i, 0\} = \frac{1-p}{W_0}, k \in [0, W_0 - 1], i \in [0, m] \quad (3)$$

$$P\{i, k | i-1, 0\} = \frac{p}{W_i}, k \in [0, W_i - 1], i \in [1, m] \quad (4)$$

$$P\{m, k | m, 0\} = \frac{p}{W_m}, k \in [0, W_m - 1] \quad (5)$$

Equation (2) indicates that the probability of reducing k to k -1 is 1, at the beginning of every time slot. Equation (3) calculated the probability that, after transmitting a frame successfully, the UE's status is transferred

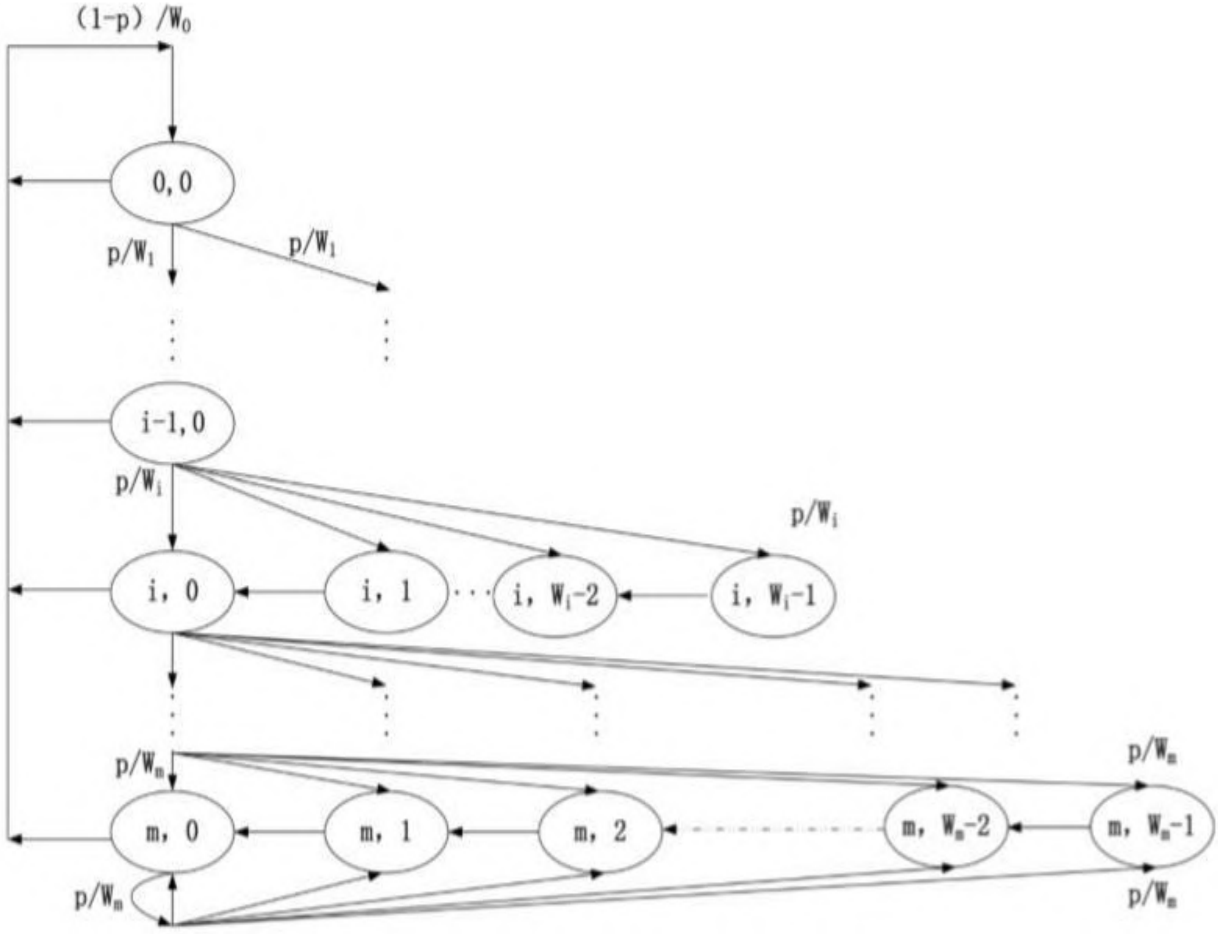


Figure 4. A state transition diagram of Markov process{s(t),b(t)}.

from (i,0) to (0,k). Similarly, Equation (4) is the probability of UE's status transferring from (i-1,0) to (i,k) after failing to transmit a frame. The probability of UE's status from (m,0) to (m,k) is expressed in Equation (5) when the allowable collision times reached the maximum value m. We assume $b_{i,k}$, presented as Equation (6) is the stationary distribution of the Markov process. According to Figure 4, we can have Equations (7) and (8). From Figure 4, we can observe that(bm,0)contains two parts of transition probability, one is from (b_{m-1},0) to(b_m,0) and the other is from (b_m,0)to (b_m,0).

$$b_{i,k} = \lim_{t\to\infty} P\{s(t) = i, b(t) = k\},$$
$$i \in [0, m], k \in [0, W_i - 1] \tag{6}$$

$$b_{i-1,0} \times p = b_{i,0} \Rightarrow b_{i,0} = p^i \times b_{0,0}, i \in [1, m] \tag{7}$$

$$\begin{aligned} &pb_{m-1,0} + pb_{m,0} = b_{m,0} \Rightarrow \\ &\quad pb_{m-1,0} = (1-p)b_{m,0} = p^m \times b_{0,0} \Rightarrow \\ &\quad b_{m,0} = \frac{p^m \times b_{0,0}}{1-p} \end{aligned} \tag{8}$$

According to the stationary distribution theorem, we can obtain (bi,k) from the following:

$$b_{i,k} = \frac{W_i - k}{W_i} \times \begin{cases} (1-p)\sum_{j=0}^{m} b_{j,0} & i = 0 \\ p \times b_{i-1,0} & 0 < i < m \\ p \times (b_{m-1,0} + b_{m,0}) & i = m \end{cases} \tag{9}$$

Because $\sum_{j=0}^{m} b_{j,0} = \frac{b_{0,0}}{1-p}$, Equation (9) can be rewritten as

$$b_{i,k} = \frac{W_i - k}{W_i} \times b_{i,0}; i \in [0, m]; k \in [0, W_i - 1] \tag{10}$$

In addition, the summation of each stationary distribution is 1; thus, we can have

$$
\begin{aligned}
1 &= \sum_{i=0}^{m} \sum_{k=0}^{W_i-1} b_{i,k} \\
&= \sum_{i=0}^{m} b_{i,0} \sum_{k=0}^{W_i-1} \frac{W_i - k}{W_i} = \sum_{i=0}^{m} b_{i,0} \times \frac{W_i + 1}{2} \qquad (11) \\
&\times \frac{b_{0,0}}{2} \left[\left(\sum_{i=0}^{m-1} (2p)^i + \frac{(2p)^m}{1-p} \right) + \frac{1}{1-p} \right]
\end{aligned}
$$

Transforming Equation (11) to another expression form, we can obtain $b_{0,0}$ as

$$
b_{0,0} = \frac{2(1-2p)(1-p)}{(1-2p)(\mathrm{CW}_{\min}+1) + \mathrm{pCW}_{\min}(1-(2p)^m)} \qquad (12)
$$

Here, we define τ as the transmitting probability, which results in

$$
\begin{aligned}
\tau &= \sum_{i=0}^{m} b_{i,0} = \frac{b_{0,0}}{1-p} \\
&= \frac{2(1-2p)(1-p)}{(1-2p)(\mathrm{CW}_{\min}+1) + \mathrm{pCW}_{\min}(1-(2p)^m)}
\end{aligned} \qquad (13)
$$

We assume that the procedure of D2D users' access satisfies Poisson distribution; thus, the probability of k D2D UEs simultaneously initiating access requests at every access slot is expressed as

$$
p_k = \frac{(\lambda/N)^k}{k!} e^{-(\lambda/N)}, \lambda > 0, k = 0, 1, 2 \ldots \qquad (14)
$$

In Equation (14), λ is the intensity of users asking for simultaneous access and N is the number of preambles. The access collision probability can be given as

$$\begin{aligned} p = p_c &= 1 - p_k(0) - p_k(1) \\ &= 1 - (1 + (\lambda/N))e^{-(\lambda/N)} \end{aligned} \tag{15}$$

Substituting Equation (15) into (13), we can obtain

$$\tau = \frac{2(1-2p_c)}{(1-2p_c)(\mathrm{CW_{min}}+1) + \mathrm{p_c CW_{min}}(1-(2p_c)^m)} \tag{16}$$

Thus, we have

$$p_c\tau(2p_c)^m - 4p_c + 3p_c\tau - 2\tau + 2 = 0 \tag{17}$$

When a large number of UEs start to access simultaneously, we know that frequent collision will occur. In this paper, we propose the control scheme to ensure the acceptable access probability , which is expressed as

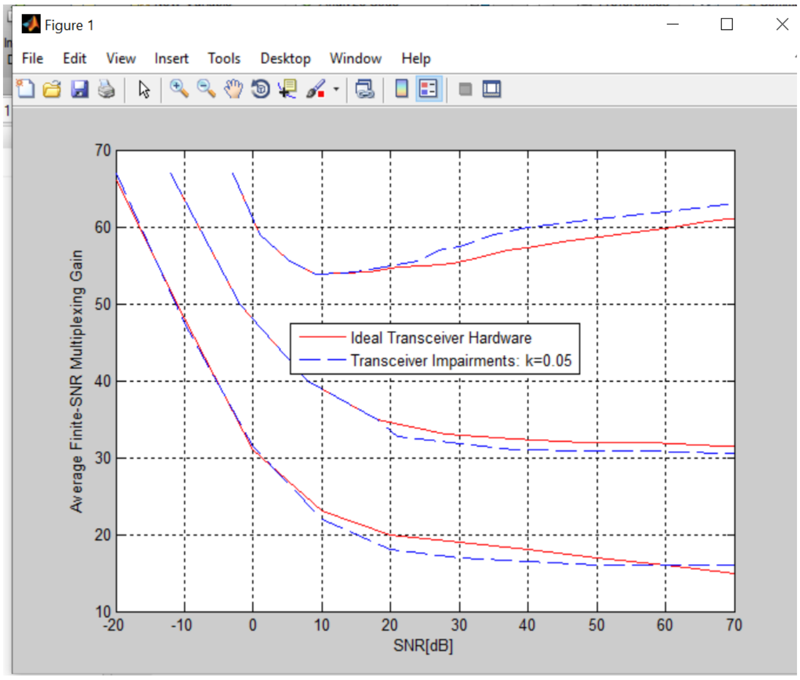
$$\alpha\tau(t) + (1-\alpha)\tau(1-t) \geq \rho \tag{18}$$

where αis the weighted factor, τ(t), and τ(t-1) is the user's access probability at t and t -1 time slot and ρ is the access threshold. Based on this constraint, we could obtain the controllable access probability denoted by P_{ctrl}, as expressed in Equation (19), which can adjust the user's access probability in the next time slot if Equation (18) is not satisfied. Otherwise, P_{ctrl}=1.

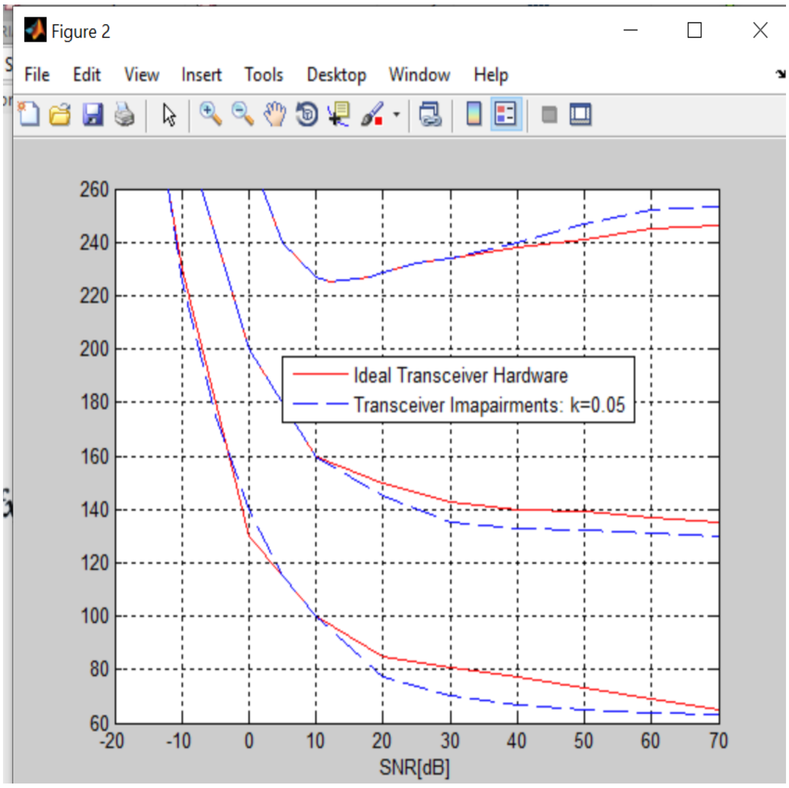
$$P_{\mathrm{ctrl}} = \frac{\alpha\tau(t) + (1-\alpha)\tau(t-1)}{\rho} \tag{19}$$

CHAPTER-11

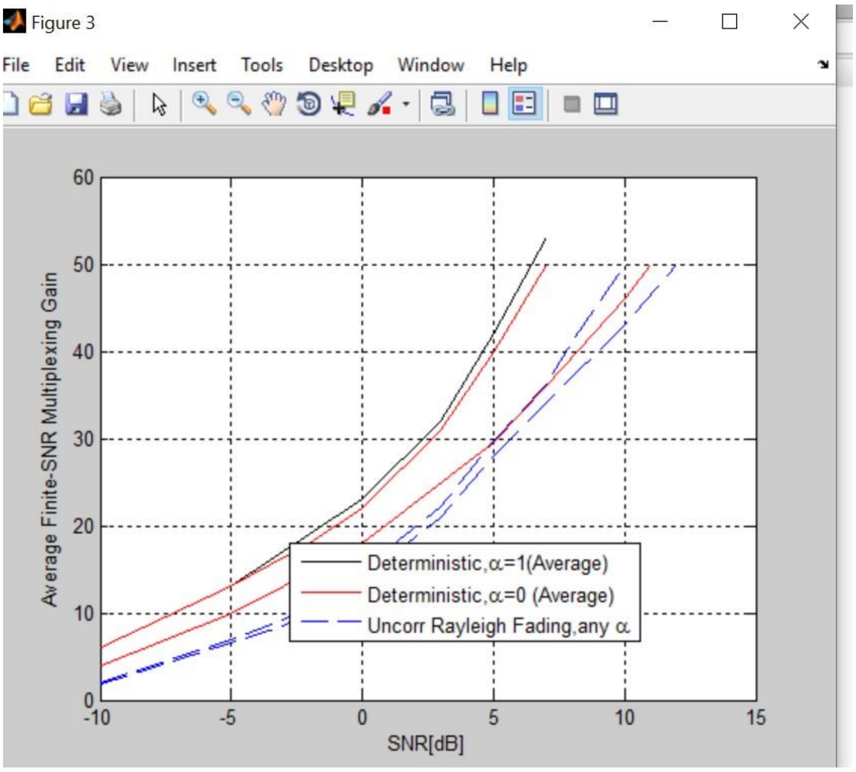
SIMULATION RESULTS AND ANALYSIS



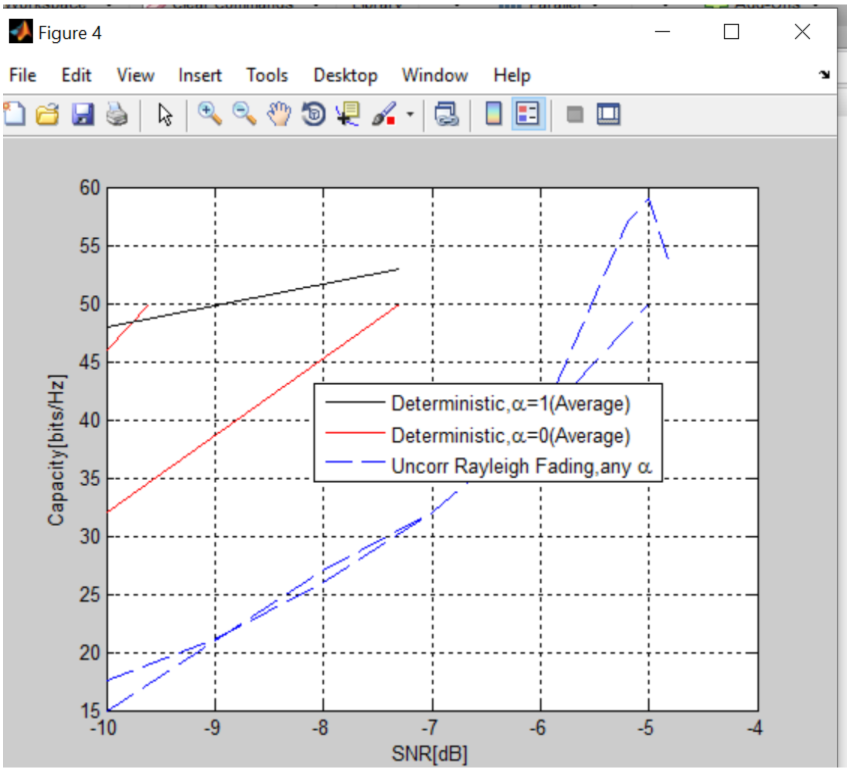
Graph 11.1.1 Average finite-SNR multiplexing gain of deterministic channels with different Nt and Nr=64 configurations



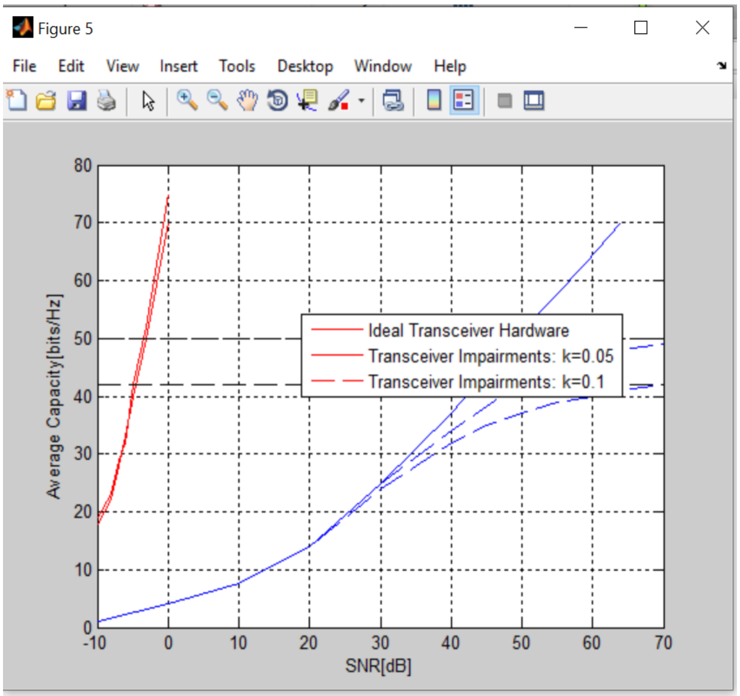
Graph 11.1.2 The capacity of uncorrelated Rayleigh fading channels with different Nt and Nr=128.



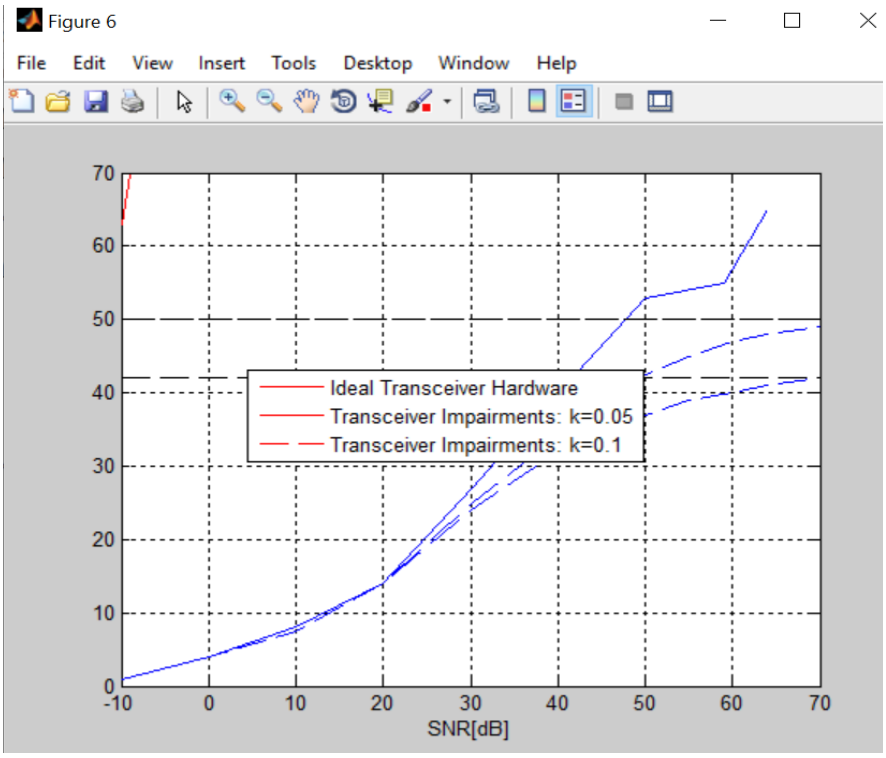
Graph 11.1.3 The capacity of uncorrelated Rayleigh fading channels with different Nt and Nr=16.



Graph 11.1.4 The average capacity of massive MIMO as a function of SNR for Nt=64 and Nr=64.



Graph 11.1.5 The average capacity of massive MIMO as a function of SNR for Nt=64 and Nr=64.



Graph 11.1.6 The average capacity of massive MIMO as a function of SNR for Nt=256 and Nr=256.

CHARACTERISATION

Multiplexing gain (M)	**Elapsed time (second)**	**Capacity limit (bit/s/Hz)**	**Maximum SNR value (dB)**
(4,4)	4,874	[35-41]	64
(16,16)	4,977	[36-42]	10
(64,64)	5,177	[42-50]	2
(128, 128)	5,681	[45-52]	0
(256,256)	8,049	[50-57]	-5

Table 11.1.7 Performance Evaluation

CHAPTER-12
CONCLUSION

In this paper, an investigation of the massive MIMO system model was presented for both deterministic and uncorrelated Rayleigh fading channels in the context of 5G mobile communications systems. An extension of analytic capacity expression was demonstrated for the uncorrelated Rayleigh fading channels taking into account SNR and subcarrier number. Moreover, two metrics are used in simulation to characterize massive MIMO; average finite SNR multiplexing gain and average capacity. It was demonstrated also that there's saturation of the massive MIMO system under the high SNR region and the finite capacity limit is independent of the channel distribution. Additionally, the massive MIMO capacity is proportional to the multiplexing gain over the whole SNR range, thus leading to an upper bound of capacity limit equal to 41 bit/s/Hz and 57 bit/s/Hz for a multiplexing gain of 4 and 256, respectively. Finally, three metrics were presented to determine the limit of massive MIMO in terms of average finite SNR, capacity and the maximum tolerable SNR for (4,4), (16,16), (64,64), (128,128) and (256,256) configurations, respectively. Therefore, by increasing the multiplexing gain a trade-off between capacity limit and tolerable SNR value is observed.

REFERENCES

[1] M. Eechkaute, A. Bourdoux, P. Donker and F. Horlin, 'Performance of emerging multi-carrier waveforms for 5G asynchronous communications', EURASIP Journal on wireless communications and networking, (2017) 2017: 29.

[2] K. Hassan, R. Gautier, I. Dayoub, M. Berbineau, E. Radoi, 'MultipleAntenna-Based Blind Spectrum Sensing in the Presence of Impulsive Noise', IEEE Transactions On Vehicular Technology, 63(5), pp. 2248 – 2257, 2014

[3] S. Kharbech, I. Dayoub, M. Zwingelstein-Colin, and E. Pierre Simon, 'Blind Digital Modulation Identification for MIMO Systems in Railway Environments With High-Speed Channels and Impulsive Noise', IEEE Transactions On Vehicular Technology, 2018, 67,(8), pp. 7370-7379.

[4] L. Lu, G. Ye Li, A. Lee Swindlehurst, A. Ashikhmin, and R. Zhang, 'An Overview of Massive MIMO: Benefits and Challenges', IEEE Journal of Selected Topics in Signal Processing, 2014, 8, (5), pp. 742- 758.

[5] E. Bjorrnson, Erik G. Larsson, and M. Debbah, 'Massive MIMO for Maximal Spectral Efficiency: How Many Users and Pilots Should Be Allocated?', IEEE Transactions on Wireless Communications, 2016, 15, (2), pp. 1293-1308,.

[6] A. Larsson, O. Edfors ,F. Tufvesson , T. Marzetta, 'Massive MIMO for next generation wireless systems', IEEE Com. Mag., 2014, 52, (2), pp.186-195.

[7] D. C. Araújo, T. Maksymyuk, A. L.F. de Almeida, T. Maciel, J. C.M. Mota1, M. Jo, 'Massive MIMO: survey and future research topics', IET Comm., 2016, 10,(15), pp. 1938–1946,.

[8] Vinel, Q. Ni, D. Staehle, and A. Turlikov, 'Capacity Analysis of Reservation-based Random Access for Broadband Wireless Access Networks', IEEE J. Sel. Areas in Communication, 2009, 27,(2), pp. 172–181.

[9] D. Zhang, K. Yu, Z. Wen, and T. Sato, 'Outage Probability Analysis of NOMA within Massive MIMO Systems', in proceeding of IEEE Vehicular Technology Conference, china, May 2016, pp. 1–5.

[10] M. Debbah and R. R. Muller, 'MIMO Channel Modeling and the Principle of Maximum Entropy', IEEE Trans. on Inf. Theory, 2005, 51,(5), pp. 1667–1690.

[11] Z. Pan, J. Lei, Y. Zhang, X. Sun, and S. Kwong, ‘Fast Motion Estimation Based on Content Property for Low-Complexity H.265/HEVC Encoder’, IEEE Trans. Broadcast, 2016, 62, (3), pp. 675– 684.

[12] R. Couillet, M. Debbah, and J. W. Silverstein, ‘A Deterministic Equivalent for the Analysis of Correlated MIMO Multiple Access Channels’, IEEE Trans. on Inf. Theory, 2011, 57,(6), pp. 3493–3514.

[13] J. W. Silverstein and A. M. Tulino, ‘Theory of Large Dimensional Random Matrices for Engineers’, in IEEE Ninth International Symposium on Spread Spectrum Techniques and Applications, Aug. 2006, pp. 458– 464.

[14] Z. Zhou, Y. Wang, Q. M. J. Wu, C. N. Yang, and X. Sun, ‘Effective and Efficient Global Context Verification for Image Copy Detection’, IEEE Trans. Inf. Forensics Security, 2017, 12,(1), pp. 48–63.

[15] J. Zhang, C. K. Wen, S. Jin, X. Gao, and K. K. Wong, ‘On Capacity of Large-Scale MIMO Multiple Access Channels with Distributed Sets of Correlated Antennas’, IEEE J. Sel. Areas Commun., 2013,(31),2, pp. 133–148.

[16] Lu, X. Gao, and C. Xiao, “A Free Deterministic Equivalent for The Capacity of MIMO MAC with Distributed Antenna Set, ” in proceeding of IEEE International Conference on Communications, Jun. 2015, pp. 1751-1756.

[17] Firag, P. J. Smith, and M. R. McKay, ‘Capacity Analysis of MIMO Three Product Channels’, in proceeding of IEEE AusCTW, Feb. 2010, pp. 13–18.

[18] E. Bjornson, P. Zetterberg, M. Bengtsson, and B. Ottersten, ‘Capacity Limits and Multiplexing Gains of MIMO Channels with Transceiver Impairments’, IEEE Communications Letters, 2013,17, (1), pp. 91-94.

[19] Paulraj, R. Nabar, D. Gore, ‘Introduction to Space-Time Wireless Communications’, (Cambridge University Press, 2003).

[20] M. Sharma, Effective Channel State Information (CSI) Feedback for MIMO Systems in Wireless Broadband Communications, Master thesis, 2014. [21] M. Listerri, ‘Impact of channel state information on the analysis and design of multi-antenna communication systems’. Dissertation thesis .2006.

A

PROJECT REPORT

On

FEW CLASS LEARNING FOR IMAGE CLASSIFICATION-AWARE DENOISING

Submitted by

Mr.K.Avinash Redddy (17K81A0486)

Mr.N.Sai Teja (17K81A0498)

Mr.V.Anirudh (17K81A04B8)

in partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

IN

Electronics and Communication Engineering

Under The Guidance of

Mr.Dr. M. Thirupathi

(B.Tech., M.Tech)

DEPARTMENT OF ECE

ST. MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

ST. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad – 500 100

NBA & NAAC A+ Accredited

CERTIFICATE

This is to certify that the project entitled "**FEW CLASS LEARNING FOR IMAGE CLASSIFICATION-AWARE DENOISING**" is being submitted by **1.Mr. K.Avinash Reddy** (17K81A0486), **2.Mr.N.Sai Teja** (17K81A0498), **3.Mr. V.Anirudh** (17K81A04B8) in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology** in **Electronics and Communication Engineering** is recorded of bonafide work carried out by them. The results embodied in this report have been verified and found satisfactory.

Mr.Dr.M.Thirupathi
Internal Guide
Department of Electronics and
Communication Engineering.

Head of the Department
Dr.B. Hari krishna ,
Department of Electronics and
Communication Engineering.

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **K.AVINASH REDDY** WITH ROLL NO.**17K81A0486, N.SAI TEJA** WITH ROLL NO.**17K81A0498**, **V.ANIRUDH** WITH ROLL NO.**17K81A04B8**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**FEW-CLASS LEARNING FOR IMAGE-CLASSIFICATION- AWARE DENOISING**

" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. ★ Kompally-500014 ★

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

DECLARATION

We, the student of **Bachelor of Technology** in Department of Electronics and Communication Engineering, session: <2017 - 2021>, St. Martin’s Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work Entitled **“FEW CLASS LEARNING FOR IMAGE CLASSIFICATION-AWARE DENOISING”** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

K Avinash Reddy	17K81A0486
N Sai Teja	17K81A0498
V Anirudh	17K81A04B8

ACKNOWLEDGEMENTS

While bringing out this project report to its final form, we came across a number of people whose contributions in various ways helped my field of research and they deserve special thanks. It is a pleasure to convey our gratitude to all of them.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our college management** for their kind support and permission to use the facilities available in the institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.Santosh Kumar Patra**, our **beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna, Head & professor,** the **Dept. of Electronics and Communication Engineering** for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our **Dept. Project Coordinator** Dr. **A. Anand, Assistant Professor, Dept.of ECE** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project guide, **Mr. M.Thirupathi, Assistant Professor,** Department of ECE and our class project coordinator **Mr. K. Karthik, Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level project coordinators **Dr.Malayadri, Professor in IT & Dr.R.Santhosh Kumar, Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and cooperation that is difficult to express in words.

1.K Avinash Reddy

2.N Sai Teja

3.V Anirudh

ABSTRACT

The accuracy of image classification networks decreases substantially in noisy situations such as low light environments. A simple solution to this problem is to integrate an image denoising network as a pre-processing operation before classification and then to retrain the classifier to improve the classification accuracy. However, this straightforward approach suffers from excessive training time because the denoising network requires the entire dataset to conduct end-to-end training. In this paper, instead of using classification accuracy as a loss function, we propose using the difference of the outputs of the hidden layer of the image classification network. This loss function, known as feature loss or perceptual loss, allows us to train the denoising network using only limited images containing extremely few classes from the dataset. The experimental results show that the proposed method dramatically improves the classification accuracy, when we use only a few classes (from 2.5% to 10% of the original dataset) for training. This approach is effective on previously unseen classes even when the image classifier network has been changed by fine-tuning.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Introduction :

A DNN (deep neural network) is a versatile framework for performing a variety of image processing and computer vision tasks. In particular, DNNs have achieved significant successes in image classification [1, 2, 3]. However, a wellknown drawback of these classifiers is that they are highly sensitive to low-level image degradations such as noise, blur, and compression artifacts. For example, noise is inevitable when applying a pretrained image classification network to images captured by surveillance cameras operated in low light conditions. However, retraining the image classifier network itself for each type of degradation is prohibitively expensive due to the extremely high computational complexity. A simple and straightforward solution to this problem is to use a DNN-based image denoising network [4, 5, 6, 7, 8] as a pre-processing step before image classification. A similar approach involves building cascading networks to address low-level vision tasks such as denoising and high-level vision asks (image classification, image retrieval, semantic segmentation) was proposed in [9, 10, 11].

Historical Development:

Liu et al. proposed a cascaded network architecture in [10] that included an image denoising network and an image classification network. Because the training of the denoising network was performed using an end-to-end approach, they adopted cross entropy as their loss function, which requires images representing all the classes in the dataset for training. In the case of ILSVRC 2012, the dataset contains 1,000 classes. Thus, the method proposed in [10] still suffers from long training times. Consequently, a natural question arises: does training the denoising network truly require the entire image dataset?

In this study, we propose a new method for training cascading image denoising and image classification networks using images belonging to only a few classes. Unlike the method in [10], which uses the classification results as the loss function, our proposed method uses the output vector of the hidden layer (feature vector) of the image classification network for the loss function. We train our denoising network to output a feature vector similar to the output feature vector of the original image. We reveal that only a few classes (from 2.5% to 10%) in the original dataset are sufficient to train a denoising network that can improve the classification accuracy even under severe noise. We also show that the proposed method performs well even when the classification network is fine-tuned to a dataset with completely unknown classes. The adopted loss function is not new. Using a feature vector for the loss function is called feature loss or perceptual loss [12] and has been used in many studies [11, 13]. In particular, the study in [11] uses feature loss to train a cascaded network similar to ours. However, in these methods, feature loss is used to reconstruct visually pleasing outputs. To the best of our knowledge, this is the first study to reveal that feature loss enables us to train a classification-aware denoising network from very few classes. The contributions of this paper are summarized as follows: • Using feature loss allows the training process to improve the classification accuracy of the denoising network using only images that represent extremely small percentages of the classes in the dataset. • Even if the image classification network has been changed by fine-tuning on images belonging to totally unseen classes, our denoising network can improve the classification accuracy on noisy images.

INTRODUCTION TO IMAGE PROCESSING

IMAGE:

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—suchas cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Fig 1.1 General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

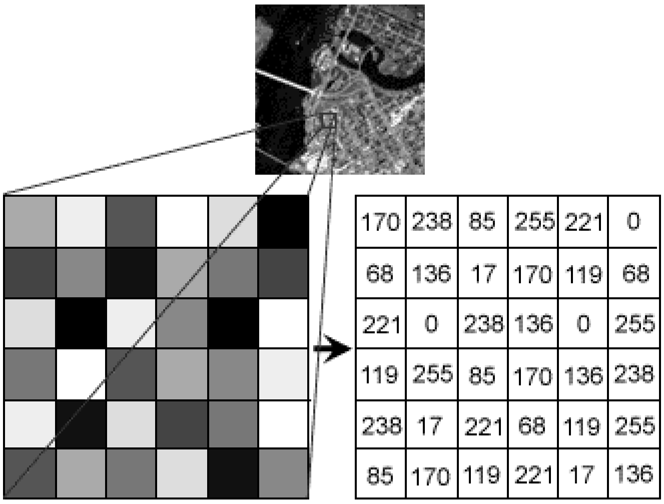


Fig 1.2 Image pixel

Each pixel has a colour. The colour is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
Transparency	Red	Green	Blue

Fig 1.3. Transparency image

IMAGE FILE SIZES:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

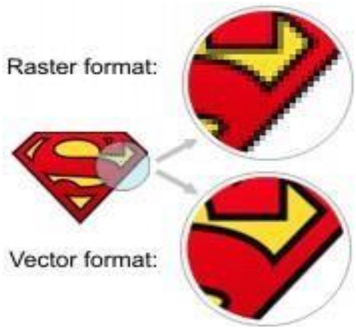


Fig 1.4 Resolution image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save themin their own native format.

RASTER FORMATS:

These formats store images as bitmaps (also known as pixmaps)

- **JPEG/JFIF:**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

- **EXIF:**

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

- **TIFF:**

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

- **PNG:**

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

- **GIF:**

GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

- **BMP:**

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM:**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

- **SVG:**

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all-purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

IMAGE PROCESSING:

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, misunderstandings and misinformation. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factors combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several newtechnological trends promise to further promote digital image processing. These include parallel processing mode practical by low-cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:

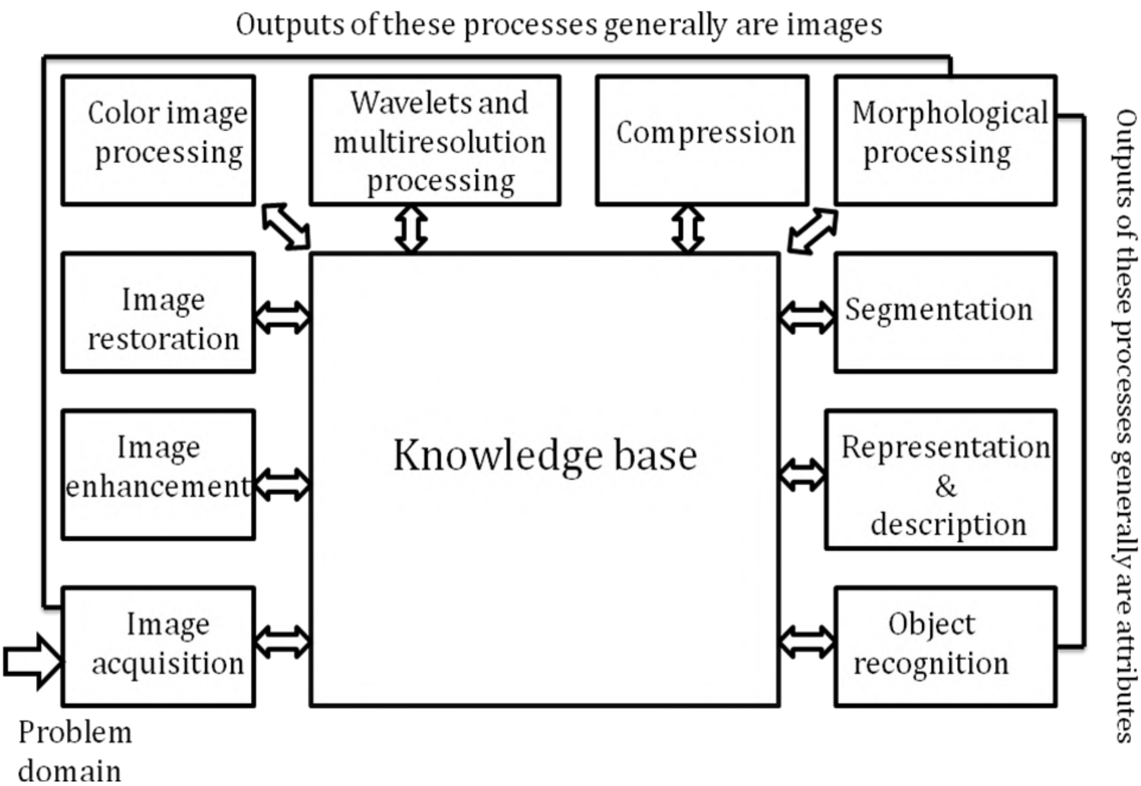


Fig 1.6 Image fundamental

Image Acquisition:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Fig 1.7 Digital camera image

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Fig 1.8 digital camera cell

Image Enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because “it looks better.” It is important to keep in mind that enhancement is a very subjective area of image processing.

Fig 1.9 Image enhancement

Image restoration:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Fig 1.10 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, whereas removal of image blur by applying a deblurring function is considered a restoration technique.

Colour image processing:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Fig 1.11 Color & Gray scale image

Wavelets and multiresolution processing:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform-based image processing since the late1950's, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

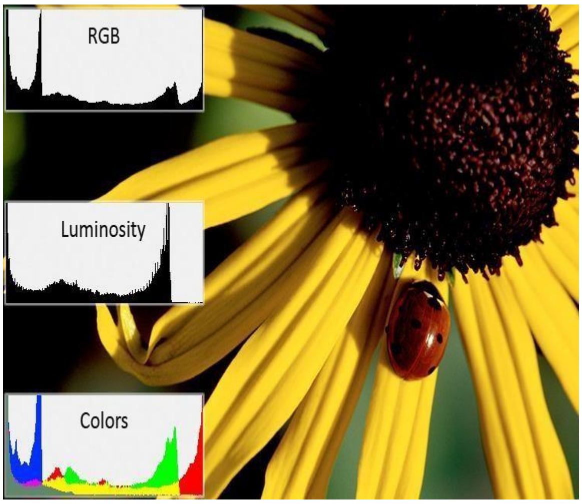


Fig .1.12 rgb histogram image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called **Multiresolution** theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

Compression:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

Morphological processing:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Fig 1.13 blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

Segmentation:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

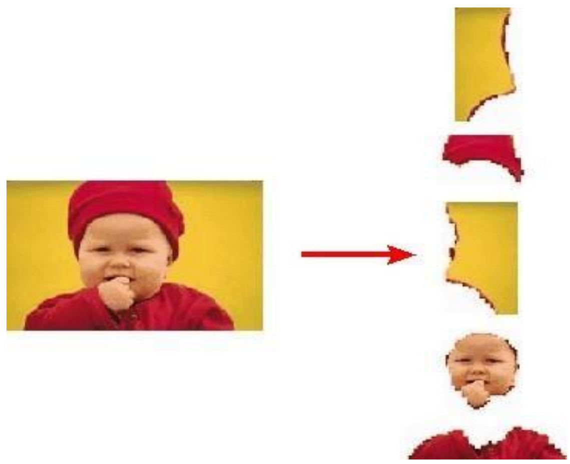


Fig 1.14 Image segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

Representation and description:

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

Object recognition:

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

Knowledgebase:

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an interrelated to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented pre-processing techniques using MATLAB. The knowledge base also can be quite complex, such as an interrelated to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion application.

This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located but it also aids in feedback operations between modules through the knowledge base. We implemented pre-processing techniques using MATLAB.

COMPONENTS OF AN IMAGE PROCESSING SYSTEM:

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

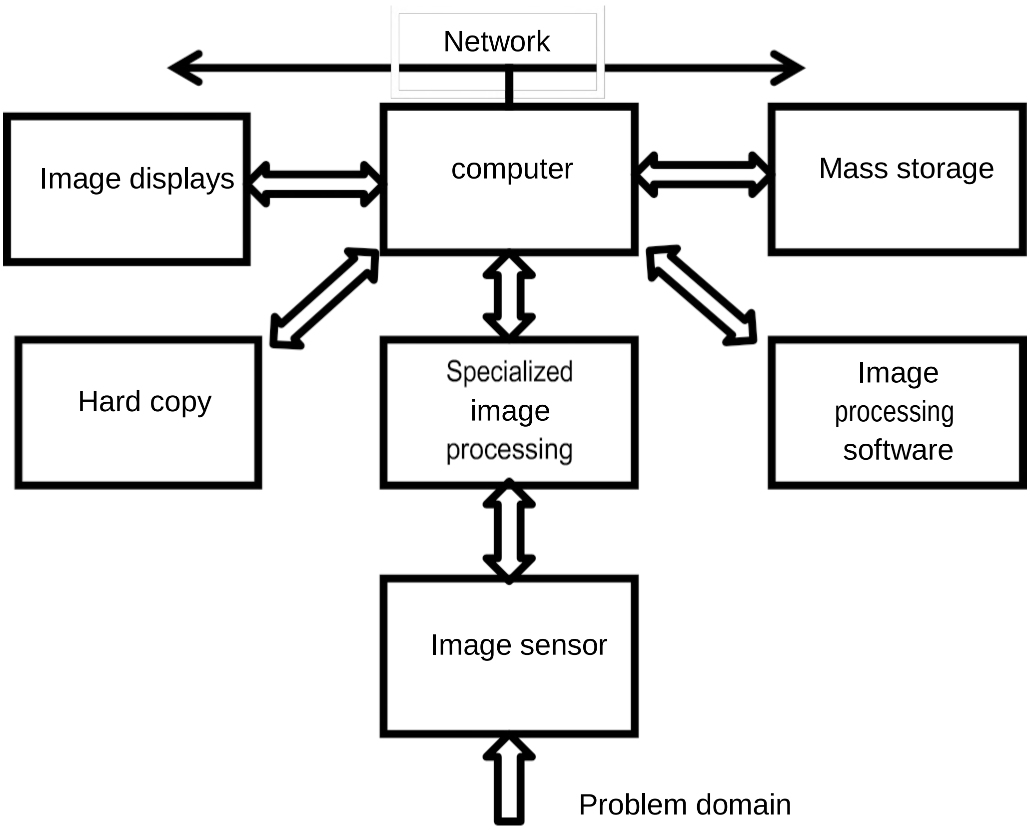


Fig 1.15 Component of image processing

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 1.24 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

- **Image sensors:**

  With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

- **Specialized image processing hardware:**

  Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle. This type of hardware sometimes is called a front-end subsystem,

- **Computer:**

  The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

- **Image processing software:**

  Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

- **Mass storage:**

  Mass storage capability is a must in image processing applications. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage for image processing applications fall into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), Gbytes (meaning giga, or one billion, bytes), and Tbytes (meaning tera, or one trillion, bytes)

  One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig. 1.24. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in "jukeboxes" are the usual media for archival applications.

- **Image displays:**

  Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are their requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

- **Hardcopy:**

Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

- **Network:**

Networking is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of optical fiber and other broadband technologies.

Color and texture are two low-level features widely used for image classification, indexing and retrieval. Color is usually represented as a histogram, which is a first order statistical measure that captures global distribution of color in an image One of the main drawbacks of the histogram- based approaches is that the spatial distribution and local variations in color are ignored. Local spatial variation of pixel intensity is commonly used to capture texture information in an image. Grayscale Co-occurrence Matrix (GCM) is a well-known method for texture extraction in the spatial domain. A GCM stores the number of pixel neighbourhoods in an image that have a particular grayscale combination. Let I be an image and let p and Np respectively denote any arbitrary pixel and its neighbour in a given direction. If GL denotes the total number of quantized gray levels and gl denotes the individual Gray levels, where gl∈{0, . . ., GL _ 1}, then each component of GCM can be written as follows:

$$\mathrm{gcm}(i, j) = \Pr((\mathrm{gl}\ \ \mathrm{gl}$$

gcm(i, j) is the number of times the gray level of a pixel p denoted by glp equals i, and the gray level of its neighbour Np denoted by glNp equals j, as a fraction of the total number of pixels in the image. Thus, it estimates the probability that the gray level of an arbitrary pixel in an image is i, and that of its neighbour is j. One GCM matrix is generated for each possible neighbourhood direction, namely, 0, 45, 90 and 135.Average and range of 14 features like Angular Second Moment, Contrast, Correlation, etc., are generated by combining all the four matrices to get a total of 28 features. In the GCM approach for texture extraction, color information is completely lost since only pixel gray levels are considered.

To incorporate spatial information along with the color of image pixels, a feature called color correlogram has recently been proposed. It is a three-dimensional matrix that represents the probability of finding pixels of any two given colors at a distance ‘d’ apart Auto correlogram is a variation of correlogram, which represents the probability of finding two pixels with the same color at a distance ‘d’ apart. This atproach can effectively represent color distribution in an image. However, correlogram features do not capture intensity variation. Many image databases often contain both color as well as gray scale images. The color correlogram method does not constitute a good descriptor in such databases.

Another method called Color Co-occurrence Matrix (CCM) has been proposed to capture color variation in an image. CCM is represented as a three-dimensional matrix, where color pair of the pixels p and Np are captured in the first two dimensions of the matrix and the spatial distance ‘d’ between these two pixels is captured in the third dimension. This approach is a generalization of the color correlogram and reduces to the pure color correlogram for d = 1. CCM is generated using only the Hue plane of the HSV (Hue, Saturation and Intensity Value) color space. The Hue axis is quantized into HL number of levels. If individual hue values are denoted by hl, where $\mathrm{hl} \in \{0, \ldots, HL\}$, then each component of CCM can be written as follows:

$$\mathrm{ccm}(i, j) = \Pr((\mathrm{hl}_{\mathrm{p}}, \mathrm{hl}_{N_{\mathrm{p}}}) = (i, j))$$

Four matrices representing neighbours at angles 0, 90, 180 and 270 are considered. This approach was further extended by separating the diagonal and the non-diagonal components of CCM to generate a Modified Color Co-occurrence Matrix (MCCM). MCCM, thus, may be written as follows: MCCM = (CCMD; CCMND) .

Here, CCMD and CCMND correspond to the diagonal and off-diagonal components of CCM. The main drawback of this approach is that, like correlogram, it also captures only color information and intensity information is completely ignored.

An alternative approach is to capture intensity variation as a texture feature from an image and combine it with color features like histograms using suitable weights. One of the challenges of this approach is to determine suitable weights since these are highly application dependent. In certain applications like Content-based Image Retrieval (CBIR), weights are often estimated from relevance feedback given by users.

While relevance feedback is sometimes effective, it makes the process of image retrieval user-dependent and iterative. There is also no guarantee on the convergence of the weight-learning algorithms. In order to overcome these problems, researchers have tried to combine color and texture features together during extraction.

proposed two approaches for capturing color and intensity variations from an image using the LUV color space. In the Single-channel Co-occurrence Matrix (SCM), variations for each color channel, namely, L, U and V are considered independently. In the Multi-channel Co-occurrence Matrix (MCM), variations are captured taking two channels at a time – UV, LU and LV. Since the LUV color space separates out chrominance (L and U) from luminance (V), SCM in effect, generates one GCM and two CCMs from each image independently. As a result, correlation between the color channels is lost

However, in MCM, the count of pair wise occurrences of the values of different channels of the color space is captured. Thus, each component of MCM can be written as follows:

mcmUV(i; j) = Pr((up; vNp) = (i; j))

mcmLU(i; j) = Pr((lp; uNp) = (i; j))

mcmLV(i; j) = Pr((lp; vNp) = (i; j))

Here, mcmUV(i, j) is the number of times the U chromaticity value of a pixel p denoted by up equals i, and the V chromaticity value of its neighbour Np denoted by vNp equals j, as a fraction of the total number of pixels in the image. Similarly, mcmLU(i, j) and mcmLV(i, j) are defined. One MCM matrix is generated for each of the four neighbourhood directions, namely, 0, 45, 90 and 135.

Deng and Manjunath (2001) proposed a two-stage method called JSEG, which combines color and texture after image segmentation. In the first stage, colors are quantized to the required levels for differentiating between various regions of an image. Pixel values of the regions are then replaced by their quantized color levels to form a color map. Spatial variation of color levels between different regions in the map is viewed as a type of texture composition of the image.

Yu et al. (2002) suggested the use of color texture moments to represent both color and texture of an image. This approach is based on the calculation of

Local Fourier Transformation (LFT) coefficients. Eight templates equivalent to LFT are operated over an image to generate a characteristic map of the image. Each template is a 3 · 3 filter that considers eight neighbours of the current pixel for LFT calculation. First and second order moments of the characteristic map are then used to generate a set of features.

In this paper, we propose an integrated approach for capturing spatial variation of both color and intensity levels in the neighbourhood of each pixel using the HSV color space. In contrast to the other methods, for each pixel and its neighbour, the amount of color and intensity variation between them is estimated using a weight function. Suitable constraints are satisfied while choosing the weight function for effectively relating visual perception of color and the HSV color space properties. The color and intensity variations are represented in a single composite feature known as Integrated Color and Intensity Co-occurrence Matrix (ICICM). While the existing schemes generally treat color and intensity separately, the proposed method provides a composite view to both color and intensity variations in the same feature. The main advantage of using ICICM is that it avoids the use of weights to combine individual color and texture features. We use ICICM feature in an image retrieval application from large image databases.

Early result on this work was reported in (Vadivel et al., 2004a). In the next section, we describe the proposed feature extraction technique after introducing some of the properties of the HSV color space. Choice of quantization levels for color and intensity axes, selection of parameter values and a brief overview of the image retrieval application

Integrated color and intensity co-occurrence matrix:

We propose to capture color and intensity variation around each pixel in a two-dimensional matrix called Integrated Color and Intensity Co-occurrence Matrix (ICICM). This is a generalization of the Grayscale Co-occurrence Matrix and the Color Co-occurrence Matrix techniques. For each pair of neighbouring pixels, we consider

their contribution to both color perception as well as gray level perception to the human eye. Some of the useful properties of the HSV color space and their relationship to human color perception are utilized for extracting this feature. In the next sub-section, we briefly explain relevant properties of the HSV color space. In the subsequent subsection, we describe how the properties can be effectively used for generating ICICM.

HSV color space:

HSV Color space: Basically, there are three properties or three dimensions of color that being hue, saturation and value HSV means Hue, Saturation and Value. It is important to look at because it describes the color based on three properties. It can create the full spectrum of colors by editing the HSV values. The first dimension is the Hue. Hue is the other name for the color or the complicated variation in the color. The quality of color as determined by its dominant wavelength. This Hue is broadly classified into three categories. They are primary Hue, Secondary Hue and Teritiary Hue. The first and the foremost is the primary Hue it consists of three colors they are red, yellow and blue. The secondary Hue is formed by the combination of the equal amount of colors of the primary Hue and the colors of the secondary Hue which was formed by the primary Hue are Orange, Green and violet. The remaining one is the teritiary Hue is formed by the combination of the primary Hue and the secondary Hue. The limitless number of colors are produced by mixing the colors of the primary Hue in different amounts. Saturation is the degree or the purity of color. Then the second dimension is the saturation. Saturation just gives the intensity to the colors. The saturation and intensity drops just by mixing the colors or by adding black to the color. By adding the white to the color in spite of more intense the color becomes lighter. Then finally the third dimension is the Value. The value is the brightness of the color. When the value is zero the color space is totally black with the increase in the color there is also increase in the brightness and shows the various colors. The value describes the contrast of the color. That means it describes the lightness and darkness of the color. As similar to the saturation this value consists of the tints and shades. Tints are the colors with the added white and shades are the colors with the added black.

Properties of the HSV color space:

Sensing of light from an image in the layers of human retina is a complex processwith rod cells contributing to scotopic or dim-light vision and cone cells to photopic or bright-light vision (Gonzalez and Woods, 2002). At low levels of illumination, only the rod cells are excited so that only gray shades are perceived. As the illumination level increases, more and more cone cells are excited, resulting in increased color perception. Various color spaces have been introduced to represent and specify colors in a way suitable for storage, processing or transmission of color information in images. Out of these, HSV is one of the models that separate out the luminance component (Intensity) of a pixel color from its chrominance components (Hue and Saturation). Hue represents pure color, which is perceived when incident light is of sufficient illumination and contains a single wavelength. Saturation gives a measure of the degree by which a pure color is diluted by white light. For light with low illumination, corresponding intensity value in the HSV color space is also low.

The HSV color space can be represented as a Hexa cone, with the central vertical axis denoting the luminance component, I (often denoted by V for Intensity Value). Hue, is a chrominance component defined as an angle in the range [0,2p] relative to the red axis with red at angle 0, green at 2p/3, blue at 4p/3 and red again at 2p. Saturation, S, is the other chrominance component, measured as a radial distance from the central axis of the hexacone with value between 0 at the center to 1 at the outer surface. For zero saturation, as the intensity is increased, we move from black to white through various shades of gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the color represented by its hue. When saturation is near 0, all the pixels in an image look alike even though their hue values are different.

As we increase saturation towards 1, the colors get separated out and are visually perceived as the true colors represented by their hues. Low saturation implies presence of a large number of spectral components in the incident light, causing loss of color information even though the illumination level is sufficiently high. Thus, for low values of saturation or intensity, we can approximate a pixel color by a gray level while for higher saturation and intensity, the pixel color can be approximated by its hue. For low intensities, even for a high saturation, a pixel color is close to its gray value. Similarly, for low saturation even for a high value of intensity, a pixel is perceived as gray. We use these properties to estimate the degree by which a pixel contributes to color perception and gray level perception.

One possible way of capturing color perception of a pixel is to choose suitable thresholds on the intensity and saturation. If the saturation and the intensity are above their respective thresholds, we may consider the pixel to have color dominance; else, it has gray level dominance. However, such a hard thresholding does not properly capture color perception near the threshold values. This is due to the fact that there is no fixed level of illumination above which the cone cells get excited. Instead, there is a gradual transition from scotopic to photopic vision. Similarly, there is no fixed threshold for the saturation of cone cells that leads to loss of chromatic information at higher levels of illumination caused by color dilution. We, therefore, use suitable weights that vary smoothly with saturation and intensity to represent both color and gray scale perception for each pixel.

NON-INTERVAL QUANTIZATION:

Due to the large range for each component by directly calculating the characteristics for the retrieval then the computation will be very difficult to ensure rapid retrieval. It is essential to quantify HSV space component to reduce computation and improve efficiency. At the same time, because the human eye to distinguish colors is limited, do not need to calculate all segments. Unequal interval quantization according the human color perception has been applied on H, S, V components.

Based on the color model of substantial analysis, we divide color into eight parts. Saturation and intensity is divided into three parts separately in accordance with the human eyes to distinguish. In accordance with the different colors and subjective color perception quantification, quantified hue(H), saturation(S) and value(V)

In accordance with the quantization level above, the H, S, V three-dimensional feature vector for different values of with different weights to form one dimensional feature vector and is given by the following equation:

G = Qs*Qv*H+Qv*s+V

Where Qs is the quantized series of S and Qv is the quantized series of V. And now by setting Qs = Qv = 3, Then G = 9H+3S+V

$$H = \begin{cases} 0 & if \quad h \in [316, 20] \\ 1 & if \quad h \in [21, 40] \\ 2 & if \quad h \in [41, 75] \\ 3 & if \quad h \in [76, 155] \\ 4 & if \quad h \in [156, 190] \\ 5 & if \quad h \in [191, 270] \\ 6 & if \quad h \in [271, 295] \\ 7 & if \quad h \in [296, 315] \end{cases} \qquad S = \begin{cases} 0 & if \quad s \in [0, 0.2) \\ 1 & if \quad s \in [0.2, 0.7) \\ 2 & if \quad s \in [0.7, 1) \end{cases} \qquad V = \begin{cases} 0 & if \quad v \in [0, 0.2) \\ 1 & if \quad v \in [0.2, 0.7) \\ 2 & if \quad v \in [0.7, 1) \end{cases}$$

In this way three component vector of the HSV from one dimensional vector, which quantize the whole color space for the 72 kinds of the main colors. So, we can handle 72 bins of one-dimensional histogram. This qualification is effective in reducing the images by the effect of the light intensity, but also reducing the computational time and complexity.

1.10IMAGE RETRIEVAL:

Image retrieval is nothing, but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords or the descriptions to the images so that the retrieval can be performed. Manual image annotation is time consuming, expensive and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc.,

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others. one can today generate volumes of images with content as diverse as family get-togethers and national park visits. Low-cost storage and easy Web hosting has fueled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchable image data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real-world image search system designers.

As far as technological advances are concerned, growth in Content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real-world implications, applications, and constraints of the technology. Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future.

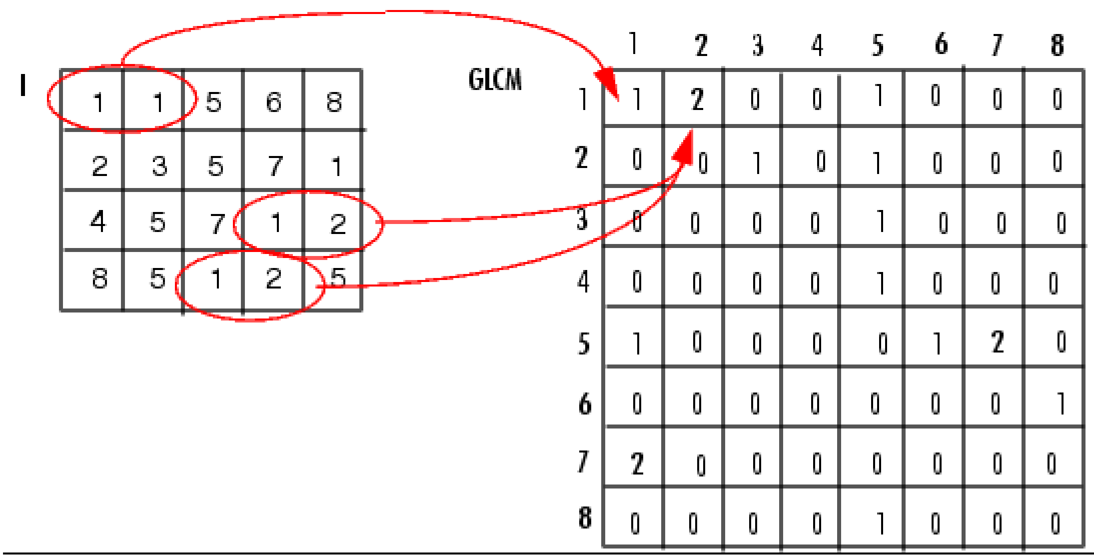
An image retrieval system designed to serve a personal collection should focus on features such as personalization, flexibility of browsing, and display methodology. For example, Google's Picasa system [Picasa 2004] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods. The notable few applications include an FPGA implementation of a color-histogram-based image retrieval system [Kotoulas and Andreadis 2003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 2003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 2002].

Discussion. Regardless of the nature of the collection, as the expected user-base grows, factors such as concurrent query support, efficient caching, and parallel and distributed processing of requests become critical. For future real-world image retrieval systems, both software and hardware approaches to address these issues are essential. More realistically, dedicated specialized servers, optimized memory and storage support, and highly parallelizable image search algorithms to exploit cluster computing powers are where the future of large-scale image search hardware support lies.

1.11.OVERVIEW OF TEXTURE:

We all know about the term Texture but for defining it is a hard time. One can differentiate the two different Textures by recognizing the similarities and differences. Commonly there are three ways for the usage of the Textures:

Based on the Textures the images can be segmented to differentiate between already segmented regions or to classify them. We can reproduce Textures by producing the descriptions. The texture can be analyzed in three different ways. They are Spectral, Structural and Statistical:



Digital image processing

Background:

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An important characteristic underlying the design of image processing systems is the significant level of testing & experimentation that normally is required before arriving at an acceptable solution. This characteristic implies that the ability to formulate approaches &quickly prototype candidate solutions generally plays a major role in reducing the cost & time required to arrive at a viable system implementation.

What is DIP

An image may be defined as a two-dimensional function f(x, y), where x & y are spatial coordinates, & the amplitude off at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital image by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensor, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis& computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting & somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low-level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low- level process is characterized by the fact that both its inputs & outputs are images.

Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images, but its outputs are attributes extracted from those images. Finally higher- level processing involves “Making sense” of an ensemble of recognized objects, as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

What is an image?

An image is represented as a two-dimensional function f (x, y) where x and y are spatial co-ordinates and the amplitude of 'f' at any pair of coordinates (x, y) is called the intensity of the image at that point.

Gray scale image:

A grayscale image is a function I (xylem) of the two spatial coordinates of the image plane.

I(x, y) is the intensity of the image at the point (x, y) on the image plane.

I (xylem) takes non-negative values assume the image is bounded by a rectangle [0, a] ×[0, b] I: [0, a] × [0, b] → [0, info)

Color image:

It can be represented by three functions, R (xylem) for red, G (xylem) for green *and* B (xylem) for blue.

An image may be continuous with respect to the x and y coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate's values is called sampling. Digitizing the amplitude values is called quantization.

Coordinate convention:

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image f (x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size M X N. The values of the coordinates (xylem) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates.

In many image processing books, the image origin is defined to be at (xylem)=(0,0).The next coordinate values along the first row of the image are (xylem)=(0,1).It is important to keep in mind that the notation (0,1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Following figure shows the coordinate convention. Note that x ranges from 0 to M-1 and y from 0 to N-1 in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (xylem) the toolbox uses the notation (race) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate topples, (alb), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentation refers to the coordinates. Less frequently the toolbox also employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refers to rows. This is the opposite of our use of variables x and y.

Image as Matrices:

The preceding discussion leads to the following representation for a digitized image function:

f (0,0)   f(0,1)   ………..   f (0, N-1)

f (1,0)   f (1,1)   …………   f (1, N-1)

f (xylem)=   .   .   .

.   .   .

f (M-1,0)   f(M-1,1)................. f (M-1, N-1)

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements. A digital image can be represented naturally as a MATLAB matrix:

f (1,1)   f (1,2)........... f (1, N)

f (2,1)   f (2,2)........... f (2, N)

.   .   .

f =   .   .   .

f (M,1) f(M,2) ……. f (M, N)

Where f (1,1) = f (0,0) (note the use of a monoscope font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation f (p, q) denotes the element located in row p and the column q. For example, f (6,2) is the element in the sixth row and second column of the matrix f. Typically we use the letters M and N respectively to denote the number of rows and columns in a matrix. A 1xN matrix is called a row vector whereas an Mx1 matrix is called a column vector. A 1x1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using mono-scope characters. We use conventional Roman, italic notation such as f(x ,y), for mathematical expressions

Reading Images:

Images are read into the MATLAB environment using function imread whose syntax is

Imread (‘filename’)

Format Name	**Description**	**Recognized Extension**
TIFF	Tagged image file format	.tif, .tiff
JPEG	Joint Photograph Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	.xwd

Table 1.1: Image Formats

Here filename is a spring containing the complete of the image file (including any applicable extension). For example, the command line

>> f = imread (‘8. jpg’);

Reads the JPEG (above table) image chest Xray into image array f. Note the use of single quotes (‘) to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output If a semicolon is not included. MATLAB displays the results of the operation(s) specified in that line. The prompt symbol (>>) designates the beginning of a command line, as it appears in the MATLAB command window.

Data Classes:

Although we work with integers coordinates the values of pixels themselves are not restricted to be integers in MATLAB. Table above list various data classes supported by MATLAB and IPT are representing pixels values. The first eight entries in the table are refers to as numeric data classes. The ninth entry is the char class and, as shown, the last entry is referred to as logical data class.

All numeric computations in MATLAB are done in double quantities, so this is also a frequent data class encounter in image processing applications. Class unit 8 also is encountered frequently, especially when reading data from storages devices, as 8-bit images are most common representations found in practice. These two data classes, classes logical, and, to a lesser degree, class unit 16 constitute the primary data classes on which we focus. Many ipt functions however support all the data classes listed in table. Data class double requires 8 bytes to represent a number uint8 and int 8 require one byte each, uint16 and int16 requires 2bytes and unit 32.

Name	**Description**
Double	Double _ precision, floating_ point numbers the Approximate.
Uint8	unsigned 8_bit integers in the range [0,255] (1byte per Element).
Uint16	unsigned 16_bit integers in the range [0, 65535] (2byte per element).
Uint 32	unsigned 32_bit integers in the range [0, 4294967295] (4 bytes per element).
Int8	signed 8_bit integers in the range [-128,127] 1 byte per element)
Int 16	signed 16_byte integers in the range [32768, 32767] (2 bytes per element).
Int 32	Signed 32_byte integers in the range [-2147483648, 21474833647] (4 byte per element).

Single   single _precision floating _point numbers with values

In the approximate range (4 bytes per elements)

Char   characters (2 bytes per elements).

Logical   values are 0 to 1 (1byte per element).

Int 32 and single required 4 bytes each. The char data class holds characters in Unicode representation. A character string is merely a 1*n array of characters logical array contains only the values 0 to 1, with each element being stored in memory using function logical or by using relational operators.

Image Types:

The toolbox supports four types of images:

1. Intensity images.
2. Binary images.
3. Indexed images.
4. R G B images.

Most monochrome image processing operations are carried out using binary or intensity images, so our initial focus is on these two image types. Indexed and RGB colour images.

Intensity Images:

An intensity image is a data matrix whose values have been scaled to represent intentions. When the elements of an intensity image are of class unit8, or class unit 16, they have integer values in the range [0,255] and [0, 65535], respectively. If the image is of class double, the values are floating point numbers. Values of scaled, double intensity images are in the range [0, 1] by convention.

Binary Images:

Binary images have a very specific meaning in MATLAB.A binary image is a logical array 0s and1s.Thus, an array of 0s and 1s whose values are of data class, say unit8, is not considered as a binary image in MATLAB. A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, we create an array B using the statement.

B=logical (A)

If A contains elements other than 0s and 1s. Use of the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.

Using relational and logical operators also creates logical arrays.

To test if an array is logical, we use the I logical function: islogical(c).

If c is a logical array, this function returns a 1. Otherwise returns a 0. Logical array can be converted to numeric arrays using the data class conversion functions.

Indexed Images:

An indexed image has two components:

A data matrix integer, x

A color map matrix, map

Matrix map is an m*3 arrays of class double containing floating point values in the range [0, 1]. The length m of the map is equal to the number of colors it defines. Each row of map specifies the red, green and blue components of a single color. An indexed images uses "direct mapping" of pixel intensity values color map values. The color of each pixel is determined by using the corresponding value the integer matrix x as a pointer into map. If x is of class double, then all of its components with values less than or equal to 1 point to the first row in map, all components with value 2 point to the second row and so on. If x is of class units or unit 16, then all components value 0 point to the first row in map, all components with value 1 point to the second and so on.

RGB Image:

An RGB color image is an M*N*3 array of color pixels where each color pixel is triplet corresponding to the red, green and blue components of an RGB image, at a specific spatial location. An RGB image may be viewed as "stack" of three gray scale images that when fed in to the red, green and blue inputs of a color monitor

Produce a color image on the screen. Convention the three images forming an RGB color image are referred to as the red, green and blue components images. The data class of the components images determines their range of values. If an RGB image is of class double the range of values is [0, 1].

Similarly, the range of values is [0,255] or [0, 65535]. For RGB images of class units or unit 16 respectively. The number of bits use to represents the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8bit image, the corresponding RGB image is said to be 24 bits deep.

Generally, the number of bits in all component images is the same. In this case the number of possible color in an RGB image is (2^b) ^3, where b is a number of bits in each component image. For the 8bit case the number is 16,777,216 colors

Chapter 2

LITERATURE SURVEY

Literature survey on research area:

In this study, we propose a new method for training cascading image denoising and image classification networks using images belonging to only a few classes. Unlike the method in [10], which uses the classification results as the loss function, our proposed method uses the output vector of the hidden layer (feature vector) of the image classification network for the loss function. We train our denoising network to output a feature vector similar to the output feature vector of the original image. We reveal that only a few classes (from 2.5% to 10%) in the original dataset are sufficient to train a denoising network that can improve the classification accuracy even under severe noise. We also show that the proposed method performs well even when the classification network is fine-tuned to a dataset with completely unknown classes. The adopted loss function is not new. Using a feature vector for the loss function is called feature loss or perceptual loss [12] and has been used in many studies [11, 13]. In particular, the study in [11] uses feature loss to train a cascaded network similar to ours. However, in these methods, feature loss is used to reconstruct visually pleasing outputs. To the best of our knowledge, this is the first study to reveal that feature loss enables us to train a classification-aware denoising network from very few classes. The contributions of this paper are summarized as follows: • Using feature loss allows the training process to improve the classification accuracy of the denoising network using only images that represent extremely small percentages of the classes in the dataset. • Even if the image classification network has been changed by fine-tuning on images belonging to totally unseen classes, our denoising network can improve the classification accuracy on noisy images.

Review of related literature:

Image denoising is one of the most basic undertakings in image preparing for better examination and vision. There are numerous sorts of noise which can diminish the nature of images. The Speckle noise which can be displayed as multiplicative noise, primarily happens in different imaging framework because of arbitrary variety of the pixel esteems. It very well may be defined as the duplication of arbitrary qualities with the pixel esteem. Images are a characteristic route for people to consider spatial data, and burrow ital images are a characteristic portrayal of spatial information. Like every recorded signal, advanced images are regularly ruined by noise, expanding the trouble with which human spectators or PC calculations can extricate the valuable basic data. Despite the fact that noise can be relieved by improved image securing equipment, in certain modalities, for example, intelligible imaging, the noise is an innate aspect of the imaging cycle. Fig. 1.1 Image denoising model. There are two principle purposes for getting this gauge. To start with, the noise sifting can be proceeded as a prehandling venture for additional machine investigation, for example, scene division, object identification, or visual following. Besides, denoised images are simpler to decipher by human spectators, helping in errands, for example, arranging ice types in SAR images, or surveying blood vessel malady in ultrasound images. Multiplicative noise or spot noise happened in different imaging frameworks because of irregular variety of pixel esteems. Albeit various reclamation strategies were proposed in writing like Wiener filtering and Lee filtering to denoise such sort of uproarious images, anyway these techniques are not giving promising outcomes as far as PSNR, MSE and SNR. If there should arise an occurrence of denoising the attribute of the framework just as the sort of noise is known heretofore. The image s(x,y) is obscured by the direct activities causing the noise n(x,y) to include or duplicate with the image. The uproarious image at that point goes through a denoising strategy and produces the denoised image z(x,y). How close the image z(x,y) is to the first image relies upon the noise levels and the denoising calculation use.

CONCLUSION:

This examination work presents an extensive survey of literature on image denoising based on prior work. Image Noise is random variation of brightness or color in an image. It can be produced by any circuitry such as sensor, scanner or digital camera. Image noise is an undesirable 37 INTERNATIONAL JOURNAL OF INNOVATIVE TRENDS IN ENGINEERING (IJITE) ISSN: 2395-2946 ISSUE: 93, VOLUME 69, NUMBER 01, SEPTEMBER 2020 signal, it’s produce by image capturing device that add extra information. In many cases, it reduces image quality and is especially significant when the objects being imaged are small and have relatively low contrast. This random variation in image brightness is designated noise. This noise can be either image dependent or image independent

CHAPTER-3

PROJECT DESCRIPTION

Existing Method:

In [10], Liu et al. proposed connecting a denoising network and an image classification network and revealed that cascading two networks is mutually beneficial: improving both the classification accuracy and the visual appearance of the denoised image. They used VGG16 as the image classification network and a U-Net [14]-based network as the denoising network. The loss function of the cascaded network, which is the weighted sum of cross entropy and mean squared error (MSE). (1where xi is a i-th clean image, and yi is a corresponding noisy image obtained by adding Gaussian noise with a standard deviation of σ to each pixel in xi. Let ui be a one-hot vector of the ground truth label of xi. VGG16 and the image denoising network are denoted by C(·) and D(·; Θ), respectively, and Θ represents all the trainable parameters of D. The first term of the loss function is the cross entropy e(·) between the classification result of the noisy image and the corresponding grand truth u. Thus, we can improve the classification accuracy by minimizing this term. The second term is used to improve the subjective quality of the denoised images. The balance between these terms is controlled by the parameter λ.

Proposed Method:

In this section, we introduce a new learning method for the denoising network to improve the classification accuracy of the cascaded network that requires only a few data classes. Our cascaded network, shown in Fig. 1, consists ofthe denoising network and the image classification network. To overcome Fig. 1. The network architectureof the proposed method. We use a feature vector for our loss function instead of the final classification result. the problem of CDnN, we use the output of the hidden layer in VGG16 (feature vector) to train the denoising network. We use a 4,096-dimensional vector immediately before the 1,000- class classification of VGG16 as our feature vector. We split VGG16 into two parts before and after the final layer and denote them as featureNet and classNet. Thus, our feature vector can be considered as the output of featureNet. The loss function of the proposed method.

BLOCK DIAGRAM:

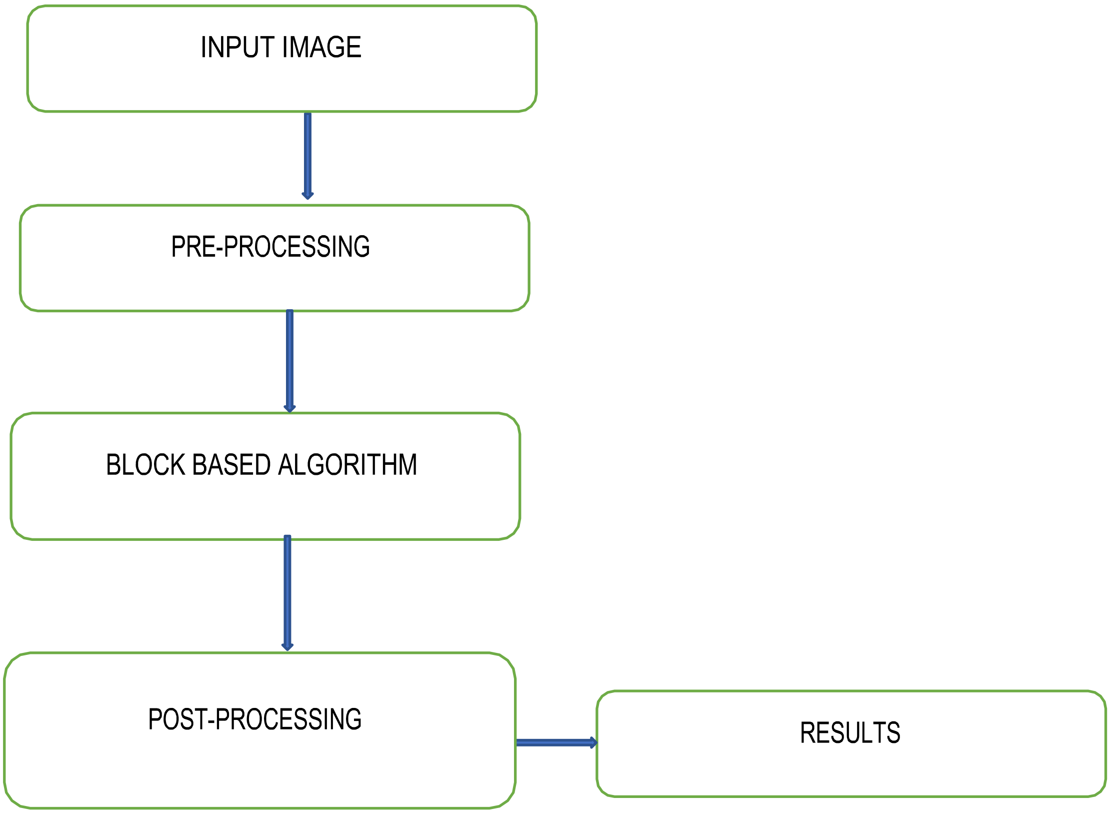


Fig 3.1: Block Diagram

Image denoising :

We used the ILSVRC2012 training and validation sets to train the cascaded network and the test images to evaluate the classification accuracy. A preprocessing operation (cropping and pixel value shifting) for VGG16 [1] was applied during the training procedure. First, we obtained noisy input images by adding additive Gaussian noise (σ = 15, 30, 45, 60) to all the images in the dataset. The top-1 and top-5 classification accuracies of VGG16 for the noisy images are shown in the leftmost column of Table 2. The results show that the accuracy decreases rapidly as the noise level increases. Then, we applied DnCNN [4], CDnN, and FDnN (ours) to the noisy images and obtained the corresponding classification accuracies on the denoised images. We use sets of classes (25, 50, 75, and 100) randomly chosen from the ILSVRC2012 training set. Then, we chose 650 and 10 images randomly from each class for training and validation, respectively. The learning rate for the denoising network was initial set to 0.001 and multiplied by 0.2 after 30, 60 and 90 epochs. The training was completed at 100 epochs. We chose the model that yielded the best validation accuracy for subsequent testing. We trained all the denoising networks with each noise level. Table 2 shows all the resulting classification accuracies, revealing the effectiveness of the proposed method (VGG16 + FDnN) on every class and at each noise level. Notably, when only a few classes are used for training, the accuracy gap between FDnN and CDnN increases. Fig. 2 shows the original, noisy, and two denoised images by DnCNN and FDnN. DnCNN (c), successfully denoises the images, but they appear somewhat over-smoothed. In contrast, the denoised images by FDnN (d), appear to still contain some noise. Table 2 shows that the accuracy is higher when using FDnN than when using DnCNN. These results suggest that the remaining noise contributes to improving the classification accuracy

Steps for the Proposed Method:

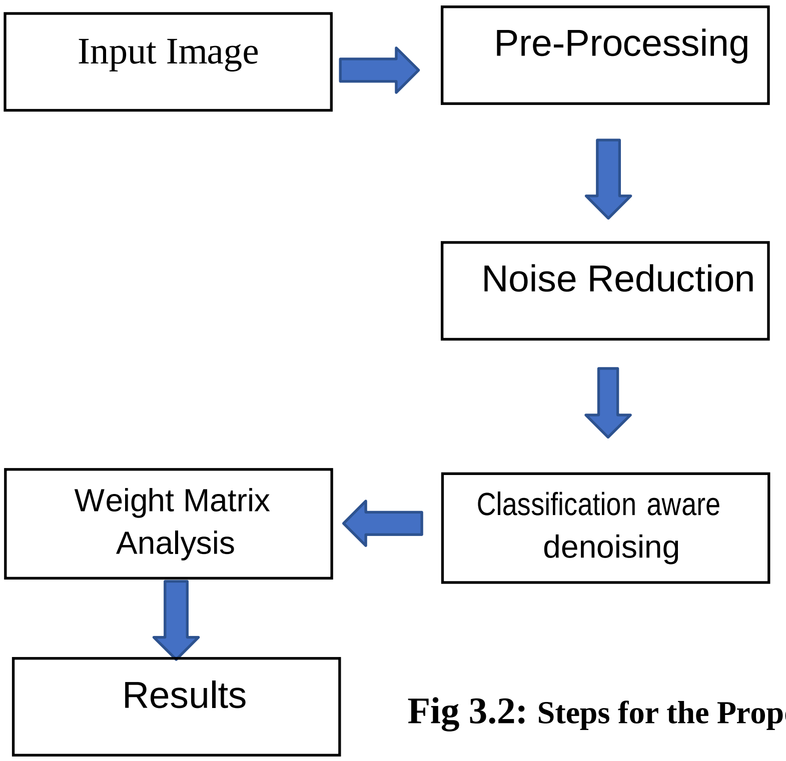


Fig 3.2: Steps for the Proposed Method

METHODOLOGY:

In this section, we introduce a new learning method for the denoising network to improve the classification accuracy of the cascaded network that requires only a few data classes. Our cascaded network, shown in Fig. 1, consists of the denoising network and the image classification network. To overcome

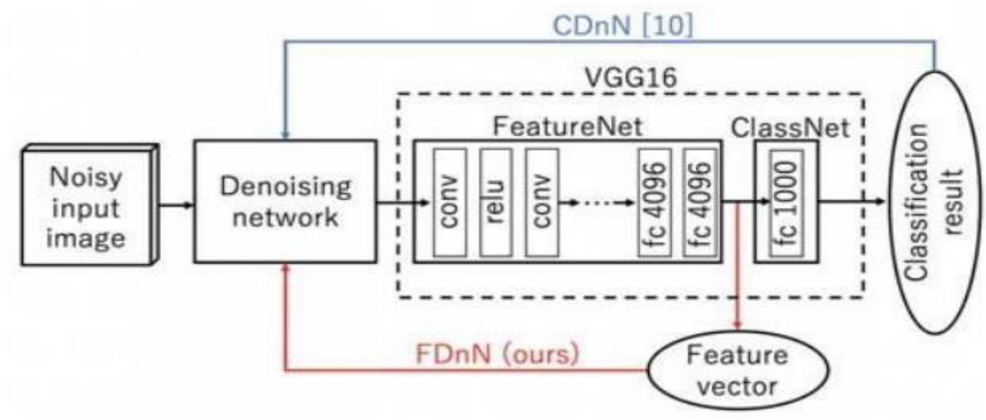


Fig. 1. The network architecture of the proposed method. We use a feature vector for our loss function instead of the final classification result.

Fig 3.3:Network Architecture Of Proposed Method

the problem of CDnN, we use the output of the hidden layer in VGG16 (feature vector) to train the denoising network. We use a 4,096-dimensional vector immediately before the 1,000- class classification of VGG16 as our feature vector. We split VGG16 into two parts before and after the final layer and denote them as featureNet and classNet. Thus, our feature vector can be considered as the output of featureNet. The loss function of the proposed method is defined as follows:

$$L_{\mathrm{f}}(\Theta) = \sum_i \| F(D(\boldsymbol{y}_i; \Theta)) - F(\boldsymbol{x}_i)) \|_2^2,$$

where F(·) is a map corresponding to featureNet. Eq. (3) shows that our loss function is defined by the MSE between the denoised and the original image. We denote the denoising network that is optimized by the above loss function as the feature-oriented denoising network (FDnN). Because the feature vector potentially contains more information than the classification results, we expect that training with the loss function shown in Eq. (3) should improve the classification accuracy for unknown classes. We also conduct some preliminary experiments to find the best layer from which to extract the feature vector; however, we found no other layers that improve the classification performance other than the 4,096-dimensional feature vector described above.

CHAPTER 4

Software Introduction:

Introduction to MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others. Areasin which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

The MATLAB system:

The MATLAB system consists of five main parts

Development Environment:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

MATLAB Mathematical Function Library:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

The MATLAB Language:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both “programming in the small” to rapidly create quick and dirty throw-away programs, and “programming in the large” to create large and complex application programs.

Graphics:

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully

customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

The MATLAB Application Program Interface (API):

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

GRAPHICAL USER INTERFACE (GUI):

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.) and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension.fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.
- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type

guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted,

GUIDE opens a new (i.e., blank) window.

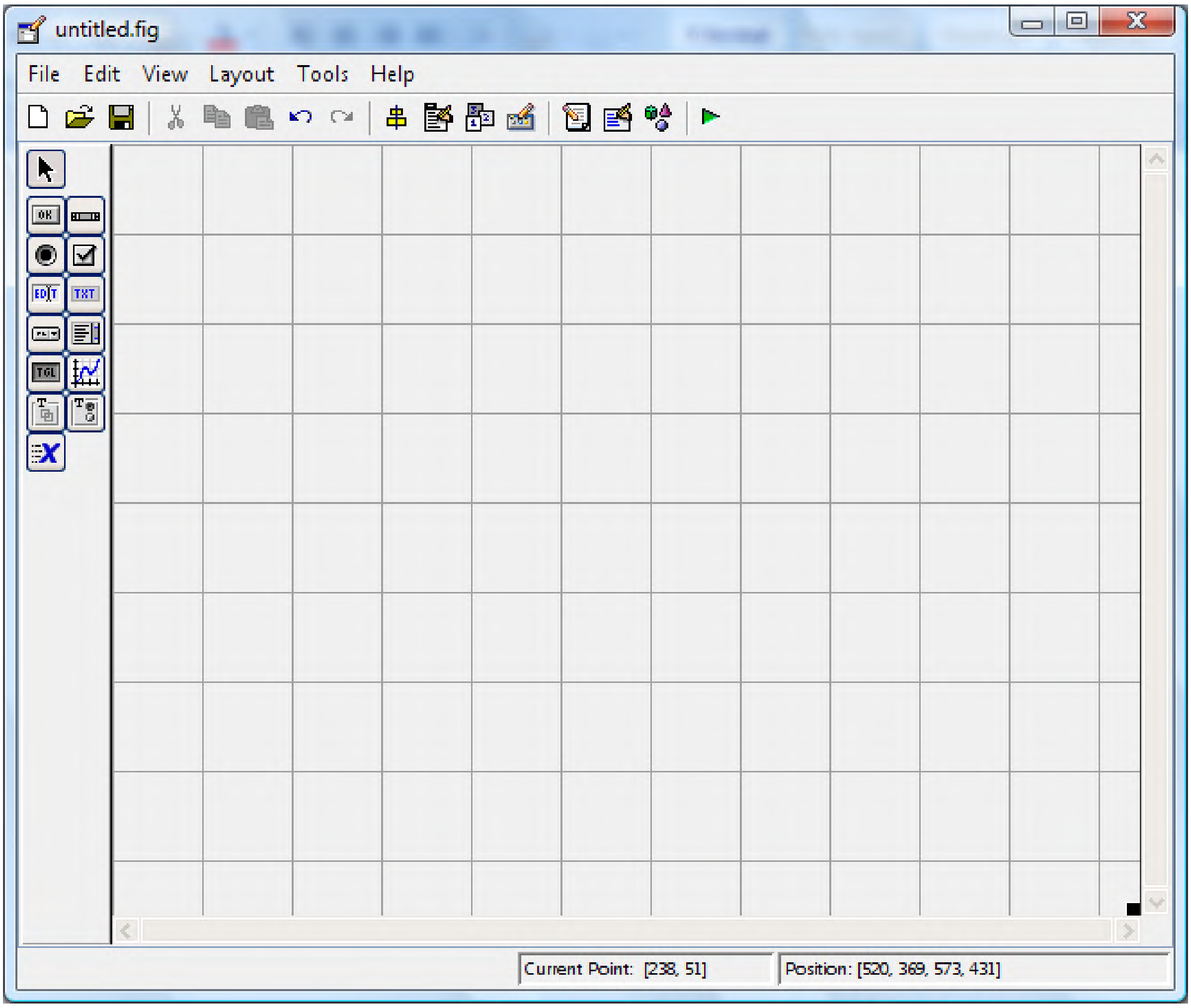


Fig 4.1: GUI

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

Getting Started

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

Introduction - describes the components of the MATLAB system.

Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

DEVELOPMENT ENVIRONMENT

4.5.1Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop. On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB. The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, andresizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History

- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example, !emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentations for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type help browser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Content's tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favourite's tab - View a list of documents you previously designated as favourites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favourites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and who's.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace as from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a.mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to run MATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

MANIPULATING MATRICES

Entering Matrices

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Dürer's matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

MATLAB displays the matrix you just entered.

```
A =

16    3    2   13

5   10   11    8

9    6    7   12

4   15   14    1
```

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to itsimply as A.

Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3 -99 0.0001

9.6397238 1.60210e-20 6.02252e23

1i -3.14159j 3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

Operators

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

Table 4.1: Operators

Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built in. They are part of the MATLAB core, so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sign, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.14159265...
I	Imaginary unit, $\sqrt{-1}$
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

Table 4.2: Functions

GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied, and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or howthey work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

GUI Development Environment

The process of implementing a GUI involves two basic tasks.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of its

children (uicontrols and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the

callbacks, which are defined as subfunctions. This M-file is referred to as the

application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

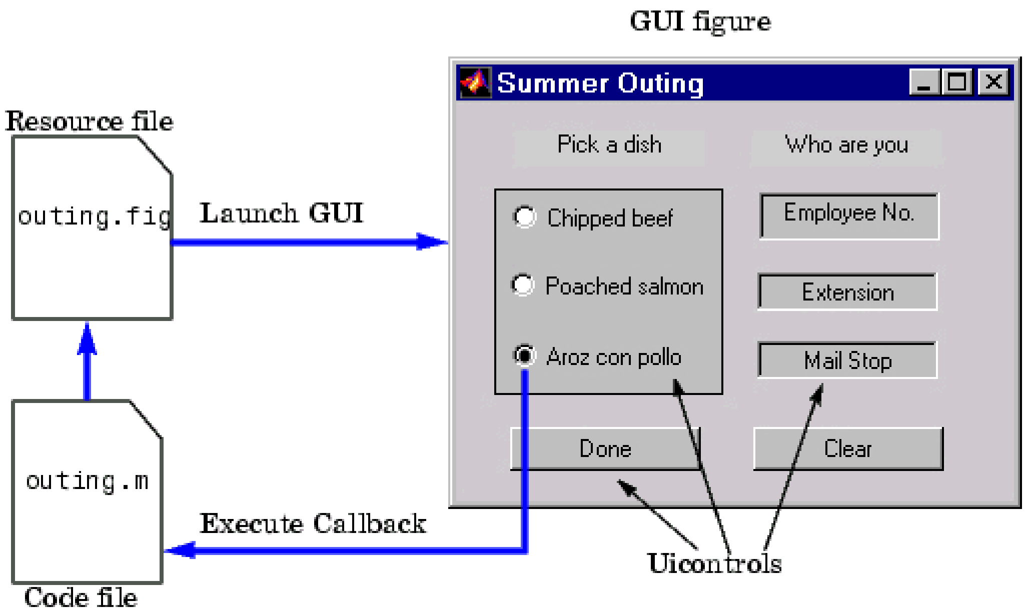


FIG 3.7.2 graphical user blocks

Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques used in the application M-file.

Application Examples - a collection of examples that illustrate techniques which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- Listboxes

- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle buttonis depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)

button_state = get(h,'Value');

if button_state == get(h,'Max')

% toggle button is pressed

elseif button_state == get(h,'Min')

% toggle button is not pressed

end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);

a(:,:,2) = rand(16,128);

a(:,:,3) = rand(16,128);

set(h,'CData',a)
```

Radio Buttons

Radio buttons are similar to checkboxes but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behaviour

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

function mutual_exclude(off)

set(off,'Value',0)

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

This structure is an input argument to all radio button callbacks.

function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)

off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];

mutual_exclude(off)

% Continue with callback

.

.

.

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked ornot checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

function checkbox1_Callback(h,eventdata,handles,varargin)

if (get(h,'Value') == get(h,'Max'))

% then checkbox is checked-take approriate action

else

% checkbox is not checked-take approriate action

end

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

function edittext1_Callback(h,eventdata, handles,varargin)

user_string = get(h,'string');

% proceed with callback...

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback (h, eventdata,handles,varargin)
user_entry = str2double(get(h,'string'));
if isnan(user_entry)
errordlg('You must enter a numeric value','Bad Input','modal')
end
% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menu bar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menu bar does not cause the editable text call back routine to execute. This behaviour is consistent with the respective platform conventions. Clicking on other components in the GUI execute the call back.

Static Text

Static text controls display lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the call back routine associated with it.

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no call back routines associated with them and only ui controls can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The List box Top property defines which string in the list displays as the topmost item when the list box is not large enough to display all list entries. List box Top is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Non integer values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

Listboxes differentiate between single and double clicks on an item and set the figure Selection Type property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure SelectionType property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure, you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback (h, eventdata,handles,varargin)

val = get(h,'Value');

switch val

case 1

% The user selected the first item

case 2

% The user selected the second item

% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback (h, eventdata,handles,varargin)
val = get(h,'Value');
string_list = get(h,'String');
selected_string = string_list{val}; % convert from cell array to string
% etc.
```

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled, and its label (set by the string property) is grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uicontrol objects but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes ButtonDownFcn property to define the callback.

Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

axes (handles. axes1)

makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to targeta different axes. See GUI with Multiple Axes for and example that uses two axes.

You can do this using the axes command and the handles structure.

Figure

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control.

Chapter 5

RESULTS

- At first we will get the input image as ouput in the first figure widow. Now we can see some noise in the image.

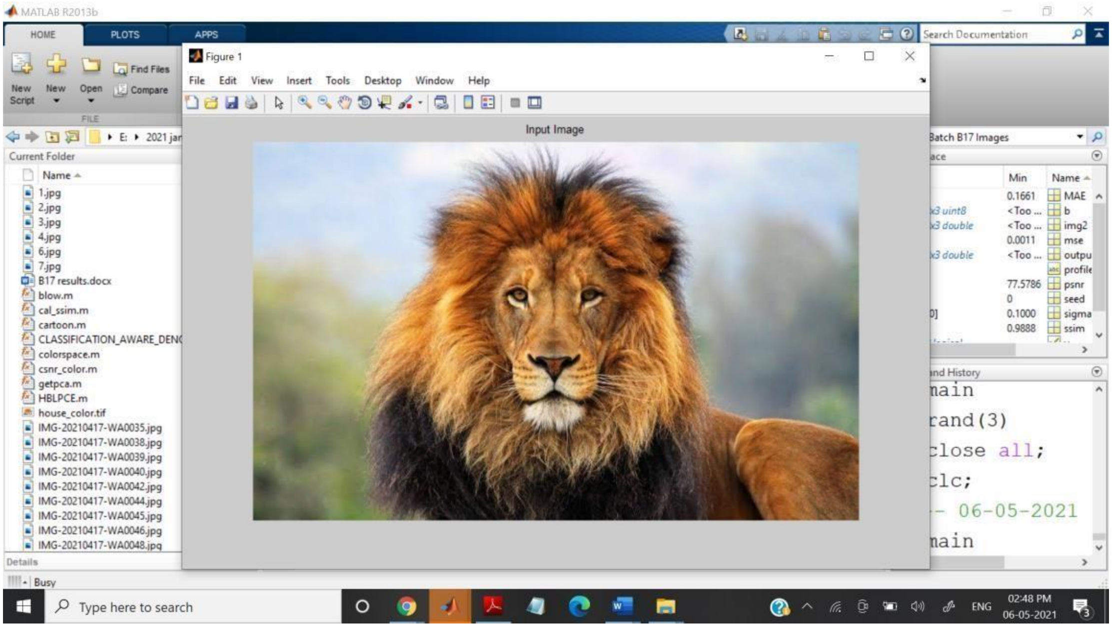


Fig 5.1:Input Image

- In the second figure the histogram equalization occurs for the given image

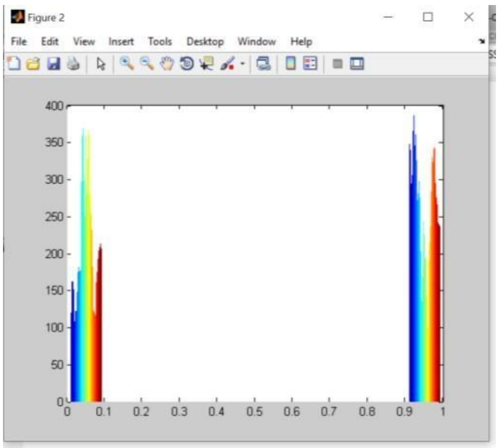


Fig 5.2:Histogram Equaliztion

- In the third figure window left side image is the noise added image which is more than the input image.

5.1 Final Output:

In the right side image i.e., Classification aware denoising image output this is the final output which is an required image

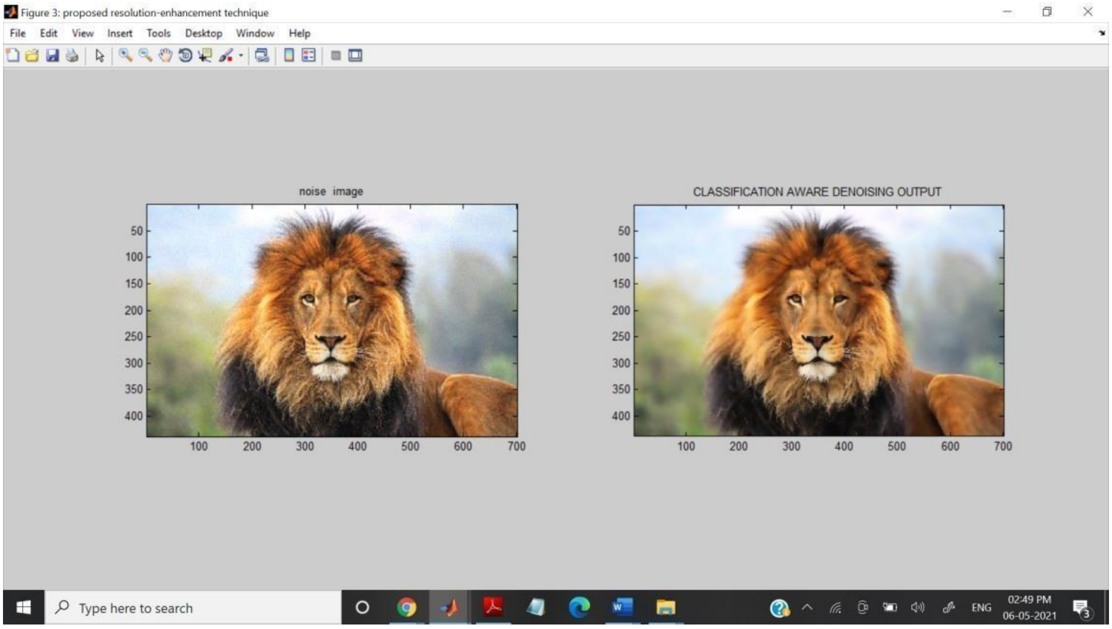


Fig 5.3: a) Noised Image

b) Classification aware denoising image

Chapter 6

LIMITATIONS & ADVANTAGES

Limitations:

- Requires a large amount of data — if you only have thousands of example, deep learning is unlikely to outperform other approaches.
- The disadvantage is that the amount of operations of such a model is comparatively huge. It takes a lot of time not only to train a model, but also to use it. Unless you can speed the computation up using e.g. CUDA cores of NVidia GPU, it's not very practical. Even with CUDA technology, you may probably forget e.g. real-time processing of video frames. So if you are able to train a simpler model and reach good enough.
- CNN do not encode the position and orientation of object. Lack of ability to be spatially invariant to the input data. Lots of training data is required.

Advantages:

- The main advantage of DNN compared to its predecessors is that it automatically detects the important features without any human supervision.
- The huge advantage of multi-layered models such as the DNNs is that on each layer, different level of visual information is processed.
- Lower layers process (detect) very local features; just small parts of curves etc. The higher you get, the more complex features are concerned. And you can still interpret the functionality of the network relatively well.
- Commonly, to adapt a model to new purpose, its lower layers are kept and you only train the higher ones to infer the features for the particular case. That speeds the training up a lot.

Chapter 7

CONCLUSION & FUTURE ENHANCEMENT

CONCLUSION:

- In this paper, we proposed a new method for training an image-classification-aware denoising method that requires only a very small subset of the classes in the original dataset.
- The experimental results show that the proposed method drastically improves the classification accuracy under noisy conditions even when we apply the denoised network to a dataset containing completely unseen classes.
- Our method provides a plug-and-play denoising network for image classification networks in various severe environments without retraining the whole classification network.

FUTURE ENHANCEMENT:

- The increasing demand for high image quality in mobile devices brings forth the need for better computational enhancement techniques, and image denoising in particular.

CODE

```
%%%%%FEW-CLASS LEARNING FOR IMAGE-CLASSIFICATION-
AWARE DENOISING
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%Impulsive Noise

clc;
clear all;
close all;

profile    = 'fast';
seed       = 0;
randn('state', seed );

% [filename, pathname] = uigetfile('*.*','*.png' ,'Pick an Image');
%     if isequal(filename,0) || isequal(pathname,0)
%        warndlg('User pressed cancel')
%     else
%
%         a=imread(filename);
%          if size(a,3)==3
%         b=double(a);
%         figure,imshow(mat2gray(b)),title('Input Color Image');
%          else
%               b=double(a);
%         figure,imshow(b),title('Input Gray Scale Image');
%          end
%     end
     b=imread('3.jpg');
    figure,imshow(b),title('Input Image');
    %%%%%%%%%% histogram
technique%%%%%%%%%%%%%%%%%%
    v=im2bw(b);
     figure,hist(v);

clc;

  img2 = double(imread('3.jpg'))/255;
  img2 = double(img2);
img2 = img2+0.09*rand(size(img2));
```

```
img2(img2<0) = 0;
img2(img2>1) = 1;

% Set parameters.
w     = 5;       % matrix width
sigma = [3 0.1]; % standard deviations
output_super=kron(w,sigma);
%CLASSIFICATION_AWARE_DENOISING
output_super =
CLASSIFICATION_AWARE_DENOISING(img2,w,sigma);

figure,; clf;
set(gcf,'Name','proposed resolution-enhancement technique');
subplot(1,2,1); imagesc(mat2gray(img2));
axis image; colormap gray;
title('noise  image');
subplot(1,2,2); imagesc(mat2gray(output_super));
axis image; title('CLASSIFICATION AWARE DENOISING OUTPUT');
drawnow;

   [psnr,mse,MAE,ssim] = measerr(img2,output_super);
   disp('display PSNR value')
    fprintf('\npsnr: %7.2f ', psnr(:,:,1));
 fprintf('\nmse: %9.7f ', mse(:,:,1));
   fprintf('\nMAE: %7.2f ', MAE(:,:,1));
 fprintf('\nssim: %9.7f ', ssim(:,:,1));
```

REFERENCES

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in International Conference on Learning Representations, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in IEEE Conference on Computer Vision and Pattern Recognition, June 2016, pp. 770–778.

[3] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2018, pp. 7132– 7141.

[4] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3142–3155, July 2017.

[5] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for cnn-based image denoising," IEEE Transactions on Image Processing, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

[6] T. Remez, O. Litany, R. Giryes, and A. M. Bronstein, "Class-aware fully convolutional gaussian and poisson denoising," IEEE Transactions on Image Processing, vol. 27, no. 11, pp. 5707–5722, Nov. 2018.

[7] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2018, pp. 2472–2481.

[8] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in Proceedings of International Conference on Computer Vision, 2017.

[9] B. V. Somasundaran, R. Soundararajan, and S. Biswas, "Image denoising for image retrieval by cascading a deep quality assessment network," in IEEE International Conference on Image Processing, Oct. 2018, pp. 525–529.

A

PROJECT REPORT

On

ADVANCED RAILWAY TRACK FAULT DETECTION AND REPORTING OVER INTERNET OF THINGS(IOT)

Submitted by

Mr. K. Vasanth Kumar (17K81A0487)

Ms. P. Susmitha (17K81A04A1)

Ms. T. Navya Sree (17K81A04B5)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

Electronics And Communication Engineering

Under the Guidance of

Mr. P. Pavan Kumar

Assistant Professor

DEPARTMENT OF ECE

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

BONAFIDE CERTIFICATE

This is to certify that the project entitled "**Advanced Railway Track Fault Detection and Reporting Over Internet OF Things (IoT)**", is being submitted by **Mr. K. Vasanth Kumar** (17K81A0487), **Ms. P. Susmitha** (17K81A04A1), **Ms. T. Navya Sree** (17K81A04B5) in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Head of the Department

Mr. Pavan Kumar
Department of ECE

Dr. B. Hari Krishna
Department of ECE

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT K. VASANTH WITH ROLL NO.17K81A0487, P.SUSMITHA WITH ROLL NO.17K81A04A1, T.NAVYA WITH ROLL NO.17K81A04B5, OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "ADVANCED RAILWAY TRACK FAULT DETECTION AND REPORTING OVER INTERNET OF THINGS" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT

DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of **Bachelor of Technology** in Department of Electronics and Communication Engineering, session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled **Advanced Railway Track Fault Detection and Reporting Over Internet OF Things (IoT)** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

K. Vasanth Kumar	P. Susmitha	T. Navya Sree
17K81A0487	17K81A04A1	17K81A04B5

Place: Secunderabad
Date:

// ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr. P. Santosh Kumar Patra**, **our beloved Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr. B. Hari Krishna**, **Head &Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept. Project coordinator Mr.K.Karthik** for his involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time to come.

Finally, we are highly grateful and obliged to our college level Project Guide **Mr. P. Pavan Kumar**, **Assistant Professor in ECE** advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

ABSTRACT

This project concern to a process for monitoring the condition of train tracks and more specifically has the object of the identification of defects detected by monitoring equipment on the tracks to be examine to allow maintenance crews to subsequently find these defects. When the supply is given to the device, the DC motor gets start through relay driver circuit. Two IR sensors are fixed in front of the train which is used to find out the crack on the track. Each sensor will produce the signal related to the position with the rail. If the track position is normal both the sensor gives the constant sensed output. If any sensor misses their output condition to fail then there is defect on that side. The sensor will inform this by giving alarm and sends information to the smart phone android app in IOT based via WIFI Module. Ultrasonic sensor is used to detect the obstacle in the track if any obstacle detected means the Arduino automatically transmits the message to nearby station via IOT using WIFI module.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

OVERVIEW OF THE PROJECT

In Indian railway scenario, most commercialized transport is being carried out by the railway network and therefore any problem in the same has the capacity to induce major damage to the economy-notwithstanding the societal effect of loss of life. This paper put forward a cost effective yet robust solution to the problem of railway crack detection utilizing a method that is unique in the sense. This project discusses the technical and design aspects in detail and also provides the proposed robust crack detection.

Present industry is increasingly shifting towards automation. The two basis components of current industrial automations are programmable controller and robot. In order to support the monotonous work and to serve the mankind, there is a general tendency to develop an intelligent operation.PIC Microcontroller is the heart of the device which handles all the sub devices connected over it. The microcontroller has been used here. It has flash type reprogrammable memory and some peripheral devices to play this project perform. IR sensor is used to detect the crack in rail road, when the crack is detected motor of the train will stop thus the train stops and this information is sent as a message to nearby station by using Wi-Fi module. Then Ultrasonic is used for the surveying process. Then other important component is PIR sensor it is used to detect the presence of humans in track. A practical methodology to non-destructively localize cracks and estimate the sizes of the cracks in beam-type structures using changes in frequencies. Utilizing simple components like IR Transmitter and Receiver based crack detector assembly is very useful in railway crack detection.

Crack detection method has been explored which uses microwave sensors to observe the rail surface. When a cracked rail is encountered along the route, the circuit will be interrupted or open, resulting in a changed flux pattern. Crack detection method in railway system utilizes the change in infrared emission of the rail surface during the passage of a train wheel. Initial data from this infrared method are dispense from studies of both laboratory-based three-point bend specimen and a short section of rail. This paper proposes a work of detecting the crack in the railway track using IOT and also presents the details of the implementation results of the robust crack detection system utilizing simple components inclusive of a Wi-fi module and IR, Ultrasonic based crack detector assembly. The proposed methodology has been designed for robust implementation in the Indian scenario.

STATEMENT OF THE PROJECT

Indian Railways have been in the news albeit for wrong reasons. With the rapid increase in passenger and goods traffic, the frequency of train accidents is increasing very fast. This has raised serious doubts in the public mind about safety of Rail travel and the general health of the railway network.

The credibility of an organisation with a long and proud history of nation building seriously eroded. In such a situation it is but natural to ask where the Railways are heading. On an average the Railways report 20 major collisions, 350 derailments and around 80 level crossing accidents in a year.

There are several factors which are responsible for increasing number of railway accidents; some outstanding being gaps or cracks in the railway tracks and also some of them will try to cross the railway track without noticing the train and thus accidents happen.

Worried about the increasing rate of accidents and loss of life and property resulting from these accidents. Railway is one of the most significant transportation modes of our country but it is a matter of great sorrow that, railway tracks of our country are very prone. The cracks and other problems with the rails generally go unnoticed due to improper maintenance and irregular manual track line monitoring that is being carried out in the current situation.

Nowadays system have some limitations, if the bridge or track damaged, that information goes to railway authority people, they notifies and informs to the corresponding trains it will takes more time informing those information.

AIMS AND OBHECTIVE OF THE STUDY

The main objective is to define any railway track fault using this system, which is implemented in effective and will also function efficiently. This method will be helpful in regular track checking as it is more convenient than the handheld checking system. The current system has a railway labourers walking on the railway tracks and detecting the fault manually. In this project we are using arduino for control action to make system we are connecting Ultrasonic sensors and IR sensors to detect the obstacle and we are connecting dc motors, wi-fi module is used. This requires a lot of time and labour. So we are using a Ultrasonic sensors and IR sensors for railway track crack detection. The testing vehicle consists of motors driven by a motor driver. The IR, and Ultrasonic sensor which is connected to the Arduino. Message will be sent to the nearby station i.e through telnet app via wi-fi module

SCOPE OF THE STUDY

Through our proposed system, we need to establish more modern and secure railway system. Present industry is increasingly shifting towards automation. The two basis components of current industrial automations are programmable controller and robot. The microcontroller has been used here. IR sensor is used to detect the crack in rail road. The Ultrasonic sensor is used to detect the object and at what distance the object is from rail.

The main scope of the project is to detect mis-alignment and the gaps on the track. It also detect the object present in-front of the train at a distance. The mis-alignments and gaps are detected using IR sensors. Similarly the objects in-front of the train is identified using Ultrasonic sensor. By this we can reduce the physical errors present on the track.

MATERIAL REQUIREMENT

HARDWARE REQUIREMENT

- Arduino UNO
- Ultrasonic Sensor
- IR sensor

- L293D Motor driver
- Motor
- WI-FI Module
- Liquid Crystal Display
- Power Supply
- Battery

SOFTWARE REQUIREMENT

- Arduino IDE
- Proteus Software

PROCUMENT OF EQUIPMENT

We brought all the required hardware components in a electronic manufacturing shop and before assembling and interfacing hardware components to Arduino UNO microcontroller, we connected individual sensors to Arduino in Proteus Software and understood working of every individual sensor in Proteus Software.

Now, we assembled all sensors and interfaced to Arduino UNO micro-controller using connecting wires through bread board and we wrote program to work all the hardware sensors in Arduino IDE, we understood the working of all sensors in a practical way

We calculated distance of object i.e object detection and gap i.e crack detection and displayed on LCD and also updated on Server i.e in Telnet app. So, it will be visible to Railway Authority. If sensors detects object or any gap in the railway track then the train will stop.

CHAPTER 2
LITERATURE SURVEY

LITERATURE REVIEW ON RESEARCH AREA

The main objective of the track crack detection and health monitoring in base station between two signals any track is broken, cut the track, and any fault means signal is transmitted to signal engineer, because 5volt power will be passing to track. This technique used only for base station. Microcontroller: ATmega162 The ATmega162 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega162 achieves throughputs approaching 1 MIPS per MHz allowing the system designed to optimize power consumption versus processing speed. IR sensors are also used to identify the crack in the railway.

Existing System: According to Indian Railways currently the method used in railways for crack detection is an inspection by foot, Trollies, locomotives, and rear vehicles. There are two types of track recording car one mechanical and the other electronic. These cars make a continuous record of track geometry under load conditions. The car moves on the track and checks if there is any wear and tear on the tracks. They also use IR sensor, but it is not very accurate because manual work is not possible for every railway network. A system gang patrolling is also used which is implemented by foot along the tracks during the rainy season. Saurabh Srivastava and his teammates have proposed a vehicle for detecting crack using IR sensor array assembly which tracks the exact location of faulty tracks and informs to the nearby railway station. The team from SRM group of colleges Lucknow has proposed a system that detects cracks using Arduino mega power by solar panel along with the laser source. They have also used GSM and GPS module to give an exact location of the crack to authorities via SMS and a link to open location on Google Maps. Rizvi Aliza Raza has presented a model in IJARIIT which is capable of taking pictures of the track and compare it with the old database and also sends a message to the authorities about the fault detected.

REVIEW ON RELATED LITERATURE

The cracks and other problems with the rails generally go unnoticed due to improper maintenance and irregular manual track line monitoring that is being carried out in the current situation. Nowadays system have some limitations, if the bridge or track damaged, that information goes to railway authority people, they notify and informs to the corresponding trains it will take more time informing the information. In the literature survey, the commonly employed rail crack detection schemes in foreign countries are usually ultrasonic or eddy current based techniques which are the reasonably good accuracy in most cases. However, the one characteristic which the above-mentioned methods have in common is that they are both expensive, which makes them ineligible for implementation in the current Indian scenario. Also, the ultrasonic can only inspect the core of materials; that is, the method cannot check for surface and near surface cracking where many of the faults are located. Many of the most serious defects that can develop in the rail head can be very difficult to detect using the currently available inspection equipment. This system is mainly concerned in identifying the cracks in railway tracks and helps to prevent the accidents without manual power. It’s not only concentrated on finding damaged tracks but also helpful to find out the derailment and the exact place where it is. In this technical solution offered by many companies in the detection of cracks in rails involve periodic maintenance coupled with occasional monitoring usually once a month or in a similar time frame. But the robotics possesses the inherent advantage of facilitating monitoring of rail tracks on a daily basis during nights, when the usual train traffic is suspended. Further, that the simplicity of this idea and easy availability of the components make for implementation on a large scale with very little initial investment.

CONCLUSIONS ON REVIEWS

The simplicity of this work ensures robustness of operation and also the design has been carefully modified to permit rugged operation. Another disadvantage that can be attributed to the conventional commercially available testing equipment’s is that they are heavy which poses a practical limitation. This important disadvantage has been rectified in robotics project as the design is simple and sensible enabling the device to be easily portable. While designing the mechanical parts of the robot, due consideration has been given to the variable nature of the tracks and the unique challenges possessed by the deviations in the Indian scenario. For example, in areas near road crossings the outer part of the track is usually covered with cement. Also, there is always the problem of rocks obstructing the path on the inside parts of the rails. So the specialized wheels that have been provided in robot that has taken into account and are specifically designed to overcome this aforementioned problem. The railway track crack detection is used to detect the crack whiles the train running on the track. The proposed system is used to detect the crack on railway track before 10km.

CHAPTER 3

PROJECT DESIGN

BLOCK DIAGRAM OF THE PROJECT

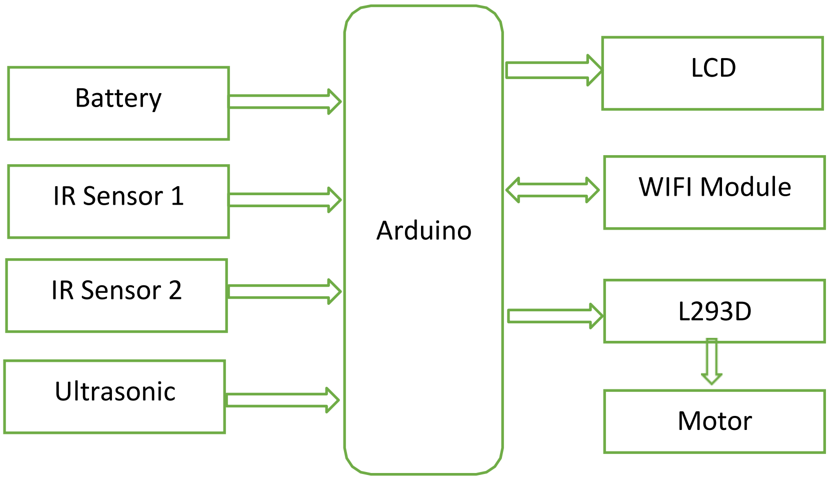


Fig.3.1 Block diagram of the project

HARDWARE REQUIREMENT

- Arduino UNO
- ESP8266 Wi-Fi Module
- Liquid Crystal Display
- Ultrasonic Sensor
- IR Sensor
- L293D Driver motor
- Motor

ARDUINO UNO

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The ATmega8 microcontroller contains 32 general purpose working registers. As shown in the below figure these registers are directly connected to

ALU. Two registers can carry one single instruction consequently in one clock cycle.

Specifications:

Table 3.3: Atmega328 specifications

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 Ma
DC Current on 3.3V Pin	50 Ma
Flash Memory	32 KB (0.5 KB is used for Boot loader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready- made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board. The key features are −

- Arduino boards are able to read Analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).

- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.

BOARD DESCRIPTION

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

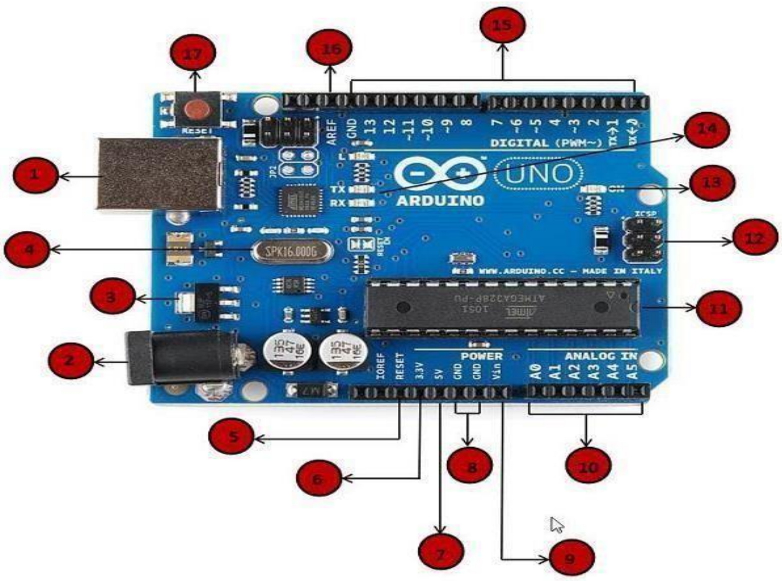


Fig 3.3.1 Board Description

TABLE 3.3.1 Board Description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit.

	•Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

PIN DESCRIPTION OF ATMEGA328

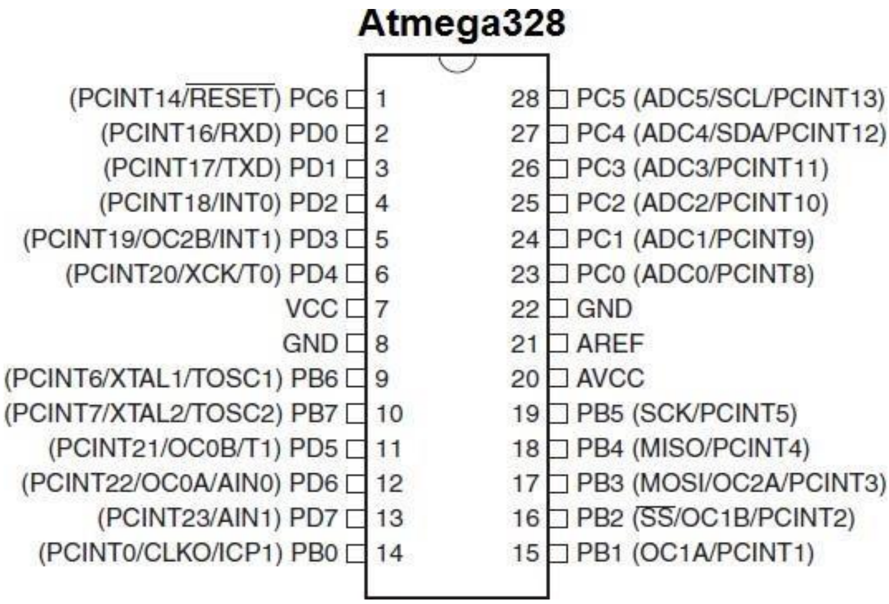


Fig 3.3.2 Pin description pf ATMEGA328

ADVANTAGES OF ARDUINO

- It is cheap
- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the

prevailing programming language libraries and may be extended and changed. For beginners, it is very simple to use.

APPLICATIONS

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model .

WI-FI MODULE

The ESP8266 is a low-cost Wi-Fi module that can be integrated easily into IoT devices. We've featured several projects using this module, such as **How To Make Smart Home Electronics: A Smart Mailbox** and **How To Read Your Arduino's Mind: Building A Childproof Lock**. This tutorial will walk you through setting up ESP8266 Wi-fi module which can be used with Arduino. The ESP8266 comes in many models with different functionalities. We'll be focusing on the ESP8266 ESP-01 module, the most common and basic one available.

What is ESP8266?

The ESP8266 is a small Wi-Fi module built around the ESP8266 chip that can connect your microcontroller to the internet wirelessly for a very small cost. It can be a great option for Internet of Things (IoT) projects, but can be difficult to work with for beginner hobbyists who do not have prior experience with the module. In this tutorial, we hope to show you how to interface the ESP8266 with an Arduino and perform some basic functions like connecting it to a Wi-Fi network.

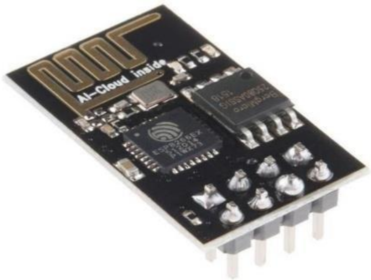


Fig 3.4.1 ESP8266 Wi-Fi Module

FEATURES OF ESP8266

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 1MB Flash Memory
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)

The first feature to notice about the ESP8266 is its awkwardly spaced header pins. The module has 8 pins that serve different functions, but they are packed in a 4×2 arrangement that makes plugging the module into a breadboard impossible. This means that to prototype projects on a breadboard, you'll need male-female jumper wires to connect the pins on the ESP8266 to rows on the breadboard. If you'd like to make your prototyping more compact, you can also purchase breadboard breakouts for the ESP8266 such as thisone. For prototyping, I chose to just use jumper wires.

STRUCTURE AND PIN CONFIGURATION

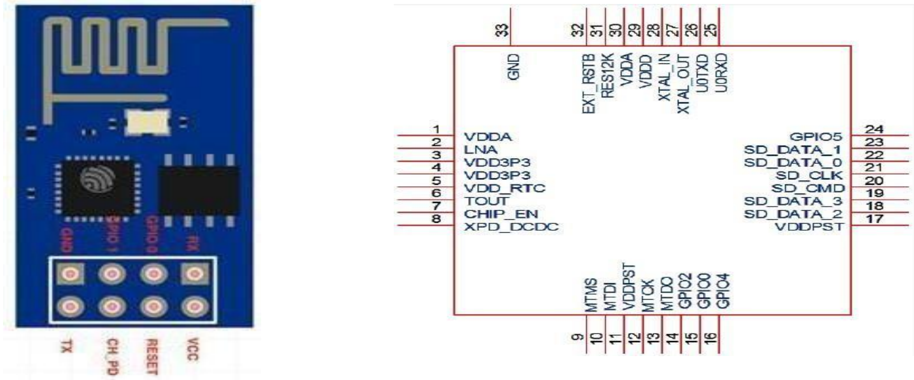


Fig 3.4.3 structure and pin configuration

RX: UART serial communication receive pin

GPIO 0: GPIO pin (unused in this project)

GPIO 1: GPIO pin (unused in this project)

GND: Connection to Ground

VCC: Connection to 3.3V VCC (VCC cannot exceed 3.3V!)

RESET: Reset pin (pull down to reset)

CH_PD: Chip enable and power down pin

TX: UART serial communication transmit pin

Note that the maximum voltage input for the ESP8266 is 3.3V. Any input voltage greater than 3.3V will damage the module! To program settings on the ESP8266, we'll first need to connect it to a serial terminal on a computer through which we can send it special commands. Settings that we'll have to program include, for example, the SSID and password for the wi-fi network the module will be connected to. To connect the ESP8266 to a computer and configure its settings, we'll need a USB to serial adapter with 3.3V logic, along with a serial terminal program. Fortunately for us, we have the Arduino and the Arduino IDE's serial monitor! This means that we'll just have to connect the ESP8266 module to the Arduino and upload a custom sketch to the Arduino.

HOW TO CONNECT ESP8266 WI-FI MODULE

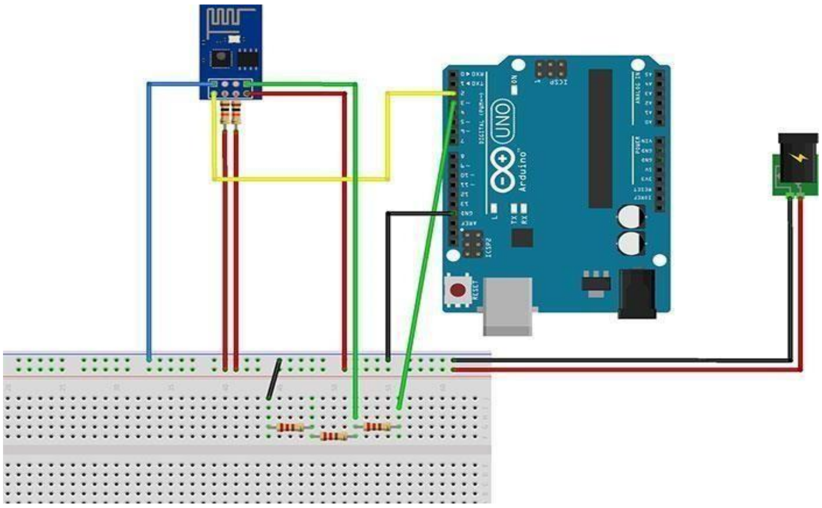


Fig 3.4.4 connection of ESP8266 wi-fi module

You will need the following parts:

- Arduino Uno or similar module
- ESP8266 ESP-01 module
- 2 x 1kΩ resistors for CH_PD and RESET pull-up
- 3 x 220Ω resistors for serial line voltage divider
- 3.3V regulated power supply for ESP8266 (you cannot plug the power line into the Arduino's 3.3V out pin because the ESP8266 draws a lot of current!)
- Breadboard and jumper wires

A couple of features of this circuit stand out immediately.

First, the ESP8266 module is powered by a 3.3V regulated power supply. As described earlier, the ESP8266 has a maximum voltage input rating of 3.3 volts. Using a power supply rated higher than this recommended 3.3 volts could fry the module! When connecting the ESP8266 to this power supply, you should not only connect the VCC and Ground pins to the power supply lines, but also connect the power supply's ground line to the Arduino's ground pin. Since the Arduino will be powered by the USB connection to the laptop, creating a common ground essentially creates a common reference (you can think of it as a baseline) to compare voltages and thereby interpret digital high and low signals.

Second, the receive (RX) line for the ESP8266 module is connected to the output of a resistor voltage-divider circuit. We do this to shift the serial communication logic level (the highs and lows of the digital signals that make up the serial communications) from a logic high of 5 volts on the Arduino to a logic high of 3.3 volts on the ESP8266 module. Again, the ESP8266 is specified with 3.3V logic, so connecting the module's receive line directly to the Arduino's transmit line could damage the device. We do not need to shift levels on the ESP8266's transmit line because the module's 3.3V logic high is a high enough voltage to also register as a logic high on the Arduino.

Finally, there are two pull-up resistors on the ESP8266's CH_PD and RESET pins. These are specified by the module's creators to ensure that the device functions properly while it is being used. To reset the ESP8266 or disable it, these pins must be pulled down, but since we do not want those processes to occur, we will instead pull these pins high through some 10kΩ pull-up resistors.

APPLICATIONS

- IOT Projects
- Access Point Portals
- Wireless Data logging
- Smart Home Automation
- Learn basics of networking
- Portable Electronics
- Smart bulbs and Sockets

16 * 2 ALPHANUMERIC LCD

WHAT IS LCD

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

BLOCK DIAGRAM

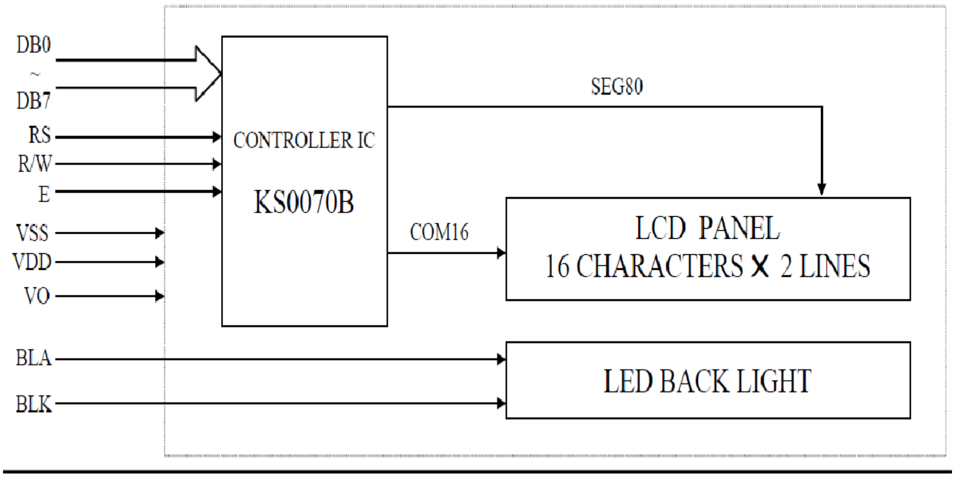


Fig 3.5.2 Block diagram of 16*2 LCD

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.

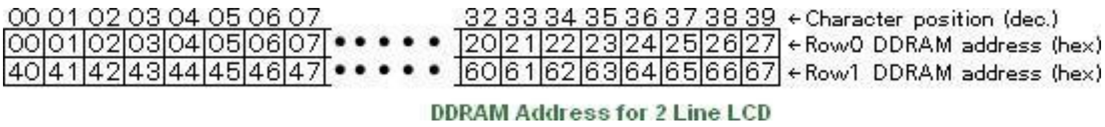


Fig.3.5.3 DDRAM addresses of 2 Line LCD

CHARACTER GENERATOR ROM

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM.

BUSY FLAG

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e. BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

INSTRUCTION REGISTER (IR) AND DATA REGISTER (DR)

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g. LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD. Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

16 X 2 ALPHANUMERIC LCD MODULE FEATURES

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

SCHEMATIC DIAGRAM

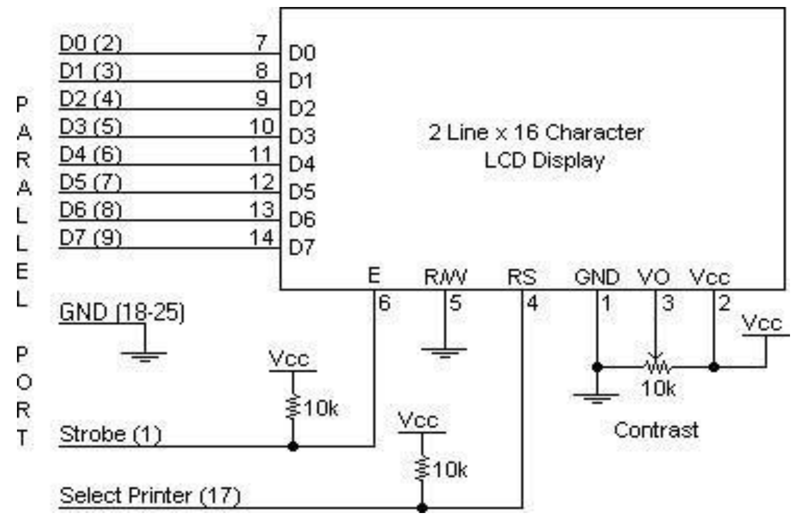


Fig 3.5.7 Schematic Diagram of LCD

SPECIFICATIONS

Table 3.5.8 Specifications

Connector Pin Assignment:

Pin	*Symbol*	*Function*	*Pin*	*Symbol*	*Function*
14	DB7	Data Bus Line	6	E	Enable
13	DB6	Data Bus Line	5	R/W	Read/Write
12	DB5	Data Bus Line	4	RS	Register select input
11	DB4	Data Bus Line	3	Vcontrast	Contrast (0V for max contrast)
10	DB3	Data Bus Line	2	**GND (0V)**	Power supply
9	DB2	Data Bus Line	1	**Vcc +5V**	Power supply
8	DB1	Data Bus Line			
7	DB0	Data Bus Line			

Note1: Pin 1 is +5V and pin 2 is GND! This is different from most other HD44780 compatible models.
Note2: Pin 14 is marked with a “1” on the back of the PCB. Do not get confused by this.

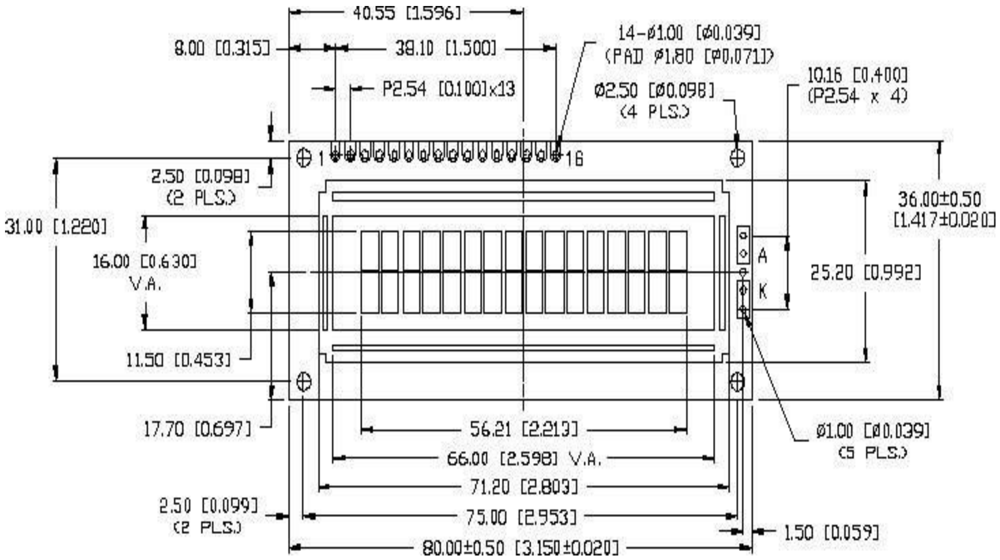


Fig.3.5.9 Connector pin diagram

CIRCUIT DESCRIPTION

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore, by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore, we hard wire the R/W line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result, we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly. The 2-line x 16-character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

APPLICATIONS

- In most of the applications that's have only small values to show, uses the LCD.
- Most of the commercial meters use this module to represent the data output.
- In the toys and developing projects, it is still vastly in use.
- In black and white printers, it helps to show the printer settings and status.

ULTRASONIC SENSOR

WHAT IS ULTRA SONIC SENSOR?

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal.

Fig 3.6.1 Ultrasonic sensor

WORKING PRINCIPLE

As shown above the **HC-SR04 Ultrasonic (US) sensor** is a 4pin module, whose pin names are VCC, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications, where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below

Fig 3.6.2 Ultrasonic sensor working principle

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic-wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

PIN CONFIGURATION

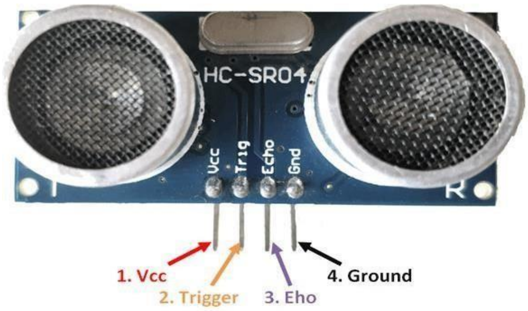


Fig 3.6.3 Pin Configuration

Table 3.6.3 Pin Configuration

Pin Number	Pin Name	Description
1	VCC	The VCC pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.

3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

HOW TO USE THE HC-SR04 ULTRASONIC SENSOR

HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used.

Power the Sensor using a regulated +5V through the VCC ad Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor.

The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information, the distance is measured as explained in the above heading.

SPECIFICATIONS

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz

APPLICATIONS

- Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot path finding robot etc.
- Medical ultrasonography.
- Burglar Alarms.
- Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water

INFRARED SENSOR

WHAT IS IR SENSOR?

IR sensor is an electronic device, that emits the light in order to sense some object of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation. These types of radiations are invisible to our eyes, but infrared sensor can detect these radiations.

Fig 3.7.1 IR Sensor

WORKING PRINCIPLE OF IR SENSOR

There are different types of infrared transmitters depending on their wavelengths, output power and response time. An IR sensor consists of an IR LED and an IR Photodiode, together they are called as Photo-Coupler or Opto-Coupler.

IR Transmitter or IR LED

Infrared Transmitter is a light emitting diode (LED) which emits infrared radiations called as IR LED's. Even though an IR LED looks like a normal LED, the radiation emitted by it is invisible to the human eye.

Fig 3.7.2.1 Infrared LED

IR Receiver or Photodiode

Infrared receivers or infrared sensors detect the radiation from an IR transmitter. IR receivers come in the form of photodiodes and phototransistors. Infrared Photodiodes are different from normal photo diodes as they detect only infrared radiation. Below image shows the picture of an IR receiver or a photodiode,

Fig 3.7.2.2 Infrared Photodiode

Different types of IR receivers exist based on the wavelength, voltage, package, etc. When used in an infrared transmitter – receiver combination, the wavelength of the receiver should match with that of the transmitter.

An Infrared Sensor works in the following sequence:

- IR source (transmitter) is used to emit radiation of required wavelength.
- This radiation reaches the object and is reflected back.
- The reflected radiation is detected by the IR receiver.
- The IR Receiver detected radiation is then further processed based on its intensity. Generally, IR Receiver output is small and amplifiers are used to amplify the detected signal.

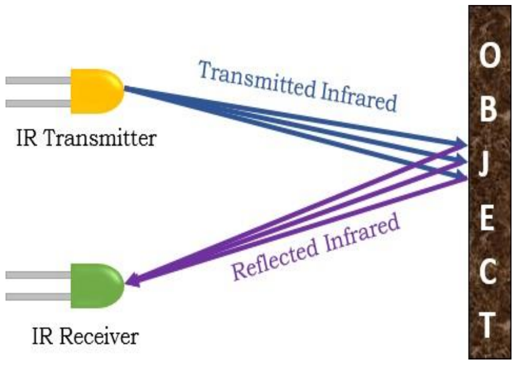


Fig 3.7.2.3 Working of IR Sensor

When the IR transmitter emits radiation, it reaches the object and some of the radiation reflects back to the IR receiver. Based on the intensity of the reception by the IR receiver, the output of the sensor defines.

Principle of operation

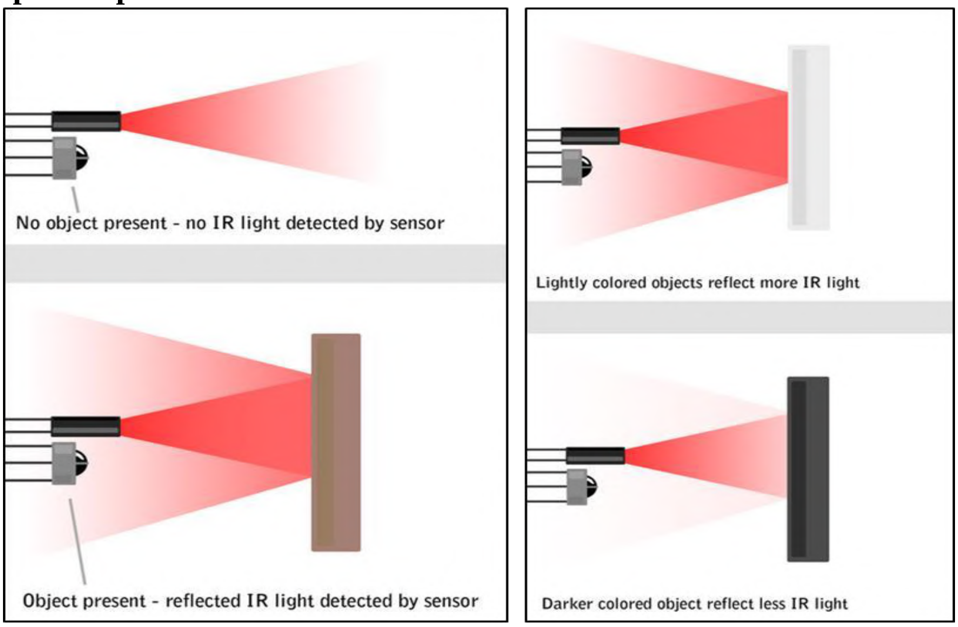


Fig. 3.7.2.4 Principle of operation

The IR transmitter sends an infrared signal that, in case of a reflecting surface (e.g. white color), bounces off in some directions including that of the IR receiver that captures the signal detecting the object.

When the surface is absorbent (e.g., black color) the IR signal isn't reflected and the object cannot be detected by the sensor. This result would occur even if the object is absent.

TYPES OF INFRARED SENSOR

IR sensors can be classified in two types based on presence of IR source:

- Active Infrared Sensor
- Passive Infrared Sensor

Active Infrared Sensor

Active Infrared Sensor contains both transmitter and receiver. Most of the cases LED or laser diode is used as source. LED for non-imaging IR sensor and laser diode for imaging IR sensor are used.

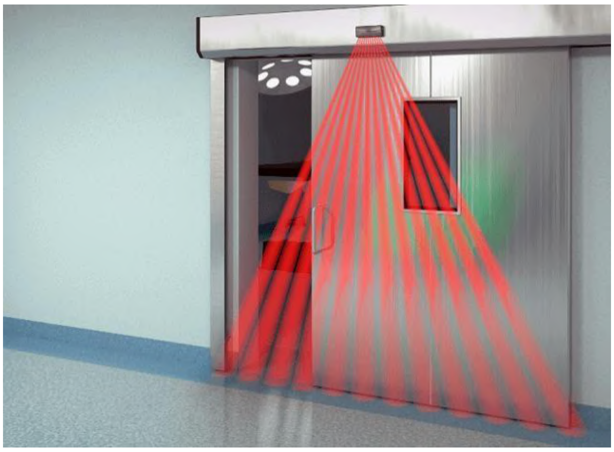


Fig. 3.7.3.1 Active Infrared Sensor

Active IR Sensor works by radiating energy, received and detected by detector and further processed by signal processor in order to fetch information required. Examples of Active IR Sensor: Break Beam Sensor, Reflectance Sensor.

Passive Infrared Sensor

Passive Infrared Sensor contains detectors alone. There won't be a transmitter component. These type of sensors use object as IR source/ transmitter. Object radiates energy and it is detected by IR receivers. A Signal processor is then used to interpret the signal to fetch information required. Example of Passive IR Sensor: Thermocouple-Thermopile, Bolometer, Pyro-Electric Detector, etc.

Fig3.7.3.2 Passive Infrared Sensor

There are two types of Passive Infrared Sensor:

- Thermal Infrared Sensor
- Quantum Infrared Sensor

Thermal Infrared Sensor

Thermal Infrared sensors are independent of wavelength. They use heat as energy source. Thermal detectors are slow with their detection time and response time.

Fig 3.7.3.3 Infrared Sensor as Heat Sensing Device

Quantum Infrared Sensor

Quantum Infrared Sensor are dependent on wavelengths. They have high detection time and response time. These type of IR sensors require frequent cooling for precise measurement.

PIN CONFIGURATION

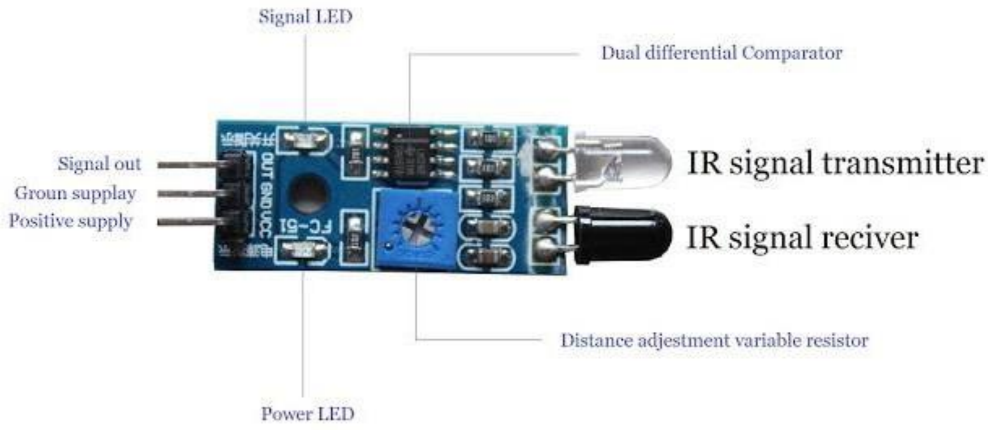


Fig 3.7.4 Pin configuration

Table 3.7.4 Pin Configuration

Pin No.	Pin Name	Description
1	VCC	Power Supply Input
2	GND	Power Supply Ground
3	OUT	Active High Output

SPECIFICATIONS

- 5VDC Operating voltage
- I/O pins are 5V and 3.3V compliant
- Range: Up to 20cm
- Adjustable Sensing range
- Built-in Ambient Light Sensor
- 20mA supply current

Applications of Infrared Sensor

- Proximity Sensor
- Item Counter
- Human Body Detection
- Gas Analyzers
- Water Analysis
- Rail Safety

L293D MOTOR DRIVER

WHAT IS L293D MOTOR DRIVER?

L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motor with a single L293D IC.

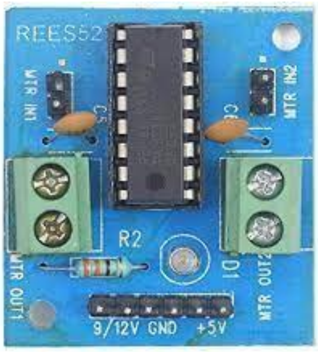


Fig 3.8.1 L293D Motor Driver

WORKING OF L293D

It works on the concept of H-bridge. It is a circuit which allows the voltage to be flown in either direction. As voltage need to change its direction for being able to rotate the motor in clockwise or anticlockwise direction, Hence H-bridge IC are ideal for driving a DC motor.

In a single L293D chip there are two h-Bridge circuit inside the IC which can rotate two dc motor independently. Due its size it is very much used in robotic application for controlling DC motors.

There are two Enable pins on l293d. Pin 1 and pin 9, for being able to drive the motor, the pin 1 and 9 need to be high. For driving the motor with left H-bridge you need to enable pin 1 to high. And for right H-Bridge you need to make the pin 9 to high. If anyone of the either pin1 or pin9 goes low then the motor in the corresponding section will suspend working. It’s like a switch.

NOTE: you can simply connect the pin16 VCC (5v) to pin 1 and pin 9 to make them high.

PIN CONFIGURATION

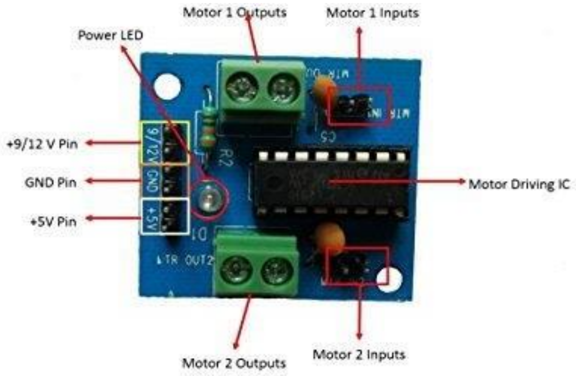


Fig 3.8.3 Pin Configuration

Table 3.8.3 L293D Pin Configuration

Pin Number	Pin Name	Description
1	Enable 1,2	This pin enables the input pin Input 1(2) and Input 2(7)
2	Input 1	Directly controls the Output 1 pin. Controlled by digital circuits
3	Output 1	Connected to one end of Motor 1
4	Ground	Ground pins are connected to ground of circuit (0V)
5	Ground	Ground pins are connected to ground of circuit (0V)
6	Output 2	Connected to another end of Motor 1
7	Input 2	Directly controls the Output 2 pin. Controlled by digital circuits
8	Vcc2 (Vs)	Connected to Voltage pin for running motors (4.5V to 36V)
9	Enable 3,4	This pin enables the input pin Input 3(10) and Input 4(15)
10	Input 3	Directly controls the Output 3 pin. Controlled by digital circuits
11	Output 3	Connected to one end of Motor 2
12	Ground	Ground pins are connected to ground of circuit (0V)
13	Ground	Ground pins are connected to ground of circuit (0V)
14	Output 4	Connected to another end of Motor 2
15	Input 4	Directly controls the Output 4 pin. Controlled by digital circuits
16	Vcc2 (Vss)	Connected to +5V to enable IC function

Where to use L293D IC

The L293D is a popular 16-Pin Motor Driver IC. As the name suggests it is mainly used to drive motors. A single L293D IC is capable of running two DC motors at the same time; also the direction of these two motors can be controlled independently. So if you have motors which has operating voltage less than 36V and operating current less than 600mA, which are to be controlled by digital circuits like Op-Amp, 555 timers, digital gates or even Micron rollers like Arduino, PIC, ARM etcthis IC will be the right choice for you.

HOW TO USE A L293D MOTOR DRIVER IC

Using this L293D motor driver IC is very simple. The IC works on the principle of Half H-Bridge, let us not go too deep into what H-Bridge means, but for now just know that H bridge is a setup which is used to run motors both in clock wise and anticlockwise direction. As said earlier this IC is capable of running two motors at the any direction at the same time.

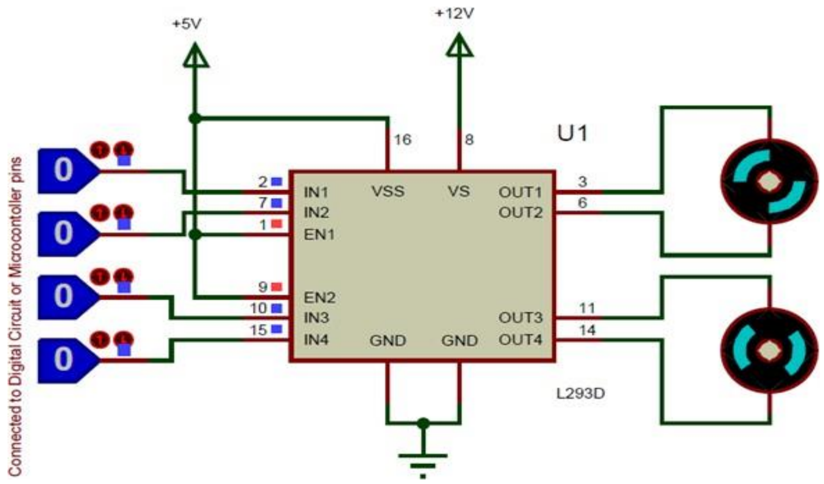


Fig 3.8.4 Motor driver IC

All the Ground pins should be grounded. There are two power pins for this IC, one is the Vss(Vcc1) which provides the voltage for the IC to work, this must be connected to +5V. The other is Vs(Vcc2) which provides voltage for the motors to run, based on the specification of your motor you can connect this pin to anywhere between 4.5V to 36V, here I have connected to +12V.

The Enable pins (Enable 1,2 and Enable 3,4) are used to Enable Input pins for Motor 1 and Motor 2 respectively. Since in most cases we will be using both the motors both the pins are held high by default by connecting to +5V supply. The input pins Input 1,2 are used to control the motor 1 and Input pins 3,4 are used to control the Motor 2. The input pins are connected to the any Digital circuit or microcontroller to control the speed and direction of the motor.

SPECIFICATIONS

- Can be used to run Two DC motors with the same IC.
- Speed and Direction control is possible
- Motor voltage Vcc2 (Vs): 4.5V to 36V
- Maximum Peak motor current: 1.2A
- Maximum Continuous Motor Current: 600mA
- Supply Voltage to Vcc1(vss): 4.5V to 7V
- Transition time: 300ns (at 5Vand 24V)
- Automatic Thermal shutdown is available
- Available in 16-pin DIP, TSSOP, SOIC packages

APPLICATIONS

- Used to drive high current Motors using Digital Circuits
- Can be used to drive Stepper motors
- High current LED’s can be driven
- Relay Driver module (Latching Relay is possible)

MOTORS

WHAT IS MOTOR?

Motor is a device that creates motion, not an engine; it usually refers to either an electrical motor or an internal combustion engine.

It may also refer to:

- Electric motor, a machine that converts electricity into a mechanical motion
- AC motor, an electric motor that is driven by alternating current

- Synchronous motor, an alternating current motor distinguished by a rotor spinning with coils passing magnets at the same rate as the alternating current and resulting magnetic field which drives it
- Induction motor, also called a squirrel-cage motor, a type of asynchronous alternating current motor where power is supplied to the rotating device by means of electromagnetic induction

- DC motor, an electric motor that runs on direct current electricity
- Brushed DC electric motor, an internally commutated electric motor designed to be run from a direct current power source
- Brushless DC motor, a synchronous electric motor which is powered by direct current electricity and has an electronically controlled commutation system, instead of a mechanical commutation system based on brushes
- Electrostatic motor, a type of electric motor based on the attraction and repulsion of electric charge
- Servo motor, an electric motor that operates a servo, commonly used in robotics
- Internal fan-cooled electric motor, an electric motor that is self-cooled by a fan, typically used for motors with a high energy density

TYPES OF MOTORS

Industrial motors come in a variety of basic types. These variations are suitable for many different applications. Naturally, some types of motors are more suited for certain applications than other motor types are. This document will hopefully give some guidance in selecting these motors.

AC Motors

The most common and simple industrial motor is the three phase AC induction motor, sometimes known as the "squirrel cage" motor. Substantial information can be found about any motor by checking its (nameplate).

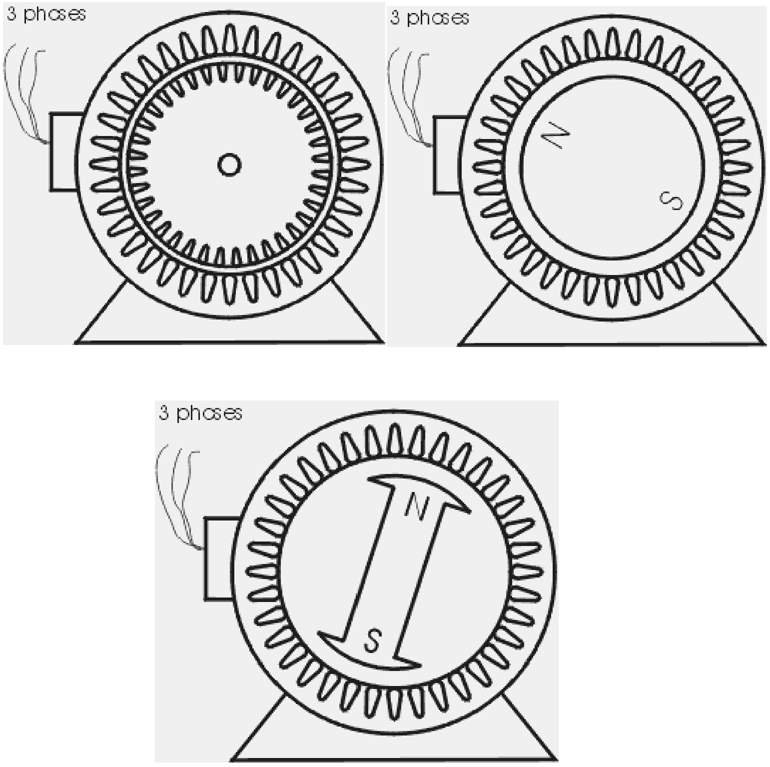


Fig 3.9.2 AC Motors

Advantages

- Simple Design
- Low Cost
- Reliable Operation
- Easily Found Replacements
- Variety of Mounting Styles
- Many Different Environmental Enclosures

Simple Design

The simple design of the AC motor -- simply a series of three windings in the exterior (stator) section with a simple rotating section (rotor). The changing field caused by the 50 or 60 Hertz AC line voltage causes the rotor to rotate around the axis of the motor.

The speed of the AC motor depends only on three variables:

1. The fixed number of winding sets (known as poles) built into the motor, which determines the motor's base speed.
2. The frequency of the AC line voltage. Variable speed drives change this frequency to change the speed of the motor.
3. The amount of torque loading on the motor, which causes slip.

Low Cost

The AC motor has the advantage of being the lowest cost motor for applications requiring more than about 1/2 hp (325 watts) of power. This is due to the simple design of the motor. For this reason, AC motors are overwhelmingly preferred for fixed speed applications in industrial applications and for commercial and domestic applications where AC line power can be easily attached. Over 90% of all motors are AC induction motors. They are found in air conditioners, washers, dryers, industrial machinery, fans, blowers, vacuum cleaners, and many, many other applications.

Reliable Operation

The simple design of the AC motor results in extremely reliable, low maintenance operation. Unlike the DC motor, there are no brushes to replace. If run in the appropriate environment for its enclosure, the AC motor can expect to need new bearings after several years of operation. If the application is well designed, an AC motor may not need new bearings for more than a decade.

Easily Found Replacements

The wide use of the AC motor has resulted in easily found replacements. Many manufacturers adhere to either European (metric) or American (NEMA) standards. (For Replacement Motors)

Variety of Mounting Styles

AC Motors are available in many different mounting styles such as:

- Foot Mount
- C-Face
- Large Flange
- Vertical
- Specialty

DC Motors

The brushed DC motor is one of the earliest motor designs. Today, it is the motor of choice in the majority of variable speed and torque control applications.

Advantages

- Easy to understand design
- Easy to control speed
- Easy to control torque
- Simple, cheap drive design

Easy to understand design

The design of the brushed DC motor is quite simple. A permanent magnetic field is created in the stator by either of two means:

- Permanent magnets
- Electro-magnetic windings

If the field is created by permanent magnets, the motor is said to be a "permanent magnet DC motor" (PMDC). If created by electromagnetic windings, the motor is often said to be a "shunt wound DC motor" (SWDC). Today, because of cost-effectiveness and reliability, the PMDC motor is the motor of choice for applications involving fractional horsepower DC motors, as well as most applications up to about three horsepower.

At five horsepower and greater, various forms of the shunt wound DC motor are most commonly used. This is because the electromagnetic windings are more cost effective than permanent magnets in this power range.

Caution: If a DC motor suffers a loss of field (if for example, the field power connections are broken), the DC motor will immediately begin to accelerate to the top speed which the loading will allow. This can result in the motor flying apart if the motor is lightly loaded. The possible loss of field must be accounted for, particularly with shunt wound DC motors.

Opposing the stator field is the armature field, which is generated by a changing electromagnetic flux coming from windings located on the rotor. The magnetic poles of the armature field will attempt to line up with the opposite magnetic poles generated by the stator field. If we stopped the design at this point, the motor would spin until the poles were opposite one another, settle into place, and then stop -- which would make a pretty useless motor!

However, we are smarter than that. The section of the rotor where the electricity enters the rotor windings is called the commutator. The electricity is carried between the rotor and the stator by conductive graphite-copper brushes (mounted on the rotor) which contact rings on stator. Imagine power is supplied:

The motor rotates toward the pole alignment point. Just as the motor would get to this point, the brushes jump across a gap in the stator rings. Momentum carries the motor forward over this gap. When the brushes get to the other side of the gap, they contact the stator rings again and -- the polarity of the voltage is reversed in this set of rings! The motor begins accelerating again, this time trying to get to the opposite set of poles. (The momentum has carried the motor past the original pole alignment point.) This continues as the motor rotates.

In most DC motors, several sets of windings or permanent magnets are present to smooth out the motion.

Easy to control speed

Controlling the speed of a brushed DC motor is simple. The higher the armature voltage, the faster the rotation. This relationship is linear to the motor's maximum speed.

The maximum armature voltage which corresponds to a motor's rated speed (these motors are usually given a rated speed and a maximum speed, such as 1750/2000 rpm) are available in certain standard voltages, which roughly increase in conjunction with horsepower. Thus, the smallest industrial motors are rated 90 VDC and 180 VDC. Larger units are rated at 250 VDC and sometimes higher.

Specialty motors for use in mobile applications are rated 12, 24, or 48 VDC. Other tiny motors may be rated 5 VDC.

Most industrial DC motors will operate reliably over a speed range of about 20:1 -- down to about 5-7% of base speed. This is much better performance than the comparable AC motor. This is partly due to the simplicity of control, but is also partly due to the fact that most industrial DC motors are designed with variable speed operation in mind, and have added heat dissipation features which allow lower operating speeds.

Easy to control torque

In a brushed DC motor, torque control is also simple, since output torque is proportional to current. If you limit the current, you have just limited the torque which the motor can achieve. This makes this motor ideal for delicate applications such as textile manufacturing.

Simple, cheap drive design

The result of this design is that variable speed or variable torque electronics are easy to design and manufacture. Varying the speed of a brushed DC motor requires little more than a large enough potentiometer. In practice, these have been replaced for all but sub-fractional horsepower applications by the SCR and PWM drives, which offer relatively precisely control voltage and current. Common DC drives are available at the low end (up to 2 horsepower) for under US$100 -- and sometimes under US$50 if precision is not important.

Large DC drives are available up to hundreds of horsepower. However, over about 10 horsepower careful consideration should be given to the price/performance trade-offs with AC inverter systems, since the AC systems show a price advantage in the larger systems. (But they may not be capable of the application's performance requirement's).

Disadvantages

- Expensive to produce
- Can't reliably control at lowest speeds
- Physically larger
- High maintenance
- Dust

WORKING OF DC MOTOR

In any electric motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. As you are well aware of from playing with magnets as a kid, opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion.

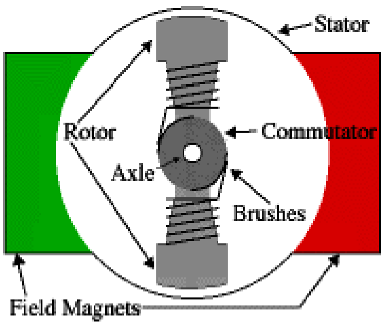


Fig 3.9.3.1 Working od DC motor

Principle

When a rectangular coil carrying current is placed in a magnetic field, a torque acts on the coil which rotates it continuously.

When the coil rotates, the shaft attached to it also rotates and thus it is able to do mechanical work.

Every DC motor has six basic parts -- axle, rotor (a.k.a., armature), stator, commutator, field magnet(s), and brushes. In most common DC motors (and all that Beamer will see), the external magnetic field is produced by high-strength permanent magnets. The stator is the stationary part of the motor -- this includes the motor casing, as well as two or more permanent magnet pole pieces. The rotor (together with the axle and attached commutator) rotate with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a common motor layout -- with the rotor inside the stator (field) magnets.

The geometry of the brushes, commentator contacts, and rotor windings are such that when power is applied, the polarities of the energized winding and the stator magnet(s) are misaligned, and the rotor will rotate until it is almost aligned with the stator's field magnets. As the rotor reaches alignment, the brushes move to the next commentator contacts, and energize the next winding. Given our example two-pole motor, the rotation reverses the direction of current through the rotor winding, leading to a "flip" of the rotor's magnetic field, driving it to continue rotating.

In real life, though, DC motors will always have more than two poles (three is a very common number). In particular, this avoids "dead spots" in the commutator. You can imagine how with our example two-pole motor, if the rotor is exactly at the middle of its rotation (perfectly aligned with the field magnets), it will get "stuck" there. Meanwhile, with a two-pole motor, there is a moment where the commutator shorts out the power supply (i.e., both brushes touch both commutator contacts simultaneously). This would be bad for the power supply, waste energy, and damage motor components as well. Yet another disadvantage of such a simple motor is that it would exhibit a high amount of torque "ripple" (the amount of torque it could produce is cyclic with the position of the rotor).

Construction and Working

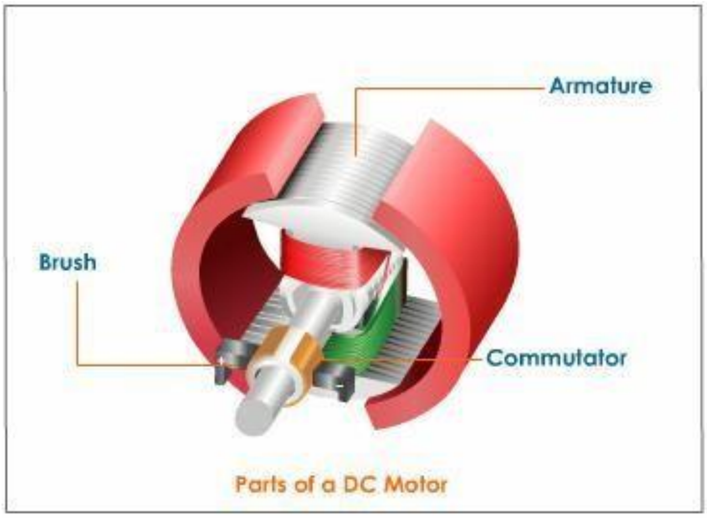


Fig 3.9.3.2 Parts of DC motor

Parts of a DC Motor

Armature

A D.C. motor consists of a rectangular coil made of insulated copper wire wound on a soft iron core. This coil wound on the soft iron core forms the armature. The coil is mounted on an axle and is placed between the cylindrical concave poles of a magnet.

Commutator

A commutator is used to reverse the direction of flow of current. Commutator is a copper ring split into two parts C_1 and C_2. The split rings are insulated form each other and mounted on the axle of the motor. The two ends of the coil are soldered to these rings. They rotate along with the coil. Commutator rings are connected to a battery. The wires from the battery are not connected to the rings but to the brushes which are in contact with the rings.

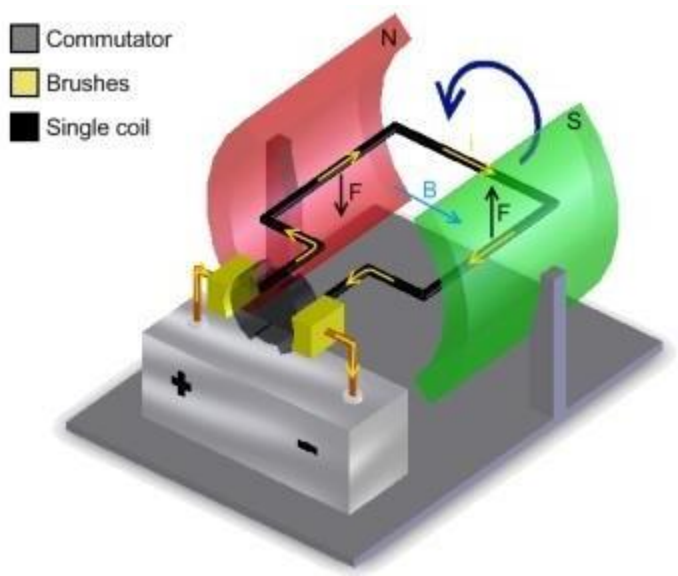


Fig 3.9.4 DC Motor

Brushes

Two small strips of carbon, known as brushes press slightly against the two split rings, and the split rings rotate between the brushes.

The carbon brushes are connected to a D.C. source.

PARAMETRS OF THE DC MOTRS

1. Direction of rotation
2. Motor Speed
3. Motor Torque
4. Motor Start and Stop

Direction of Rotation

A DC Motor has two wires. We can call them the positive terminal and the negative terminal, although these are pretty much arbitrary names (unlike a battery where these polarities are vital and not to be mixed!). On a motor, we say that when the + wire is connected to + terminal on a power source, and the - wire is connected to the - terminal source on the same power source, the motor rotates clockwise (if you are looking towards the motor shaft). If you reverse the wire polarities so that each wire is connected to the opposing power supply terminal, then the motor rotates counter clockwise. Notice this is just an arbitrary selection and that some motor manufacturers could easily choose the opposing convention. As long as you know what rotation you get with one polarity, you can always connect in such a fashion that you get the direction that you want on a per polarity basis.

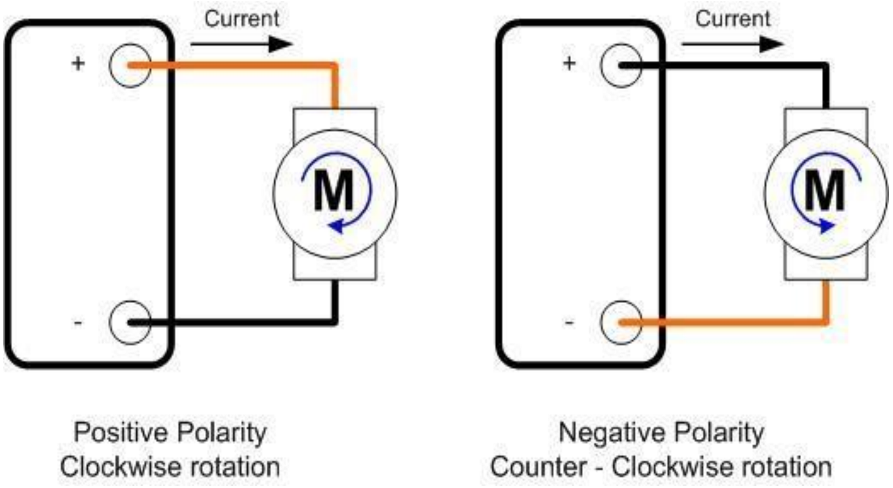


Fig 3.9.5.1 Direction of rotation

DC Motor Rotation vs Polarity

Facts:

- DC Motor rotation has nothing to do with the voltage magnitude or the current magnitude flowing through the motor.
- DC Motor rotation does have to do with the voltage polarity and the direction of the current flow.

DC Motor Speed

Whereas the voltage polarity controls DC motor rotation, voltage magnitude controls motor speed. Think of the voltage applied as a facilitator for the strengthening of the magnetic field. In other words, the higher the voltage, the quicker will the magnetic field become strong. Remember that a DC motor has an electromagnet and a series of permanent magnets. The applied voltage generates a magnetic field on the electromagnet portion. This electromagnet field is made to oppose the permanent magnet field. If the electromagnet field is very strong, then both magnetic entities will try to repel each other from one side, as well as atract each other from the other side. The stronger the induced magnetic field, the quicker will this separation/attaction will try to take place. As a result, motor speed is directly proportional to applied voltage.

Motor Speed Curve

One aspect to have in mind is that the motor speed is not entirely lineal. Each motor will have their own voltage/speed curve. One thing I can guarantee from each motor is that at very low voltages, the motor will simply not move. This is because the magnetic field strength is not enough to overcome friction. Once friction is overcome, motor speed will start to increase as voltage increase.

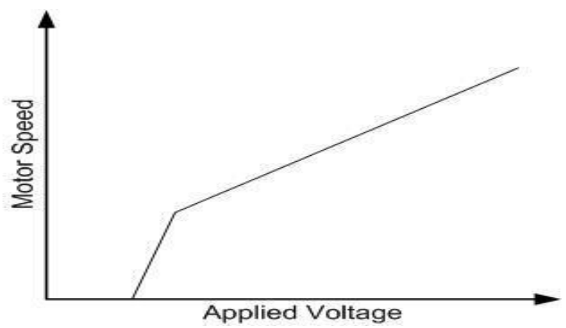


Fig 3.9.5.2 Motor speed curve

Motor Torque

In the previous segment I kind of described speed as having to do with the strength of the magnetic field, but this is in reality misleading. Speed has to do with how fast the magnetic field is built and the attraction/repel forces are installed into the two magnetic structures. Motor strength, on the other hand, has to do with magnetic field strength. The stronger the electromagnet attracts the permanent magnet, the more force is exerted on the motor load.

Per example, imagine a motor trying to lift 10 pounds of weight. This is a force that when multiplied by a distance (how much from the ground we are lifting the load) results in WORK. This WORK when exerted through a predetermined amount of time (for how long we are lifting the weight) gives us power. But whatever power came in, must come out as energy can not be created or destroyed. So that you know, the power that we are supplying to the motor is computed by

$P = IV$

Where P is power, I is motor current and V is motor voltage

Hence, if the voltage (motor speed) is maintained constant, how much load we are moving must come from the current. As you increase load (or torque requirements) current must also increase.

Motor Loading

One aspect about DC motors which we must not forget is that loading or increase of torque can not be infinite as there is a point in which the motor simply can not move. When this happens, we call this loading "Stalling Torque". At the same time this is the maximum amount of current the motor will see, and it is refer to Stalling Current. Stalling deserves a full chapter as this is a very important scenario that will define a great deal of the controller to be used. I promise I will later write a post on stalling and its intricacies.

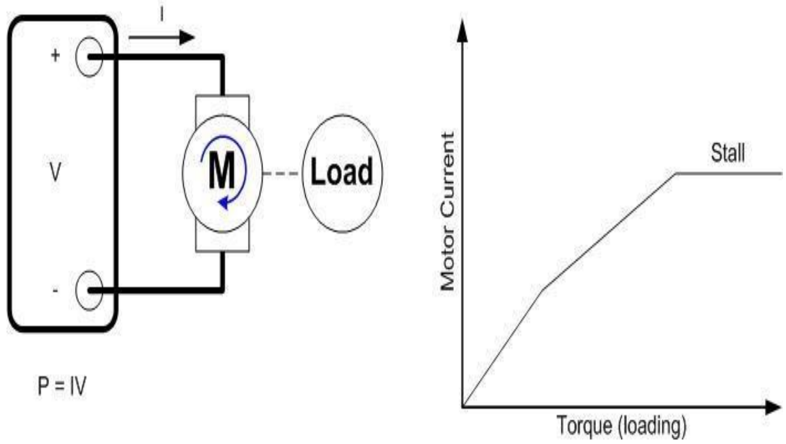


Fig 3.9.5.3 Motor loading

Motor Start and Stop

You are already well versed on how to control the motor speed, the motor torque and the motor direction of rotation. But this is all fine and dandy as long as the motor is actually moving. How about starting it and stopping it? Are these trivial matters? Can we just ignore them or should we be careful about these aspects as well? You bet we should!

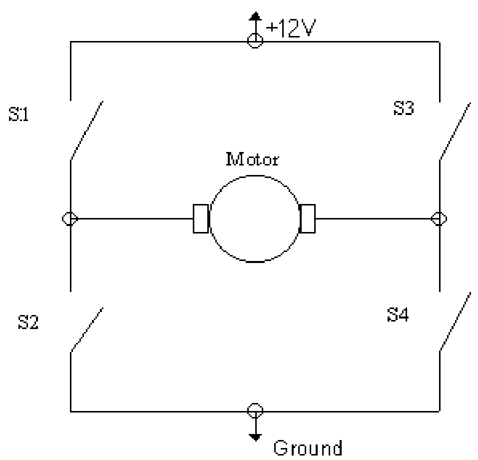
Starting a motor is a very hazardous moment for the system. Since you have an inductance whose energy storage capacity is basically empty, the motor will first act as an inductor. In a sense, it should not worry us too much because current can not change abruptly in an inductor, but the truth of the matter is that this is one of the instances in which you will see the highest currents flowing into the motor. The start is not necessarily bad for the motor itself as in fact the motor can easily take this Inrush Current. The power stage, on the other hand and if not properly designed for, may take a beating.

Once the motor has started, the motor current will go down from inrush levels to whatever load the motor is at. Per example, if the motor is moving a few gears, current will be proportional to that load and according to torque/current curves.

Stopping the motor is not as harsh as starting. In fact, stopping is pretty much a breeze. What we do need to concern ourselves is with how we want the motor to stop. Do we want it to coast down as energy is spent in the loop, or do we want the rotor to stop as fast as possible? If the later is the option, then we need braking. Braking is easily accomplished by shorting the motor outputs. The reason why the motor stops so fast is because as a short is applied to the motor terminals, the Back EMF is shorted. Because Back EMF is directly proportional to speed, making Back EMF = 0, also means making speed = 0.

MOTORDRIVER CIRCUIT

The name "H-Bridge" is derived from the actual shape of the switching circuit which control the motion of the motor. It is also known as "Full Bridge". Basically, there are four switching elements in the H-Bridge as shown in the figure below.



H-Bridge Switch

Switches Closed	Motor
S1 S3	Off
S2 S4	Off
S1 S4	Forward
S3 S2	Reverse

Fig 3.10.1 H bridge

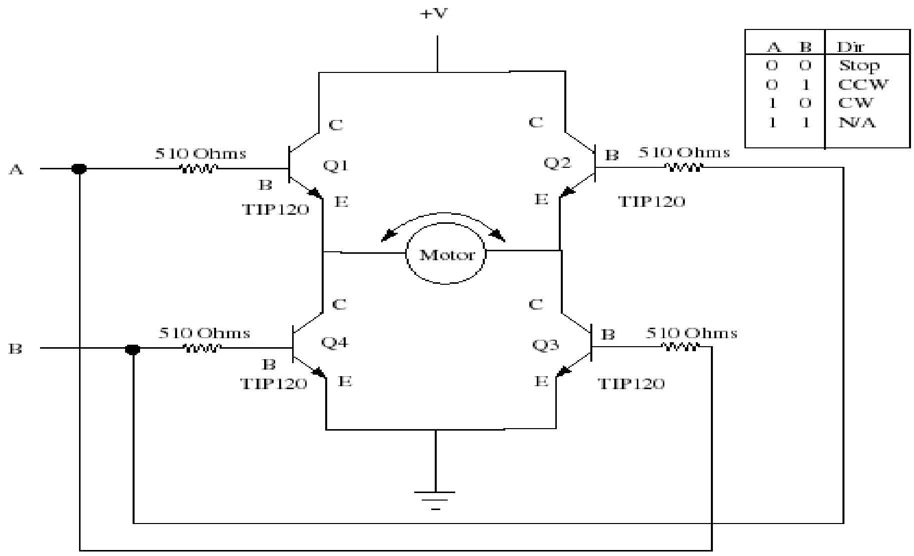


Fig 3.10.2 circuit diagram of motor driver

As you can see in the figure above there are four switching elements named as "High side left", "High side right", "Low side right", "Low side left". When these switches are turned on in pairs motor changes its direction accordingly. Like, if we switch on High side left and Low side right then motor rotate in forward direction, as current flows from Power supply through the motor coil goes to ground via switch low side right. This is shown in the figure below.

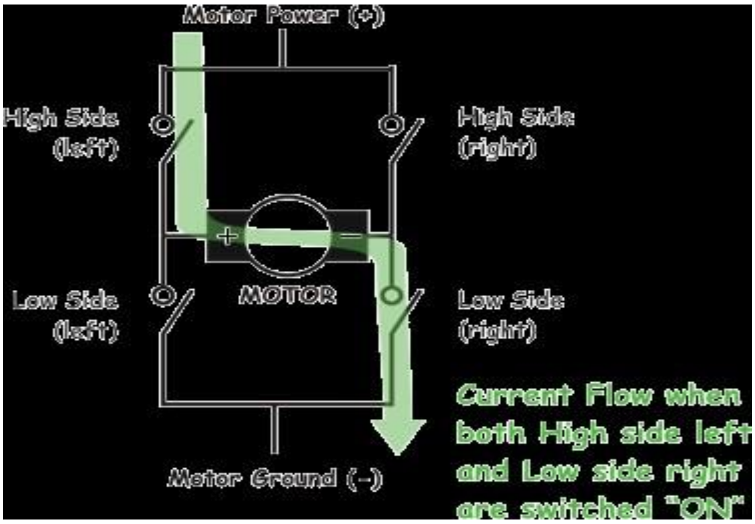


Fig 3.10.3 Working of motor driver

Similarly, when you switch on low side left and high side right, the current flows in opposite direction and motor rotates in backward direction.

This is the basic working of H-Bridge. We can also make a small truth table according to the switching of H-Bridge explained above.

Truth Table				
High Left	High Right	Low Left	Low Right	Description
On	Off	Off	On	Motor runs clockwise
Off	On	On	Off	Motor runs anti-clockwise
On	On	Off	Off	Motor stops or decelerates
Off	Off	On	On	Motor stops or decelerates

Table 3.10 Truth table of motor driver

As already said, H-bridge can be made with the help of transistors aswell as MOSFETs, the only thing is the power handling capacity of the circuit.If motors are needed to run with high current then lot of dissipation is there.

So head sinks are needed to cool the circuit.

CHAPTER 4

EMBEDDED SYSTEMS

EMBEDDED SYSTEMS

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer. The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes. The applications software on such processors is sometimes referred to as firmware

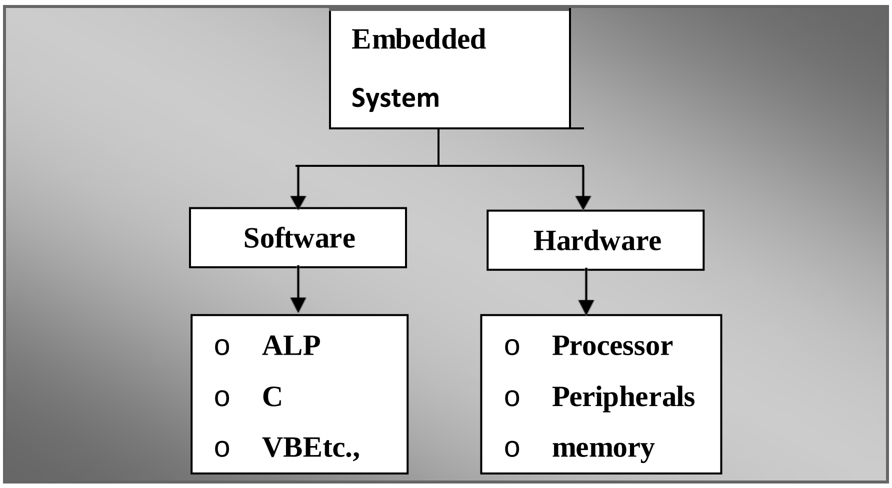


Fig 4.1 Block diagram of Embedded System

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

Memory: It is used to store data or address.

Peripherals: These are the external devices connected

Processor: It is an IC which is used to perform some task

APPLICATIONS OF EMBEDDED SYSTEMS

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

Processors are classified into four types

- Micro Processor (µp)
- Micro controller (µc)
- Digital Signal Processor (DSP)
- Application Specific Integrated Circuits (ASIC)

MICRO PROCESSOR (µp)

WHAT IS MICRO PROCESSOR?

A silicon chip that contains a CPU, In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most work stations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection system for automobiles.

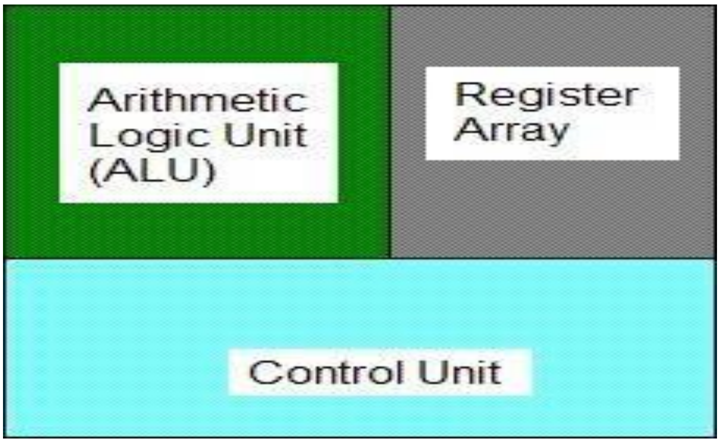


Fig.4.3.1 Three basic elements of a microprocessor

WORKING OF MICRO PROCESSOR

A microprocessor accepts binary data as input, processes that data, and then provides output based on the instructions stored in the memory. The data is processed using the microprocessor's ALU (arithmetical and logical unit), control unit, and a register array. The register array processes the data via a number of registers that act as temporary fast access memory locations. The flow of instructions and data through the system is managed by the control unit.

MICRO CONTROLLER

WHAT IS MICRO CONTROLLER?

A microcontroller is a compact integrated circuit designed to govern a specific operation in an embedded system. A typical microcontroller includes a processor, memory and input/output (I/O) peripherals on a single chip.

Sometimes referred to as an embedded controller or microcontroller unit (MCU), microcontrollers are found in vehicles, robots, office machines, medical devices, mobile radio transceivers, vending machines and home appliances, among other devices. They are essentially simple miniature personal computers (PCs) designed to control small features of a larger component, without a complex front-end operating system (OS).

WORKING OF MICRO CONTROLLER

A microcontroller is embedded inside of a system to control a singular function in a device. It does this by interpreting data it receives from its I/O peripherals using its central processor. The temporary information that the microcontroller receives is stored in its data memory, where the processor accesses it and uses instructions stored in its program memory to decipher and apply the incoming data. It then uses its I/O peripherals to communicate and enact the appropriate action.

Microcontrollers are used in a wide array of systems and devices. Devices often utilize multiple microcontrollers that work together within the device to handle their respective tasks.

For example, a car might have many microcontrollers that control various individual systems within, such as the anti-lock braking system, traction control, fuel injection or suspension control. All the microcontrollers communicate with each other to inform the correct actions. Some might communicate with a more complex central computer within the car, and others might only communicate with other microcontrollers. They send and receive data using their I/O peripherals and process that data to perform their designated tasks.

DIGITAL SIGNAL PROCESSORS

A digital signal processor (DSP) is a specialized microprocessor chip, with its architecture optimized for the operational needs of digital signal processing. DSPs are fabricated on MOS integrated circuit chips. They are widely used in audio signal processing, telecommunications, digital image processing, radar, sonar and speech recognition systems, and in common consumer electronic devices such as mobile phones, disk drives and high-definition television(HDTV) products.

The goal of a DSP is usually to measure, filter or compress continuous real-world Analog signals. Most general-purpose microprocessors can also execute digital signal processing algorithms successfully, but may not be able to keep up with such processing continuously in real-time. Also, dedicated DSPs usually have better power efficiency, thus they are more suitable in portable devices such as mobile phones because of power consumption constraints. DSPs often use special memory architectures that are able to fetch multiple data or instructions at the same time. DSPs often also implement data compression technology, with the discrete cosine transform(DCT) in particular being a widely used compression technology in DSPs.

APPLICATION SPECIFIC INTEGRATED CIRCUITS(ASIC)

An application-specific integrated circuit (ASIC) is an integrated circuit(IC) chip customized for a particular use, rather than intended for general-purpose use. For example, a chip designed to run in a digital voice recorder or a high-efficiency bit coin miner is an ASIC. Application-specific standard product(ASSP) chips are intermediate between ASICs and industry standard integrated circuits like the7400 series or the4000 series. ASIC chips are typically fabricated using metal-oxide semiconductor(MOS) technology, as MOS integrated circuit chips.

HARVARD ARCHITECTURE

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark Irelay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

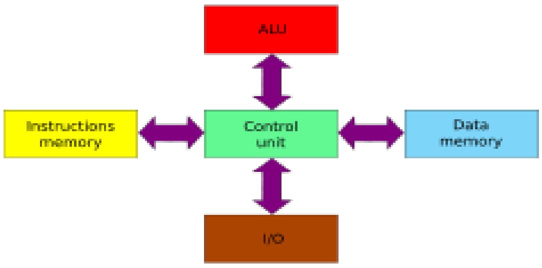


Fig.4.7 Harvard Architecture

USES OF THE HARVARD ARCHITECTURE

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where trade-offs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

Digital signal processors(DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behaviour must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. TexasInstrumentsTMS320C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.

Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARMCortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

VON-NEUMANN ARCHITECTURE:

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. They cannot be performed at the same time. The von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit(CPU) and a single separate storage structure("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A stored-program digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture. The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

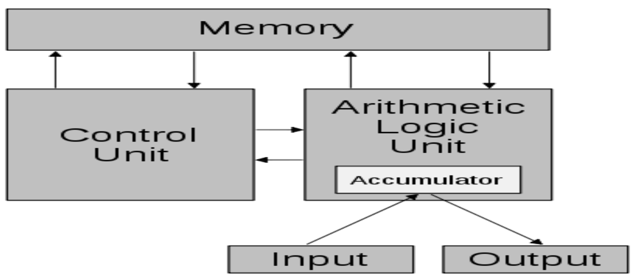


Fig4.7.2 Schematic of the Von-Neumann Architecture

SOFTWARE EXPLANATION

SOFTWARE REQUIREMENTS:

- Arduino software
- Programming language
- Proteus simulation

ARDUINO SOFTWARE

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments.

A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

WHY ARDUINO?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step-by-step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows. Wrap it up in an easy-to- use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

Open source and extensible hardware - The plans of the Arduino boards are published under

- a Creative Commons license, so experienced circuit designers c an make their own version
- Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money •Getting Started with Arduino and Genuine products: -

Install the Arduino Software (IDE) on Windows PCs-

This document explains how to install the Arduino Software (IDE) on Windows machines.

- Download the Arduino Software (IDE)
- Proceed with board specific instructions.

HOW TO DOWNLOAD THE ARDUINO SOFTWARE (IDE)

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

INSTALLATION

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.4.8.4.1 Standard USB Cable (A plug to B plug)

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig.4.8.4.2 Standard USB Cable (an A to Mini-B plug)

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

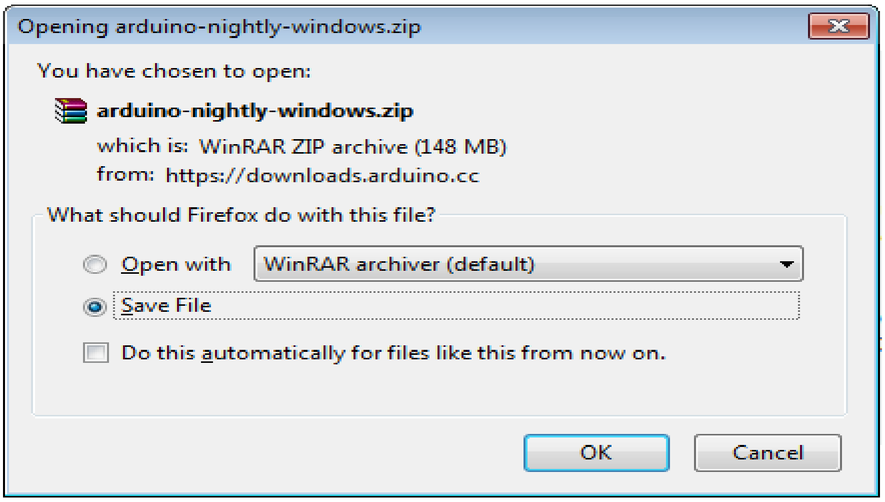


Fig.4.8.4.3 Download Arduino IDE Software Step

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

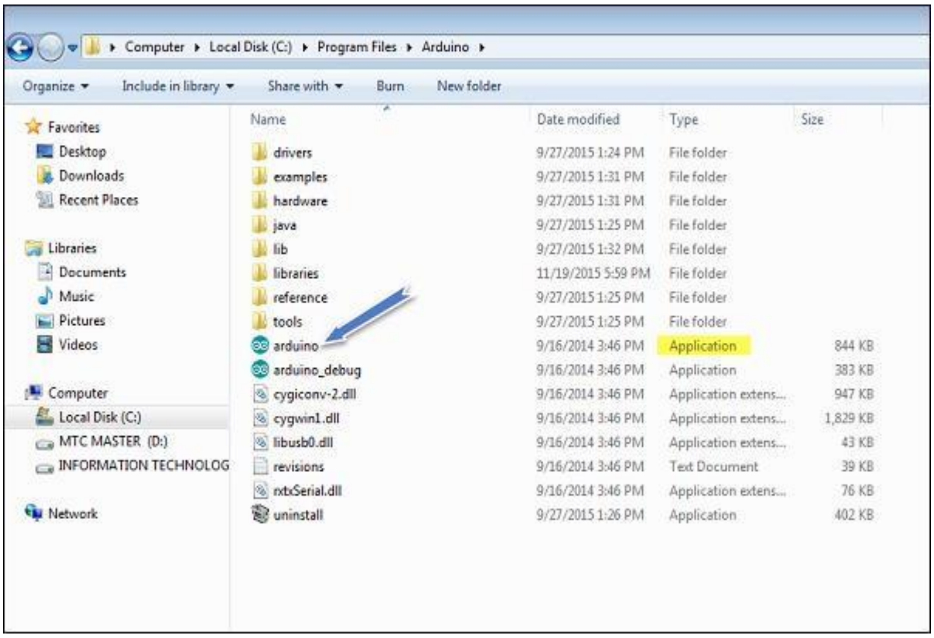


Fig.4.8.4.4 Launch Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.

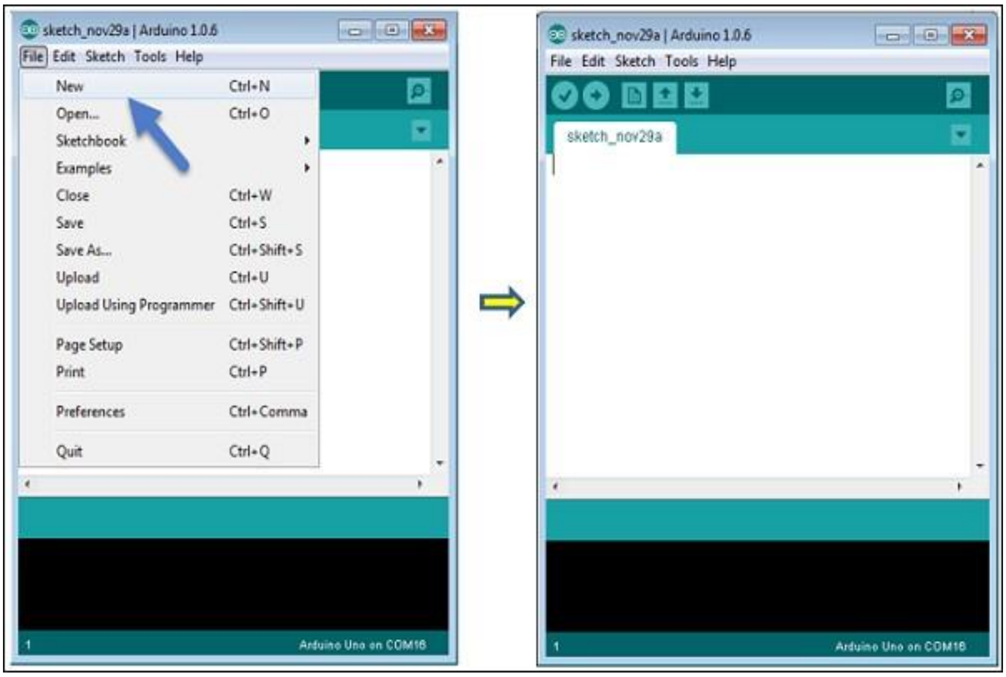


Fig.4.8.4.5 Open your first project

To open an existing project example, select File → Example → Basics → Blink.

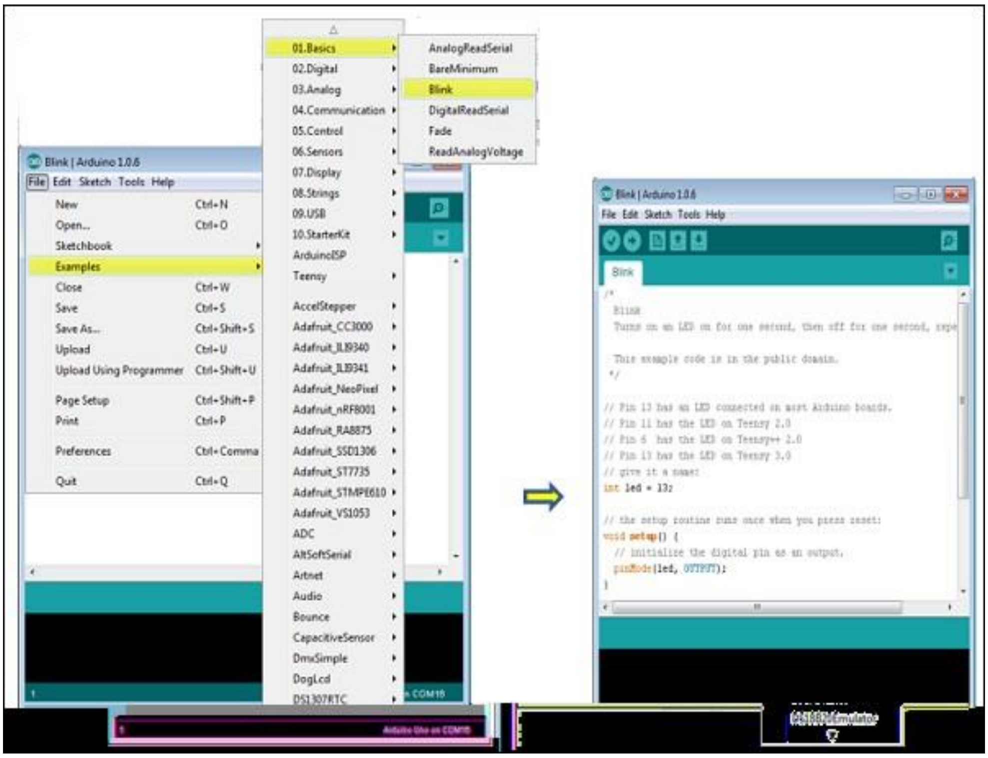


Fig.4.8.4.6 Open your first program

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

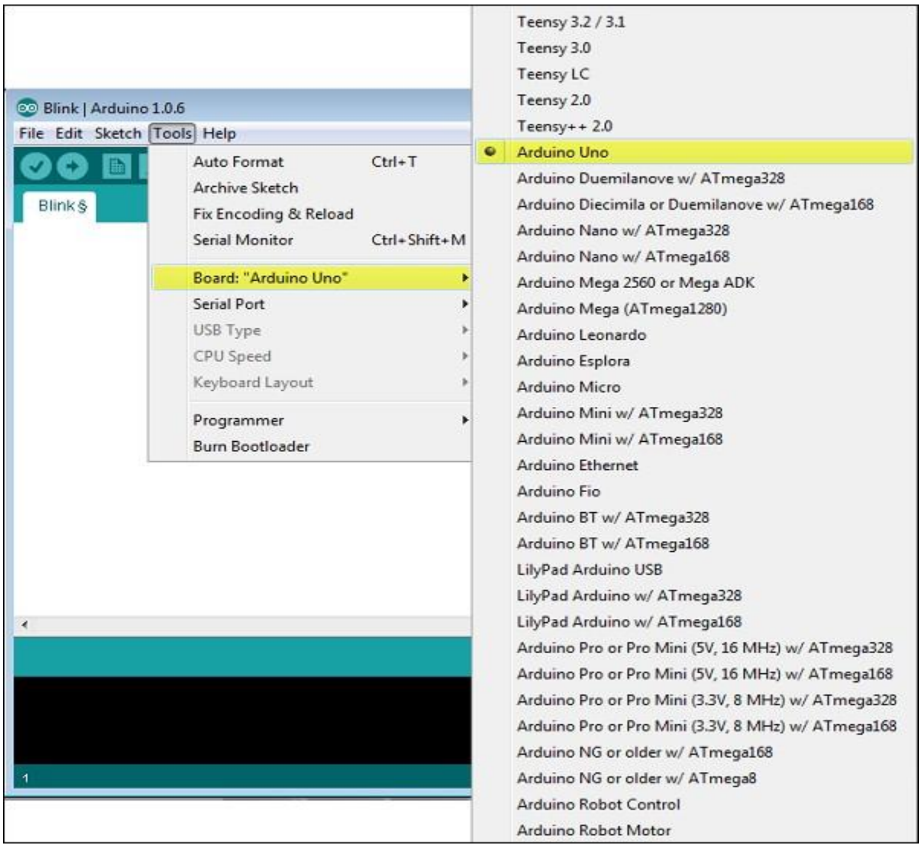


Fig.4.8.4.7 Select your Arduino board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

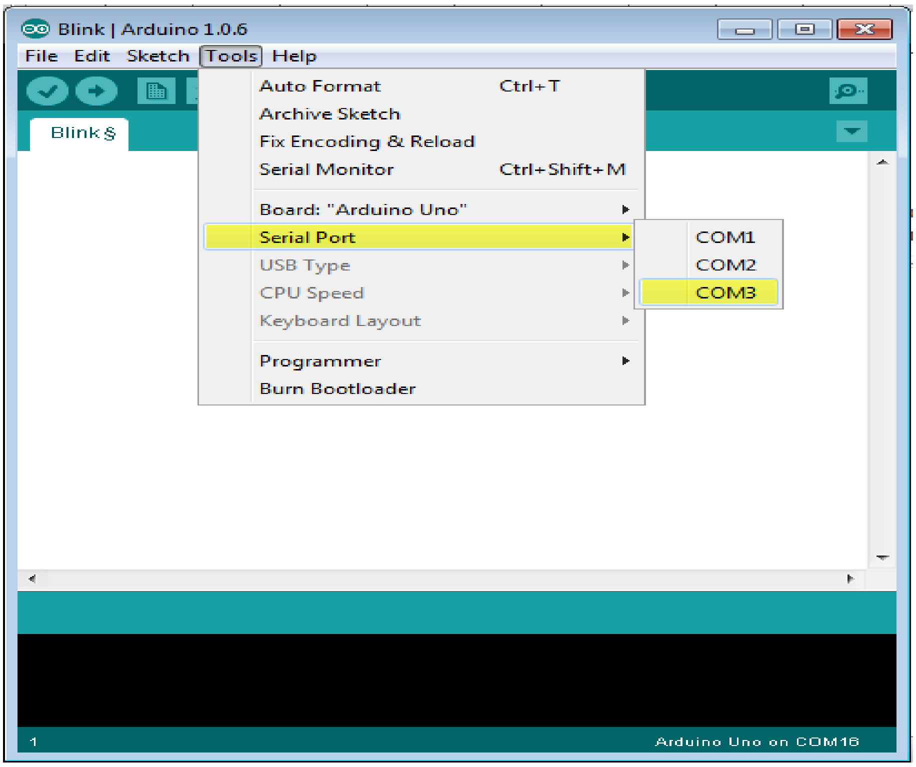


Fig.4.8.4.8 Select your serial port

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

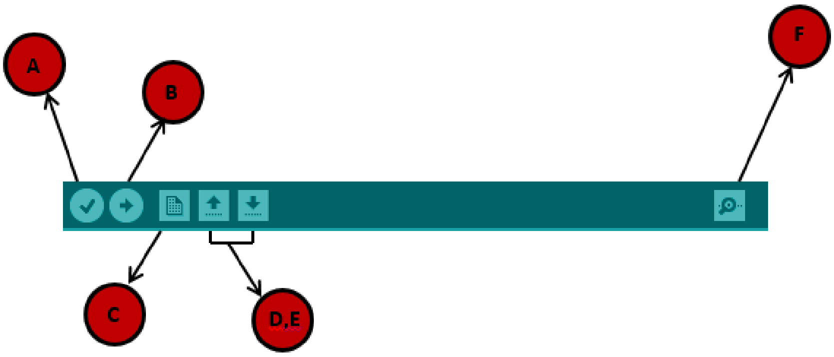


Fig.4.8.4.9 Upload the program to your board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

CONNECTING A BATTERY

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 fromwww.jameco.com.Here is what this looks like.

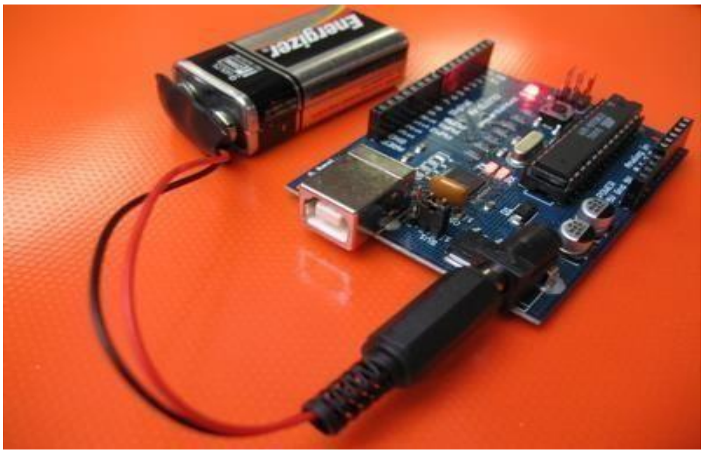


Fig.4.8.5.1 Connecting a Battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC.

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called "sketches", but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi- colons appear at the end of command lines.

```
void setup()
{
  Serial.begin(9600);
  Serial.println("Hello World");
}
void loop() {}
```

Fig.4.8.5.2 Example Program

MC PROGRAMMING LANGUAGE: EMBEDDED C

This is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc.

are prime requirements. Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

PROTEUS

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possesses 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design

Step 1: Open ISIS software and select New design in File menu

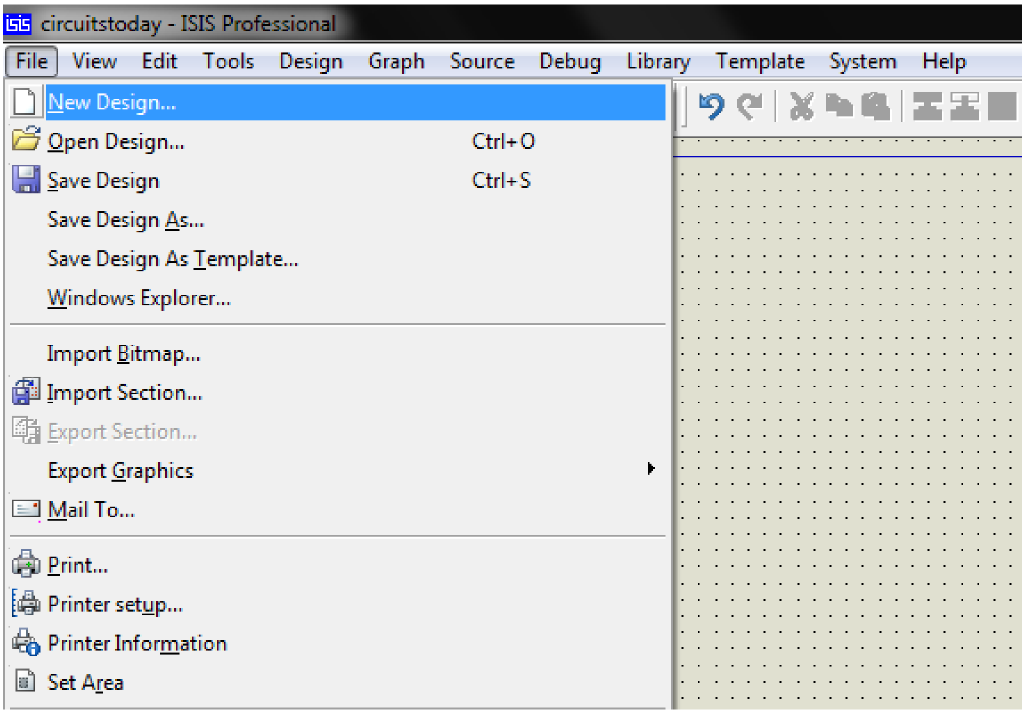


Fig 4.10.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

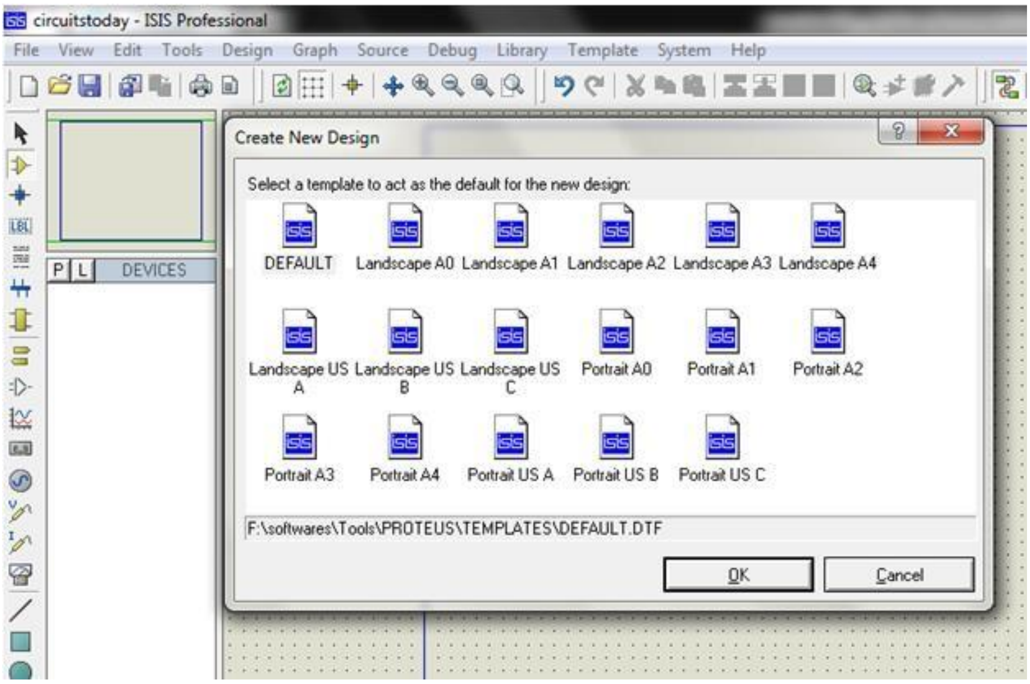


Fig 4.10.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

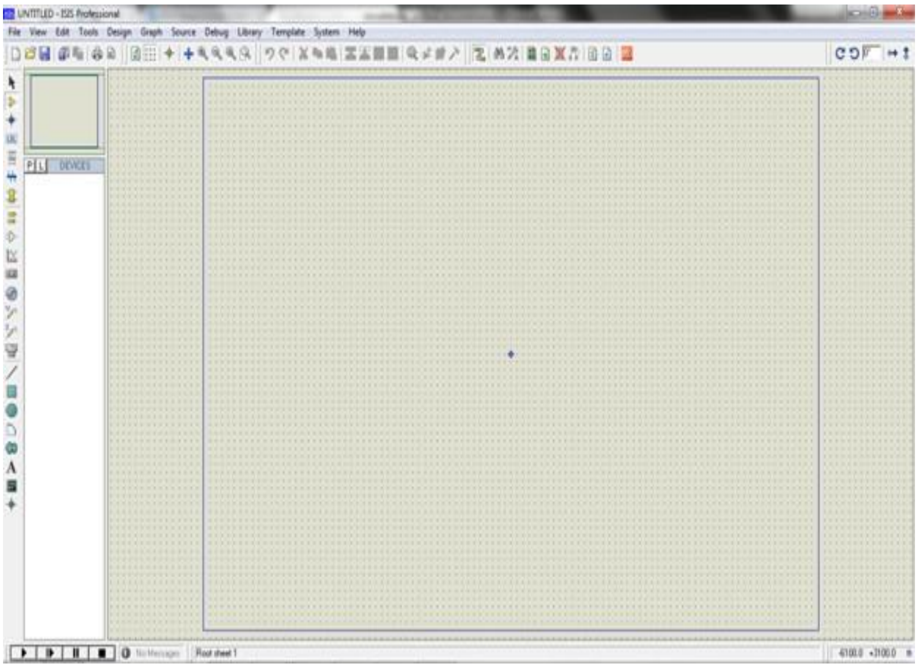


Fig 4.10.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

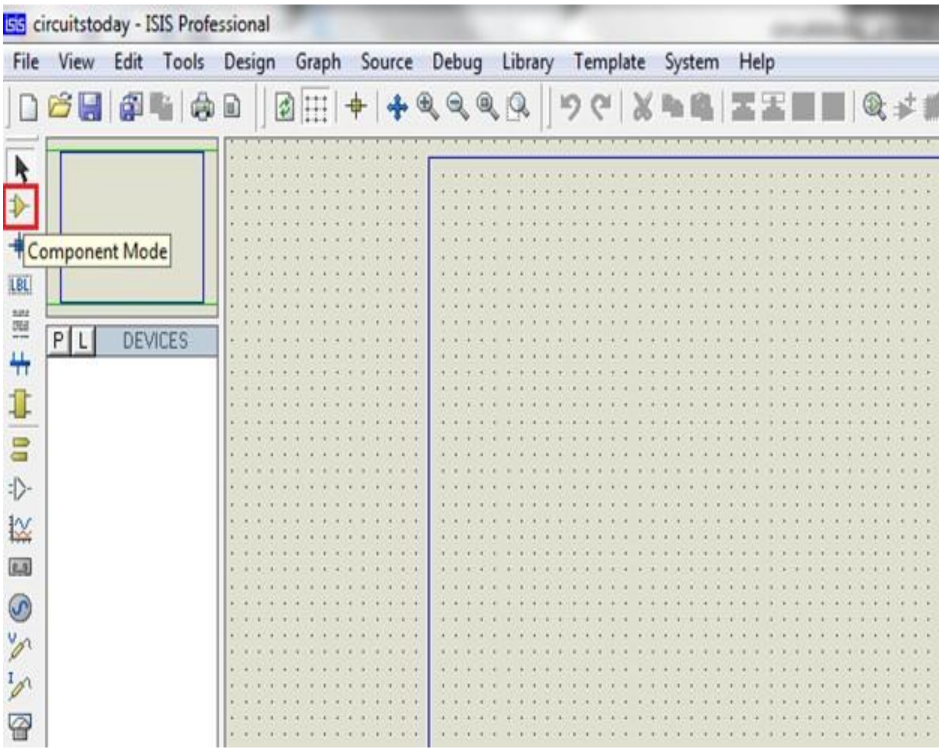


Fig4.10.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

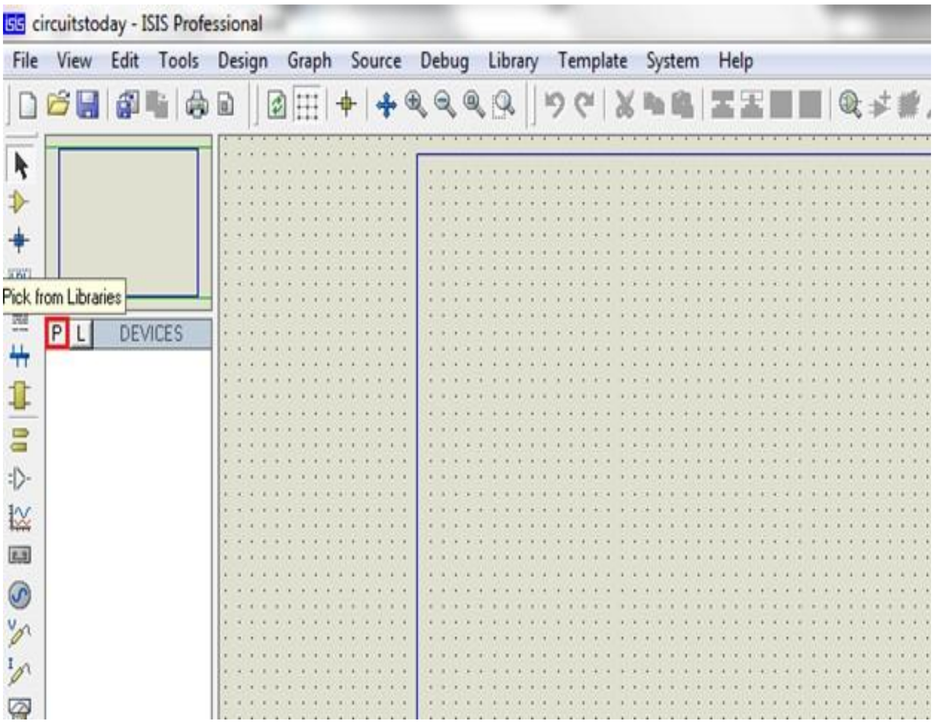


Fig 4.10.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

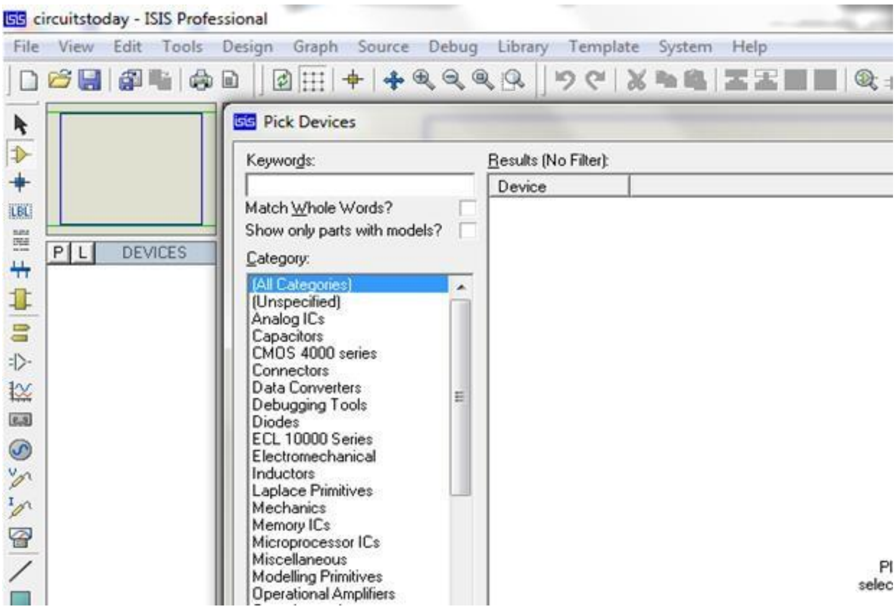


Fig 4.10.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

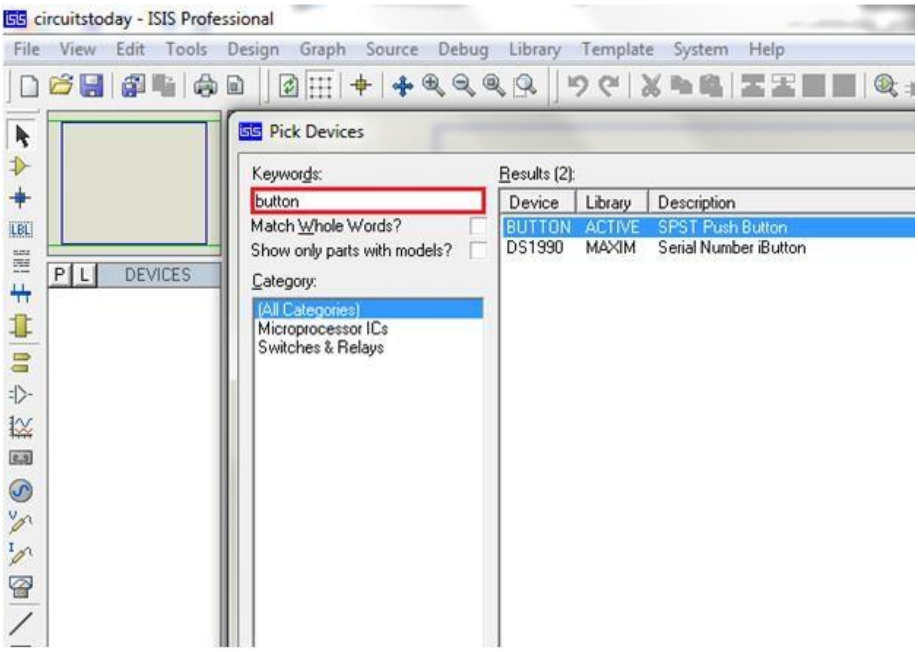


Fig 4.10.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

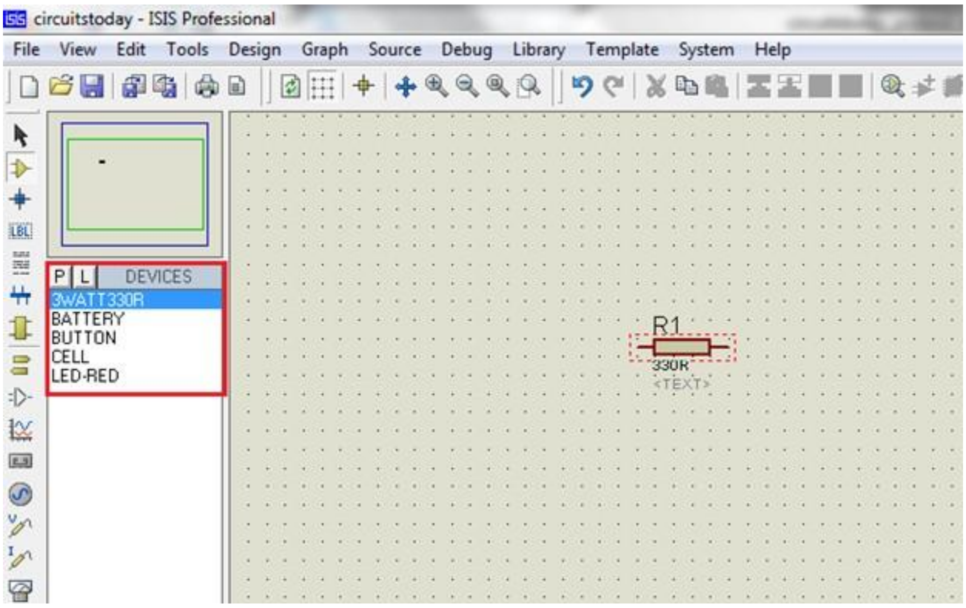


Fig 4.10.8 Component Selection

Place all the required components and route the wires i.e., make connections. Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

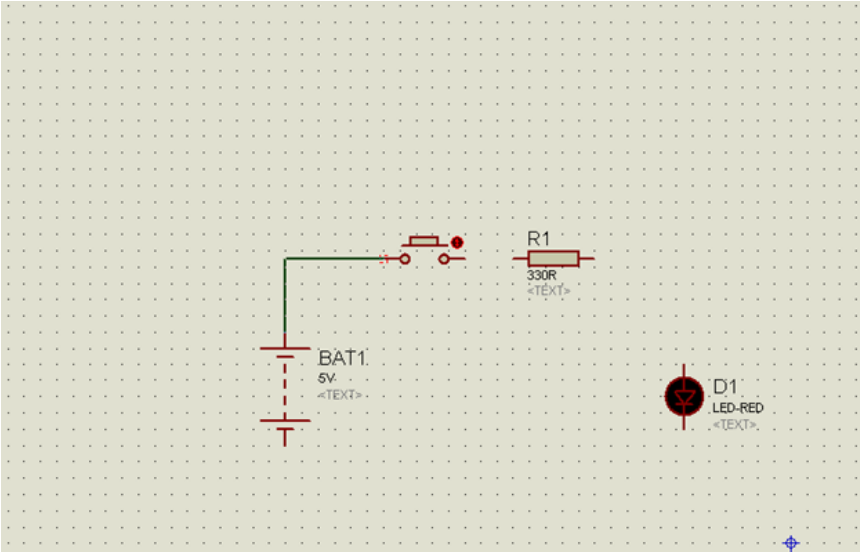


Fig 4.10.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

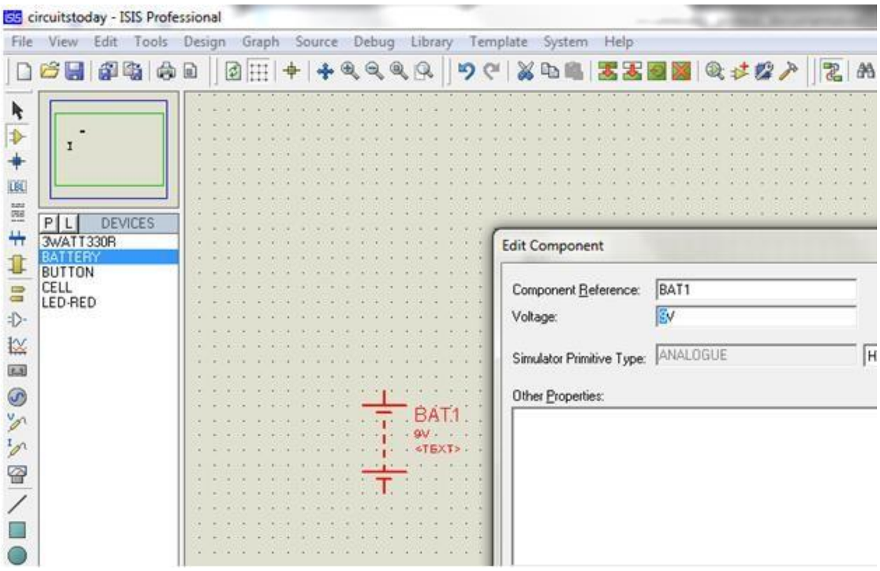


Fig 4.10.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

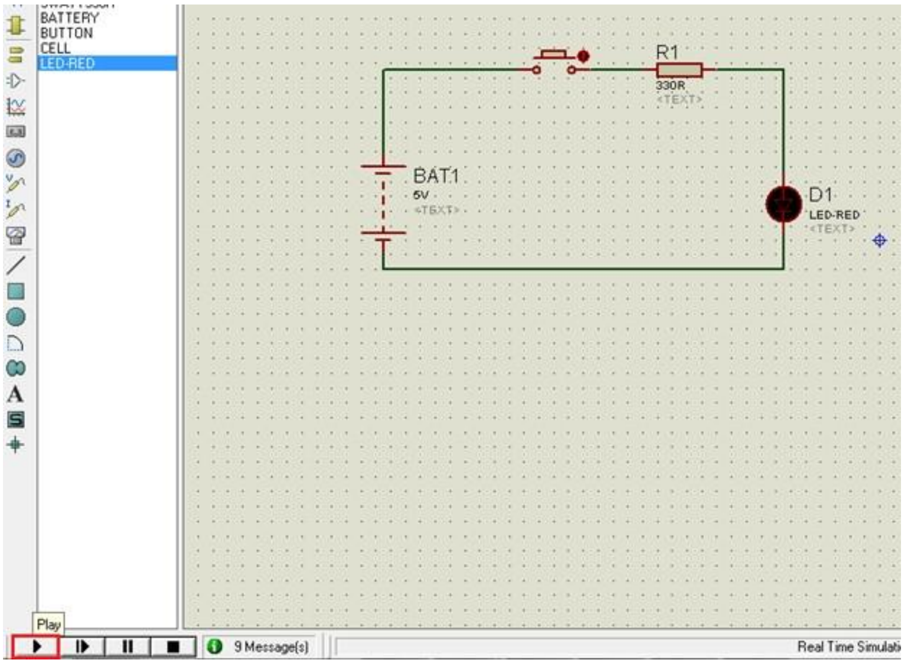


Fig 4.10.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

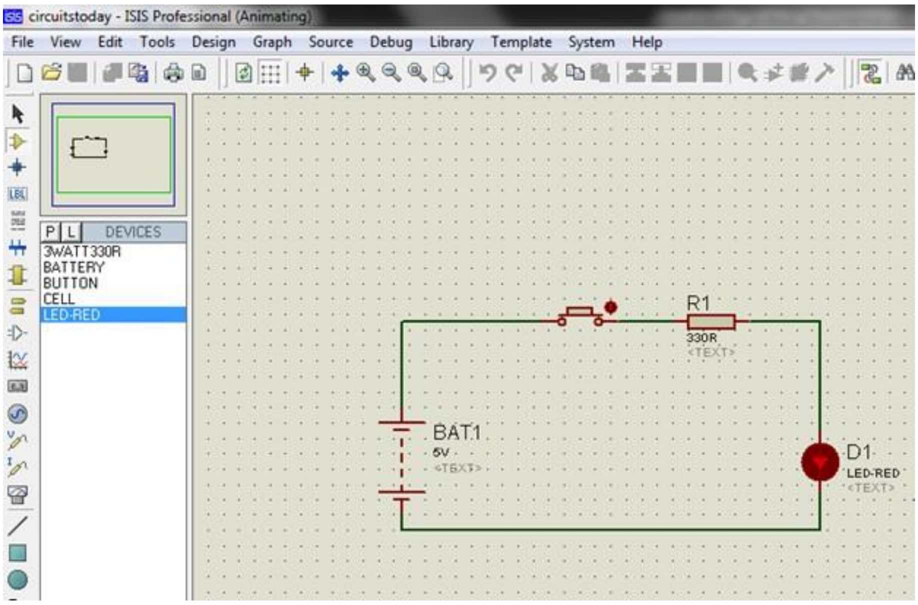


Fig 4.10.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

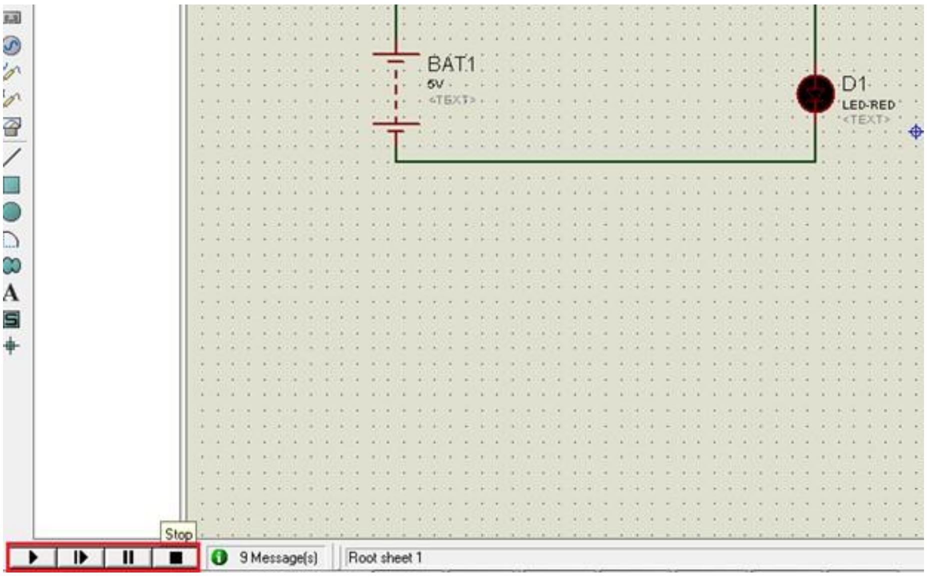


Fig 4.10.13 Simulation Step-Pause-Stop Buttons

Telnet is an application protocol used on the Internet or local area network to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection. User data is interspersed in-band with Telnet control information in an 8-bit byte oriented data connection over the Transmission Control Protocol (TCP).

Telnet was developed in 1969 beginning with RFC 15, extended in RFC 855, and standardized as Internet Engineering Task Force (IETF) Internet Standard STD 8, one of the first Internet standards. The name stands for "**teletype network**".

Historically, Telnet provided access to a command-line interface on a remote host. However, because of serious security concerns when using Telnet over an open network such as the Internet, its use for this purpose has waned significantly in favor of SSH.

CHAPTER 5

PROJECT TESTING

SOFTWARE TESTING

Software Testing is a process of executing the application with an intent to find any software bugs. It is used to check whether the application met its expectations and all the functionalities of the application is working. The final goal of testing is to check whether the application is behaving in the way it is supposed to under specified conditions. All aspects of the code are examined to check the quality of application. The primary purpose of testing is to detect software failures so that defects may be uncovered and corrected. The test cases are designed in such way that scope of finding the bugs is maximum.

SYSTEM TEST CASES

A test case is a set of test data, preconditions, expected results and post conditions, developed for a test scenario to verify compliance against a specific requirement. We have designed and executed a few test cases to check if the project meets the functional requirements.

PROJECT CODE

IMPLEMENTATION OF

CODE:

```
#include<LiquidCrystal.h>
#include<SoftwareSerial.h>
LiquidCrystal
lcd(13,12,11,10,9,8);
SoftwareSerial mySerial(2,3);//rx , tx //esp8266
int ir1=4, ir2=7;
int m1=5;
int m2=6;
int trigger_pin = A5;
int echo_pin = A4;
int time;
```

```
int distance;
void serialFlush(){
 while(Serial.available() > 0) {
 char tt = Serial.read();
 }
}
void sendwifi(String chr,unsigned int len) {
 mySerial.print("AT+CIPSEND=0,");
 mySerial.println(len-1);
 delay(2000);
 mySerial.print(chr);
 delay(2000);
}
char res[130];
char buff[130];
void setup() {
 pinMode(ir1,INPUT);
 pinMode(ir2,INPUT);
 pinMode(m1,OUTPUT);
 pinMode(m2,OUTPUT);
 digitalWrite(m1,LOW);
 digitalWrite(m2,LOW);
 pinMode(trigger_pin, OUTPUT);
 pinMode(echo_pin, INPUT);
 lcd.begin(16,2);
 lcd.clear();
 lcd.setCursor(0,0);
 lcd.print("IOT Based");
 lcd.setCursor(0,1);
```

```
lcd.print("Railway Security");
delay(2000);
Serial.begin(9600);
  mySerial.begin(115200);//esp8266
  mySerial.print("AT\r\n");
  delay(1000);
  mySerial.print("ATE0\r\n");
  delay(1000);
  mySerial.print("AT+CWMODE=3\r\n");
  delay(1000);
  mySerial.print("AT+CWSAP=\"Project\",\"project1235\",5,0\r\n\r\n");
  delay(1000);
  mySerial.print("AT+CIPMUX=1\r\n");
  delay(1000);
  mySerial.print("AT+CIPSERVER=1,23\r\n");
  delay(1000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("WATING FOR LINK");
  while(1)
  {
    if(mySerial.available())
    {
      //if(Esp.find("0,LINK"))
      if(mySerial.find("0,CONNECT"))
      {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("LINK FOUND");
```

```
          break;
           }
             }
           }
    delay(1000);
    sendwifi("WELCOME \r\n",11);
    delay(1000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("READY");
    delay(2000);
}
void loop()
{
  digitalWrite (trigger_pin, HIGH);
  delayMicroseconds (10);
  digitalWrite (trigger_pin, LOW);
  time = pulseIn (echo_pin, HIGH);
  distance = (time * 0.034) / 2;
  lcd.clear();
  lcd.setCursor(0,0);
 lcd.print("Meters: ");
 lcd.print(distance);
 delay(100);
 digitalWrite(m1,LOW);
 digitalWrite(m2,HIGH);
 delay(1000);
 if(distance==50)
 {
```

```
 sprintf(buff,"Object Detected at Location: %u\r",distance);
 sendwifi(buff,strlen(buff));
 delay(300);
 lcd.clear();
 lcd.setCursor(0,0);
 lcd.print("Object Detected");
 delay(100);
 analogWrite(m1,200);
 digitalWrite(m2,HIGH);
 delay(3000);
}
if(distance==10)
{
 sprintf(buff,"Object Approach at Location: %u\r",distance);
 sendwifi(buff,strlen(buff));
 delay(300);
 lcd.clear();
 lcd.setCursor(0,0);
 lcd.print("Object Detected");
 delay(100);
 digitalWrite(m1,LOW);
 digitalWrite(m2,LOW);
 delay(3000);
}
if(digitalRead(ir1)==LOW)
{
 sprintf(buff,"Spot Detect at Track 1 : %u\r\n",distance);
 sendwifi(buff,strlen(buff));
 delay(300);
```

```
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Spot Detect at");
  lcd.setCursor(0,1);
  lcd.print("Track 1");
  delay(100);
  digitalWrite(m1,LOW);
  digitalWrite(m2,LOW);
  delay(3000);
 }
 if(digitalRead(ir2)==LOW)
 {
  sprintf(buff,"Spot Detect at Track 2 : %u\r\n",distance);
  sendwifi(buff,strlen(buff));
  delay(300);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Spot Detect at");
  lcd.setCursor(0,1);
  lcd.print("Track 2");
  delay(100);
  digitalWrite(m1,LOW);
  digitalWrite(m2,LOW);
  delay(3000);
 }
}
```

SCHEMATIC DIAGRAM

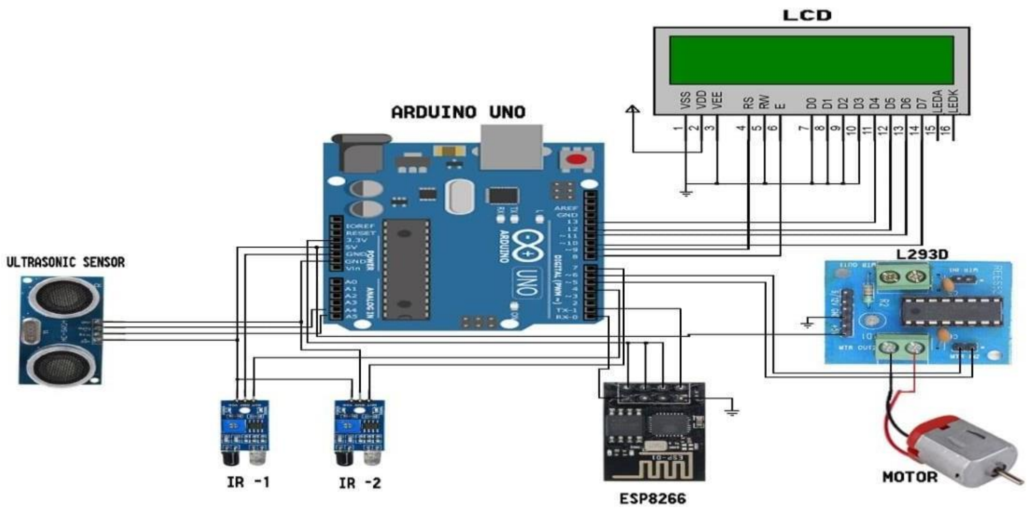


Fig 5.3.2 Schematic diagram

FLOW CHART

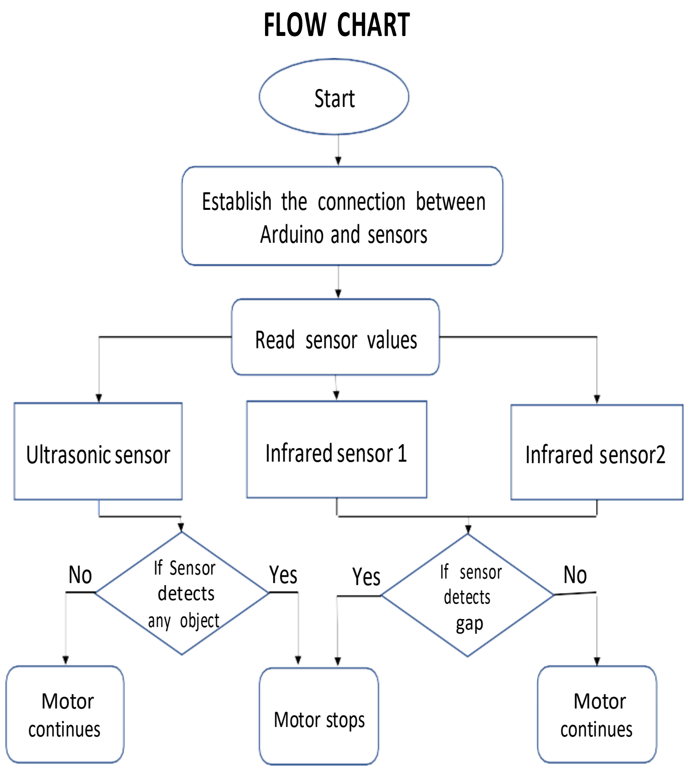


Fig 5.3.3 Flow Chart

RESULTS

- When IR1 sensor detects gap in the railway rail it sends information to the Arduino board and it is displayed on the LCD as follows

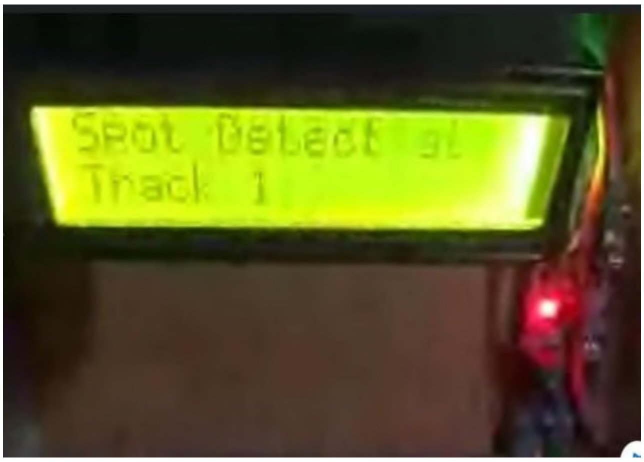


Fig 5.3.4.1 Gap Detection at track1 using IR 1

- When IR2 sensor detects gap in the railway rail it sends information to the Arduino board and is displayed on the LCD as follows

Fig 5.3.4.2 Gap Detection at track 2 using IR 2

- When Ultrasonic sensor detects object in the railway track it sends information to the Arduino board and is displayed on the LCD as follows

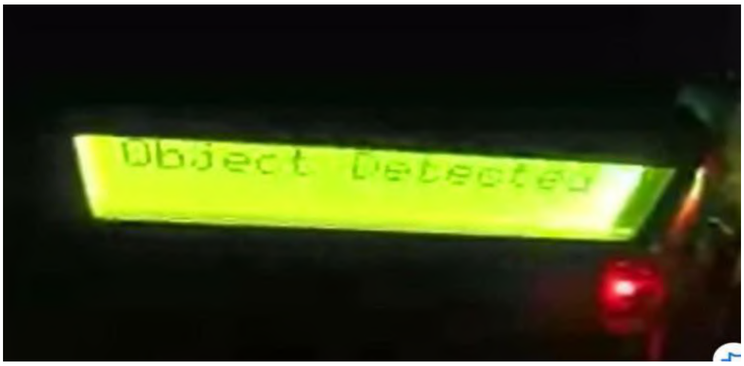


Fig 5.3.4.3 Object Detection using Ultrasonic sensor

- Fig represents the data in telnet app which is displayed when any fault is detected in the railway track through wi-fi module

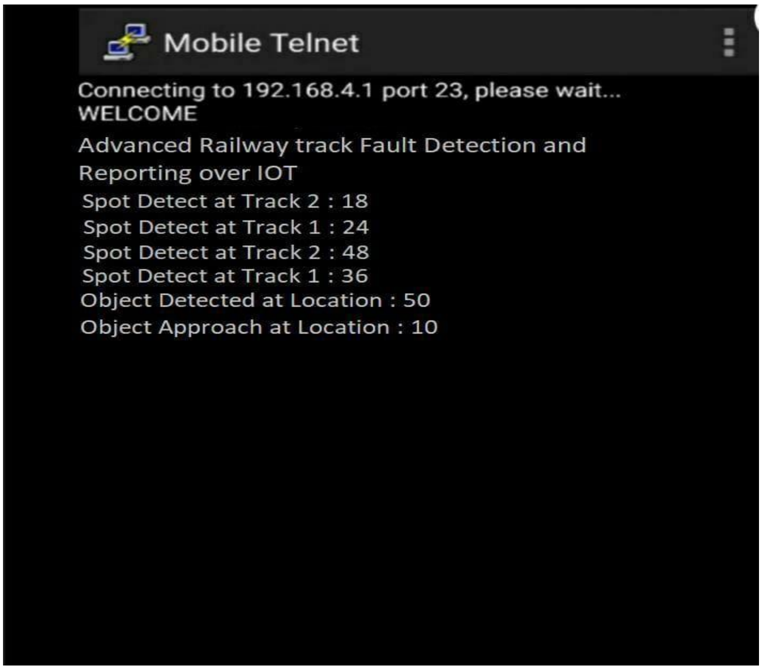


Fig. 5.3.4.4 Display of Fault in Telnet app

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION

Despite railway is the country's prime mode of transportation, the Indian Railways have to contended with doubtful safety records. By the usage of a smart way of fault detection in railway tracks, this will have a remarkable have an effect on the protection of the tracks which will assist in stopping train accidents to a very large extent. The regions where manual assessment is beyond the realm of imagination, as, in profound coal mine shafts, mountain areas, and thick woodland locales can be effectively done utilizing this framework. This system is placed on the trains so that every area gets monitored and along with the rescue team the driver also gets a notification so he would be able to make quick decisions due to which many lives would be saved. This will help in the support for checking the state of the rail tracks with no mistakes and along these lines keeping up the tracks in great condition. So, the maintenance of the tracks is easier. It also avoiding train mishaps to a very large extent.

FUTURE ENHANCEMENT

In the future, the proposed system will also be connected to networking & communication device. So, if any fault is detected the system will automatically broadcast the location and type of fault to every train coming on that route. If the broadcast is done, then the other trains can be diverted or blocked off (Stopped) to avoid any inconvenience. Along with it, the system will conjointly use deflectors, so that if there's any high curvature track then the deflectors can transmit the waves in keeping with the curve. The main objective of the system is to provide cost effective railway track crack detection system with greater accuracy and High speed information transferring after crack detection. The detected crack will be get updated through GPS and GSM.

REFERENCES

[1] Navaraja. P "Crack Detection System For Railway Track By Using Ultrasonic and PIR Sensor"International Journal of Advanced Informantion

and Communication Technology Volume-1, Issue-1, PP. 126-130 May-2014.

[2] Kim, N.Stubbs " Crack Detection in beam type structures using frequency data" Elseveir volume 2592, PP. 1-37, January 2003.

[3] Akhil N,Dinu Mohan,Fayis P,Sija Gopinath, "Railway Crack Detection System" ISSN 2395 0056 International Research Journal of Engineering

and Technology Volume-3, Issue-5, May 2016.

[4] Reenu George, Divya Jose, Gokul T G, Varun A G "Automatic broken track detection using IR transmitter and receiver" ISSN 2320-3765

International Journal of Advanced Research in Electrical Electronics and Instrumentation Engineering volume 4, Issue 4, April- 2015

[5] Prashanthi antari, Mirza sajid ali baig, Narsappa reddy "Automatic Railway Track Crack Detection using GSM & GPS" ijitech.orgISSN 2321-

8665 Vol.03,Issue.03, Pages:0365-0367, July-2015.

[6] Milev. I and Gruendig. L, "Rail track data base of German rail-The future automated maintenance", in Proc. INGEO FIG Regional Central

Eastern Eur. Conf. Eng. Surv., Bratislava, Slovakia, pp. 1–8, Nov11–13, 2004.

[7] Trehag. J, Handel. P, and Ogren. M, "Onboard estimation and classification of a railroad curvature", IEEE Trans. Instrum. Meas., vol. 59, no. 3,

pp. 653–660, Mar. 2010.

[8] Beales. L, "Track system requirements, Railway Group Standards", GC/RT5021, Railway Safety, London, Oct. 2003.

[9] Lemmon. R. T, "The influence of the number of satellites on the accuracy of RTK GPS positions", Australian Surv., vol. 44, no. 1, pp. 64–70,

1999.

[10] Qiao Jian-hua, Li Lin-sheng and Zhang Jinggang, "Design of Rail Surface Crack-detecting System Based on Linear CCD Sensor", IEEE Int.

Conf. on Networking, Sensing and Control, 2008.

[11] Vijayakumar. K , S.R. Wylie, J. D. Cullen, C.C.Wright, and A.I. AIShamma'a, "Noninvasive rail track detection system using Microwave

sensor", Journal of App. Phy., 2009.

[12] Cacciola. M, G. Megali, D. Pellicanμo, S. Calcagno, M. Versaci, and F. C. Morabito, "Rotating Electromagnetic Field for Crack Detection in

Railway Tracks", piers online, Vol. 6, NO. 3, 2010.

[13] Wojnarowski, Robert John Welles, Kenneth Brakeley Kornrumpf, and William Paul, "Electromagnetic system for railroad track crack detection

and traction enhancement"US 6,262,573 B1, Jul17, 2001.

[14] Richard J. Greene, John R. Yates and Eann A. Patterson, "Crack detection in rail using infrared methods", Opt. Eng. 46, 051013, May 2007.

[15] Hartman, G.A., "Infrared Damage Detection System (IDDS) for real time, small-scale damage monitoring, Charlotte, North Carolina 2003

A MAJOR PROJECT REPORT ON

ROBOTIC VEHICLE CONTROLLED BY HAND GESTURE USING ARDUINO

submitted by partial fulfillment of the requirement of the award of degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted

By

SadamVeekshith (17K81A04A8)

PerikaAkanksha (17K81A04A3)

PoteAkhil Kumar (17K81A04A4)

Under the Guidance of

Ms.P.Pushpa, M.Tech

Assistant Professor

ECE

St.MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

St.MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the major-project work entitled **"ROBOTIC VEHICLE CONTROLLED BY HAND GESTURE USING ARDUINO**" is a bonafide work carried out by **Sadam Veekshith (17K81A04A8), Perika Akanksha (17K81A04A3), Pote Akhil Kumar (17K81A04A4)** in partial fulfillment of the requirements for the degree of **Bachelor of Technology in Electronics & Communication Engineering** by the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2020-21.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree.

INTERNAL GUIDE	**HEAD OF THE DEPARTMENT**
Ms. P .Pushpa	Dr. B. Hari Krishna,
Assistant professor.	Professor
Department of ECE	

ACKNOWLEDGEMENT

We sincerely thank the management of our college **St. MARTIN'S ENGINEERING COLLEGE** for providing required facilities during our project work.

We derive our great pleasure in expressing our sincere gratitude to our principal **Dr.P. Santosh Kumar Patra** for his timely suggestions, which helped us to complete the project work successfully.
It is the very auspicious moment we would like to express our gratitude to **Dr. B. Hari Krishna**,Head of the Department, ECE for his consistent encouragement during the progress of this project.

We take it as a privilege to thank our overall project coordinator **Dr. A. Anand**,Professor, department of ECE for the ideas that led to complete the project work and we also thank them for their continuous guidance, support and unfailing patience, throughout the course of this work.

We sincerely thank to our project coordinator **Mr. K. Karthik,** Assistant professor**,** Department of ECE for guidance and encouragement in carrying out this project work.

We sincerely thank to our project internal guide **Ms. P. Pushpa**, assistant professor, Department of ECE for her guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level **Project CoordinatorsDr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

1. SADAM VEEKSHITH (17K81A04A8)
2. PERIKA AKANKSHA (17K81A04A3)
3. POTE AKHIL KUMAR (17K81A04A4)

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **P.AKANKSHA** WITH ROLL NO.**17K81A04A3**, **P.AKHIL** WITH ROLL NO.**17K81A04A4**, **S.VEEKSHITH** WITH ROLL NO.**17K81A04A8**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED **"ROBOTIC VEHICLE CONTROLLED BY HAND GESTURE USING PIC "**

AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD

DECLARATION

We, the students of ‘Bachelor of Technology in Department of Electronics and Communication Engineering’, Session: 2017 – 2021, St. Martin’s Engineering College, hereby declare that the project work entitled ‘ROBOTIC VEHICLE CONTROLLED BY HAND GESTURE USING ARDUINO’ is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

Sadam Veekshith	**Perika Akanksha**	**Pote Akhil Kumar**
17K81A04A8	**17K81A04A3**	**17K81A04A4**

ABSTRACT

Robotic vehicle controlled by the hand movement mainly benefits the disabled person, as those person with the hand gesture moves to the direction they Want to go without pressing any Axis. The circuit is the same as that of the old post and there is only a slight change in circuit, here we need to program the circuit. So I will be using an Arduino as the programming platform. To recognize the gestures made I will be using an accelerometer sensor. This system includes a glove which has a receiver data which will be mounted on the top with Atmega microcontroller interfaced to the accelerometer and it is supposed to be worn by the user while using this machine. The circuit on the vehicle includes WIFI(esp8266) and Driver IC to operate the motors. The commands that are received by the IC on the circuit are sent to the wifi which forwards the command to the local network application. The command from application wifi then gets transmitted to the Arduino microcontroller which decodes the command and makes the vehicle move in the direction specified by the user.

Index terms: —**Accelerometer sensor, Arduino UNO, Wifi, Gesture Control,Telnet Protocol.**

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1
INTRODUCTION

CHAPTER-1

1.Introduction:

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.

The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system.

The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes.The applications software on such processors is sometimes referred to as firmware.

The simplest devices consist of a single microprocessor (often called a "chip"), which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine.

As the embedded system is the combination of both software and hardware.

1.1Block Diagram of Embedded Systems:

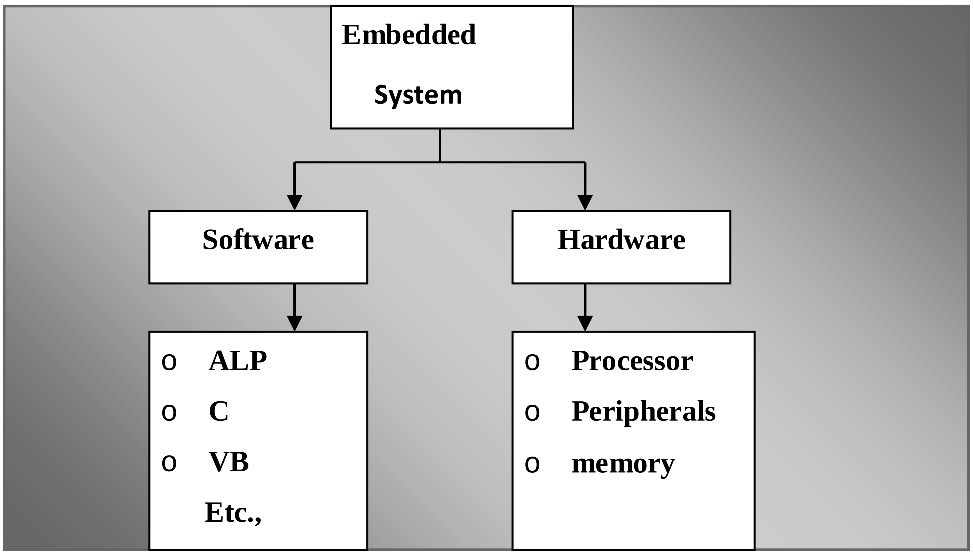


Fig.1.1: Block Diagram of Embedded System

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

1.2 Applications of embedded systems:

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems.

1.3 Micro Processor(µp):

A siliconchip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digitaldevices, from clock radios to fuel-injection systems for automobiles.

1.3.1 Processors are classified into four types like:

- ➢ Micro Processor (μp)
- ➢ Micro controller (μc)
- ➢ Digital Signal Processor (DSP)
- ➢ Application Specific Integrated Circuits (ASIC)

1.4 Three Basic Elements of a Microprocessor:

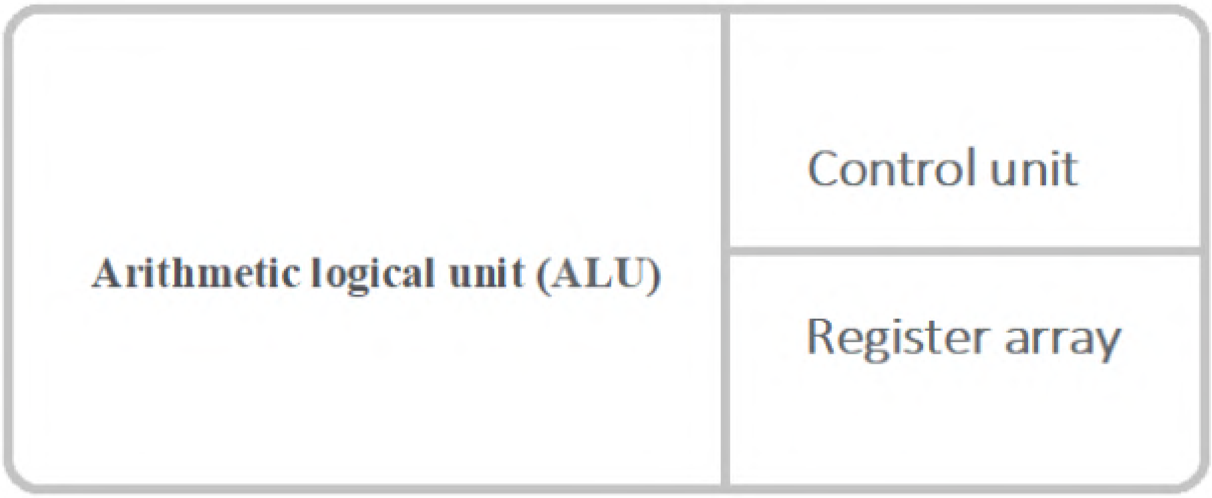


Fig.1.2: Three Basic Elements of a Microprocessor.

1.5 Harvard Architecture:

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not bootitself.

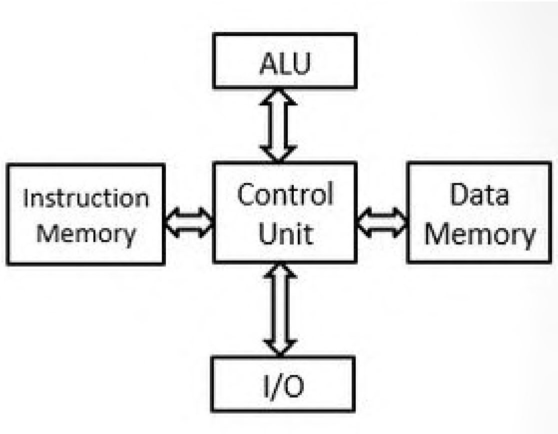


Fig.1.3: Harvard Architecture.

1.5.1 Uses of the Harvard architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

- Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behaviour must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.
- Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

1.5.2 Von-Neumann Architecture:

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. They cannot be performed at the same time. The von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientistJohn von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A stored-program digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

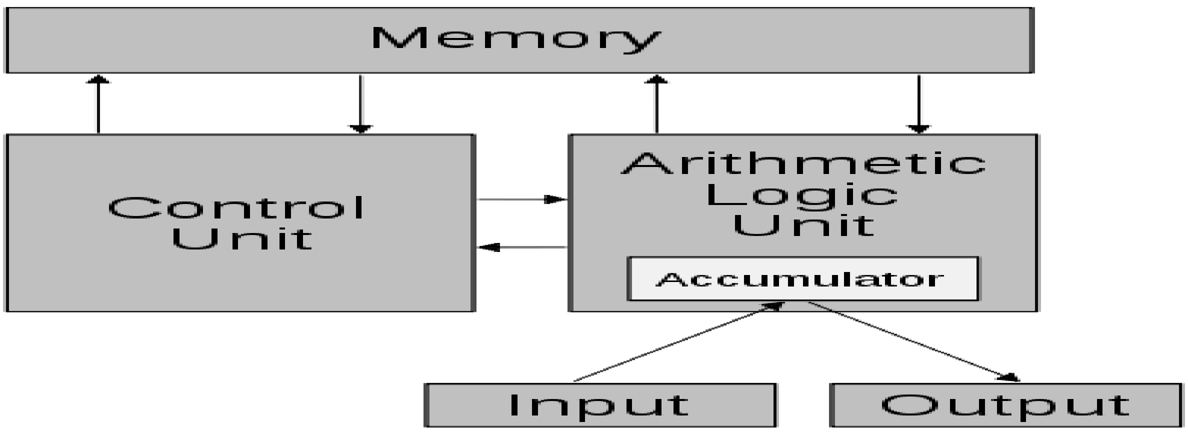


Fig.1.4: Schematic of the Von-Neumann Architecture.

1.6 Arduino:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

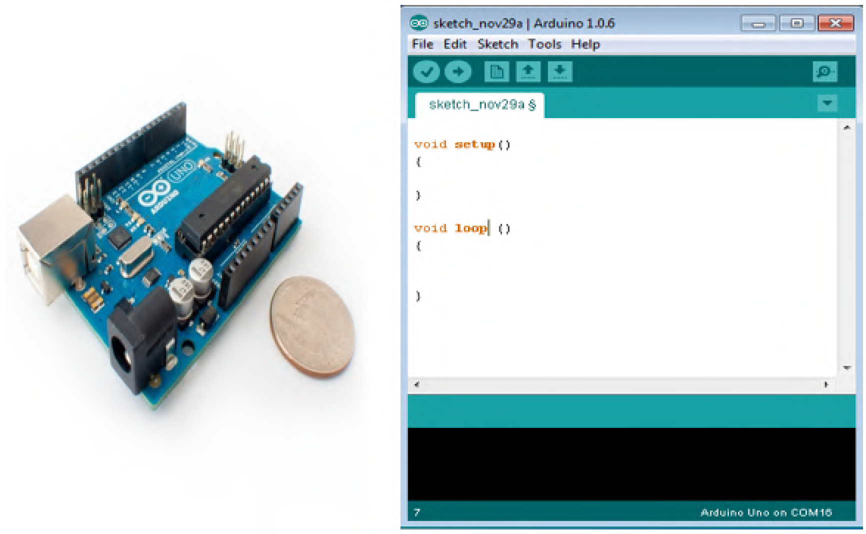


Fig 1.5 Arduino Uno and IDE.

1.6.1 Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

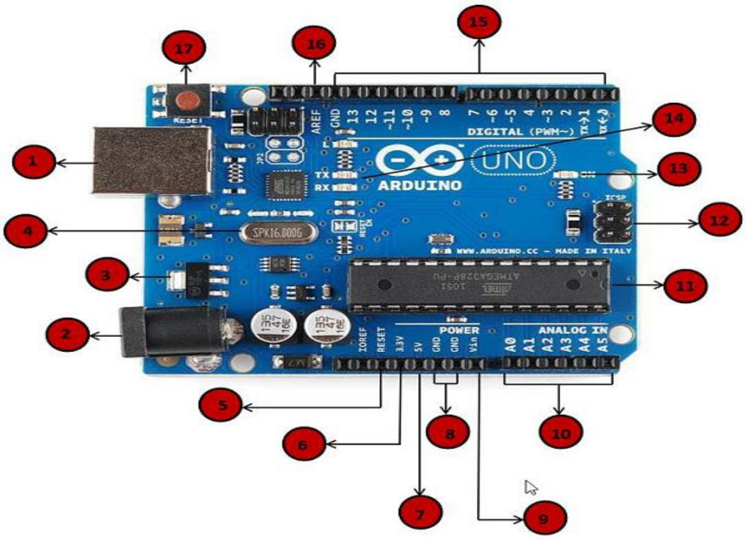


Fig 1.6.Arduino Uno.

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).

6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Table1.1:Arduino Board Description.

1.7 Advantages of Arduino:

- It is cheap
- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
- For beginners, it is very simple to use.

1.8 Applications:

Arduino is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed.

1.9 ARDUINO FAMILY

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project , check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

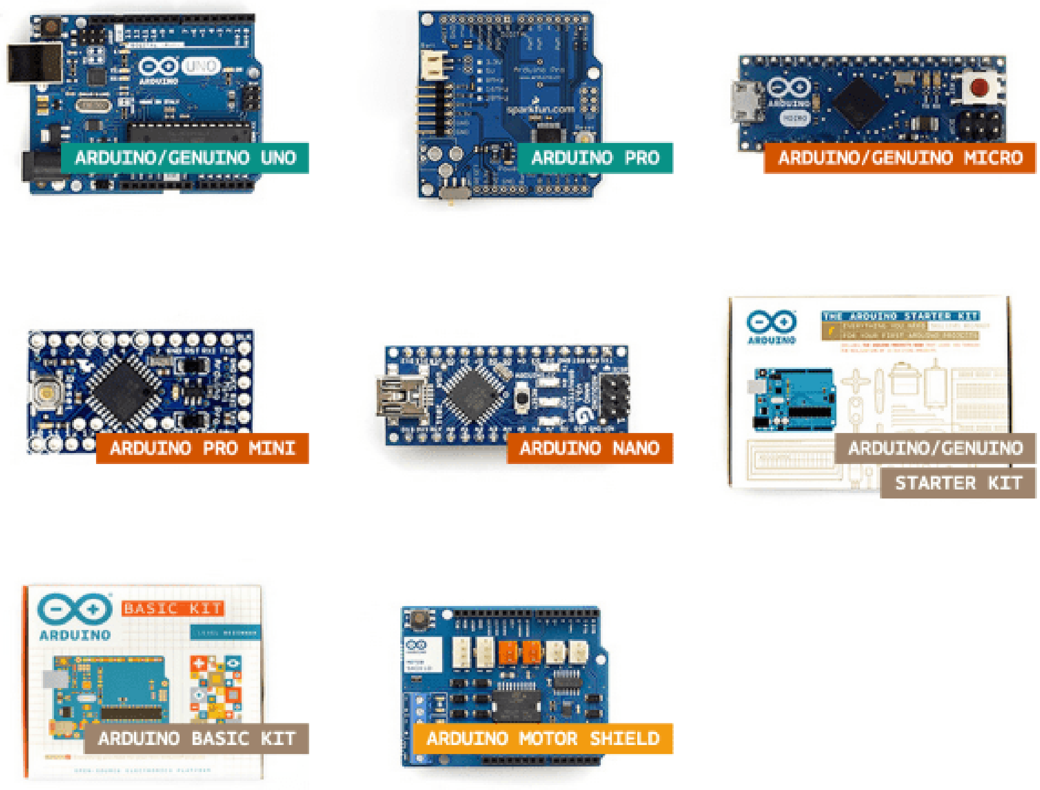


Fig.1.7: Arduino Family

1.10 SHIELDS

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

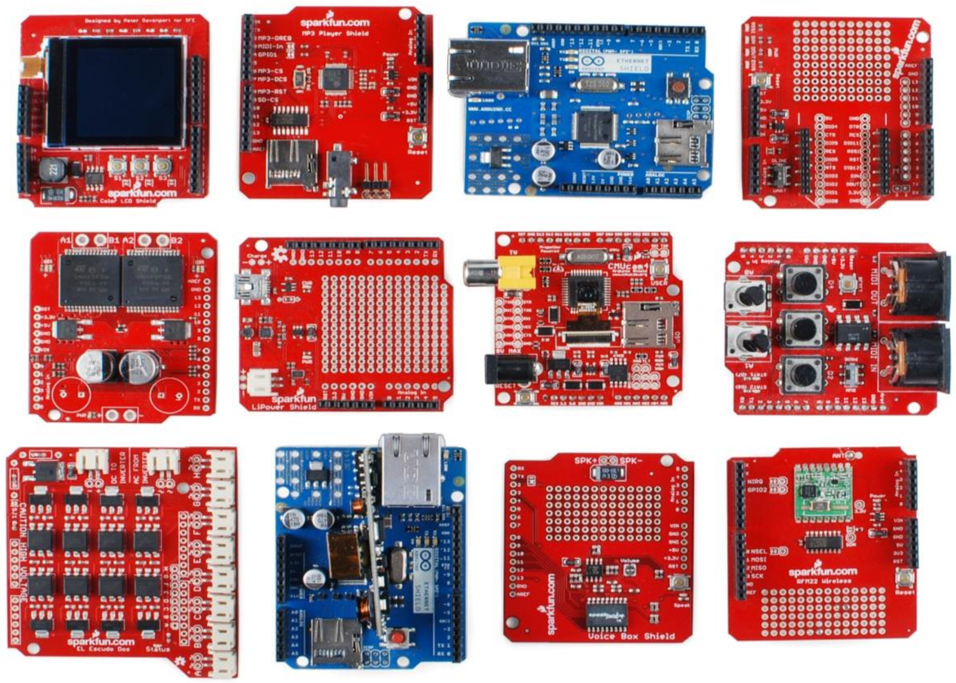


Fig.1.8: Arduino Shields

1.11 PIN DESCRIPTION OF ATMEGA328

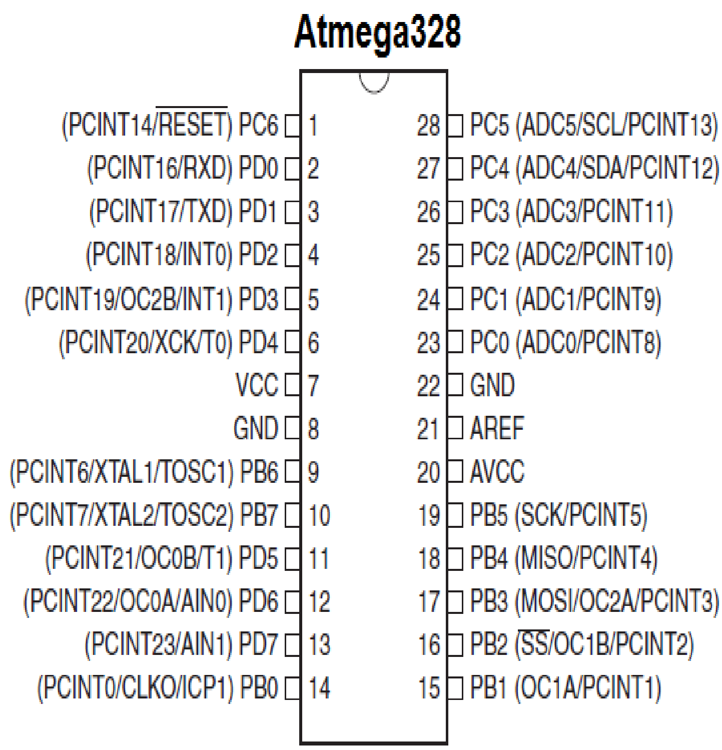


Fig.1.9: Pin description of ATMEGA328

1.12 INTRODUCTION OF POWER SUPPLY:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as “Regulated D.C Power Supply”.

1.13 BLOCK DIAGRAM OF POWER SUPPLY:

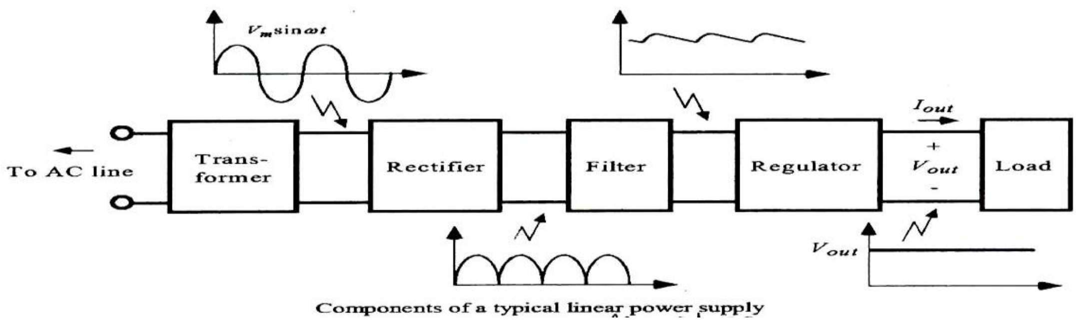


Fig.1.10: Block Diagram of Power Supply

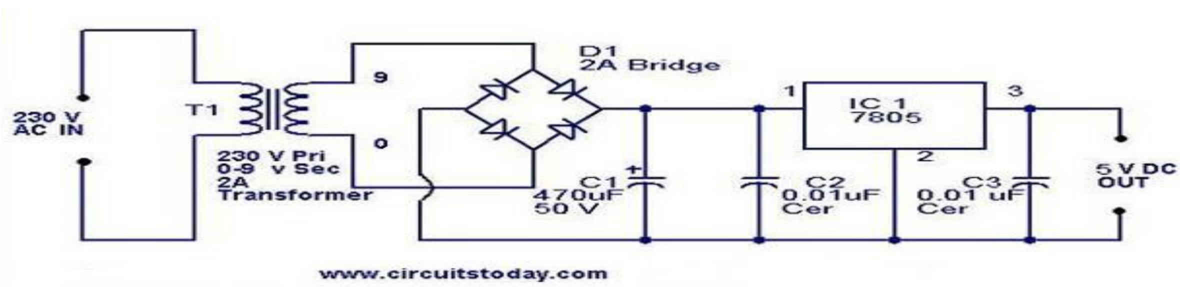


Fig.1.11: Schematic Diagram of Power Supply

1.14 DESCRIPTION OF POWER SUPPLY:

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert aa different form of energy – such as solar, mechanical, or chemical - into electrical energy.

A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered. For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

1.15 TRANSFORMER:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupledconductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

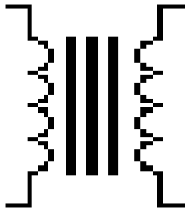


Fig.1.12: Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

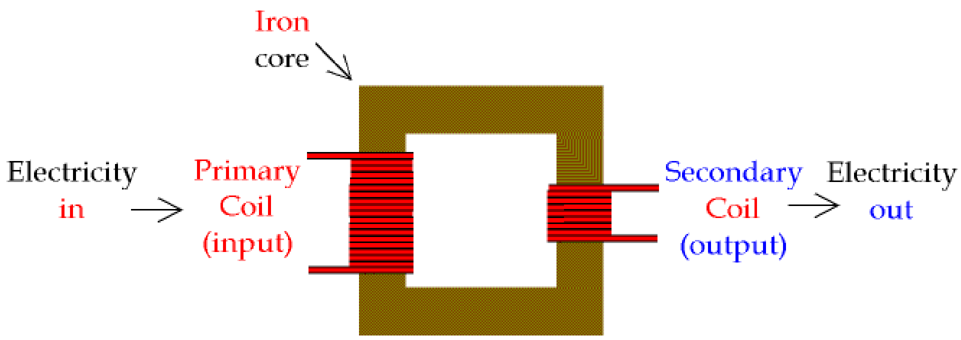
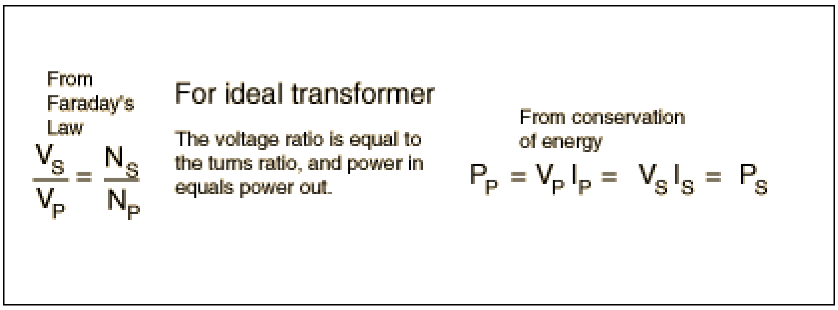


Fig.1.12(a): Transformer

1.16 Basic Principle of Transformer:

A transformer makes use of Faraday's law and the ferromagnetic properties of an ironcore to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.



1.17 Working of Transformer:

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

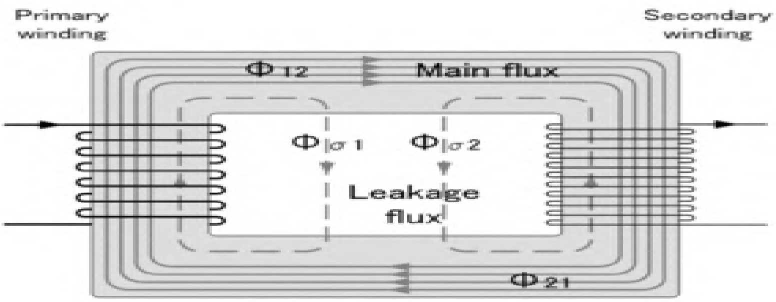


Fig.1.13: Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f. The iron core is laminated to prevent 'eddy currents' flowing in the core.

These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

1.18 Classification of Transformer:

➢ Step-Up Transformer

➢ Step-Down Transformer

1.18.1 Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn's ratio of the two sets of windings determines the amount of voltage transformation.

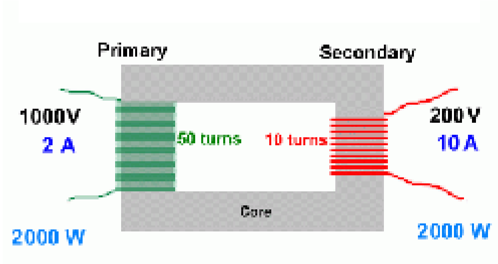


Fig.4.2.2: Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1

1.19 Software:

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

CHAPTER-2
LITERATURE SURVEY

CHAPTER-2

2.1 Literature Survey:

Robotics is a field concerned with the “intelligent connection of perception of action”. The most common manufacturing robot is the robotic arm with different degree of freedom. Today, these humanoids perform many functions to assist humans in different undertakings such as space missions, driving and monitoring high speed vehicles. They are called Humanoids because they resemble humans. The idea of this paper is to change perception of controlling robotic arm. This paper provides a way to get rid of old fashioned remote controls and gives an intuitive technique for implementation of Semi- Humanoid Gesture controlled robot. The motion sensors used are - flex and accelerometer (used in mobile phones for tilting motion). It includes two robotic arms which are exactly similar to human arms (with 5 fingers), increasing sensitivity of the system, a moving platform and a camera for capturing real time video. The prime aim of the system is to design a system in such a way that user can operate the robot with the ease through gestures. This involves movement of both robotic arms and platform in synchronous with user's hand and leg gestures respectively.

This paper illustrates the design and implementation of a robotic vehicle that can be controlled from all directions by using the hand movement with a wireless camera that is installed at the top of the vehicle to broadcast wireless live streaming to the user end. Therefore, it removes the hassle of gesture recognition and image processing technique or even the use of switches or joysticks to control the movement of a robot in different directions and provides a wireless surveillance facility to the user. There are mainly three parts of this propose system; the transmitter, the receiver and the live streaming section. The transmitter section detects the movement of the hand in x and y axis by using an accelerometer sensor, processed by Arduino Uno, and transmits those signals to the receiver section by using RF module. The receiver section receives those signals via RF module, decoded by the Arduino Uno, and the motor driver, which is connected to the Arduino Uno and moves the robotic vehicle in all directions according to the decoded signals. Finally, in order to perform live streaming operation - a common network (Wi-Fi), a laptop, Raspberry pi, and two software, namely Virtual Network Computing (VNC) and Foundation Internet Nouvelle Generation (FING) are used to ensure the surveillance feature for the user.

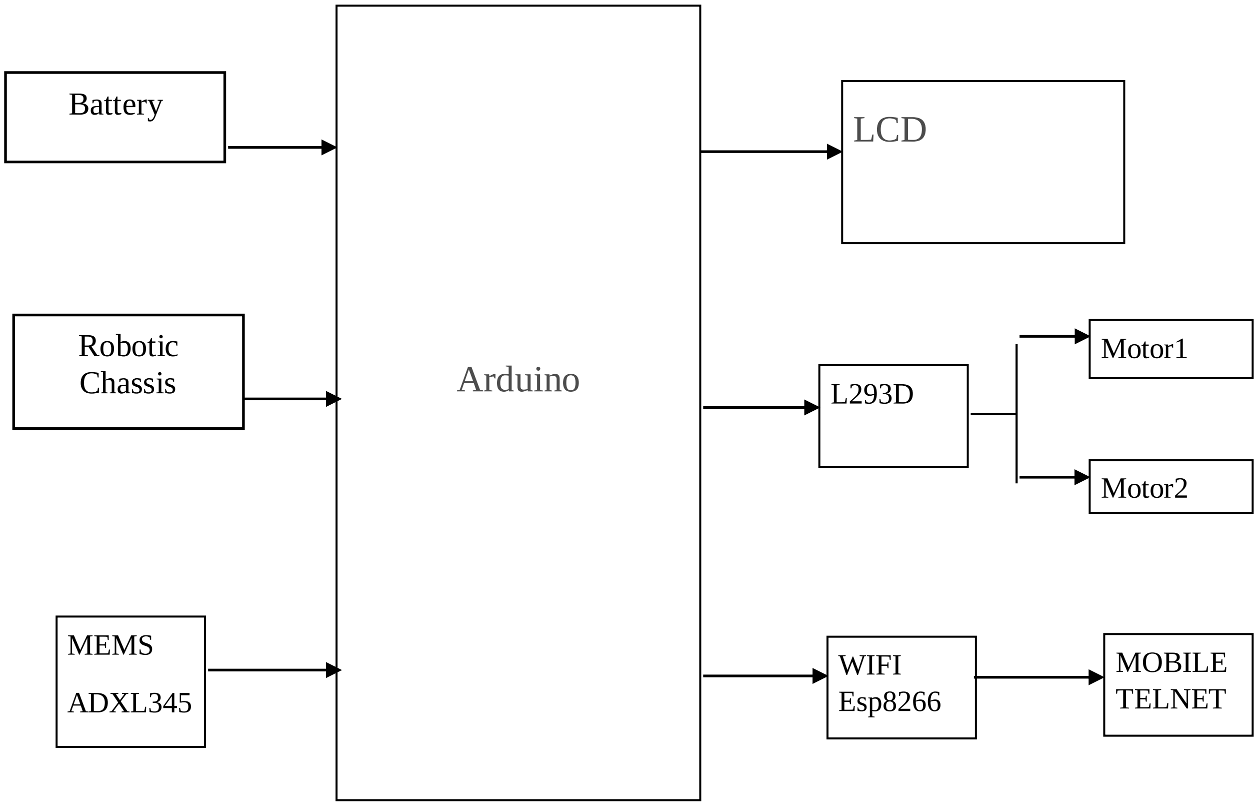
Robotics is one of the emerging fields nowadays. It can be defined as a design gadget which can assist humans in their day-to-day activities and help them by amalgamating electronics and mechanical engineering. Robots are assuming a significant number of jobs over sectors like construction, military, medical, etc. Various attempts have been implemented to make interfaces among users and PCs put together frameworks based on human gestures. These gesture based interfaces can substitute the regular interface gadgets. In the wake of making some essential robots like a line follower robot or a computer-controlled robot, human gestures can operate these types of robots and therefore an accelerometer based Hand Gesture Robot is on the rise. This technique will reduce the dichotomy between the digital and physical world. In this paper, we will see how a robot can move by using Hand Gestures.

This paper explains the solar grass cutter using hand gesture, solar powered robotic grass cutting machine that eliminates the obstacles and could be accessed from every direction by utilizing hand gesture through remote camera which is introduced to the highest point of the motor-vehicle to communicate remote gushing to the users end. The system uses a 6v batteries to drive both the motor vehicle movement and the grass cutter engine. The system also uses solar panels to charge the battery, so that no additional battery is required. The grass cutter and vehicle motors are interfaced with a microcontroller's family which controls all the motors running. It is also interfaced with ultrasonic sensor to track the object. There are essentially three parts of the framework; the transmitter; the receiver and the live gushing area.

CHAPTER-3
PROPOSED SYSTEM

CHAPTER-3

3.1 Block Diagram:



3.2 Arduino

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).

- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

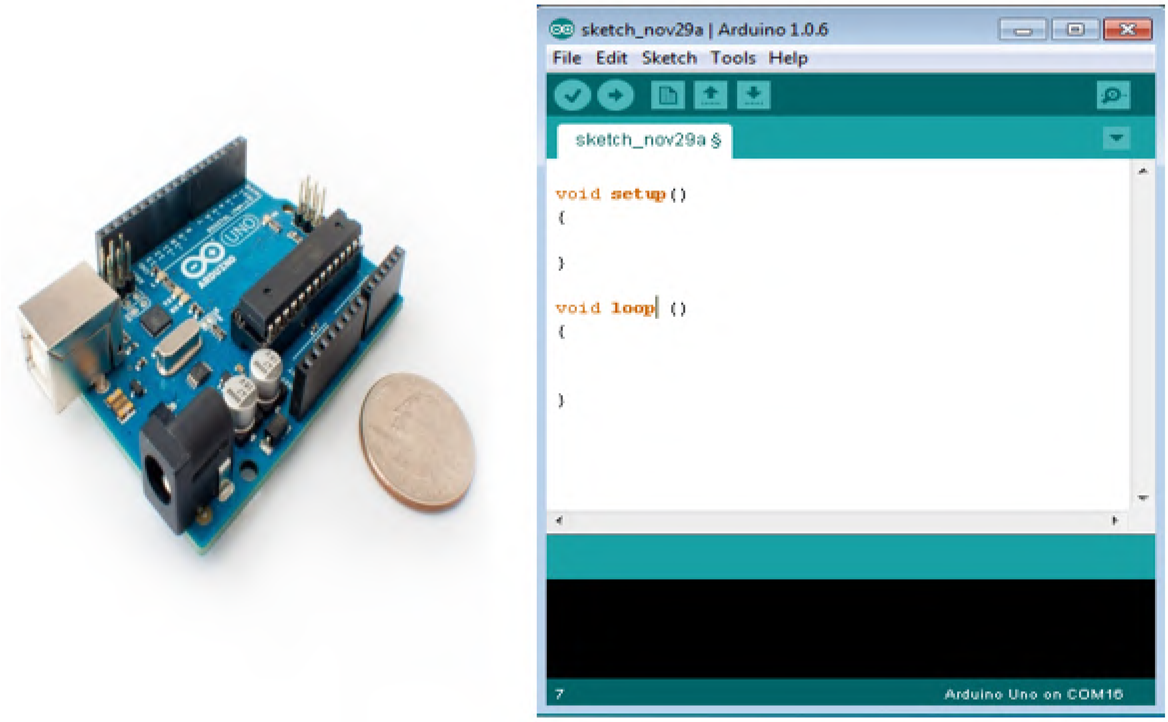


Fig 3.1 Arduino UNO and IDE

3.2.1 Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2

Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.1 Arduino boards based on ATMEGA328 microcontroller

3.3 16 * 2 Alphanumeric LCD

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

3.3.1 16 x 2 Alphanumeric LCD Module Features

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

Schematic

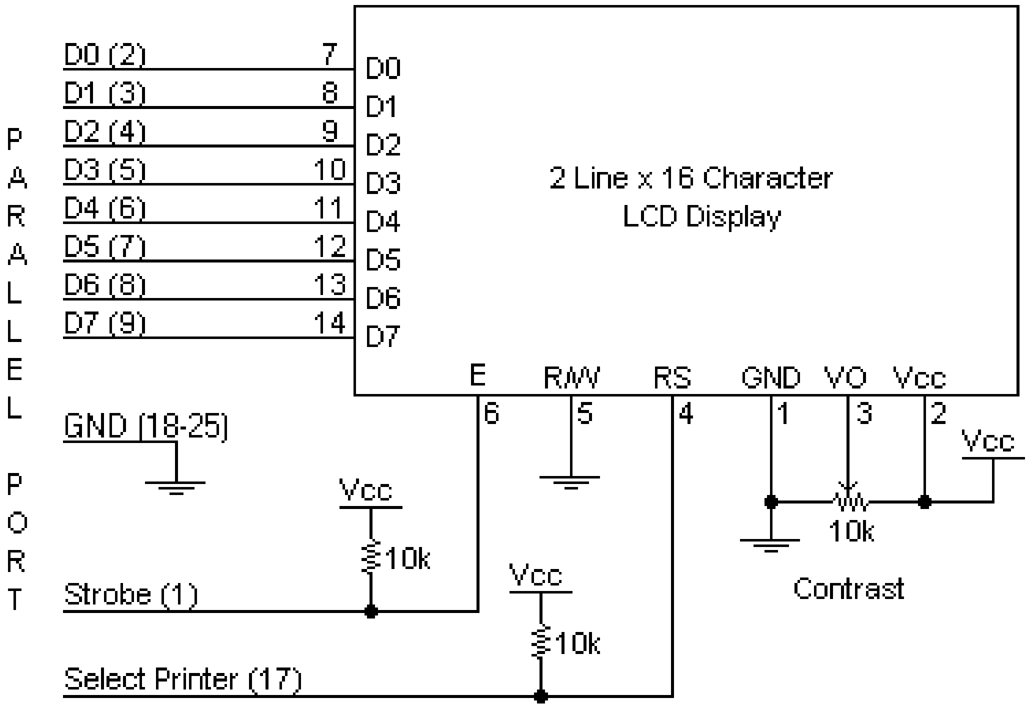


Fig 3.2 Schematic Diagram of LCD

3.3.2 Specifications

Connector Pin Assignment:

Pin	*Symbol*	*Function*	*Pin*	*Symbol*	*Function*
14	DB7	Data Bus Line	6	E	Enable
13	DB6	Data Bus Line	5	R/W	Read/Write
12	DB5	Data Bus Line	4	RS	Register select input
11	DB4	Data Bus Line	3	Vcontrast	Contrast (0V for max contrast)
10	DB3	Data Bus Line	2	**GND (0V)**	Power supply
9	DB2	Data Bus Line	1	**Vcc +5V**	Power supply
8	DB1	Data Bus Line			
7	DB0	Data Bus Line			

Note1: Pin 1 is +5V and pin 2 is GND! This is different from most other HD44780 compatible models.

Note2: Pin 14 is marked with a "1" on the back of the PCB. Do not get confused by this.

Table3.3 Specifications

3.4 Battery:

Batteries are a collection of one or more cells whose chemical reactions create a flow of electrons in a circuit. All batteries are made up of three basic components: an anode (the '-' side), a cathode (the '+' side), and some kind of electrolyte (a substance that chemically reacts with the anode and cathode).

When the anode and cathode of a battery is connected to a circuit, a chemical reaction takes place between the anode and the electrolyte. This reaction causes electrons to flow through the circuit and back into the cathode where another chemical reaction takes place.

Batteries that must be thrown away after use are known as **primary batteries**. Batteries that can be recharged are called **secondary batteries**.

Fig 3.2: Lithium Battery

Lithium polymer batteries, for example, can be recharged without batteries, your quadcopter would have to be tethered to the wall, you would have to hand crank your car, and your Xbox controller would have to plugged in all the time (like in the good old days). Batteries offer a way to store electrical potential energy in a portable container.

3.4.1 History

Historically, the word "battery" was used to describe a "series of similar objects grouped together to perform a function," as in a battery of artillery. In 1749, Benjamin Franklin first used the term to describe a series of capacitors he had linked together for his electricity experiments. Later, the term would be used for any electrochemical cells linked together for the purpose of providing electric power.

Fig 3.3 Leydan jar

Battery of Leyden Jar "capacitors" linked together (Image courtesy of Alvinrune of Wikimedia Commons)

3.5 Wifi Module:

The ESP8266 is a low-cost WiFi module that can be integrated easily into IoT devices. We've featured several projects using this module, such as How To Make Smart Home Electronics: A Smart Mailbox and How To Read Your Arduino's Mind: Building A Childproof Lock. This tutorial will walk you through setting up ESP8266 Wifi module which can be used with Arduino. The ESP8266 comes in many models with different functionalities. We'll be focusing on the ESP8266 ESP-01 module, the most common and basic one available.

3.5.1 What is ESP8266?

The ESP8266 is a small WiFi module built around the ESP8266 chip that can connect your microcontroller to the internet wirelessly for a very small cost. It can be a great option for Internet of Things (IoT) projects, but can be difficult to work with for beginner hobbyists who do not have prior experience with the module. In this tutorial, we hope to show you how to interface the ESP8266 with an Arduino and perform some basic functions like connecting it to a WiFi network.

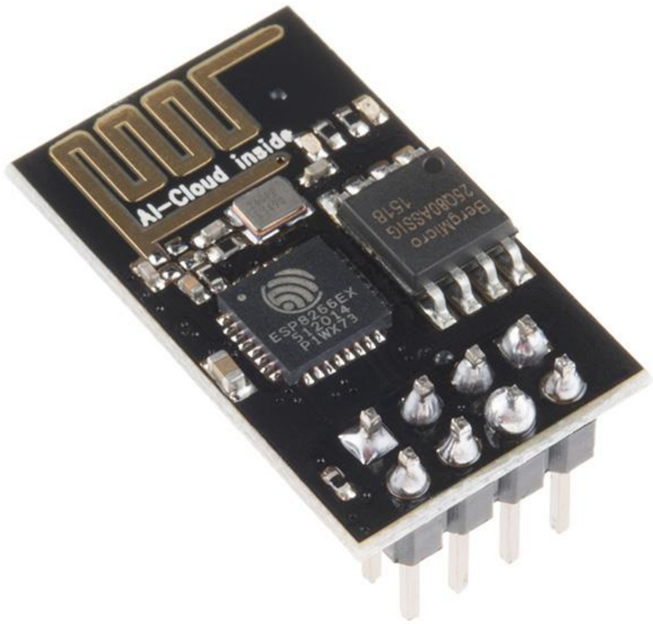


Fig 3.4. ESP8266 ESP-01 module

ESP-01 Features

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 1MB Flash Memory
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)

The first feature to notice about the ESP8266 is its awkwardly spaced header pins. The module has 8 pins that serve different functions, but they are packed in a 4×2 arrangement that makes plugging the module into a breadboard impossible. This means that to prototype projects on a breadboard, you'll need male-female jumper wires to connect the pins on the ESP8266 to rows on the breadboard. If you'd like to make your prototyping more compact, you can also purchase breadboard breakouts for the ESP8266 such as this one. For prototyping, I chose to just use jumper wires.

The pinout for the ESP8266's pins are according to the following diagram:

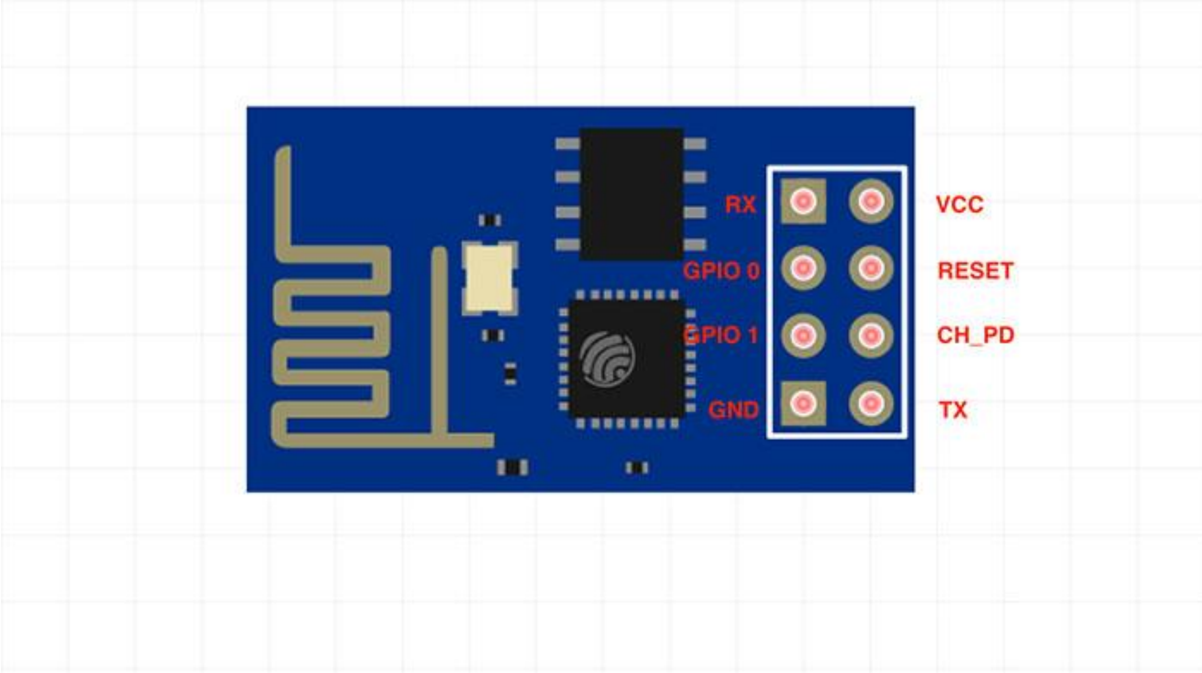


Fig 3.5 ESP8266 Pinout

RX: UART serial communication receive pin

GPIO 0: GPIO pin (unused in this project)

GPIO 1: GPIO pin (unused in this project)

GND: Connection to Ground

VCC: Connection to 3.3V Vcc (Vcc cannot exceed 3.3V!)

RESET: Reset pin (pull down to reset)

CH_PD: Chip enable and power down pin

TX: UART serial communication transmit pin

Note that the maximum voltage input for the ESP8266 is 3.3V. Any input voltage greater than 3.3V will damage the module! To program settings on the ESP8266, we'll first need to connect it to a serial terminal on a computer through which we can send it special commands. Settings that we'll have to program include, for example, the SSID and password for the wifi network the module will be connected to. To connect the ESP8266 to a computer and configure its settings, we'll need a USB to serial adapter with 3.3V logic, along with a serial terminal program. Fortunately for us, we have the Arduino and the Arduino IDE's serial monitor! This means that we'll just have to connect the ESP8266 module to the Arduino and upload a custom sketch to the Arduino.

Use the following diagram to connect the ESP8266 module to the Arduino:

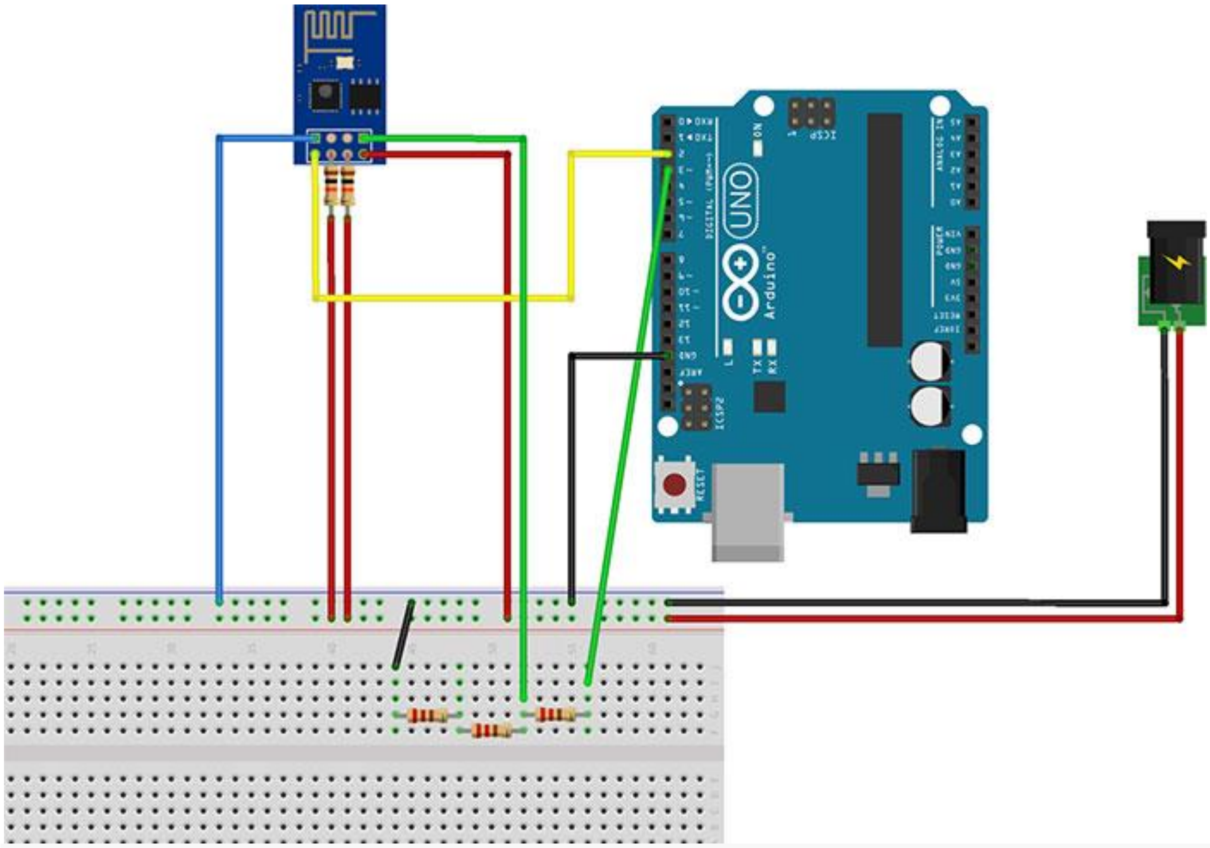


Fig 3.6 Connection between ESP8266 and Arduino Uno

You will need the following parts:

- Arduino Uno or similar module
- ESP8266 ESP-01 module
- 2 x 1kΩ resistors for CH_PD and RESET pull-up
- 3 x 220Ω resistors for serial line voltage divider
- 3.3V regulated power supply for ESP8266 (you cannot plug the power line into the Arduino's 3.3V out pin because the ESP8266 draws a lot of current!)
    - You can use something like this: https://www.sparkfun.com/products/114
- Breadboard and jumper wires

A couple of features of this circuit stand out immediately.

First, the ESP8266 module is powered by a 3.3V regulated power supply. As described earlier, the ESP8266 has a maximum voltage input rating of 3.3 volts. Using a power supply rated higher than this recommended 3.3 volts could fry the module! When connecting the ESP8266 to this power supply, you should not only connect the Vcc and Ground pins to the power supply lines, but also connect the power supply's ground line to the Arduino's ground pin. Since the Arduino will be powered by the USB connection to the laptop, creating a common ground essentially creates a common reference (you can think of it as a baseline) to compare voltages and thereby interpret digital high and low signals.

Second, the receive (RX) line for the ESP8266 module is connected to the output of a resistor voltage-divider circuit. We do this to shift the serial communication logic level (the highs and lows of the digital signals that make up the serial communications) from a logic high of 5 volts on the Arduino to a logic high of 3.3 volts on the ESP8266 module. Again, the ESP8266 is specified with 3.3V logic, so connecting the module's receive line directly to the Arduino's transmit line could damage the device. We do not need to shift levels on the ESP8266's transmit line because the module's 3.3V logic high is a high enough voltage to also register as a logic high on the Arduino.

Finally, there are two pull-up resistors on the ESP8266's CH_PD and RESET pins. These are specified by the module's creators to ensure that the device functions properly while it is being used. To reset the ESP8266 or disable it, these pins must be pulled down, but since we do not want those processes to occur, we will instead pull these pins high through some 10kΩ pull-up resistors.

3.6 MEMS:

The ADXL345 is a low-power, 3-axis MEMS accelerometer modules with both I2C and SPI interfaces. The Adafruit Breakout boards for these modules feature on-board 3.3v voltage regulation and level shifting which makes them simple to interface with 5v microcontrollers such as the Arduino.

The ADXL345 features 4 sensitivity ranges from +/- 2G to +/- 16G. And it supports output data rates ranging from 10Hz to 3200Hz.

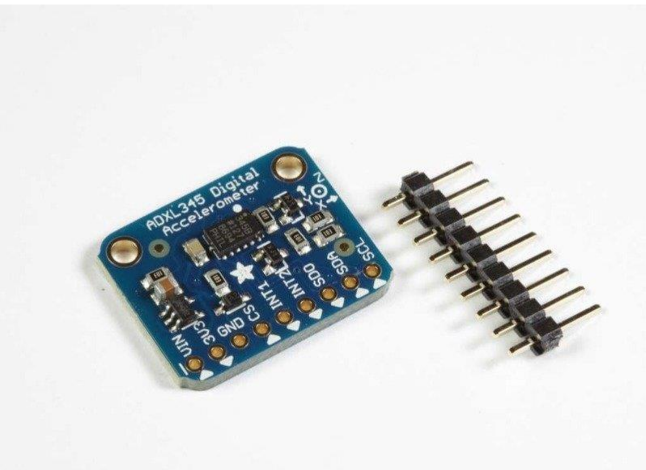


Fig3.7: ADXL345 Sensor

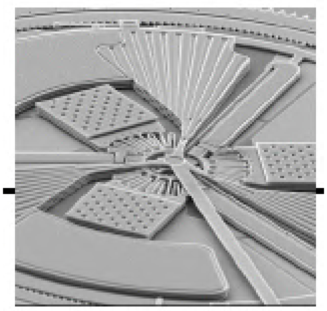
The sensor consists of a micro-machined structure on a silicon wafer. The structure is suspended by polysilicon springs which allow it to deflect smoothly in any direction when subject to acceleration in the X, Y and/or Z axis. Deflection causes a change in capacitance between fixed plates and plates attached to the suspended structure. This change in capacitance on each axis is converted to an output voltage proportional to the acceleration on that axis.

Introduction :

Microelectromechanical systems (MEMS) is the technology of the very small, and merges at the nano-scale into nano-electromechanical systems. MEMS are also referred to as micro-machines (in Japan), or Micro Systems Technology - MST (in Europe).MEMS is an emerging technology which uses the tools and techniques that were developed for the Integrated Circuit industry to build microscopic machines. These machines are built on standard silicon wafers. The real power of this technology is that many machines can be built at the same time across the surface of the wafer, with no assembly required. Since it is a photographic-like process, it is just as easy to build a million machines on the wafer as it would be to build just one.

MEMS is the integration of mechanical elements, sensors, actuators, and electronics on a common silicon substrate through micro-fabrication technology. While the electronics are fabricated using integrated circuit (IC) process sequences (e.g., CMOS, Bipolar, or BICMOS processes), the micro-mechanical components are fabricated using compatible "micro-machining" processes that selectively etch away parts of the silicon wafer or add new structural layers to form the mechanical and electromechanical devices.

Imagine a machine so small that it is imperceptible to the human eye. Imagine working machines no bigger than a grain of pollen. Imagine thousands of these machines batch fabricated on a single piece of silicon, for just a few pennies each. Imagine a world where gravity and inertia are no longer important, but atomic forces and surface science dominate. Imagine a silicon chip with thousands of microscopic mirrors working in unison, enabling the all optical network and removing the bottlenecks from the global telecommunications infrastructure. You are now entering the microdomain, a world occupied by an explosive technology known as MEMS. A world of challenge and opportunity, where traditional engineering concepts are turned upside down, and the realm of the "possible" is totally redefined.



A truly amazing MEMS device. It is a sophisticated MEMS Thermal Actuator

Fig 3.8 MEMS Thermal Actuator

MEMS are quietly changing the way you live, in ways that you might never imagine. The device that senses your car has been in an accident, and fires the airbag is a MEMS device. Most new cars have over a dozen MEMS devices, making your car safer, more energy efficient, and more environmentally friendly. MEMS are finding their way into a variety of medical devices, and everyday consumer products.

MEMS are quietly changing the way you live, in ways that you might never imagine. The device that senses your car has been in an accident, and fires the airbag is a MEMS device. Most new cars have over a dozen MEMS devices, making your car safer, more energy efficient, and more environmentally friendly. MEMS are finding their way into a variety of medical devices, and everyday consumer products.

Microelectronic integrated circuits can be thought of as the "brains" of a system and MEMS augments this decision-making capability with "eyes" and "arms", to allow Microsystems to sense and control the environment. Sensors gather information from the environment through measuring mechanical, thermal, biological, chemical, optical, and magnetic phenomena. The electronics then process the information derived from the sensors and through some decision making capability direct the actuators to respond by moving, positioning, regulating, pumping, and filtering, thereby controlling the environment for some desired outcome or purpose. Because MEMS devices are manufactured using batch fabrication techniques similar to those used for integrated circuits, unprecedented levels of functionality, reliability, and sophistication can be placed on a small silicon chip at a relatively low cost.

3.6.1 THEORY OF OPERATION

The ADXL103/ADXL203 are complete acceleration measurement systems on a single, monolithic IC. The ADXL103 is a single-axis accelerometer, and the ADXL203 is a dual-axis accelerometer. Both parts contain a polysiliconsurfacemicromachined sensor and signal conditioning circuitry to implement an open-loop acceleration measurement architecture. The output signals are analog voltages proportional to acceleration. The ADXL103/ADXL203 are capable of measuring both positive and negative accelerations to at least ±1.7 *g*. The accelerometer can measure static acceleration forces such as gravity, allowing it to be used as a tilt sensor.

The sensor is a surfacemicromachinedpolysilicon structure built on top of the silicon wafer. Polysilicon springs suspend the structure over the surface of the wafer and provide a resistance against acceleration forces. Deflection of the structure is measured using a differential capacitor that consists of independent fixed plates and plates attached to the moving mass. The fixed plates are driven by 180° out-of-phase square waves. Acceleration deflects the beam and unbalances the differential capacitor, resulting in an output square wave whose amplitude is proportional to acceleration. Phase-sensitive demodulation techniques are then used to rectify the signal and determine the direction of the acceleration. The output of the demodulator is amplified and brought off-chip through a 32 kΩ resistor. At this point, the user can set the signal bandwidth of the device by adding a capacitor. This filtering improves measurement resolution and helps prevent aliasing.

Pin configuration and function description

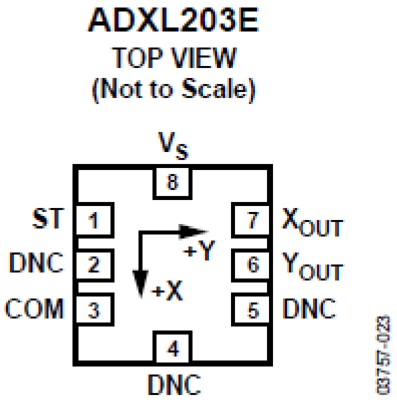


Figure 4. ADXL203 Pin Configuration

Table 5. ADXL203 Pin Function Descriptions

Pin No.	Mnemonic	Description
1	ST	Self Test
2	DNC	Do Not Connect
3	COM	Common
4	DNC	Do Not Connect
5	DNC	Do Not Connect
6	Y_{OUT}	Y Channel Output
7	X_{OUT}	X Channel Output
8	V_S	3 V to 6 V

Table3.4 ADXL203E Pin Configuration

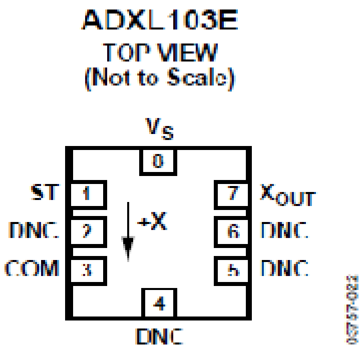


Figure 3. ADXL103 Pin Configuration

Table 4. ADXL103 Pin Function Descriptions

Pin No.	Mnemonic	Description
1	ST	Self Test
2	DNC	Do Not Connect
3	COM	Common
4	DNC	Do Not Connect
5	DNC	Do Not Connect
6	DNC	Do Not Connect
7	X_{OUT}	X Channel Output
8	V_S	3 V to 6 V

Table3.5 ADXL103E Pin Configuration.

3.7 MOBILE TELNET:

Telnet is a client-server protocol based on character-oriented data exchange over TCP connections. Telnet enables remote control of computers via text-based inputs and outputs. For this reason, a client-server connection is established as a default via the TCP protocol and port 23, where the remote-controlled device acts as a server and waits for commands. The Telnet client, the controlling instance in this process (also referred to as remote access or login), can be installed on a particular device, as well as on an ordinary computer. However, the presentation of the transmitted information differs, depending on the device. This protocol can also be used to manage applications that do not have a graphical interface.

As early as 1969 (during the nine months of work on ARPANET), the development of Telnet (Teletype Network) was practically completed. But it was only in 1973 that the protocol allowing access to remote computers first received its final specification in RFC 495 (Request for Comments). It was implemented by most platforms as the official standard of the Internet Engineering Task Force (IETF).

The kernel protocol and the basic working methods and extensions are characterized in the more recent standards RFC 854 and RFC 855.

3.7.1 How and where is Telnet used?

One of the main reasons for developing the remote protocol was that computer systems at that time were still really expensive and not easily accessible for everyone. Another reason was their extreme size, which meant that they were bound to specific locations. In order to make computer resources available at universities and companies, Unix was developed at the end of the 1960s as a suitable operating system and Telnet as an appropriate protocol service. This meant that any user that had the right authorization, could start, manage, and use applications on the powerful large computers for their own personal use.

3.7.2 Access to databases

Over many years, Telnet also played a crucial role for institutions that work with large databases: for example, in libraries, the protocol was an elementary building block of online catalogs published in the 1980s, which are better known as OPAC (Online Public Access Catalog). Initially, these digital publication databases, which followed the traditional card catalogs, were still accessible in the libraries' local networks via terminals. As the internet started becoming more and more successful, it was also possible to access them via locally usable web interfaces, whose communication means was supported by the Telnet protocol in many cases.

3.8 MOTORS

Motor is a device that creates motion, not an engine; it usually refers to either an electrical motor or an internal combustion engine.

It may also refer to:

- Electric motor, a machine that converts electricity into a mechanical motion
    - AC motor, an electric motor that is driven by alternating current
        - Synchronousmotor, an alternating current motor distinguished by a rotor spinning with coils passing magnets at the same rate as the alternating current and resulting magnetic field which drives it
        - Induction motor, also called a squirrel-cage motor, a type of asynchronous alternating current motor where power is supplied to the rotating device by means of electromagnetic induction
    - DC motor, an electric motor that runs on direct current electricity
        - Brushed DC electric motor, an internally commutated electric motor designed to be run from a direct current power source

        - Brushless DC motor, a synchronous electric motor which is powered by direct current electricity and has an electronically controlled commutation system, instead of a mechanical commutation system based on brushes
    - Electrostatic motor, a type of electric motor based on the attraction and repulsion of electric charge
    - Servo motor, an electric motor that operates a servo, commonly used in robotics
    - Internal fan-cooled electric motor, an electric motor that is self-cooled by a fan, typically used for motors with a high energy density

CHAPTER-4
SOFTWARE AND FUNCTIONS

4.1 Arduino software:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments.

A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Fidgets, MIT's Handy-board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50

- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers c an make their own version
- Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money. Getting Started with Arduino and Genuineproducts

Install the Arduino Software (IDE) on Windows PCs-

This document explains how to install the Arduino Software (IDE) on Windows machines.

✓ Download the Arduino Software(IDE)

✓ Proceed with board specific instructions.

4.1.1How to Download the Arduino Software (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

4.1.2 Installation:

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig4.1:USB cable(A plug to B plug)

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig4.2 : USB cable (A plug to Mini-B plug)

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

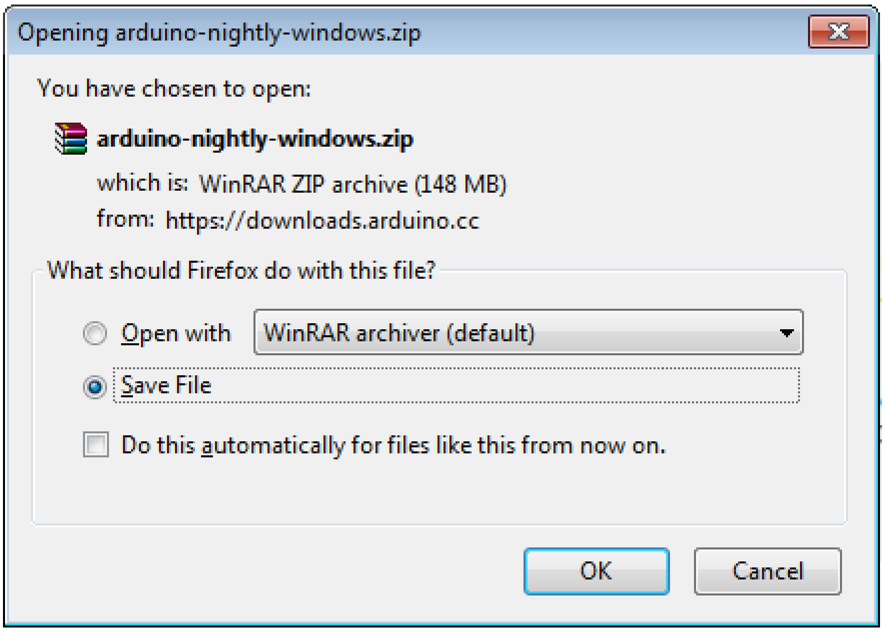


Fig4.3:Installation window of Arduino ide

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power fromthe USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

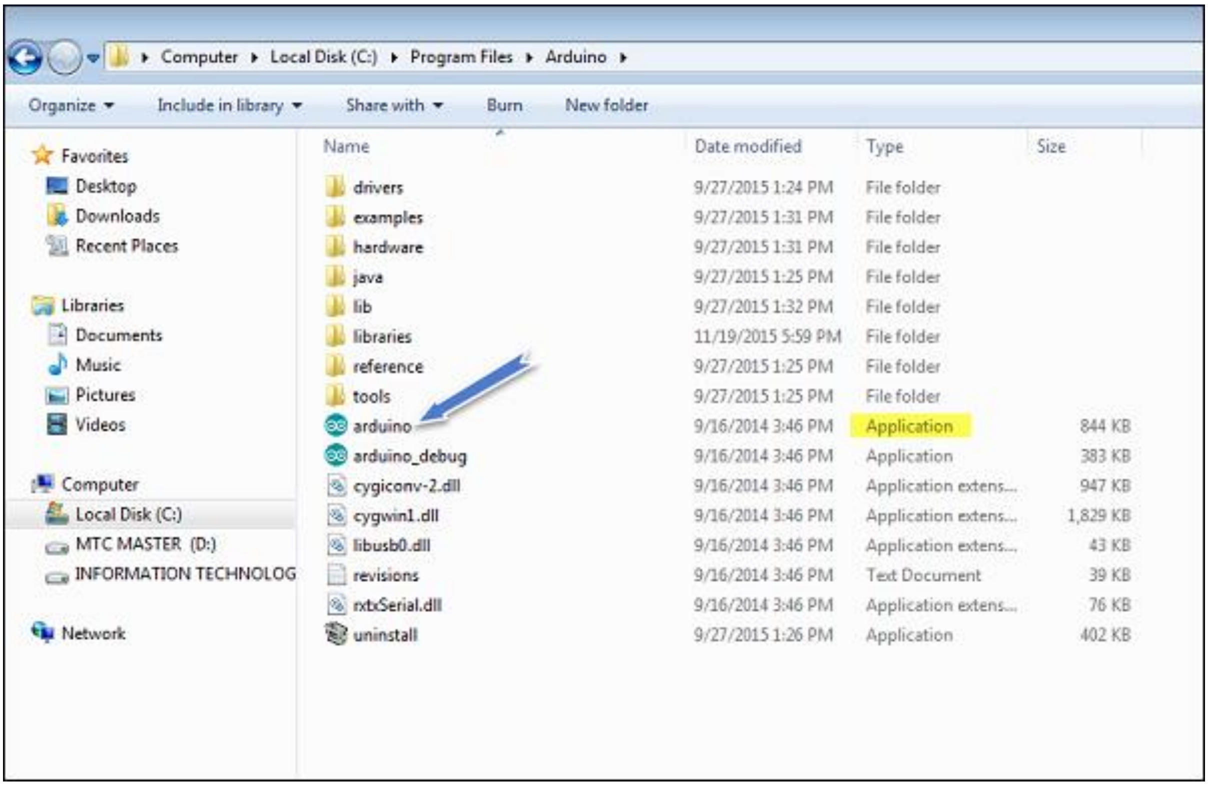


Fig4.4: Window showing how to open arduino ide software

Step 5 – Open your first project.

Once the software starts, you have two options –

- Create a new project.
- Open an existing project example.

To create a new project, select File → New

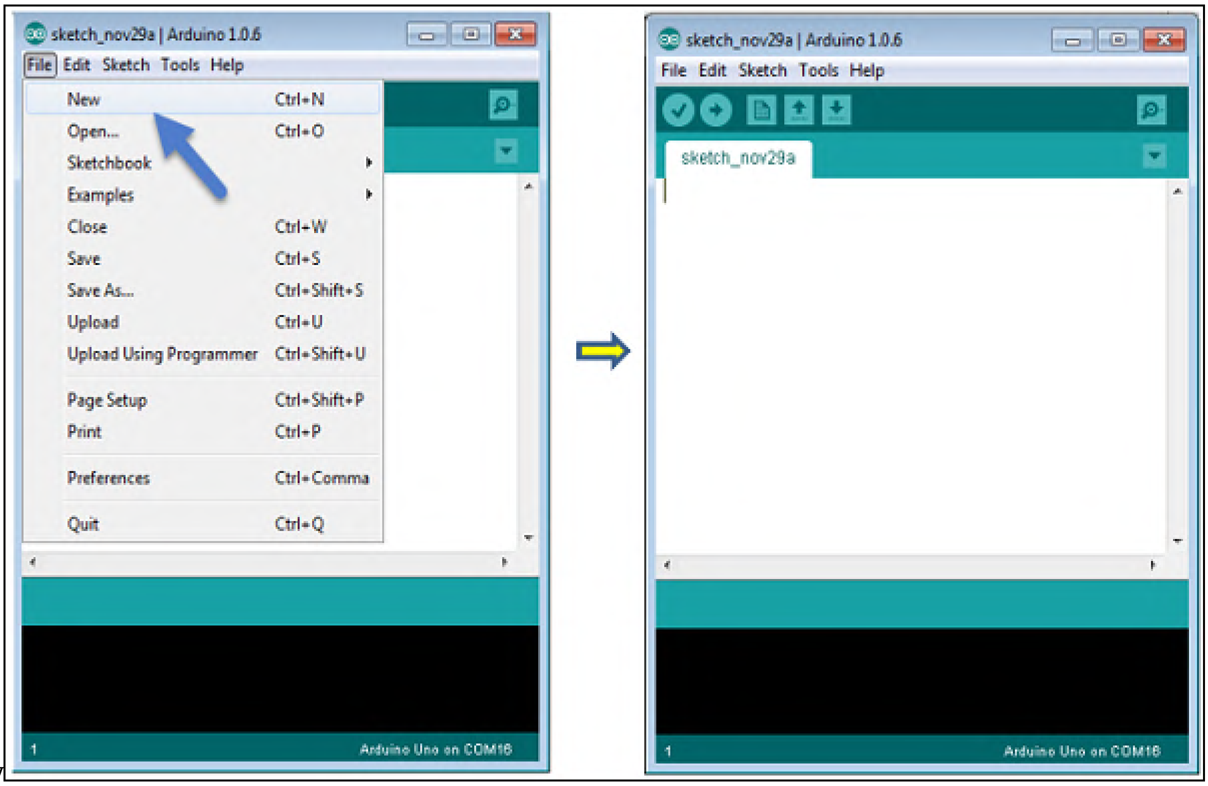


Fig4.5:Window Showing Creating New Project

To open an existing project example, select File → Example → Basics → Blink.

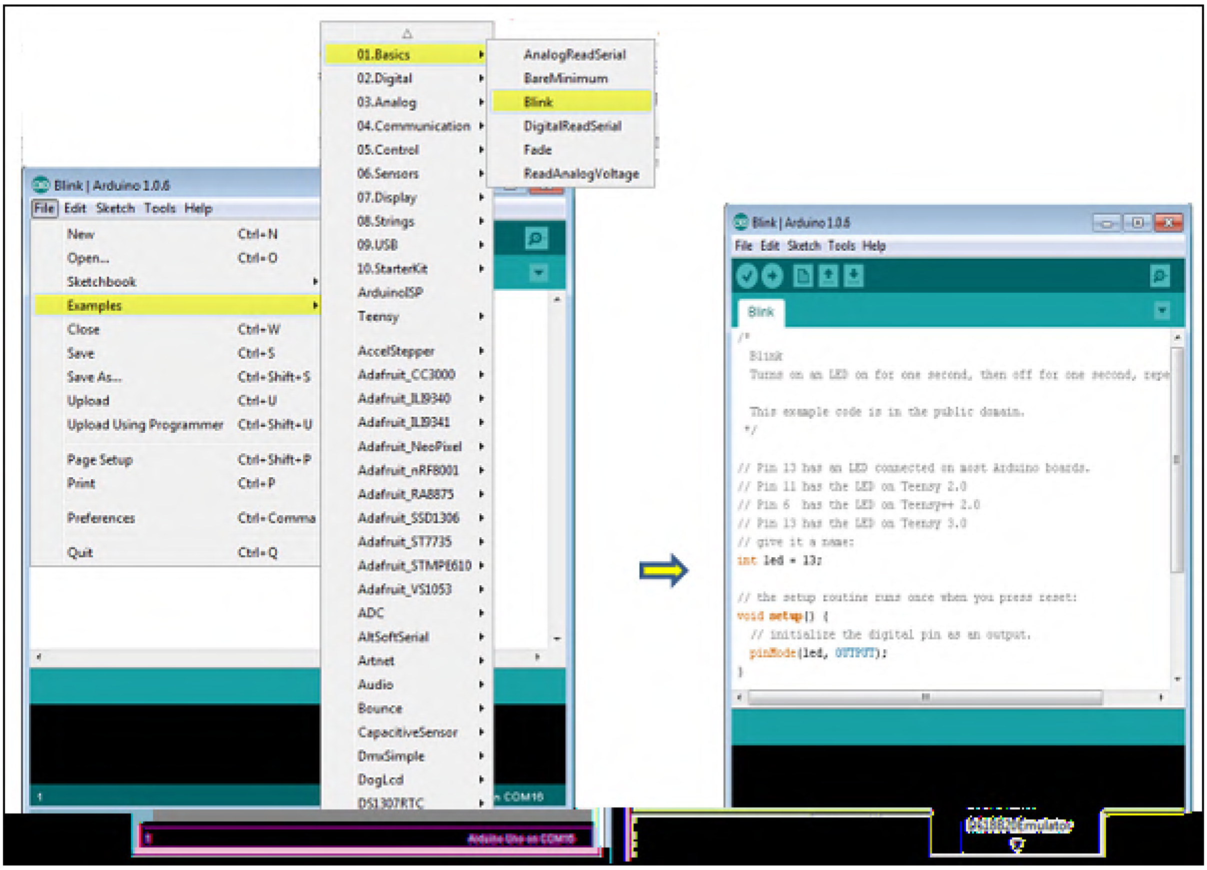


Fig.4.6:Window Showing Opening Existing Project.

Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 – Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

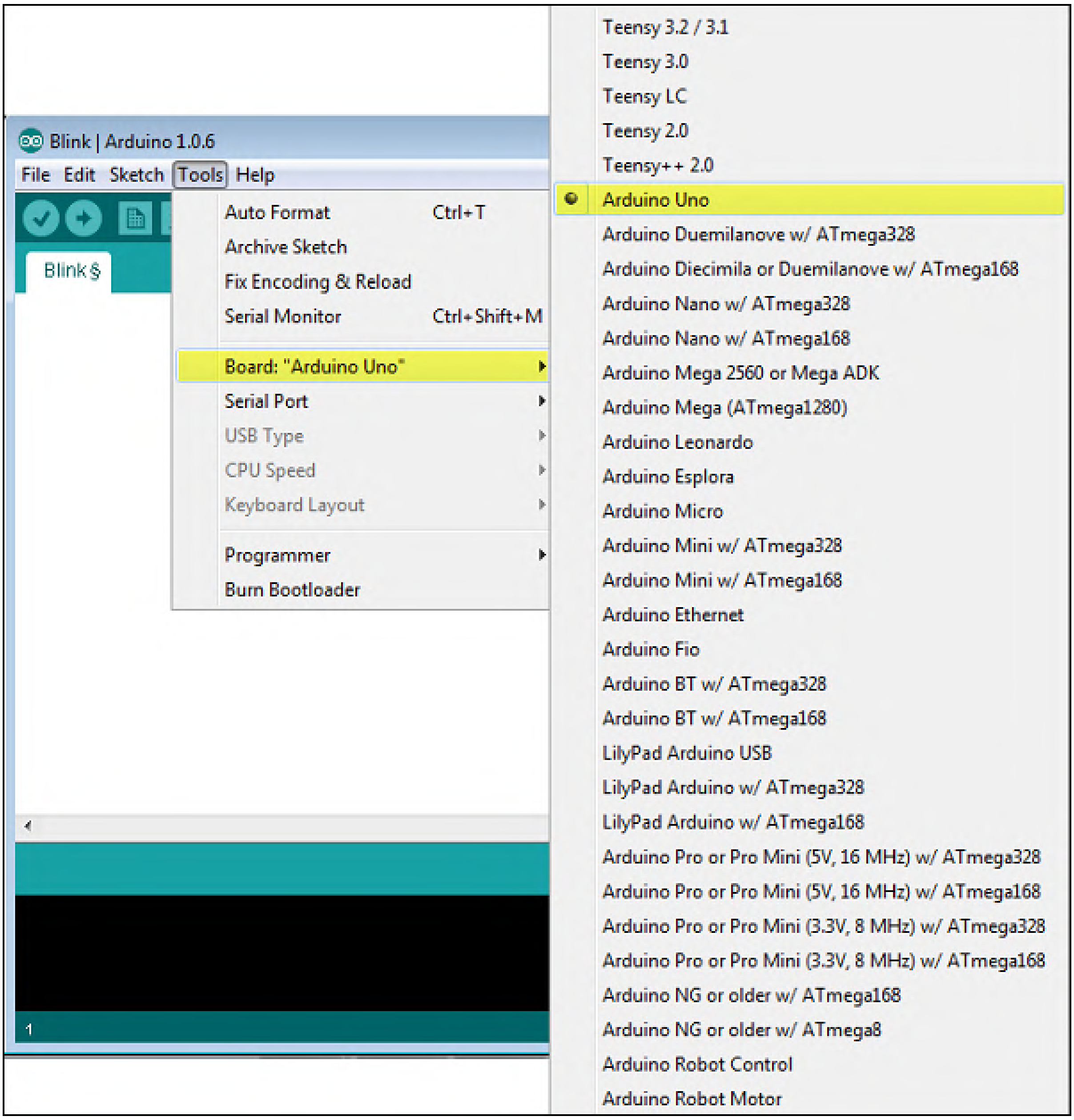


Fig4.7: Window Showing Selection of Boards

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial** Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

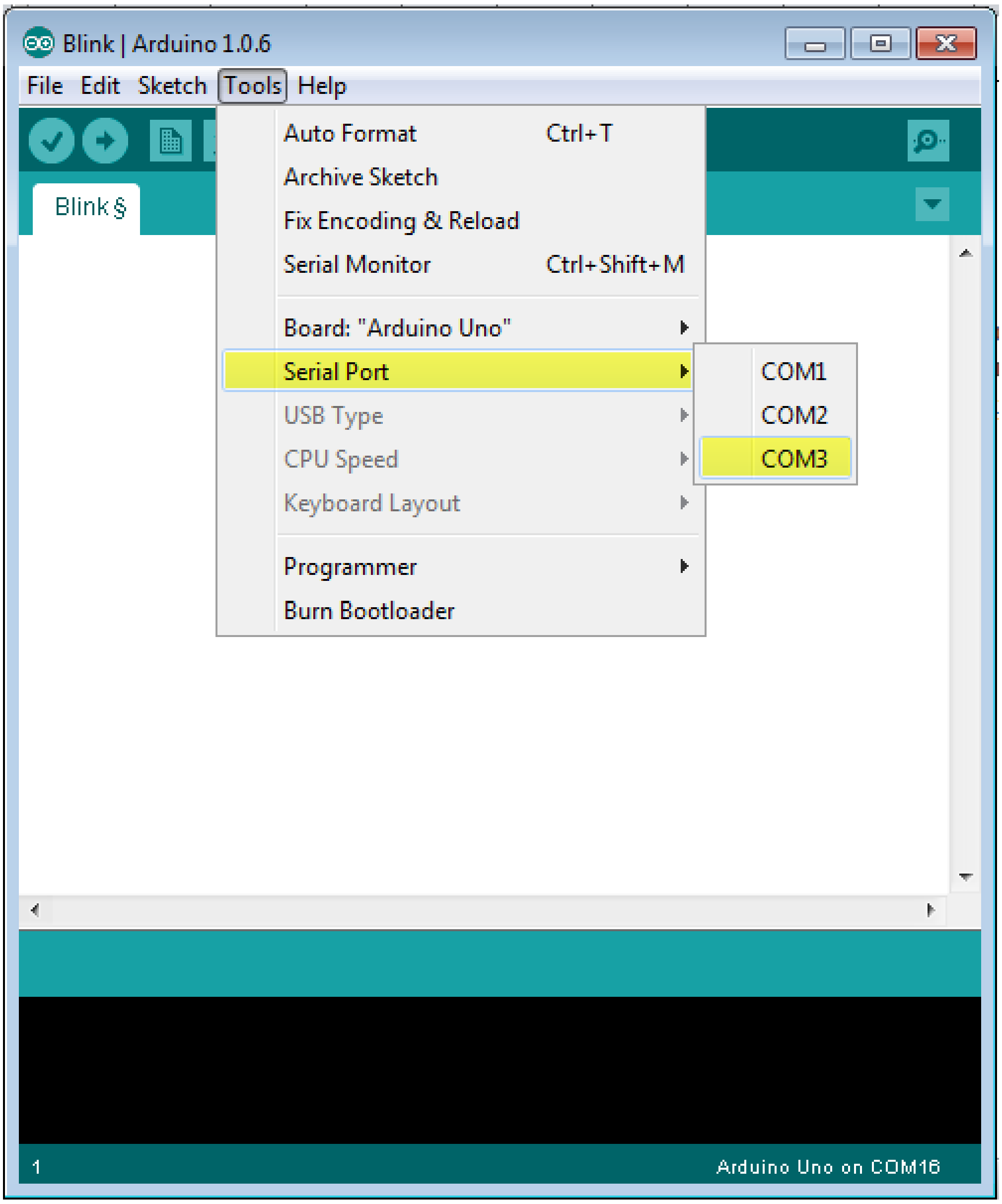


Fig4.8:Window Showing Selection of Ports

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

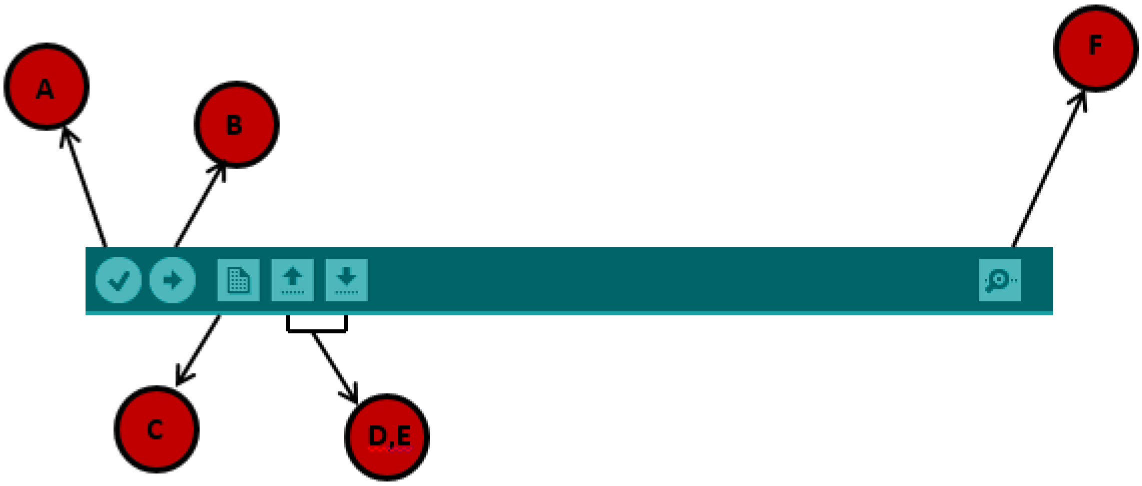


Fig4.9: Explanation of Options In Arduino Ide

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

4.2 MC Programming Language: Embedded C :

This is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements. Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

4.3 Code Implementation:

```
#include<SoftwareSerial.h>
#include<LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);
#include <SoftwareSerial.h>
SoftwareSerialmySerial(2, 3); //RX, TX respectively
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>
/* Assign a unique ID to this sensor at the same time */
Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);

int m1_1=4,m1_2=5,m2_1=6,m2_2=7;

char res[130];
char buff[130];

voidsendwifi(String chr,unsignedintlen)
{
mySerial.print("AT+CIPSEND=0,");
mySerial.println(len-1);
delay(2000);
mySerial.print(chr);
delay(2000);
}

char t;

void setup()
{
```

```
pinMode(m1_1,OUTPUT);
pinMode(m1_2,OUTPUT);
pinMode(m2_1,OUTPUT);
pinMode(m2_2,OUTPUT);
digitalWrite(m1_1,LOW);
digitalWrite(m1_2,LOW);
digitalWrite(m2_1,LOW);
digitalWrite(m2_2,LOW);
lcd.begin(16,2);
Serial.begin(9600);
mySerial.begin(115200);

lcd.begin(16,2);
lcd.clear();lcd.setCursor(0, 0);lcd.print("WELCOME");
delay(1000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("IOT Based");
delay(30);
lcd.setCursor(0,1);
lcd.print("Self Robot");
mySerial.print("AT\r\n");
delay(1000);
mySerial.print("ATE0\r\n");
delay(1000);
mySerial.print("AT+CWMODE=3\r\n");
delay(1000);
mySerial.print("AT+CWSAP=\"Project\",\"project1235\",5,0\r\n\r\n");
delay(1000);
mySerial.print("AT+CIPMUX=1\r\n");
delay(1000);
mySerial.print("AT+CIPSERVER=1,23\r\n");
delay(1000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("WATING FOR LINK");
```

```
while(1)
   {
if(mySerial.available())
     {
        //if(Esp.find("0,LINK"))
if(mySerial.find("0,CONNECT"))
       {
lcd.clear();lcd.setCursor(0, 0);lcd.print("LINK FOUND");
break;
       }


     }
   }
delay(1000);
sendwifi("WELCOME \r\n",11);
delay(500);

/* Initialise the sensor */
if(!accel.begin())
{
/* There was a problem detecting the ADXL345 ... check your connections */
lcd.print("NO Acclerometer");
while(1);
}
/* Set the range to whatever is appropriate for your project */
accel.setRange(ADXL345_RANGE_16_G);
}
char tt;
void loop(void)
{
/* Get a new sensor event */
sensors_event_t event;
```

```
accel.getEvent(&event);
/* Display the results (acceleration is measured in m/s^2) */
lcd.clear();lcd.setCursor(0,0);
lcd.print("X:"); lcd.print(event.acceleration.x);lcd.print("Z:"); lcd.print(event.acceleration.z);
lcd.setCursor(0,1);
lcd.print("Y: "); lcd.print(event.acceleration.y);
delay(100);
digitalWrite(m1_1,LOW);
digitalWrite(m1_2,LOW);
digitalWrite(m2_1,LOW);
digitalWrite(m2_2,LOW);
delay(1000);
if(mySerial.available()> 0)
    {
tt = mySerial.read();
    }
if(tt == '*')
{
  // Forward Direction
digitalWrite(m1_1,LOW);
  digitalWrite(m1_2,HIGH);
digitalWrite(m2_1,LOW);
  digitalWrite(m2_2,HIGH);
delay(2000);
}
if(tt == '#')
{
  // Backward Direction
  digitalWrite(m1_1,HIGH);
digitalWrite(m1_2,LOW);
  digitalWrite(m2_1,HIGH);
digitalWrite(m2_2,LOW);
```

```
delay(2000);
}
if(tt =='@')
{
  // Left Direction
digitalWrite(m1_1,LOW);
digitalWrite(m1_2,LOW);
digitalWrite(m2_1,LOW);
  digitalWrite(m2_2,HIGH);
delay(2000);
}
if(tt =='&')
{
  // Right Direction
digitalWrite(m1_1,LOW);
  digitalWrite(m1_2,HIGH);
digitalWrite(m2_1,LOW);
digitalWrite(m2_2,LOW);
delay(2000);
}
if(event.acceleration.z >=49 && event.acceleration.z <50 )
{
lcd.clear();lcd.setCursor(0,0);
lcd.print("Forward");
delay(100);
digitalWrite(m1_1,LOW);
  digitalWrite(m1_2,HIGH);
digitalWrite(m2_1,LOW);
  digitalWrite(m2_2,HIGH);
delay(2000);
}
if(event.acceleration.y >=9 && event.acceleration.y <10 )
```

```
{
lcd.clear();lcd.setCursor(0,0);
lcd.print("Backward");
delay(100);
  digitalWrite(m1_1,HIGH);
digitalWrite(m1_2,LOW);
  digitalWrite(m2_1,HIGH);
digitalWrite(m2_2,LOW);
delay(2000);
}
if(event.acceleration.x >=5 && event.acceleration.x <6 )
{
lcd.clear();lcd.setCursor(0,0);
lcd.print("Left");
delay(100);
digitalWrite(m1_1,LOW);
digitalWrite(m1_2,LOW);
digitalWrite(m2_1,LOW);
  digitalWrite(m2_2,HIGH);
delay(2000);
}
if(event.acceleration.y >=1 && event.acceleration.y <2)
{
lcd.clear();lcd.setCursor(0,0);
lcd.print("Right");
delay(100);
digitalWrite(m1_1,LOW);
  digitalWrite(m1_2,HIGH);
digitalWrite(m2_1,LOW);
digitalWrite(m2_2,LOW);
delay(2000);
}
```

}

CHAPTER-5
OUTPUT SCREENSHOTS

5.1 Results:

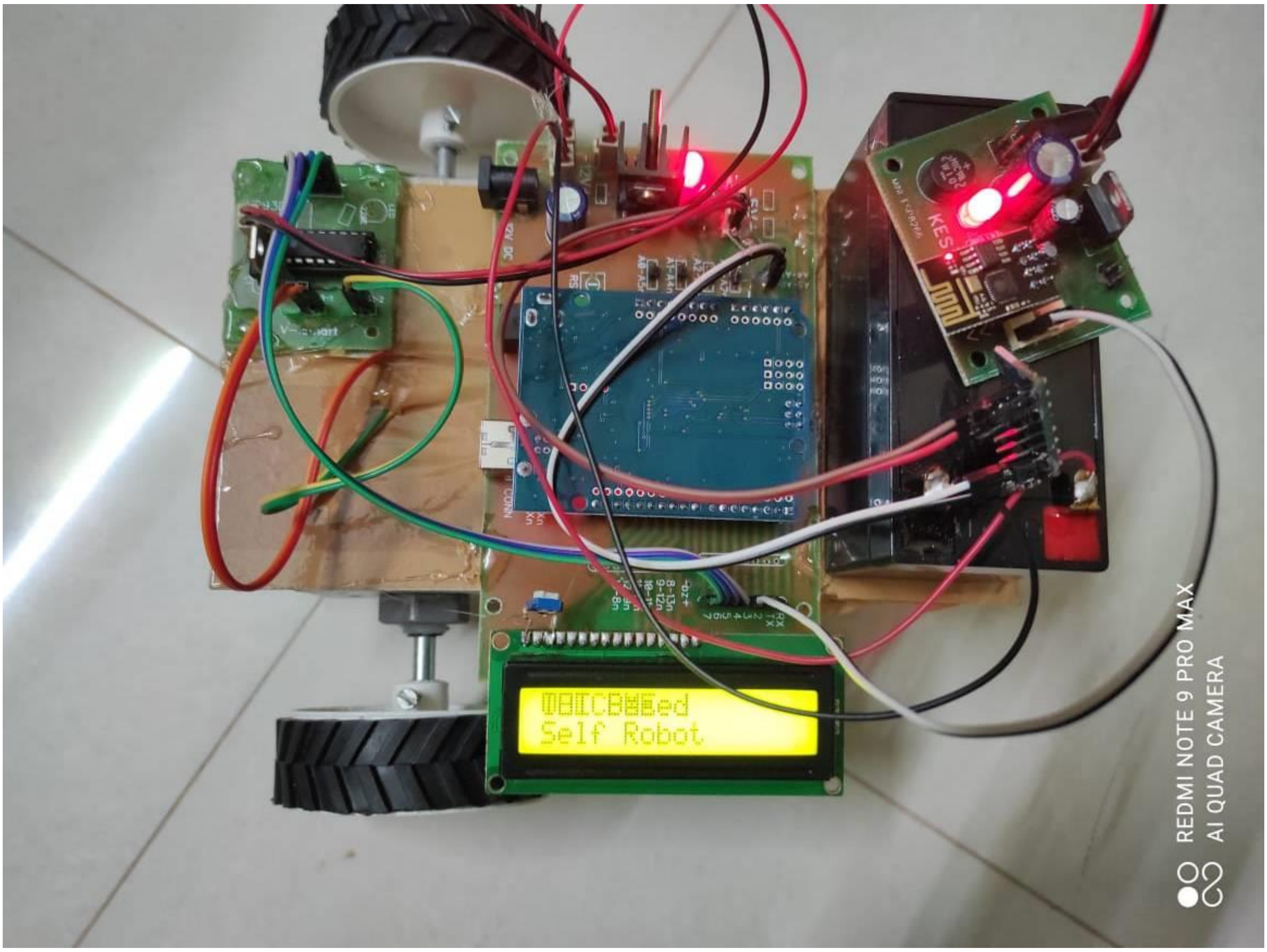


Fig 5.1:Vehicle Operating as Self Robot

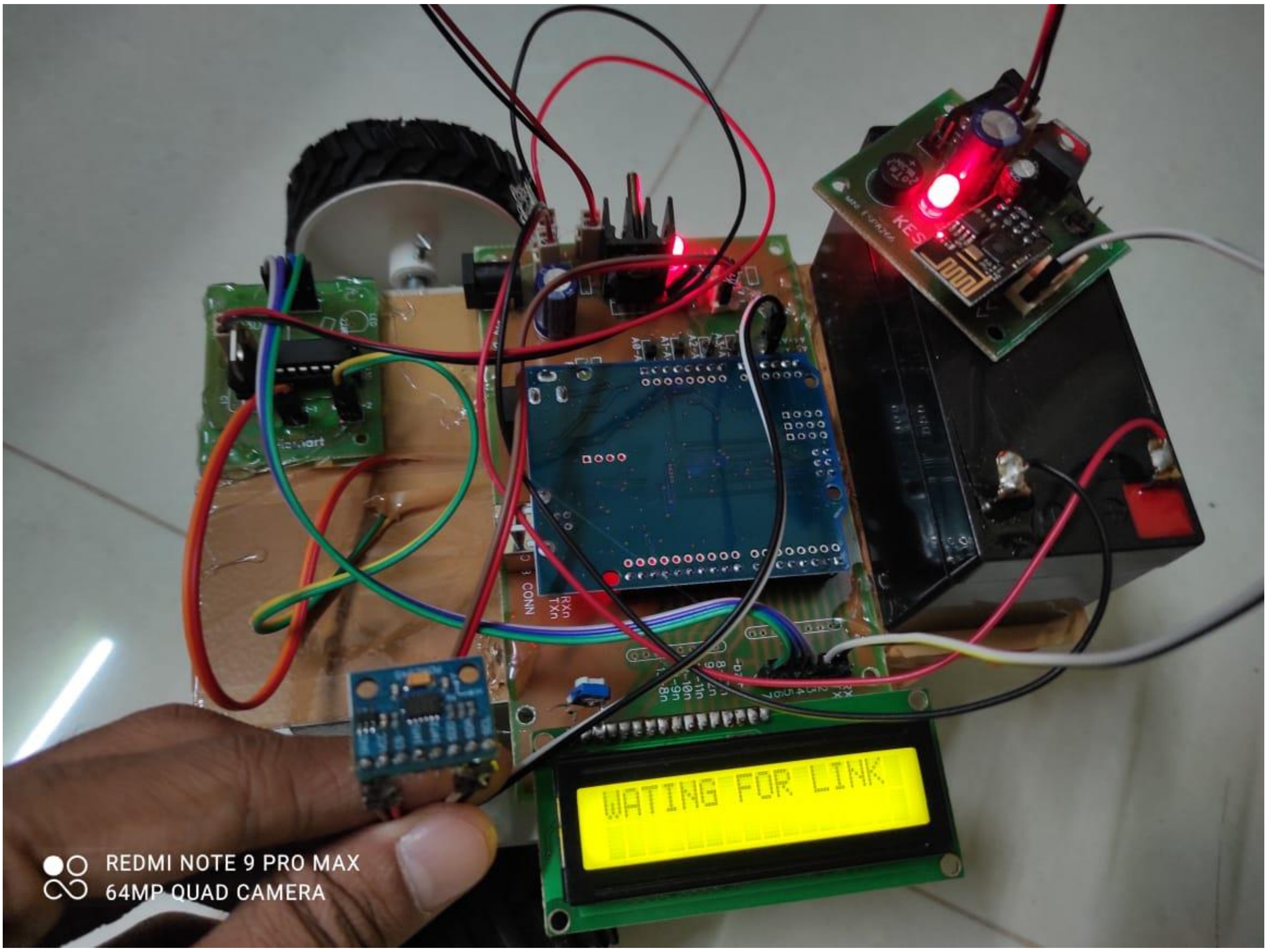


Fig:5.2:Vehicle Operating over IOT

CHAPTER-6
APPLICATIONS AND ADVANTAGES

6.1 Advantages:

There are many advantages of the gesture control technology over limitations.In this document i will clear you what are the advantages of gesture control technology.The advantages of gesture control Technology adds weight to advantages of outproject.Some of the advantages of our project are listed below

- Speed and sufficient reliable for recognition systems
- Good performance with complex background.
- The system successfully recognize static and dynamic gestures and could be applied to mobile robot.
- Simple fast and easy to implement and can be applied to real systems and play games.
- Absolutely no training is required.
- Inexpensive.

6.2 Applications:

- **Bomb Detonation**: Robots allow the close examination of devices, without endangering people. Once the device has been examined, the robot may then render the bomb inert. If not, errors can be avoided by using the robot itself to detonate the bomb and safeguard human life. It is not just bombs that the robots disable, but also devices that could detonate landmines or unexploded munitions.
- **Radioactive Environment:** When dealing with radioactive materials, robots can be used to keep humans safe from genetic mutations and fatal diseases that may occur from radiations. They can be used in labs, nuclear power plants or in areas that have been damaged by nuclear energy. There are different types of robots that are used in areas of different kinds of radiations - alpha, beta and gamma. Robots with properly automated controllers can also be used to control nuclear power plants and hence can be used to avert nuclear power plant disasters like one that occurred at Chernobyl. Robots can also be used for the disposal of radioactive waste.
- **Medical application**: Robot-mediated health care has paved the way for safe and hygienic medicine. Robots play an integral role in the battle against infectious diseases as they have the capability to disinfect a room of any bacteria and viruses within minutes. Robots, unlike humans, cannot catch infectious diseases. They are agnostic to specific infectious agents and do not contribute to medically-resistant strains. With the rise in antibiotic-resistant bacteria and outbreaks of deadly infections like Ebola, more healthcare facilities are using robots to clean and disinfect surfaces.
- **Space Rovers**: Robots are the latest intelligent systems that have helped us with outer space exploration. Robotic unmanned space crafts are used to perform multiple functions in the extra –terrestrial world; including object recognition, image evaluation and terrain modeling. They involve AI-based methods for autonomous navigation and mission planning in unknown terrain and AI-based support systems for

scientific experiments. The most famous robots used in the outer space applications are the Mars rovers of NASA.

- **Surgical Assistance and Rehabilitation Systems**: Robots today have the abilities to assist surgeons with performing complex operations, typically minimally invasive procedures to avoid infections and outbreaks. 3DHD technology provides surgeons with enhanced natural stereo visualization, along with improved reality. Robotic links also play a crucial role in the speedy recovery of patients with disabilities, including improved mobility, strength, coordination, and overall quality of life.

Some of other application are:

1. Vehicle Navigation
2. Marine Navigation
3. Fleet Management
4. Base Services
5. Auto Pilot
6. Personal Navigation
7. Travel Equipment
8. Track Equipment
9. System and Mapping Applications

6.3Future Scope:

Further developments like replace robotic vehicle by robotic arm with an IP or WiFi based camera in the receiver section. By this way perform the task, the arm can be fixed to rover that provides movement to the system wirelessly, which may be helpful for the end user to monitor or track the movement of the vehicle and performance of the robotic arm. This system provides an opportunity to increase adaptability and opens up new application areas and reduces the gap between the physical world and digital world with an output more intuitive.This kind of system will have many applications in many fields which can change the era of technology, some of them are listed below

- Bomb diffusion applications
- fireworks
- Match Industries
- Industries which include Heat Treatment.

CHAPTER-7

CONCLUSION

CHAPTER-7

7.1 Conclusion:

The objective of this paper has been achieved which was developing the hardware and software for a gesture based vehicle. Gesture based interactions allow human computer interaction to be in a natural way as well as in intuitive manner. Therefore people were able to control the robot and gripper intuitively with hand motions. This type of control could improve productivity, reduce the effects of repetitive motions and improves safety. The performed usability study points out that the system is well-suited for the pick and place operations for physically impaired people and in laboratory environments and can be used where the humans are unable to sustain in the dangerous or harsh environments. This system can also be used in defense to load or diffuse bombs under emergency or war situation and cleaning agent in sewage and drainages.

7.2 REFERENCES:

[1] Available: http://www.arduini.cc/

[2] Naveet Kumar, Neeraj Purohit, "Gesture ControlledTank Toy User Guide" URL Available

[http://www.slideshare.net/neeraj18290/wireless-gesturecontrolled-tank-toy-transmitter]Accessed 13 October 2013.

[3] Jochen Triesch and Christoph Von Der Malsburg "Robotic Gesture Recognition (1997)"URL Available

[http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.5427] Accessed 15 October 2013.

[4] "Real-Time Robotic Hand Control Using Hand Gestures" by Jagadish Lal Raheja, Radhey Shyam, G. Arun Rajsekhar and P.

Bhanu Prasad.

[5] Bhosale Prasad S., BunageYogesh B. and ShindeSwapnil V. "Hand Gesture Controlled Robot" URL Available

[6] [http://www.engineersgarage.com/contribution/accelero meter-based-hand-gesture-controlled-robot] Accessed 3 November,

2013.

[7] [http://www.robotplatform.com/howto/L293/motor_driver_1.html] Accessed 5 November, 2013.

[8] [http://en.wikipedia.org/wiki/Gesture_interface]Accessed 5 November, 2013.

[9] [http://www.wisegeek.com/what-is-a-gear-motor.htm] Accessed 6 November, 2013.

[10] [http://www.scribd.com/doc/98400320/InTech-RealTime-Robotic-Hand-Control-Using-Hand-Gestures]Accessed 6

November, 2013.

[11]Matthias rehm, Nikolausbee ,Elisabeth Andre ,Wave Like and Egyptian Accelerometer gesture.

[12] Michael Bee. “Embedded systems Glossary”.Neutrino Technical Library.Retrieved 2012-06-03.

[13] R.Cipolla and A.Pentland computer Vision for Human-machine Interaction, Cambridge University Press

A

MAJOR PROJECT REPORT
on

COMBUSTIBLE GAS LEAKAGE DETECTION WITH GSM ALERT USING ARDUINO

Submitted by

Ms. Gudise Saiteja	**(17K81A0479)**
Mr. Jaggari Manasa	**(17K81A0480)**
Mr. Meka Raja	**(17K81A0494)**
Mr. Nadimidhoddi Vikram	**(17K81A0497)**

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Mrs. Gattadi Vinatha

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)
Dhulapally, Secunderabad – 500100
NBA & NAAC A+ Accredited

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE
UGC AUTONOMOUS
Dhulapally, Secunderabad-500 100
NBA &NAAC A+ Accredited

BONAFIDE CERTIFICATE

This is to certify that the project entitled "Combustible gas detection with GSM Alert using Arduino" is being submitted by **1.Mr. Gudise Saiteja (17K81A0479), 2.Ms.Jaggari Manasa (17K81A0480) 3.Mr.Meka Raja (17K81A0494) 4.Mr.Nadimidhoddi Vikram (17K81A0497)**in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of Bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Internal Guide	**Head of the Department**
Mrs.G Vinatha Department of Electronics and Communication Engineering	Dr. Hari Krishna Department of Electronics and Communication Engineering

Internal Examiner **External Examiner**

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **G.SAI TEJA GOUD** WITH ROLL NO.**17K81A0479, J.MANASA REDDY** WITH ROLL NO.**17K81A0480**, **M.RAJA** WITH ROLL NO.**17K81A0494**, **N.VIKRAM** WITH ROLL NO.**17K81A0497**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**COMBUSTIBLE GAS LEAKAGE DETECTION WITH GSM ALERT USING ARDUINO**

" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. * Kompally-500014. *

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

DECLARATION

We, The Students of '**Bachelor of Technology in Department of Electronics and Communication Engineering**', session: 2017-2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project work entitled '**COMBUSTIBLE GAS DETECTION WITH GSM ALERT UDING ARDUINO**' is the outcome of our own Bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

Gudise Saiteja	Jaggari Manasa	Meka Raja	Vikram
17K81A0479	17K81A0480	17K81A0494	17K81A0497

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, our beloved **Principal, St. Martin's Engineering College** for his valuable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to, **Dr. Hari Krishna**, **Head &Professor** the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our dept. **Project Coordinator Dr. A. Anand** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We Sincerely thank our Project Internal Guide, **Mrs. G Vinatha, Assistant Professor,** Department of ECE and our class Project Coordinator **Mr. K Karthik, Assistant Professor,** Department of ECE for Guidance and Encouragement in carrying out this Project work. Finally, we are highly grateful and obliged to our College Level Project Coordinators **Dr.Maalyadri**, **Professor in IT** and **Dr. R. Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

1. Gudise Saiteja(17K81A0479)
2. Jaggari Manasa(17K81A0480)
3. Meka Raja(17K81A0494)
4. Nadimidhoddi Vikram(17K81A0497)

ABSRACT

Combustible gases are a very common reason for blasts and fire accidents, causing large damage to life and property. So here we propose an automated gas detection and alerting system that alert users wherever they are using SMS message. So for this purpose we use a PIC microcontroller along with Combustible gas sensor, LCD display, GSM modem for sending message and buzzer. The system is powered by a 12V power supply. The system constantly scans the gas sensor to detect leakages. As soon as gas is detected at the sensor, it produces an equivalent voltage and signals the microcontroller. The microcontroller on reading the signal checks the amount of gas detected, on detecting gas above certain level it then goes into alert mode. The system now displays the status of the event occurred on an LCD display, also sounds a buzzer to alert. It now uses the GSM modem to send an SMS message to the user/authority to inform about the situation so required action can be taken for it.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

1.1. Overview to the project:

There are numerous answers for fireplace accidents that agencies continually endorse. Smoke detectors, hearth alarms, hearth extinguishers and sprinklers are examples of those gadgets. On reflection, those devices can also alert or prevent the unfold of fire but they do not save you hearth injuries, and that alone is a main downside already. This have a look at makes a speciality of the LPG fuel and the way to save you it from causing greater injuries. There's a want to build a system that aids people's negligence of their surroundings even as stopping the begin of conflagration. The device also implements a shut-off mechanism which acts as the first line of defence inside the prevention of the coincidence ought to there be an absence of individual inside the residence. Liquefied Petroleum fuel is constituent of Butane and Propane gases, which can be distinctly inflammable in nature. The LPG is an odorless gasoline and hence the addition of Ethanethiol allows it to show case a smell throughout its leakage. An ideal gasoline sensor may be used to feel the leakage of an LPG from cars, industries, homes and different residential regions. If there is a leakage of LPG, we will effortlessly perceive by using its concentration through the gasoline sensor and by using upward push in temperature. The LPG is broadly used for home functions such as boiling, heating and cooking. some human beings can also have a low sense of scent and in such instances they'll now not be able to respond for the gasoline concentration present. Consequently, a protection primarily based LPG detection system is crucial to provide alertness, protection and protection from any harmful fuel leakage injuries. The incidents which include Kumbakonam and Bhopal fuel tragedy were the examples of the arena’s worst fuel leakage injuries. This leakage detection gadget detects the fuel leakage and additionally stops the gasoline deliver together with an alarm and a GSM alerts the required person. The fuel sensor we used right here identifies the toxic gases other than LPG and its voltage goes LOW when there's a leakage of any toxic fuel. LOW signal is despatched to a microcontroller which in flip sends those alerts to the buzzer hence, rising an alarm. After some milliseconds, the fuel leakage message is sent to the user identified mobile range thru GSM module.

INTRODUCTION ABOUT EMBEDDED SYSTEMS

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.
The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes.

The applications software on such processors is sometimes referred to as firmware.
The simplest devices consist of a single microprocessor (often called a "chip”), which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine.

As the embedded system is the combination of both software and hardware

Block diagram of embedded system:

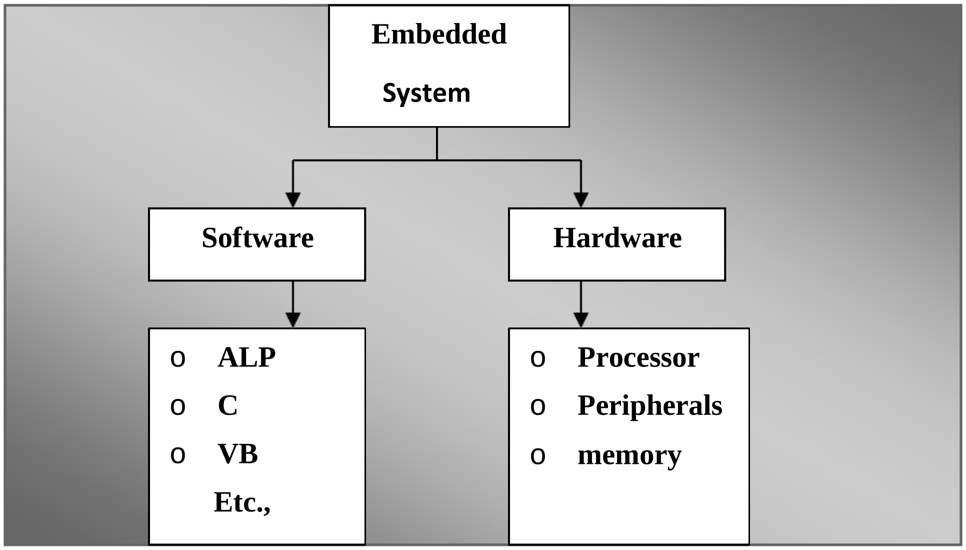


Fig.1.1: Block diagram of embedded system

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory

Applications of embedded systems:

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic system.

Micro Processor (µp):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Processors are classified into four types like:

- Micro Processor (µp)
- Micro controller (µc)
- Digital Signal Processo1r (DSP)
- Application Specific Integrated Circuits (ASIC)

Three Basic Elements of a Microprocessor:

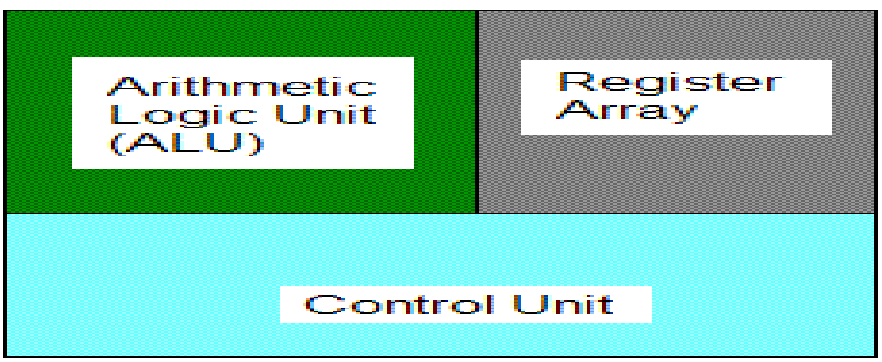


Fig.1.2: Three basic elements of a microprocessor

Harvard Architecture:

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

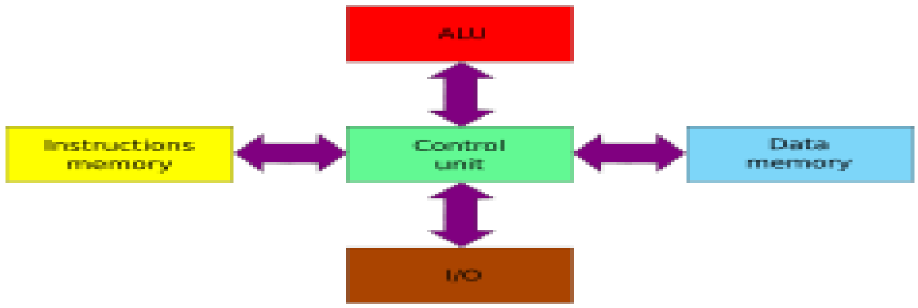


Fig.1.3: Harvard Architecture

Uses of the Harvard architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

➢ Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behaviour must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.

➢ Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

Von-Neumann Architecture

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. They cannot be performed at the same time. The von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A stored-program digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

MICRO CONTROLLER UNIT

ATMEGA328

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The ATmega8 microcontroller contains 32 general purpose working registers. As shown in the below figure these registers are directly connected to ALU. Two registers can carry one single instruction consequently in one clock cycle.

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltag	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 Ma
DC Current on 3.3V Pin	50 Ma
Flash Memory	32 KB (0.5 KB is used for Boot loader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

Specifications:

Table 1.1: Atmega328 specifications

Arduino:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

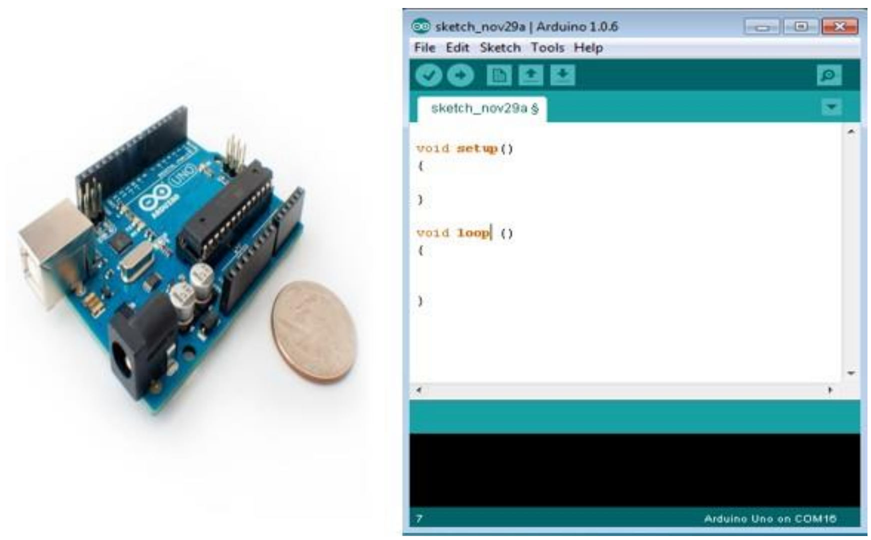


Fig 1.4 Arduino Uno hardware

Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V. Here is a list of different Arduino boards available.

Board Name	**Operating V**	**Clock Sp**	**Digital i/**	**Analog In**	**PWM**	**UART**	**Program Interface**
Arduino Uno R3	5V	16MHz	14	6	6	1	USB ATMega1
Arduino Uno R3 S	5V	16MHz	14	6	6	1	USB ATMega1
Red Board	5V	16MHz	14	6	6	1	USB via F
Arduino Pro 3.3v/8	3.3V	8MHz	14	6	6	1	FTDI-Compatibl Header
Arduino Pro 5V/16	5V	16MHz	14	6	6	1	FTDI-Compatibl Header

Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatibl Header
Arduino Pro 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatibl Header
Arduino Pro 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatibl Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatibl Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatibl Header
LilyPad Arduino main board	3.3V	8MHz	14	6	6	1	FTDI-Con
LilyPad Arduino board	3.3V	8MHz	9	4	5	0	FTDI-Compatibl Header

Table 1.2 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating V	Clock Sp	Digital i/	Analog In	PWM	UART	Programming Interface
Arduino Leona	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/1	5V	16MHz	14	6	6	1	Native USB
Pro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduin	3.3V	8MHz	14	6	6	1	Native USB

Table 1.3 Arduino boards based on ATMEGA32u4 microcontroller

Board Nam	Operating V	Clock Sp	Digital i/	Analog In	PWM	UART	Programming Interface
Arduino 2560 R3	5V	16MHz	54	16	14	4	USB ATMega16U2B
Mega Pro 3.	3.3V	8MHz	54	16	14	4	FTDI-Compatibl Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatibl Header
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatibl Header

Table 1.4 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating V	Clock Spe	Digital i/	Analog In	PWM	UART	Programming Interface
Arduino Meg R3	3.3V	84MHz	54	12	12	4	USB native

Table 1.5 Arduino boards based on AT91SAM3X8E microcontroller

Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

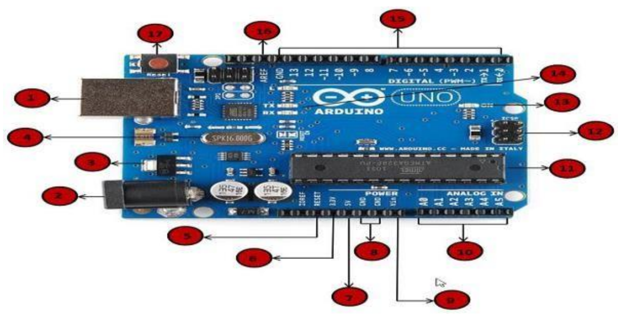


Fig:1.5 Broad description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need t connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it Barrel Jack (2).

3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and st the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate The answer is, by using the crystal oscillator. The number printed on top of the Arduino cr 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can re UNO board in two ways. First, by using the reset button (17) on the board. Second, you can c an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the from an analog sensor like the humidity sensor or temperature sensor and convert it into a value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your The main IC (integrated circuit) on the Arduino is slightly different from board to boar

	microcontrollers are usually of the ATMEL Company. You must know what IC your board has loading up a new program from the Arduino IDE. This information is available on the top of For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), could be considered as an "expansion" of the output. Actually, you are slaving the output de the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that you is powered up correctly. If this light does not turn on, then there is something wrong w connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible fo communication. Second, the TX and RX led (13). The TX led flashes with different speed sending the serial data. The speed of flashing depends on the baud rate used by the board. RX during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Modulation) output. These pins can be configured to work as input digital pins to read logic va or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins label can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference v (between 0 and 5 Volts) as the upper limit for the analog input pins.

ARDUINO FAMILY

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project , check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

Fig.1.6: Arduino Family

SHIELDS

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

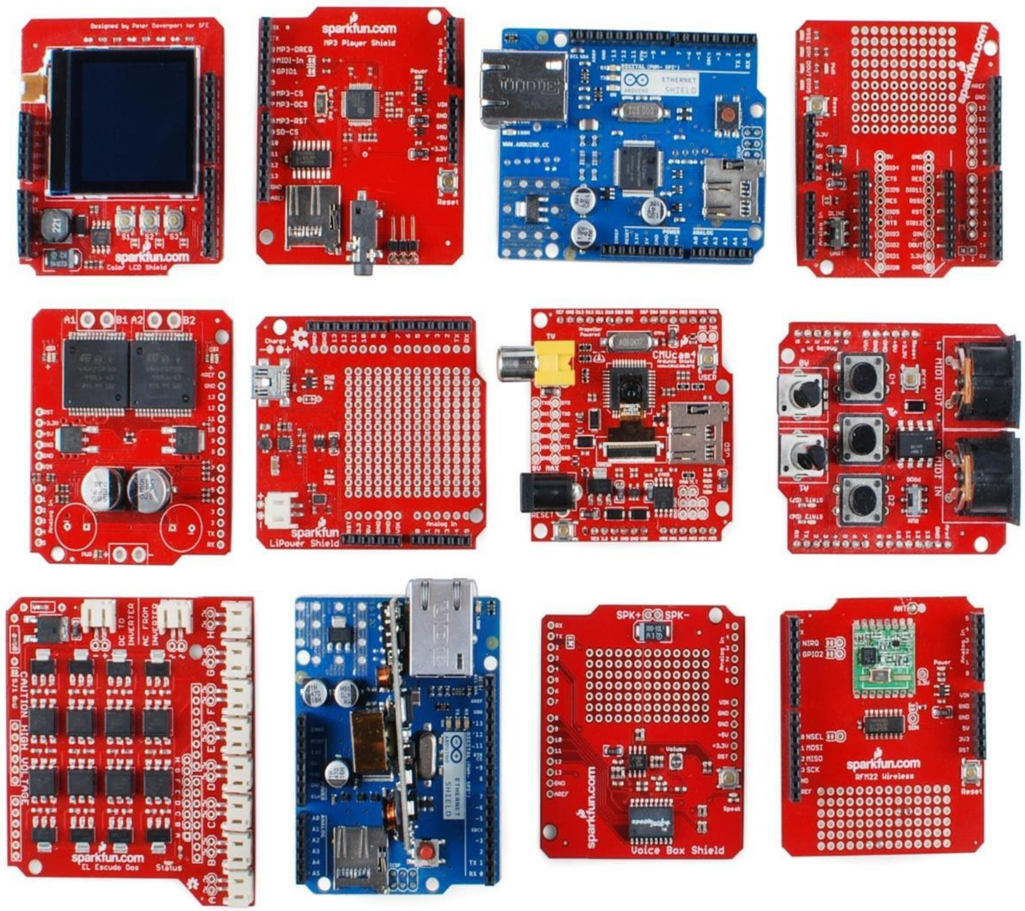


Fig.1.7: Arduino Shields

PIN DESCRIPTION OF ATMEGA328

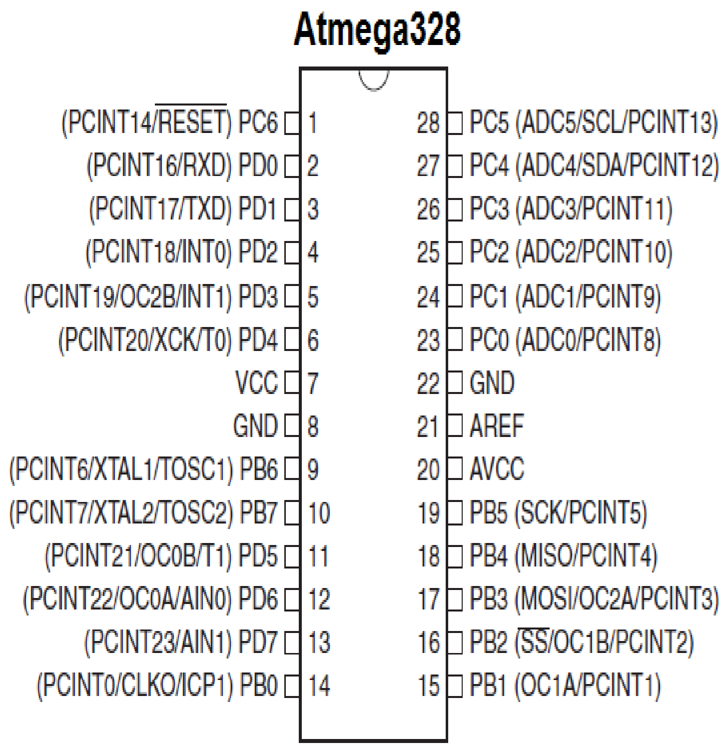


Fig.1.8: Pin description of ATMEGA328

ADVANTAGES OF ARDUINO

- It is cheap
- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
- For beginners, it is very simple to use.

CHAPTER 2
LITERATURE SURVEY

We purpose a system to come across LPG fuel leakage situation and offer a security alert to supposed customers. We right here endorse to build the gadget the use of a MQ5 gas detection sensor. If the LPG sensor senses any gas leakage from storage, gas sensor is going stumble on it this sign is monitored by using the microcontroller and it will perceive the gasoline leakage. Now the micro controller is turn on LED and Buzzer.

After few milli seconds put off microcontroller instructions driving force circuitry for exhaust fan to turn the fan on to release the gasoline out door from the room and concurrently microcontroller commands every other circuitry a relay circuitry to shut the fuel knob. In our proposed machine we designed two motive forces circuitry one to power motor to close knob. And any other is relay circuitry in which replay is used to switched on and rancid for exhaust fan. Microcontroller is programmed by using Embedded C Language. It's for the complete controller of the task. It controls the exhaust fan, LED, Buzzer and when the LPG leak takes place. The entire output ports of the microcontroller is used for this.

Literature review on research area

In 1982, the European Conference of Postal and Telecommunications Administrations (CEPT) created the Group Special Mobile (GSM) to develop a standard for a mobile telephone system that could be used across Europe. In 1987, a memorandum of understanding was signed by 13 countries to develop a common cellular telephone system across Europe. Finally the system created by SINTEF lead by Torleiv Maseng was selected.

In 1989, GSM responsibility was transferred to the European Telecommunications Standards Institute (ETSI) and phase I of the GSM specifications were published in 1990. The first GSM network was launched in 1991 by Radiolinja in Finland with joint technical infrastructure maintenance from Ericsson.

By the end of 1993, over a million subscribers were using GSM phone networks being operated by 70 carriers across 48 countries. As of the end of 1997, GSM service was available in more than 100 countries and has become the *de facto* standard in Europe and Asia.

Conclusion on the Review

- A number of reviews on the subject of gas leakage detection techniques were done in the past either as part of research papers/technical reports on a certain leak detection method and other gas related subjects.

- Ch. Manohar Raju and N. Sushma Rani, 2008; they introduce an android based automatic gas detection and indication robot. They proposed prototype depicts a mini mobile robot which is capable to detect gas leakage in hazardous places.

- Whenever there is an occurrence of gas leakage in a particular place the robot immediately read and sends the data to android mobile through wireless communication like Bluetooth.

- We developed an android application for android based smart phones which can receive data from robot directly through Bluetooth.

- The application warns with an indication whenever there is an occurrence of gas leakage and we can also control the robot movements via Bluetooth.

- Pal-Stefan Murvaya, Ioan Sileaa, 2008; they told in their survey on gas leak detection and localization techniques various ways to detect the gas leakage. They introduce some old or new technique to detect the gas. The proposed techniques in this paper are non-technical methods, hardware based methods which includes acoustic methods, optical methods and active methods.

CHAPTER 3

PROJECT DESIGN

Block Diagram

There are numerous answers for fireplace accidents that agencies continually endorse. Smoke detectors, hearth alarms, hearth extinguishers and sprinklers are examples of those gadgets. On reflection, those devices can also alert or prevent the unfold of fire but they do not save you hearth injuries, and that alone is a main downside already. This have a look at makes a speciality of the LPG fuel and the way to save you it from causing greater injuries. There's a want to build a system that aids people's negligence of their surroundings even as stopping the begin of conflagration. The device also implements a shut-off mechanism which acts as the first line of defence inside the prevention of the coincidence ought to there be an absence of individual inside the residence. Liquefied Petroleum fuel is constituent of Butane and Propane gases, which can be distinctly inflammable in nature.

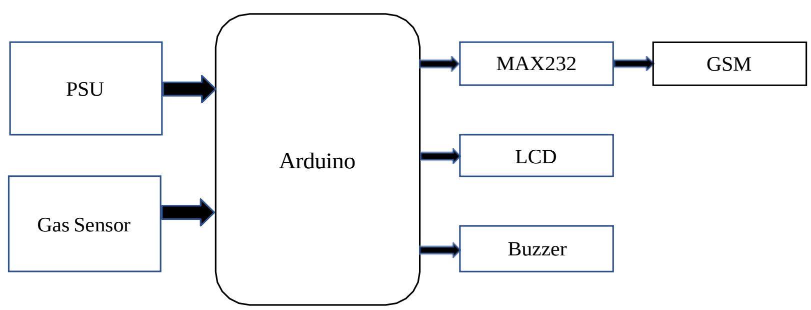


Fig.3.1 block diagram

The blocks of the circuit are:

1.Arduino Uno

2.MQ 2 gas sensor

3.Buzzer

4.LCD display

5.GSM Module

6.Power Supply Unit

1.Arduino UNO

Arduino is opensource software. Arduino Uno is a microcontroller, ATmega328Pcontain in it. It has 14 digital i/o pins and 6 analog input pin, a 16 MHz quartz crystal, a USB connection, a power jack and a reset button. Simply connect to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. Embedded C program is used for coding.

2.MQ2 gas sensor

Gas sensors main aim is to sense hazardous gases that evolve its surroundings. It detects the concentrations of combustible gas in the air and outputs its reading as an analog voltage. The sensor can measure concentrations of flammable gas of 300 to 10,000 ppm. The sensor can operate at temperatures from -20 to 50°C and consumes less than 150 mA at 5 V. This sensor is connected to LCD and when there is a leakage of gas this sensor senses and a message is displayed on the LCD indicating the leakage of gas. This sensor is also connected to a buzzer and the buzzer sounds whenever the gas leakage is detected.

3. Buzzer

Typical uses of buzzers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke.

4. LCD

The LCD is used to display when ever there is a gas leakage or fire detected in the area. When ever there is a gas leakage the Arduino that is connected to the sensor and LCD sends a message to LCD to display 'Gas leakage'. When there is fire outbreak the Arduino again sends a message to LCD to display 'Fire Detected' notifying us if there is a gas leakage or fire detected in the vicinity.

5. GSM

GSM (Global System for Mobile communications) is an open, digital cellular technology used for transmitting mobile voice and data services.

GSM (Global System for Mobile communication) is a digital mobile telephone system that is widely used in Europe and other parts of the world. GSM uses a variation of Time Division Multiple Access (TDMA) and is the most widely used of the three digital wireless telephone technologies (TDMA, GSM, and CDMA). GSM digitizes and compresses data, then sends it down a channel with two other streams of user data, each in its own time slot. It operates at either the 900 MHz or 1,800 MHz frequency band. It supports voice calls and data transfer speeds of up to 9.6 kbit/s, together with the transmission of SMS (Short Message Service).

6. Power Supply Unit

Every electrical and electronic device that we use in our day-to-day life will require a power supply. In general, we use an AC supply of 230V 50Hz, but this power has to be changed into the required form with required values or voltage range for providing power supply to different types of devices. There are various types of power electronic converters such as step-down converter, step-up converter, voltage 31 stabilizer, AC to DC converter, DC 25 to DC converter, DC to AC converter, and so on. For example, consider the microcontrollers that are used frequently for developing many embedded systems based projects and kits used in real-time applications. These microcontrollers require a 5V DC supply.

Schematic Diagram

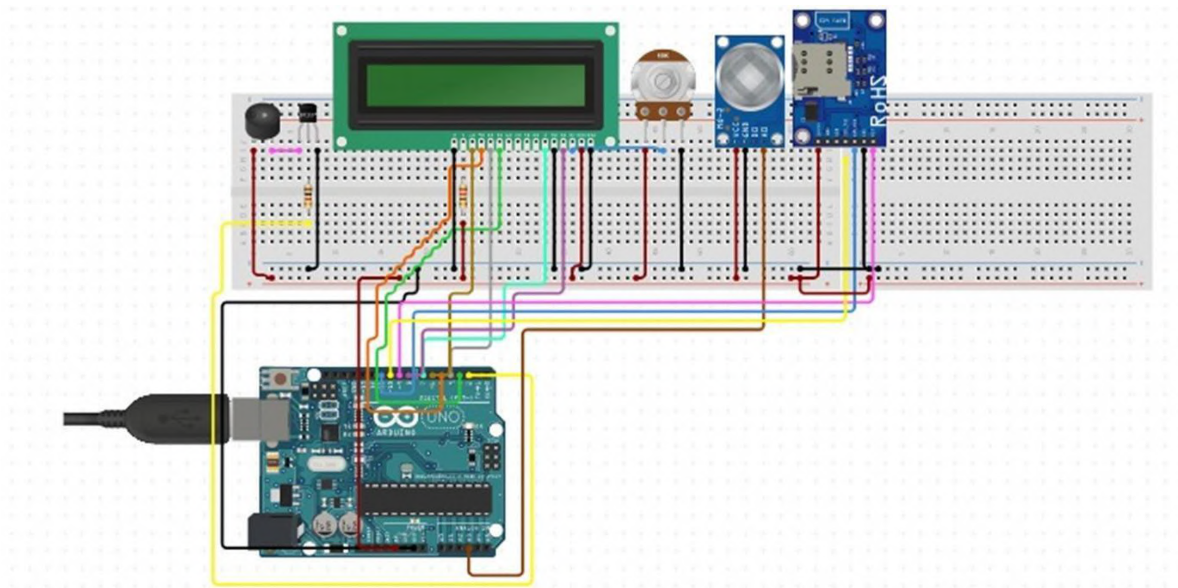


FIG:3.2 Schematic Diagram

Explanation of Schematic Diagram

The circuit mainly uses the gas sensor and Arduino to detect and smoke and gas leak. This low signal is monitored by the microcontroller and sends the signal to GSM Module to send messages as “Gas leakage” to a mobile number written in code. We can also use SIM900 or any other GSM Module instead of SIM800.

CHAPTER 4

PROJECT IMPLEMENTATION

1 Hardware Description

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

Display Data RAM

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.

00	01	02	03	04	05	06	07		32	33	34	35	36	37	38	39	← Character position (dec.)
00	01	02	03	04	05	06	07	• • • • •	20	21	22	23	24	25	26	27	← Row0 DDRAM address (hex)
40	41	42	43	44	45	46	47	• • • • •	60	61	62	63	64	65	66	67	← Row1 DDRAM address (hex)

DDRAM Address for 2 Line LCD

Character Generator ROM

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM.

Busy Flag

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i. e BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

Instruction Register (IR) and Data Register (DR)

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD.

Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

16 x 2 Alphanumeric LCD Module Features

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing

- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

GSM :

Definition of GSM:

GSM (Global System for Mobile communications) is an open, digital cellular technology used for transmitting mobile voice and data services.

GSM (Global System for Mobile communication) is a digital mobile telephone system that is widely used in Europe and other parts of the world. GSM uses a variation of Time Division Multiple Access (TDMA) and is the most widely used of the three digital wireless telephone technologies (TDMA, GSM, and CDMA). GSM digitizes and compresses data, then sends it down a channel with two other streams of user data, each in its own time slot. It operates at either the 900 MHz or 1,800 MHz frequency band. It supports voice calls and data transfer speeds of up to 9.6 kbit/s, together with the transmission of SMS (Short Message Service).

GSM Frequencies

GSM networks operate in a number of different frequency ranges (separated into GSM frequency ranges for 2G and UMTS frequency bands for 3G). Most 2G GSM networks operate in the 900 MHz or 1800 MHz bands. Some countries in the Americas (including Canada and the United States) use the 850 MHz and 1900 MHz bands because the 900 and 1800 MHz frequency bands were already allocated. Most 3G GSM networks in Europe operate in the 2100 MHz frequency band. The rarer 400 and 450 MHz frequency bands are assigned in some countries where these frequencies were previously used for first-generation systems.

GSM-900 uses 890–915 MHz to send information from the mobile station to the base station (uplink) and 935–960 MHz for the other direction (downlink), providing 124 RF channels (channel numbers 1 to 124) spaced at 200 kHz. Duplex spacing of 45 MHz is used. In some countries the GSM-900 band has been extended to cover a larger frequency range. This 'extended GSM', E-GSM, uses 880–915 MHz (uplink) and 925–960 MHz (downlink), adding 50 channels (channel numbers 975 to 1023 and 0) to the original GSM-900 band.

Time division multiplexing is used to allow eight full-rate or sixteen half-rate speech channels per radio frequency channel. There are eight radio timeslots (giving eight burst periods) grouped into what is called a TDMA frame. Half rate channels use alternate frames in the same timeslot. The channel data rate for all 8 channels is 270.833 Kbit/s, and the frame duration is 4.615 ms.

The transmission power in the handset is limited to a maximum of 2 watts in GSM850/900 and 1 watt in GSM1800/1900. GSM operates in the 900MHz and 1.8GHz bands in Europe and the 1.9GHz and 850MHz bands in the US. The 850MHz band is also used for GSM and 3G in Australia, Canada and many South American countries. By having harmonized spectrum across most of the globe, GSM's international roaming capability allows users to access the same services when travelling abroad as at home. This gives consumers seamless and same number connectivity in more than 218 countries.

Terrestrial GSM networks now cover more than 80% of the world's population. GSM satellite roaming has also extended service access to areas where terrestrial coverage is not available.

Mobile Telephony Standards

Standard	Generation	Frequency band	Throughput	
GSM	2G	Allows transfer of voice or low-volume digital data.	9.6 kbps	9.6 kbps
GPRS	2.5G	Allows transfer of voice or moderate-volume digital data.	21.4-171.2 kbps	48 kbps
EDGE	2.75G	Allows simultaneous transfer of voice and digital data.	43.2-345.6 kbps	171 kbps
UMTS	3G	Allows simultaneous transfer of voice and high-speed digital data.	0.144-2 Mbps	384 kbps

1G: The first generation of mobile telephony (written **1G**) operated using analogue communications and portable devices that were relatively large. It used primarily the following standards:

- **AMPS** (Advanced Mobile Phone System), which appeared in 1976 in the United States, was the first cellular network standard. It was used primarily in the Americas, Russia and Asia. This first-generation analogue network had weak security mechanisms which allowed hacking of telephones lines.
- **TACS** (Total Access Communication System) is the European version of the AMPS model. Using the 900 MHz frequency band, this system was largely used in England and then in Asia (Hong-Kong and Japan).
- **ETACS** (Extended Total Access Communication System) is an improved version of the TACS standard developed in the United Kingdom that uses a larger number of communication channels.

The first-generation cellular networks were made obsolete by the appearance of an entirely digital second generation.

Second Generation of Mobile Networks (2G)

The second generation of mobile networks marked a break with the first generation of cellular telephones by switching from analogue to digital. The main 2G mobile telephony standards are:

- GSM (*Global System for Mobile communications*) is the most commonly used standard in Europe at the end of the 20^{th} century and supported in the United States. This standard uses the 900 MHz and 1800 MHz frequency bands in Europe. In the United States, however, the frequency band used is the 1900 MHz band. Portable telephones that are able to operate in Europe and the United States are therefore called **tri-band**.
- **CDMA** (*Code Division Multiple Access*) uses a spread spectrum technique that allows a radio signal to be broadcast over a large frequency range.
- **TDMA** (*Time Division Multiple Access*) uses a technique of time division of communication channels to increase the volume of data transmitted simultaneously. TDMA

technology is primarily used on the American continent, in New Zealand and in the Asia-Pacific region.

With the 2G networks, it is possible to transmit voice and low volume digital data, for example text messages (**SMS**, for *Short Message Service*) or multimedia messages (**MMS**, for *Multimedia Message Service*). The GSM standard allows a maximum data rate of 9.6 kbps.

Extensions have been made to the GSM standard to improve throughput. One of these is the **GPRS** (*General Packet Radio System*) service which allows theoretical data rates on the order of 114 Kbit/s but with throughput closer to 40 Kbit/s in practice. As this technology does not fit within the "3G" category, it is often referred to as **2.5G**

The **EDGE** (*Enhanced Data Rates for Global Evolution*) standard, billed as **2.75G**, quadruples the throughput improvements of GPRS with its theoretical data rate of 384 Kbps, thereby allowing the access for multimedia applications. In reality, the EDGE standard allows maximum theoretical data rates of 473 Kbit/s, but it has been limited in order to comply with the IMT-2000 (*International Mobile Telecommunications-2000*) specifications from the ITU (*International Telecommunications Union*).

3G

The IMT-2000 (*International Mobile Telecommunications for the year 2000*) specifications from the International Telecommunications Union (ITU) defined the characteristics of **3G** (third generation of mobile telephony). The most important of these characteristics are:

1. High transmission data rate.
2. 144 Kbps with total coverage for mobile use.
3. 384 Kbps with medium coverage for pedestrian use.
4. 2 Mbps with reduced coverage area for stationary use.
5. World compatibility.
6. Compatibility of 3rd generation mobile services with second generation networks.

3G offers data rates of more than 144 Kbit/s, thereby allowing the access to multimedia uses such as video transmission, video-conferencing or high-speed internet access. 3G networks use different frequency bands than the previous networks: 1885-2025 MHz and 2110-2200 MHz.

The main 3G standard used in Europe is called **UMTS** (*Universal Mobile Telecommunications System*) and uses **WCDMA** (*Wideband Code Division Multiple Access*) encoding. UMTS technology uses 5 MHz bands for transferring voice and data, with data rates that can range from 384 Kbps to 2 Mbps. **HSDPA** (*High Speed Downlink Packet Access*) is a third generation mobile telephony protocol, (considered as "3.5G"), which is able to reach data rates on the order of 8 to 10 Mbps. HSDPA technology uses the 5 GHz frequency band and uses WCDMA encoding.

Introduction to the GSM Standard

The **GSM** (*Global System for Mobile communications*) network is at the start of the 21st century, the most commonly used mobile telephony standard in Europe. It is called as Second Generation (2G) standard because communications occur in an entirely digital mode, unlike the first generation of portable telephones. When it was first standardized in 1982, it was called as **Group Special Mobile** and later, it became an international standard called **"Global System for Mobile communications"** in 1991.

In Europe, the GSM standard uses the 900 MHz and 1800 MHz frequency bands. In the United States, however, the frequency band used is the 1900 MHz band. For this reason, portable telephones that are able to operate in both Europe and the United States are called **tri-band** while those that operate only in Europe are called **bi-band**.

The GSM standard allows a maximum throughput of 9.6 kbps which allows transmission of voice and low-volume digital data like text messages (**SMS**, for *Short Message Service*) or multimedia messages (**MMS**, for *Multimedia Message Service*).

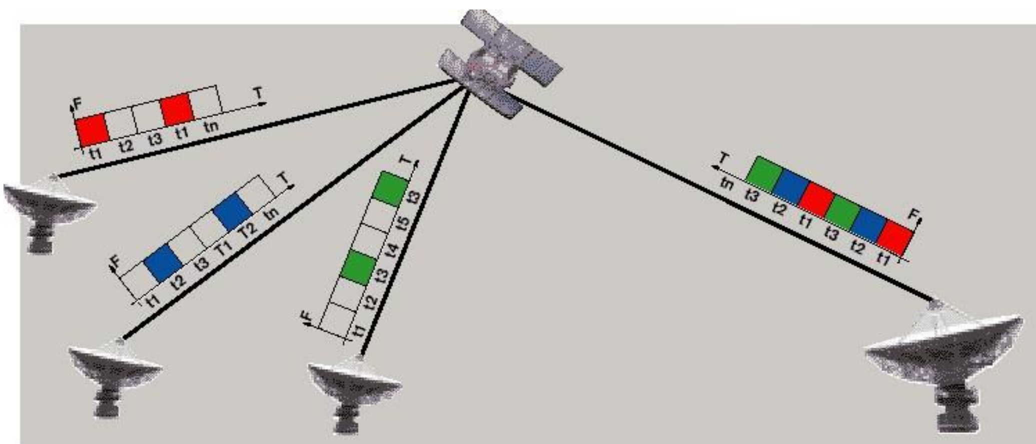
GSM Standards:

GSM uses narrowband TDMA, which allows eight simultaneous calls on the same radio frequency. There are three basic principles in multiple access, FDMA (Frequency Division Multiple Access), TDMA (Time Division Multiple Access), and CDMA (Code Division Multiple Access). All three principles allow multiple users to share the same physical channel. But the two competing technologies differ in the way user sharing the common resource.

TDMA allows the users to share the same frequency channel by dividing the signal into different time slots. Each user takes turn in a round robin fashion for transmitting and receiving over the channel. Here, users can only transmit in their respective time slot.

TDMA in brief:

In late1980's, as a search to convert the existing analog network to digital as a means to improve capacity, the cellular telecommunications industry association chose TDMA over FDMA. Time Division Multiplex Access is a type of multiplexing where two or more channels of information are transmitted over the same link by allocating a different time interval for the transmission of each channel. The most complex implementation using TDMA principle is of GSM's (Global System for Mobile communication). To reduce the effect of co-channel interference, fading and multipath, the GSM technology can use frequency hopping, where a call jumps from one channel to another channel in a short interval.



Time Division Multiple Access

Fig:4.1.1 Time Division Multiple Access

TDMA systems still rely on switch to determine when to perform a handoff. Handoff occurs when a call is switched from one cell site to another while travelling. The TDMA handset constantly monitors the signals coming from other sites and reports it to the switch without caller's awareness. The switch then uses this information for making better choices for handoff at appropriate times. TDMA handset performs hard handoff, i.e., whenever the user moves from one site to another, it breaks the connection and then provides a new connection with the new site.

Advantages of TDMA:

There are lots of advantages of TDMA in cellular technologies.

1. It can easily adapt to transmission of data as well as voice communication.
2. It has an ability to carry 64 kbps to 120 Mbps of data rates. This allows the operator to do services like fax, voice band data and SMS as well as bandwidth intensive application such as multimedia and video conferencing.
3. Since TDMA technology separates users according to time, it ensures that there will be no interference from simultaneous transmissions.
4. It provides users with an extended battery life, since it transmits only portion of the time during conversations. Since the cell size grows smaller, it proves to save base station equipment, space and maintenance.

TDMA is the most cost effective technology to convert an analog system to digital.

Disadvantages of TDMA:

One major disadvantage using TDMA technology is that the users has a predefined time slot. When moving from one cell site to other, if all the time slots in this cell are full the user might be disconnected. Likewise, if all the time slots in the cell in which the user is currently in are already occupied, the user will not receive a dial tone.

The second problem in TDMA is that it is subjected to multipath distortion. To overcome this distortion, a time limit can be used on the system. Once the time limit is expired, the signal is ignored.

The concept of cellular network

Mobile telephone networks are based on the concept of **cells**, circular zones that overlap to cover a geographical area. Cellular networks are based on the use of a central transmitter-receiver in each cell, called a "**base station**" (or *Base Transceiver Station*, written **BTS**). The smaller the radius of a cell, the higher is the available bandwidth.

So, in highly populated urban areas, there are cells with a radius of a few hundred meters, while huge cells of up to 30 kilometers provide coverage in rural areas.

In a cellular network, each cell is surrounded by 6 neighbouring cells (thus a cell is generally drawn as a hexagon). To avoid interference, adjacent cells cannot use the same frequency. In practice, two cells using the same frequency range must be separated by a distance of two to three times the diameter of the cell.

Architecture of the GSM Network

In a GSM network, the user terminal is called a **mobile station**. A mobile station is made up of a **SIM** (*Subscriber Identity Module*) card allowing the user to be uniquely identified and a mobile terminal. The terminals (devices) are identified by a unique 15-digit identification number called **IMEI** (*International Mobile Equipment Identity*). Each SIM card also has a unique (and secret) identification number called **IMSI** (*International Mobile Subscriber Identity*). This code can be protected using a 4-digit key called a *PIN code*.

The SIM card therefore allows each user to be identified independently of the terminal used during communication with a base station. Communications occur through a radio link (air interface) between a mobile station and a base station.

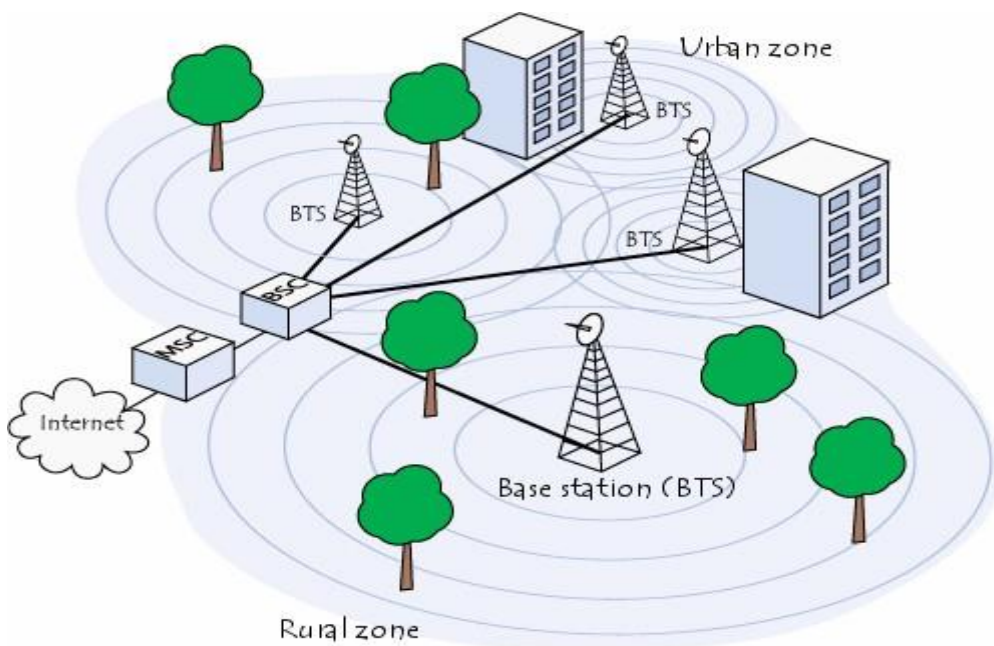


Fig:4.1.2 GSM Network

All the base stations of a cellular network are connected to a **base station controller** (**BSC**) which is responsible for managing distribution of the resources. The system consisting of the base station controller and its connected base stations is called the **Base Station Subsystem** (**BSS**).

Finally, the base station controllers are themselves physically connected to the **Mobile Switching Centre** (**MSC**), managed by the telephone network operator, which connects them to the public telephone network and the Internet. The MSC belongs to a **Network Station Subsystem** (**NSS**), which is responsible for managing user identities, their location and establishment of communications with other subscribers. The MSC is generally connected to databases that provide additional functions:

1. The **Home Location Register** (**HLR**) is a database containing information (geographic position, administrative information etc.) of the subscribers registered in the area of the switch (MSC).
2. The **Visitor Location Register** (**VLR**) is a database containing information of users other than the local subscribers. The VLR retrieves the data of a new user from the HLR of the user's subscriber zone. The data is maintained as long as the user is in the zone and is deleted when the user leaves or after a long period of inactivity (terminal off).
3. The **Equipment Identify Register** (**EIR**) is a database listing the mobile terminals.
4. The **Authentication Centre** (**AUC**) is responsible for verifying user identities.
5. The cellular network formed in this way is designed to support mobility via management of *handovers* (movements from one cell to another).

Finally, GSM networks support the concept of **roaming** i.e., movement from one operator network to another.

Introduction to Modem:

Modem stands for *modulator-demodulator.*

A modem is a device or program that enables a computer to transmit data over telephone or cable lines. Computer information is stored digitally, whereas information transmitted over telephone lines is transmitted in the form of analog waves. A modem converts between these two forms.

Fortunately, there is one standard interface for connecting external modems to computers called *RS-232*. Consequently, any external modem can be attached to any computer that has an RS-232 port, which almost all personal computers have. There are also modems that come as an expansion board that can be inserted into a vacant expansion slot. These are sometimes called *onboard* or *internal modems*.

While the modem interfaces are standardized, a number of different protocols for formatting data to be transmitted over telephone lines exist. Some, like CCITT V.34 are official standards, while others have been developed by private companies. Most modems have built-in support for the more common protocols at slow data transmission speeds at least, most modems can communicate with each other. At high transmission speeds, however, the protocols are less standardized.

Apart from the transmission protocols that they support, the following characteristics distinguish one modem from another:

- **Bps:** How fast the modem can transmit and receive data. At slow rates, modems are measured in terms of baud rates. The slowest rate is 300 baud (about 25 cps). At higher speeds, modems are measured in terms of bits per second (bps). The fastest modems run at 57,600 bps, although they can achieve even higher data transfer rates by compressing the data. Obviously, the faster the transmission rate, the faster the data can be sent and received. It should be noted that the data cannot be received at a faster rate than it is being sent.
- **Voice/data:** Many modems support a switch to change between voice and data modes. In data mode, the modem acts like a regular modem. In voice mode, the modem acts like a regular telephone. Modems that support a voice/data switch have a built-in loudspeaker and microphone for voice communication.
- **Auto-answer:** An auto-answer modem enables the computer to receive calls in the absence of the operator.
- **Data compression:** Some modems perform data compression, which enables them to send data at faster rates. However, the modem at the receiving end must be able to decompress the data using the same compression technique.
- **Flash memory:** Some modems come with *flash memory* rather than conventional ROM which means that the communications protocols can be easily updated if necessary.

- **Fax capability:** Most modern modems are fax modems, which mean that they can send and receive faxes.

GSM Modem:

A GSM modem is a wireless modem that works with a GSM wireless network. A wireless modem behaves like a dial-up modem. The main difference between them is that a dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves.

Fig:4.1.3 GSM Modem

A GSM modem can be an external device or a PC Card / PCMCIA Card. Typically, an external GSM modem is connected to a computer through a serial cable or a USB cable. A GSM modem in the form of a PC Card / PCMCIA Card is designed for use with a laptop computer. It should be inserted into one of the PC Card / PCMCIA Card slots of a laptop computer. Like a GSM mobile phone, a GSM modem requires a SIM card from a wireless carrier in order to operate.
A SIM card contains the following information:

- Subscriber telephone number (MSISDN)
- International subscriber number (**IMSI, International Mobile Subscriber Identity)**
- State of the SIM card

- Service code (operator)
- Authentication key
- PIN (***Personal Identification Code***)
- PUK (***Personal Unlock Code***)

Computers use AT commands to control modems. Both GSM modems and dial-up modems support a common set of standard AT commands. In addition to the standard AT commands, GSM modems support an extended set of AT commands. These extended AT commands are defined in the GSM standards. With the extended AT commands, the following operations can be performed:

- Reading, writing and deleting SMS messages.
- Sending SMS messages.
- Monitoring the signal strength.
- Monitoring the charging status and charge level of the battery.
- Reading, writing and searching phone book entries.

The number of SMS messages that can be processed by a GSM modem per minute is very low i.e., about 6 to 10 SMS messages per minute.

Introduction to AT Commands

AT commands are instructions used to control a modem. AT is the abbreviation of ATtention. Every command line starts with "AT" or "at". That's the reason, modem commands are called AT commands. Many of the commands that are used to control wired dial-up modems, such as ATD (Dial), ATA (Answer), ATH (Hook control) and ATO (Return to online data state) are also supported by GSM modems and mobile phones.

Besides this common AT command set, GSM modems and mobile phones support an AT command set that is specific to the GSM technology, which includes SMS-related commands like AT+CMGS (Send SMS message), AT+CMSS (Send SMS message from storage), AT+CMGL (List SMS messages) and AT+CMGR (Read SMS messages).

It should be noted that the starting "AT" is the prefix that informs the modem about the start of a command line. It is not part of the AT command name. For example, D is the actual AT command name in ATD and +CMGS is the actual AT command name in AT+CMGS.

Some of the tasks that can be done using AT commands with a GSM modem or mobile phone are listed below:

- Get basic information about the mobile phone or GSM modem. For example, name of manufacturer (AT+CGMI), model number (AT+CGMM), IMEI number (International Mobile Equipment Identity) (AT+CGSN) and software version (AT+CGMR).
- Get basic information about the subscriber. For example, MSISDN (AT+CNUM) and IMSI number (International Mobile Subscriber Identity) (AT+CIMI).
- Get the current status of the mobile phone or GSM/GPRS modem. For example, mobile phone activity status (AT+CPAS), mobile network registration status (AT+CREG), radio signal strength (AT+CSQ), battery charge level and battery charging status (AT+CBC).
- Establish a data connection or voice connection to a remote modem (ATD, ATA, etc).
- Send and receive fax (ATD, ATA, AT+F*).
- Send (AT+CMGS, AT+CMSS), read (AT+CMGR, AT+CMGL), write (AT+CMGW) or delete (AT+CMGD) SMS messages and obtain notifications of newly received SMS messages (AT+CNMI).
- Read (AT+CPBR), write (AT+CPBW) or search (AT+CPBF) phonebook entries.
- Perform security-related tasks, such as opening or closing facility locks (AT+CLCK), checking whether a facility is locked (AT+CLCK) and changing passwords(AT+CPWD). (Facility lock examples: SIM lock [a password must be given to the SIM card every time the mobile phone is switched on] and PH-SIM lock [a certain SIM card is associated with the mobile phone. To use other SIM cards with the mobile phone, a password must be entered.])
- Control the presentation of result codes / error messages of AT commands. For example, the user can control whether to enable certain error messages (AT+CMEE) and whether error messages should be displayed in numeric format or verbose format (AT+CMEE=1 or AT+CMEE=2).

- Get or change the configurations of the mobile phone or GSM/GPRS modem. For example, change the GSM network (AT+COPS), bearer service type (AT+CBST), radio link protocol parameters (AT+CRLP), SMS center address (AT+CSCA) and storage of SMS messages (AT+CPMS).
- Save and restore configurations of the mobile phone or GSM/GPRS modem. For example, save (AT+CSAS) and restore (AT+CRES) settings related to SMS messaging such as the SMS center address.
- It should be noted that the mobile phone manufacturers usually do not implement all AT commands, command parameters and parameter values in their mobile phones. Also, the behavior of the implemented AT commands may be different from that defined in the standard. In general, GSM modems, designed for wireless applications, have better support of AT commands than ordinary mobile phones.

Basic concepts of SMS technology

1. Validity Period of an SMS Message

An SMS message is stored temporarily in the SMS center if the recipient mobile phone is offline. It is possible to specify the period after which the SMS message will be deleted from the SMS center so that the SMS message will not be forwarded to the recipient mobile phone when it becomes online. This period is called the validity period. A mobile phone should have a menu option that can be used to set the validity period. After setting it, the mobile phone will include the validity period in the outbound SMS messages automatically.

2. Message Status Reports

Sometimes the user may want to know whether an SMS message has reached the recipient mobile phone successfully. To get this information, you need to set a flag in the SMS message to notify the SMS center that a status report is required about the delivery of this SMS message. The status report is sent to the user mobile in the form of an SMS message.

A mobile phone should have a menu option that can be used to set whether the status report feature is on or off. After setting it, the mobile phone will set the corresponding flag in the outbound SMS messages for you automatically. The status report feature is turned off by default on most mobile phones and GSM modems.

3. Message Submission Reports

After leaving the mobile phone, an SMS message goes to the SMS center. When it reaches the SMS center, the SMS center will send back a message submission report to the mobile phone to inform whether there are any errors or failures (e.g. incorrect SMS message format, busy SMS center, etc). If there is no error or failure, the SMS center sends back a positive submission report to the mobile phone. Otherwise it sends back a negative submission report to the mobile phone. The mobile phone may then notify the user that the message submission was failed and what caused the failure.

If the mobile phone does not receive the message submission report after a period of time, it concludes that the message submission report has been lost. The mobile phone may then send the SMS message again to the SMS center. A flag will be set in the new SMS message to inform the SMS center that this SMS message has been sent before. If the previous message submission was successful, the SMS center will ignore the new SMS message but send back a message submission report to the mobile phone. This mechanism prevents the sending of the same SMS message to the recipient multiple times.

Sometimes the message submission report mechanism is not used and the acknowledgement of message submission is done in a lower layer.

4 .Message Delivery Reports

After receiving an SMS message, the recipient mobile phone will send back a message delivery report to the SMS center to inform whether there are any errors or failures (example causes: unsupported SMS message format, not enough storage space, etc). This process is transparent to the mobile user. If there is no error or failure, the recipient mobile phone sends back a positive delivery report to the SMS center. Otherwise it sends back a negative delivery report to the SMS center.

If the sender requested a status report earlier, the SMS center sends a status report to the sender when it receives the message delivery report from the recipient. If the SMS center does not receive the message delivery report after a period of time, it concludes that the message delivery report has been lost. The SMS center then ends the SMS message to the recipient for the second time.

Sometimes the message delivery report mechanism is not used and the acknowledgement of message delivery is done in a lower layer.

GAS SENSOR

- A sensor is a technological device that detects / senses a signal, physical condition and chemical compounds.
- It is also defined as any device that converts a signal from one form to another.

Sensors are mostly electrical or electronic.

- Gas sensor is a subclass of chemical sensors.
- Gas sensor measures the concentration of gas in its vicinity. Gas sensor interacts with a gas to measure its concentration. Each gas has a unique breakdown voltage i.e. the electric field at which it is ionized. Sensor identifies gases by measuring these voltages. The concentration of the gas can be determined by measuring the current discharge in the device.

applications

- Process control industries
- Environmental monitoring
- Boiler control
- Fire detection
- Alcohol breath tests
- Detection of harmful gases in mines
- Home safety
- Grading of agro-products like coffee and spices

Gas sensing technologies:

- Metal Oxide Based Gas Sensors
- Capacitance Based Gas Sensors

- Acoustic Wave Based Gas Sensors
- Calorimetric Gas Sensors
- Optical gas sensors
- Electrochemical gas sensors

types

1. **Metal Oxide Based Gas Sensors**

- Metal oxide sensors are also known as chemiresistors.
- The detection principle of resistive sensors is based on change of the resistance of a thin film upon adsorption of the gas molecules on the surface of a semiconductor.
- The gas-solid interactions affect the resistance of the film because of the density of electronic
- species in the film.

Capacitance Based Gas Sensors

- They measure the change in dielectric constant of films between the electrodes as a function of the gas concentration.
- The capacitive sensor relies on inter-digitated electrode structures, which correspond to the two plates of a standard capacitor, to monitor changes of the dielectric coefficient of the film.
- The simple theory behind it is if the dielectric constant of the film is lower than that of the analyte, the capacitance will increase and vice versa.

Acoustic Wave Based Gas Sensors

- Sound based gas sensors are known as acoustic wave based gas sensors.

- To launch the acoustic waves, this type of sensor use piezoelectric material either in the thin film form or in bulk form which has one or more transducers on its surface.

Carbon monoxide gas sensor

- It can either be battery-operated or AC powered.
- Mostly the sensor will not sound an alarm at lower concentrations (e.g. 100 ppm). The alarm will sound within a few minutes at 400 ppm. So the function is specific to concentration-time. Figure shows simple carbon monoxide sensor.

Carbon monoxide gas sensor (Contd.)

Carbon monoxide sensor can be of different types such as:

- Semiconductor sensor
- Electrochemical sensor
- Digital sensor
- Biomimetic sensor (chem-optical or gel cell sensor)

BUZZER

1. Magnetic Transducer

Magnetic transducers contain a magnetic circuit consisting of a iron core with a wound coil and a yoke plate, a permanent magnet and a vibrating diaphragm with a movable iron piece. The diaphragm is slightly pulled towards the top of the core by the magnet's magnetic field. When a positive AC signal is applied, the current flowing through the excitation coil produces a fluctuating magnetic field, which causes the diaphragm to vibrate up and down, thus vibrating air. Resonance amplifies vibration through resonator consisting of sound hole(s) and cavity and produces a loud sound.

2. Magnetic Buzzer (Sounder)

Buzzers like the TMB-series are magnetic audible signal devices with built-in oscillating circuits. The construction combines an oscillation circuit unit with a detection coil, a drive coil and a magnetic transducer. Transistors, resistors, diodes and other small devices act as circuit devices for driving sound generators. With the application of voltage, current flows to the drive coil on primary side and to the detection coil on the secondary side. The amplification circuit, including the transistor and the feedback circuit, causes vibration. The oscillation current excites the coil and the unit generates an AC magnetic field corresponding to an oscillation frequency. This AC magnetic field magnetizes the yoke comprising the magnetic circuit. The oscillation from the intermittent magnetization prompts the vibration diaphragm to vibrate up and down, generating buzzer sounds through the resonator.

Specifications:

Rated Voltage: A magnetic buzzer is driven by 1/2 square waves (V o-p).

Operating Voltage: For normal operating. But it is not guaranteed to make the minimum Sound Pressure Level (SPL) under the rated voltage.

Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.

Direct Current Resistance: The direct current resistance is measured by ammeter directly.

Sound Output: The sound output is measured by decibel meter. Applying rated voltage and 1/2 square waves, and the distance of 10 cm.

Rated Frequency: A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.

Operating Temp. : Keep working well between -30℃ and +70℃.

How to choose:

Driving methods: AX series with built drive circuit will be the best choice when we cannot provide frequency signal to a buzzer, it only needs direct current.

Dimension: Dimension affects frequency, small size result in high frequency.

Voltage: Depend on V o-p (1/2 square waves)

Fixed methods: From the highest cost to the lowest- DIP, wires/ connector, SMD.

Soldering methods: AS series is soldered by hand, the frequency is lower because of the holes on the bottom. On the other hand, we suggest AC series for the reflow soldering, the reliability is better.

How to choose a buzzer

There are many different kinds of buzzer to choose, first we need to know a few parameters, such as voltage, current, drive method, dimension, mounting type, and the most important thing is how much SPL and frequency we want.

Operating voltage: Normally, the operating voltage for a magnetic buzzer is from 1.5V to 24V, for a piezo buzzer is from 3V to 220V. However, in order to get enough SPL, we suggest giving at least 9V to drive a piezo buzzer.

Consumption current: According to the different voltage, the consumption current of a magnetic buzzer is from dozens to hundreds of mill amperes; oppositely, the piezo type saves much more electricity, only needs a few mill amperes, and consumes three times current when the buzzer start to work.

Driving method: Both magnetic and piezo buzzer have self drive type to choose. Because of the internal set drive circuit, the self drive buzzer can emit sound as long as connecting with the direct current. Due to the different work principle, the magnetic buzzer need to be driven by 1/2 square waves, and the piezo buzzer need square waves to get better sound output.

Dimension: The dimension of the buzzer affects its SPL and the frequency, the dimension of the magnetic buzzer is from 7 mm to 25 mm; the piezo buzzer is from 12 mm to 50 mm, or even bigger.

Connecting way: Dip type, Wire type, SMD type, and screwed type for big piezo buzzer are usually seen.

Sound Pressure Level (SPL): Buzzer is usually tested the SPL at the distance of 10 cm, if distance double, the SPL will decay about 6 dB; oppositely, the SPL will increase 6 dB when the distance is shortened by one time. The SPL of the magnetic buzzer can reach to around 85 dB/ 10 cm; the piezo buzzer can be designed to emit very loud sound, for example, the common siren, are mostly made of piezo buzzer.

Introduction of Piezo Buzzer

Specifications:

Rated Voltage: A piezo buzzer is driven by square waves (V p-p).

Operating Voltage: For normal operating. But it is not guaranteed to make the minimum SPL under the rated voltage.

Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.

Capacitance: A piezo buzzer can make higher SPL with higher capacitance, but it consumes more electricity.

Sound Output: The sound output is measured by decibel meter. Applying rated voltage and square waves, and the distance of 10 cm.

Rated Frequency: A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.

Operating Temp.: Keep working well between -30℃ and +70℃.

How to choose:

Driving methods: AZ-xxxxS-x series with built drive circuit will be the best choice when we cannot provide frequency signal to a buzzer, it only needs direct current. Besides, there are different tone nature for you to choose, such as continuous, fast pulse, and slow pulse.

Dimension: Dimension affects frequency, small size result in high frequency.

Voltage: Driven by square waves (V p-p), the higher voltage results in the higher SPL.Pin Pitch: The numerous spec. for the piezo buzzers lead to the difficulty in finding a spec. in facsimile, therefore we suggest that you can firstly choose a spec. with the same pitch and similar frequency. Introduction of Micro Speaker

How to choose:

The factors which affect the SPL: the square measure of diaphragm, the amplitude of vibration, magnetic field intensity, power, impedance, resonant chamber, the pattern and the thickness of diaphragm, and the holes.

Power vs. SPL: Suppose all the conditions are the same, increasing the power dose not mean the SPL will increase as well. We need to revise the diaphragm and the sound coil to load the higher power, but it leads to lower SPL instead.

Dimension vs. SPL: A larger speaker can vibrate more air, therefore it provides higher SPL. In addition, the thicker speaker can give wider amplitude of vibration which also leads to higher SPL.

Acoustics: What we request most is how much SPL a micro speaker can output.

Matching: It will be better to provide the power slight higher than the rated power for the enlarged circuit.

Question for mechanism:

The volume of the resonant chamber: The general problem of the consuming products is that the resonant chambers are not big enough. We can only try to find space to enlarge the volume of the resonant chamber.

Airtight: The front and back sound fields of the speaker should be separated to avoid neutralization.

Shock absorber: When a speaker works the vibration will also happen at the same time. In order to reduce interference, it will do good to have some material between speaker and case to absorb the shock.

Mounting: The speakers are usually fixed on the case. Firmly fixed is important especially for the iron housing or the large size to avoid separating in the drop test.

How to choose the speaker

Dimension: To the micro speaker, size has decisive influence on its volume. 5mm difference of diameter might result in double or half area of diaphragm, therefore the SPL is quite different. Besides, the thicker speaker has more space to vibrate the air, and usually has bigger magnet, so it will be more powerful to push the air and emit louder sound.

Power: Mainly refer to how much power can a speaker bear, there is no direct relation to the SPL. The speaker with larger power needs to use thicker diaphragm and sound coil to bear larger power, which will lead to lower efficiency (SPL). Therefore, according to the mechanical design, try to select a larger speaker which matches the outputting power from the amplified circuit, then the best SPL would be emitted.

Impedance: Higher impedance can save more electricity, however, the SPL and the loaded power will go down. The reason is that we have to use thinner wire or to coil more, the front makes the power lower, and the after leads to heaviness and low efficiency.

The material of diaphragm: Most speakers (diameter less than 50mm) use mylar diaphragms, which are easily finished, cheaper and waterproof. However, mylar diaphragm is not good at heatproof and the sound is stiff.The patterns of diaphragm: The speaker with concentric circles diaphragm is good for the speech sounds. Generally, the SPL is good at the frequency before 5-6 KHz, but will dramatically decrease after 6 KHz. On the other hand, the speaker with radiate diaphragm has average frequency response. Supposing other conditions are all the same, the SPL of radiate diaphragm will lower than the concentric circles one at the frequency before 6 KHz.

POWER SUPPLY UNIT

INTRODUCTION:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as “Regulated D.C Power Supply”.

BLOCK DIAGRAM OF POWER SUPPLY:

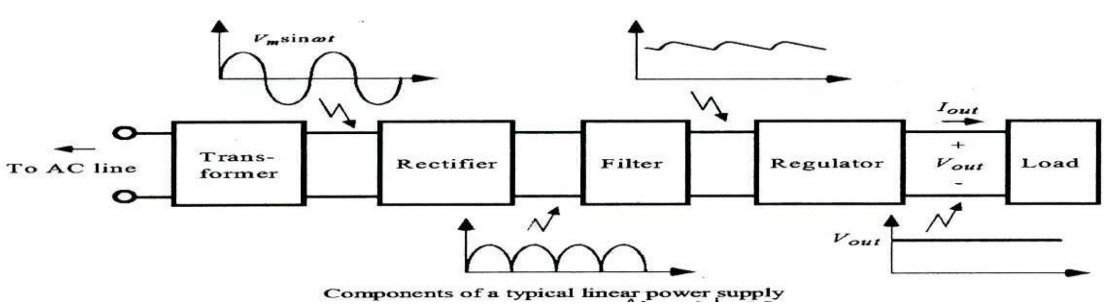


Fig.4.2.1: Block Diagram of Power Supply

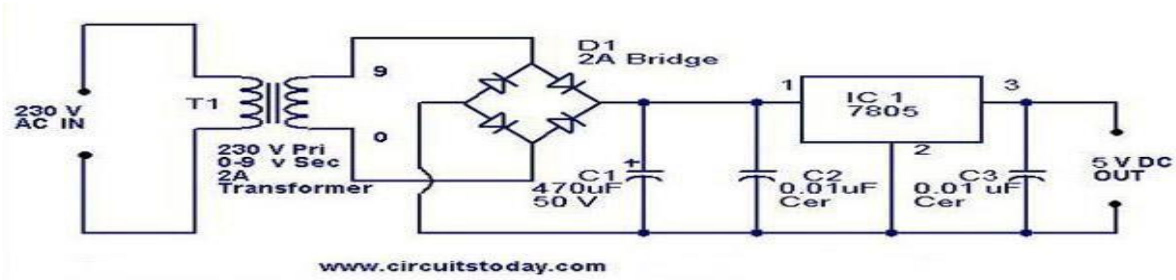


Fig.4.2.2: Schematic Diagram of Power Supply

DESCRIPTION OF POWER SUPPLY:

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a a different form of energy – such as solar, mechanical, or chemical - into electrical energy.

A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered. For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

SOFTWARE EXPLANATION:

Software Requirements

- Proteus simulation
- Arduino software
- Programming language

Arduino software:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals

- has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50

- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use forbeginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers c an make their own version
- Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money
- Getting Started with Arduino and Genuino

products:- Install the Arduino Software (IDE) on

Windows PCs-

…This document explains how to install the Arduino Software (IDE) on Windows machines.

✓ Download the Arduino Software(IDE)

✓ Proceed with board specific instructions.

How to Download the Arduino Software (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

Installation:

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.

To open an existing project example, select File → Example → Basics → Blink.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24 V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

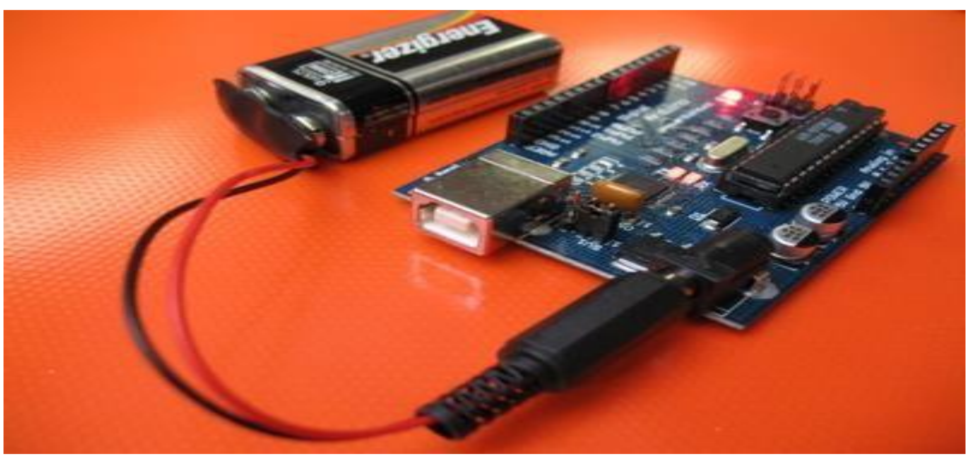


Fig.4.3.: Arduino with battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called “sketches”, but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

MC Programming Language: Embedded C

This is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements. Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

Proteus:

Proteus:

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing.

ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.

The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

CHAPTER-5

PROJECT TESTING

SOFTWARE TESTING

Software Testing is a process of executing the application with an intent to find any software bugs. It is used to check whether the application met its expectations and all the functionalities of the application is working. The final goal of testing is to check whether the application is behaving in the way it is supposed to under specified conditions. All aspects of the code are examined to check the quality of application. The primary purpose of testing is to detect software failures so that defects may be uncovered and corrected. The test cases are designed in such way that scope of finding the bugs is maximum.

TESTING LEVELS

There are various testing levels based on the specificity of test.

- Unit testing: Unit testing refers to tests conducted on a section of code in order to verify the functionality of that piece of code. This is done at the function level.
- Integration Testing: Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Its primary purpose is to expose the defects associated with the interfacing of modules.
- System Testing: System testing tests a completely integrated system to verify that the system meets its requirements.

Acceptance testing: Acceptance testing tests the readiness of application, satisfying all requirements.

Performance testing: Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or device such as response time or millions of instructions per second etc.

SYSTEM TEST CASES

A test case is a set of test data, preconditions, expected results and post conditions, developed fora test scenario to verify compliance against a specific requirement. We have designed and executed a few test cases to check if the project meets the functional requirements.

CODE IMPLEMENTATION

```
#include <LiquidCrystal.h> //LCD
#include <SoftwareSerial.h> //COMMINCATION GSM MODULE

LiquidCrystal lcd(13,12,11,10,9,8);  //RS 13,EN 12, D4 11 ,D5 10,D6 9,D7 8
SoftwareSerial mySerial(2,3);//rx , tx  //GSM

int gas = A5;
int buz=7;
void serialFlush(){
while(Serial.available() > 0) {
char tt = Serial.read();
}
}
void sendmsg(char *num,char * msg)
{
Serial.print("AT+CMGS=\"");
Serial.print(num);
Serial.println("\"");delay(800);
Serial.println(msg);delay(800);
Serial.write(0x1a);delay(2000);
}
const char* number = "8885522598\0";  /*Give the required mobile num to send the msg*/


void setup(void)
```

```
{

Serial.begin(9600);
mySerial.begin(9600);
lcd.begin(16,2);
lcd.clear();lcd.setCursor(0, 0);lcd.print("WELCOME");     // 1st romw 0th colom
delay(1000);
pinMode(gas, INPUT);
pinMode(buz, OUTPUT);

digitalWrite(buz,HIGH);//OFF STATE
Serial.println("AT");delay(1000);
Serial.println("AT+CMGF=1");delay(1000);
Serial.println("AT+CNMI=1,2,0,0");delay(1000);
Serial.println("AT+CSMP=17,167,0,16");delay(1000);
delay(1000);

lcd.clear();lcd.setCursor(0, 0);lcd.print("READY");

}

void loop()
{
int gas_data = digitalRead(gas);
Serial.println(gas_data);
lcd.clear();
lcd.setCursor(0, 0); lcd.print("GASDETECTION GSM ");
delay(2000);
lcd.clear();
lcd.setCursor(0, 1); lcd.print("G:");lcd.print(gas_data);lcd.print("     ");
delay(2000);
```

```
if(gas_data == LOW)
{
lcd.clear();lcd.setCursor(0, 1);lcd.print("GAS DETECTED");
mySerial.println("GAS DETECTED"); delay(500);
lcd.clear();lcd.setCursor(0, 0);lcd.print("GSM INIT");
Serial.println("GSM CONN ... ");
Serial.println("AT");delay(1000);
Serial.println("AT+CMGF=1");delay(1000);
Serial.println("AT+CNMI=1,2,0,0");delay(1000);
Serial.println("AT+CSMP=17,167,0,0");delay(1000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("READY");
Serial.println("GSM DONE");
sendmsg(number,"GAS DETECTED");
delay(3000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("MESSAGE SENT");
digitalWrite(buz,LOW);
delay(1000);
}
else
{
digitalWrite(buz,HIGH);
delay(500);
}
}
```

CHAPTER 6

RESULT ANALYSIS

RESULTS

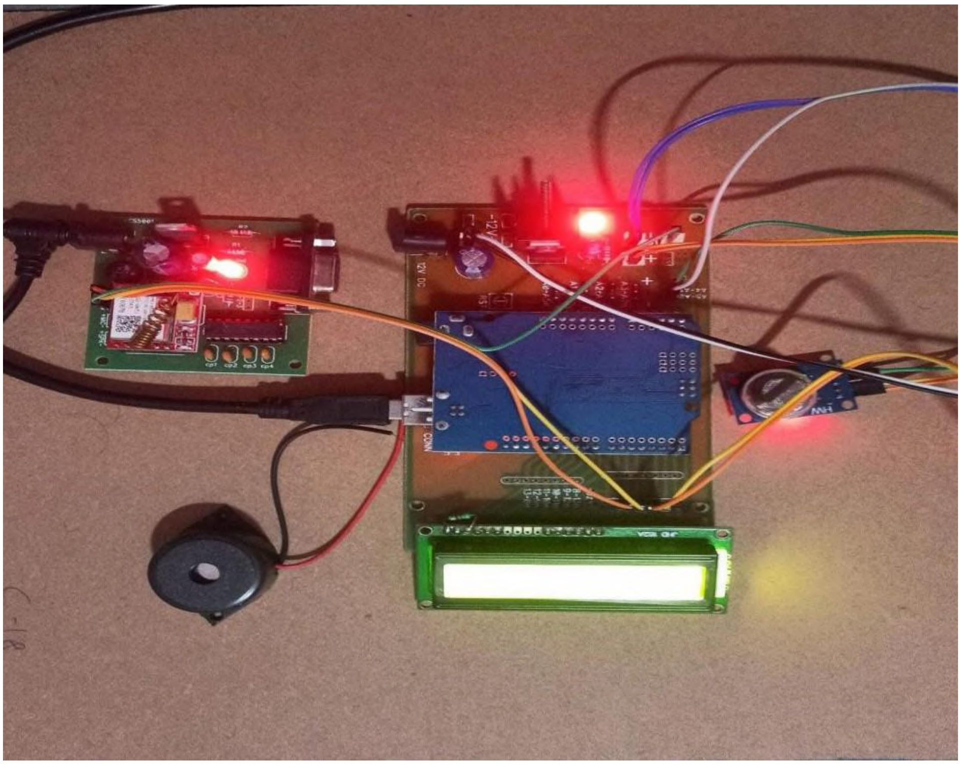


Fig:6.1 The Hardware setup of the proposed system

DISPLAYING NORMAL READING VALUES ON THE LCD

Gas sensor and Buzzer senses the concentration of the harmful gases and the level of concentrated gas in the Environment continuously to the Arduino Uno. Now the Arduino Uno which is brain of this project takes the information from sensors and transmit that values to the LCD for displaying them. The below r represents LCD display that displaying the normal ridings of the sensor and normal range of concentrated harmful gases.

Fig:6.2 Danger gas uploading

DISPLAYING OF WARNING MESSAGES

Here the Gas Sensor and Buzzer consists of potentiometer using this we can set the threshold. When the readings from gas sensor exceeds the set threshold it will display warning messages on the LCD.

Fig:6.3 Gas Detected

Fig:6.4 Displaying Warning message sent to the user

APPLICATIONS AND ADVANTAGES

6.2.1APPLICATION OF THE PROJECT

- Used in Gas Industries, Houses, Labs, Hospitals etc
- It is used when mining of coal and other materials to detect.
- Used in municipal gas distribution.
- Used in nuclear power stations.

ADVANTAGES OF THE PROJECT

- Low cost
- Low power consumption
- High accuracy
- It also detects alcohol so it is used as liquor tester.
- The sensor has excellent sensitivity combined with a quick response time.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

Future scope

- Additional features include call will initiated to telephone number.
- Temperature display during periods times.
- Another very interesting and significant improvement would be to accommodate multiple receiver MODEMS.
- Multilingual display can be another added variation in the project.

Conclusion

- Hence, **Gas leakage detection** is essential to prevent accidents and to save human lives. This project presented Gas leakage detection and alert system.
- This system triggers LED and buzzer to alert people when **Gas leakage** is detected.
- This system is very simple yet reliable.The gas leakage detector we used can detect the **Gas leakage** in **2 seconds**.

Publication

- Online Mega International Conference- “Smart Modernistic in Electronics and Communication”(ICSMEC-21)
- Accepted in ICSMEC-21 Reference number is ICSMEC21-0010.

REFERENCE

[1].D. Norris, Smart Home Automation Based on IOT and Android Technology M.Abivandhana1, K.Divya2, D.Gayathri3, R.RuhinKouser4 Student1, 2, 3, Assistant Professor4 Department of CSE Kingston Engineering College, Katpadi, Vellore, India

[2].International Journal of Innovative Studies in Sciences and Engineering Technology (IJISSET)IoT Based Home Automation Using Raspberry PI ISSN 2455-4863 (Online) www.ijisset.org Volume: 3 Issue: 4 | April 2017

[3] Mark Murphy (2011): Beginning Android 3, Après. ISBN-13 (pbk): 978-1-4302-3297-1 ISBN-13 (electronic): 978-1-4302-3298-8

[4] Addison-Wesley (2011): Android Wireless Application Development, 2nd edition ISBN-13: 978-0-321-74301-5 ISBN-10: 0-321-74301-6

[5] Wikipedia(2009). HomeAutomation. From http://en. Wikip edia. org/wiki/Home_automation. Retrievedon 20/5/2018

[6]Martin Bates(2006). Interfacing PIC Microcontrollers Embedded Designby Interactive Simulation. Newnes, London.

[7] S. Alam, M. M. R. Chowdhury, and J. Noll, "Senaas: An event-driven sensor virtualization approach for internet of things cloud," in Networked Embedded Systems for Enterprise Applications (NESEA), 2010 IEEE International Conference on, November 2010.

[8] Z. Schelby, K. Hartke, and C. Bormann, (Aug. 28, 2013) „„,Constrained application protocol (CoAP),‛‛‛ CoRE Working Group Internet-Draft. [Online].

[9] Ch. Manohar Raju, N. Sushma Rani. An android based automatic gas detection and indication robot. In International Journal of Computer Engineering and Applications. 2014; 8(1).

[10] Zhao Yang, Mingliang Liu, Min Shao, Yingjie Ji Research on leakage detection and analysis of leakage point in the gas pipeline system. In Open Journal of Safety Science and Technology; 2011.

[11] S Shyamaladevi, V G Rajaramya, P Rajasekar, P Sebastin Ashok. ARM7 based automated high performance system for lpg refill booking & leakage detection. 2014; 3(2).

[12] Pal-Stefan Murvaya, Ioan Sileaa. A survey on gas leak detection and localization techniques.

[13] Falohun A.S., Oke A.O., Abolaji B.M. Dangerous gas detection using an integrated circuit and MQ-9. In International Journal of Computer Applications. 2016; 135(7).

[14] Ashish Shrivastava, Ratnesh Prabhaker, Rajeev Kumar, Rahul Verma. GSM based gas leakage detection system. In International Journal of Technical Research and Applications. 2013; 1(2).

A

MAJOR PROJECT REPORT

on

DESIGN AND IMPLEMENTATION OF SMART MITTEN FOR DEAF AND DUMB PEOPLE

Submitted by

1) Mr. Y.Lokesh Reddy	(17K81A04J0)
2) Mr. Vedvikas Reddy Magam	(17K81A04H7)
3) Ms. P.Pravalika	(17K81A04G2)
4) Ms. Vure Supriya	(17K81A04H8)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Mrs VIDADALA SRIJA

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500100

NBA &NAAC A+ Accredited

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE
UGC AUTONOMOUS
Dhulapally, Secunderabad-500 100
NBA &NAAC A+ Accredited

BONAFIDE CERTIFICATE

This is to certify that the project entitled "Recursive Block Based Key Point Matching Copy Move Image Forgery Detection" is being submitted by **1.Mr. Y.Lokesh Reddy 17K81A04J0, 2.Mr.Vedvikas Reddy Magam 17K81A04H7 3.Ms. P.Pravalika 17K81A04G2 4.Ms. Vure Supriya 17K81A04H8** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OFTECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Internal Guide	**Head of the Department**
Ms.Vidadala Srija Department of Electronics and Communication Engineering	Dr. Hari Krishna Department of Electronics and Communication Engineering
Internal Examiner	**External Examiner**

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT Y.LOKESH REDDY WITH ROLL NO.17K81A04J0, VEDVIKAS REDDY MAGAM WITH ROLL NO.17K81A04H7, P.PRAVALIKA WITH ROLL NO.17K81A04G2, VURE SUPRIYA WITH ROLL NO.17K81A04H8, OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "DESIGN AND IMPLEMENTATION OF SMART MITTEN FOR DEAF AND DUMB PEOPLE" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of '**Bachelor of Technology in Department of Electronics and Communication Engineering**', session: 2017-2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project work entitled '**Design and Implementation of Smart Mitten For Deaf and Dumb People**' is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

Y.Lokesh Reddy	Vedvikas Reddy Magam	P.Pravalika	Vure Supriya
17K81A04J0	17K81A04H7	17K81A04G2	17K81A04H8

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success**.**

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr. P. Santosh Kumar Patra**, our beloved **Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B. Hari Krishna**, **Head & Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our department **Project Coordinator, Dr. A. Anand Kumar, Professor** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our **Class Project Coordinator, Mrs. A. Soumya,** Department of ECE and our project **Internal guide, Ms.Vidadala Srija, Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level **Project Coordinators Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. Y.Lokesh Reddy
2. Vedvikas Reddy Magam
3. P.Pravalika
4. Vure Supriya

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Communication is the only way by which we are able to express our thoughts among the people. Normalpeople can convey their thoughts effectively by establishing the conversation between them. But in our society there are lot of people who are physically disable that means (deaf and dumb) are not able to communicate effectively. Because of their disability they are not able to stand in race with the normal people. Some of the people have problem regarding hearing and some are not able to talk so they lag behind the normal people. Generally these persons use the sign language for the communication but they find some problem in communication with those is not able to understand sign language. So there is problem between normal people and physically disable people. This system has main purpose to reduce the communication gap between two communities. The main aim of our proposed project is to developed the cost effective system where disable people can communicate with normal people by using hand glove. This means that communication is not barrierbetween two communities by using smart glove. So disable can also able to grow in their respective field. Usingsuch system by disable people can make nation grow.

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION:

One of the important problems that our society faces is that people with disabilities are finding it hard to come up with the fast-growing technology. The access to communication technologies has become essential for the handicapped people. Deaf and dumb people use sign language for communication but they find difficulty in communicating with others who don't understand sign language. Sign language is an expressive and natural way for communication between normal and dumb people. In this project Force Sensor Plays the major role, which are placed on fingers and it changes resistance depending on the amount of force on the sensor. The proposed algorithm states that four force sensors for four fingers that each sensor has range of voltage and each range indicates one message, the respective message will be displayed on LCD board and vocalized by speaker. Force sensor is load cells, transducers that convert force into measurable electrical outputs. Sensors convert the known applied load into voltage and Arduino microcontroller will receive this sensed data and show output message on the LCD board and vocalized by speaker. The message that has been used is Hungry, Need Water.

This decreasing ratio of Literate and Employed Deaf and Dumb population is a result of the physical disability of hearing for deaf people and disability of speaking for dumb people so it yields to lack of communication between normal person and Deaf and Dumb Person. It actually becomes the same problem of two persons which knows two different language, no one of them knows any common language so it becomes a problem to talk with each other and so they require a translator physically which may not be always convenient to arrange and this same kind of problem occurs in between the Normal Person and the Deaf person or the Normal Person and the Dumb person [2-4]. To overcome this problem, we introduce a unique application. Our application model is a desirable Interpreter which translates. Natural English Sentences as, a text input by Normal Person for Deaf Person and Sign Language, in form of Gesture by a Dumb Person to Synthesized English Words which have a corresponding meaning in Sign Language which interprets a particular thing, as an Audio Output for Normal Person. This will help Normal and Deaf and dumb communities by removing the communication gap between them [5-8]. The sign language is an important and only method of communication for deaf-dumb persons. As sign language is a formal language employing a system of hand gesture for communication (by the deaf).

1.2 EMBEDDED SYSTEMS:

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.

The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes. The applications software on such processors is sometimes referred to as firmware

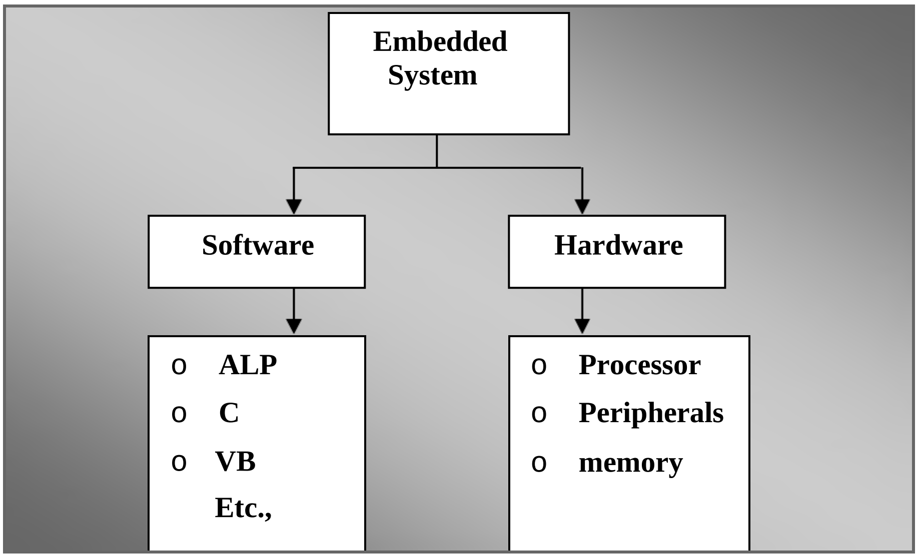


Figure 1.1 Block diagram of Embedded System

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

Memory: It is used to store data or address.

Peripherals: These are the external devices connected

Processor: It is an IC which is used to perform some task

Applications of embedded systems

Manufacturing and process control

Construction industry

Transport

Buildings and premises

Domestic service

Communications

Office systems and mobile equipment

Banking, finance and commercial

Medical diagnostics, monitoring and life support

Testing, monitoring and diagnostic systems

Processors are classified into four types like:

Micro Processor (µp)

Micro controller (µc)

Digital Signal Processor (DSP)

Application Specific Integrated Circuits (ASIC)

Micro Processor (µp):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Three basic characteristics differentiate microprocessors:

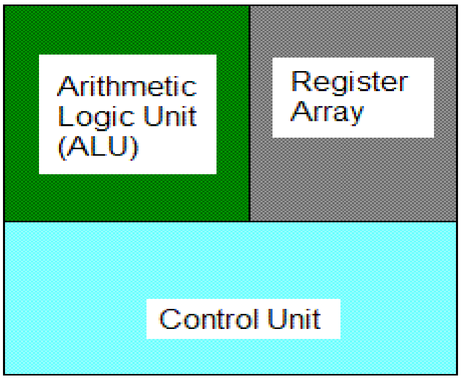
Instruction set: The set of instructions that the microprocessor can execute.

Bandwidth : The number of bits processed in a single instruction.

Clock speed : Given in megahertz (MHz), the clock speed determines how many instructions per second the processor can execute.

In both cases, the higher the value, the more powerful the CPU. For example, a 32-bit microprocessor that runs at 50MHz is more powerful than a 16-bit microprocessor that runs at 25MHz. In addition to bandwidth and clock speed, microprocessors are classified as being either RISC (reduced instruction set computer) or CISC (complex instruction set computer).

A microprocessor has three basic elements, as shown above. The ALU performs all arithmetic computations, such as addition, subtraction and logic operations (AND, OR, etc). It is controlled by the Control Unit and receives its data from the Register Array. The Register Array is a set of registers used for storing data. These registers can be accessed by the ALU very quickly. Some registers have specific functions - we will deal with these later. The Control Unit controls the entire process. It provides the timing and a control signal for getting data into and out of the registers and the ALU and it synchronizes the execution of instructions (we will deal with instruction execution at a later date).



Three Basic Elements of a Microprocessor

Micro Controller (µc):

A microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.

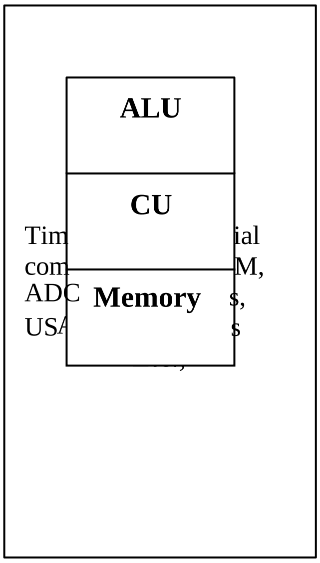


Figure 1.2 Block Diagram of Micro Controller (µc)

Digital Signal Processors (DSPs):

Digital Signal Processors is one which performs scientific and mathematical operation. Digital Signal Processor chips - specialized microprocessors with architectures designed specifically for the types of operations required in digital signal processing. Like a general-purpose microprocessor, a DSP is a programmable device, with its own native instruction code. DSP chips are capable of carrying out millions of floating point operations per second, and like their better-known general-purpose cousins, faster and more powerful versions are continually being introduced. DSPs can also be embedded within complex "system-on-chip" devices, often containing both analog and digital circuitry.

Application Specific Integrated Circuit (ASIC)

ASIC is a combination of digital and analog circuits packed into an IC to achieve the desired control/computation function

ASIC typically contains

- CPU cores for computation and control
- Peripherals to control timing critical functions
- Memories to store data and program
- Analog circuits to provide clocks and interface to the real world which is analog in nature
- I/Os to connect to external components like LEDs, memories, monitors etc.

2.2 Computer Instruction Set

There are two different types of computer instruction set there are:

1. RISC (Reduced Instruction Set Computer) and
2. CISC (Complex Instruction Set computer)

2.2.1 Reduced Instruction Set Computer (RISC)

A RISC (reduced instruction set computer) is a microprocessor that is designed to perform a smaller number of types of computer instruction so that it can operate at a higher speed (perform more million instructions per second, or millions of instructions per second). Since each instruction type that a computer must perform requires additional transistors and circuitry, a larger list or set of computer instructions tends to make the microprocessor more complicated and slower in operation.

Besides performance improvement, some advantages of RISC and related design improvements are:

- A new microprocessor can be developed and tested more quickly if one of its aims is to be less complicated.
- Operating system and application programmers who use the

microprocessor's instructions will find it easier to develop code with a smaller instruction set.

- The simplicity of RISC allows more freedom to choose how to use the space on a microprocessor.
- Higher-level language compilers produce more efficient code than formerly because they have always tended to use the smaller set of instructions to be found in a RISC computer.

2.2.2 RISC characteristics

- **Simple instruction set:**

In a RISC machine, the instruction set contains simple, basic instructions, from which more complex instructions can be composed.

- **Same length instructions.**

Each instruction is the same length, so that it may be fetched in a single operation.

- **1machine-cycleinstructions.**

Most instructions complete in one machine cycle, which allows the processor to handle several instructions at the same time. This pipelining is a key technique used to speed up RISC machines.

2.2.3 Complex Instruction Set Computer (CISC)

CISC, which stands for **Complex Instruction Set Computer,** is a philosophy for designing chips that are easy to program and which make efficient use of memory. Each instruction in a CISC instruction set might perform a series of operations inside the processor. This reduces the number of instructions required to implement a given program, and allows the programmer to learn a small but flexible set of instructions.

The advantages of CISC

At the time of their initial development, CISC machines used available technologies to optimize computer performance.

- Microprogramming is as easy as assembly language to implement, and much less expensive than hardwiring a control unit.
- The ease of micro-coding new instructions allowed designers to make CISC machines upwardly compatible: a new computer could run the same programs as earlier computers because the new computer would contain a superset of the instructions of the earlier computers.
- As each instruction became more capable, fewer instructions could be used to implement a given task. This made more efficient use of the relatively

slow main memory.

- Because micro program instruction sets can be written to match the constructs of high-level languages, the compiler does not have to be as complicated.

The disadvantages of CISC

Still, designers soon realized that the CISC philosophy had its own problems, including:

- Earlier generations of a processor family generally were contained as a subset in every new version --- so instruction set & chip hardware become more complex with each generation of computers.
- So that as many instructions as possible could be stored in memory with the least possible wasted space, individual instructions could be of almost any length---this means that different instructions will take different amounts of clock time to execute, slowing down the overall performance of the machine.
- Many specialized instructions aren't used frequently enough to justify their existence --- approximately 20% of the available instructions are used in a typical program.
- CISC instructions typically set the condition codes as a side effect of the instruction. Not only does setting the condition codes take time, but programmers have to remember to examine the condition code bits before a subsequent instruction changes them.

2.3 Memory Architecture

There two different type’s memory architectures there are:

- Harvard Architecture
- Von-Neumann Architecture

2.3.1 Harvard Architecture

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

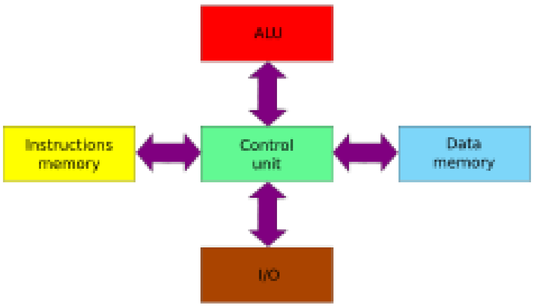


Figure 1.3 Harvard Architecture

Modern uses of the Harvard architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

- Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behavior must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.
- Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM

Cortex-M3 processor (not all ARM chips have Harvard architecture).

- Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

2.3.2 Von-Neumann Architecture

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. The von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A stored-program digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

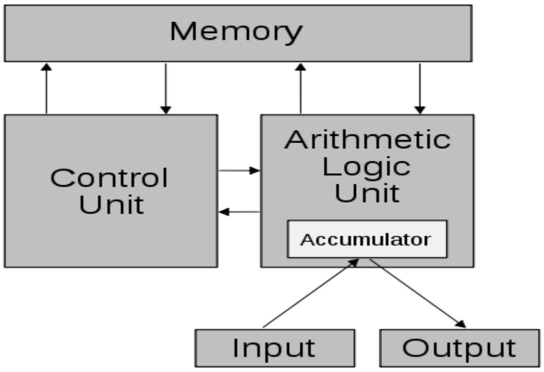


Figure 1.4 Schematic of the Von-Neumann Architecture.

Basic Difference between Harvard and Von-Neumann Architecture

The primary difference between Harvard architecture and the Von Neumann architecture is in the Von Neumann architecture data and programs are stored in the same memory and managed by the same information handling system.

Whereas the Harvard architecture stores data and programs in separate memory devices and they are handled by different subsystems.

In a computer using the Von-Neumann architecture without cache; the central processing unit (CPU) can either be reading and instruction or writing/reading data to/from the memory. Both of these operations cannot occur simultaneously as the data and instructions use the same system bus.

In a computer using the Harvard architecture the CPU can both read an instruction and access data memory at the same time without cache. This means that a computer with Harvard architecture can potentially be faster for a given circuit complexity because data access and instruction fetches do not contend for use of a single memory pathway.

Today, the vast majority of computers are designed and built using the Von Neumann architecture template primarily because of the dynamic capabilities and efficiencies gained in designing, implementing, operating one memory system as opposed to two. Von Neumann architecture may be somewhat slower than the contrasting Harvard Architecture for certain specific tasks, but it is much more flexible and allows for many concepts unavailable to Harvard architecture such as self-programming, word processing and so on.

CHAPTER 2

LITERATURE SURVEY AND EXISTING METHODS

2.1 Literature Survey:

Generally, deaf & dumb people use sign language for the communication. Therefore, it is necessary to have a gesture recognition or sign language detection system to reduce the communication gap between two communities by using glove. The data of force sensor is given to input pin of microcontroller at transmitter side. We get the corresponding voice output of each gesture and display of recognized gesture in form of text on the LCD display at receiver side. As novelty, we are appending home automation. The corresponding home appliances are controlled by using gesture and operated through relay circuit. The main purpose of smart glove an ease of sharing basic ideas, minimize communication gap and an easier collaboration for the hard of hearing people. The main aim of our project is to develop the cost-effective system. So, disable can able to grow in their respective field. Every Normal human being sees, listens and then reacts to the situations by speaking himself out. But there are some human beings those who are not able to speak or listen, but they try to react through actions most of time normal people are not able to understand what they want to say. This application will help for both of them to communicate with each other. It consists of several parts, in part one with the help of hand gestures the signs will be detected by the sensors and the output will be given. Communication between speakers and non-speakers i.e. dumb and deaf people can be problematic, inconvenient etc. This project attempts to bridge the communication gap by designing a portable glove that captures the user’s gestures and outputs the translated text on a device. The glove is equipped with flex sensors, to measure the flexion of the fingers, and the contact between fingers. The glove’s Raspberry Pi microcontroller analyses the sensor readings to identify the gesture from a library of learned gestures. So, we are going to design and built a glove to be worn on the right hand that translate sign language into spoken English. Every person's hand is a unique size and shape, and we aimed to create a device that could provide reliable translations regardless of those differences. Our device uses five Spectra Symbol Flex-Sensors that we use to quantify how much each finger is bent. These sensors are read, averaged, and arranged into packets using a

Raspberry-pi microcontroller. These packets are then sent serially to a Raspberry-pi to be run in conjunction with a Python script. The user creates data sets of information from the glove for each gesture that should eventually be translated, and the algorithm trains over these datasets to predict later at runtime what a user is signing.

2.2 Existing Systems:

Based on Image processing technology, a device that enables Voice for the mute was made. Basically, the gestures are captured using webcam and converted into text using Image processing. This text is then converted to speech using text to speech convertor. But only alphabets were captured in this device, so further research was done to search for better devices.

The system based on "Two Way Wireless Data Communication and American Sign Language Translator Glove for Images Text and Speech Display on Mobile Phone "develops applicative architecture of hand glove. It records the gestures made by a deaf and mute people, converts them into a meaningful text and transmits them to remote areas with help of Bluetooth, GSMCDMA and Internet modules. Then GUI successfully transmits them to an authorized mobile phone and simultaneously displays the responses transmitted by that phone on the same screen . The device "Sign Language Interpreter Using a Smart Glove" uses hand gestures along with the facial expressions and the body language to convey the intended message. The smart glove recognizes the signing gesture and text representation of the signed gesture is created. Then text is converted to audio output.

The system for "Two Way Communication between Deaf and Dumb People and Normal People" uses video processing approach. Video processing involves frame formation from videos, finding region of interest (ROI) and mapping of images with language knowledgebase. Gestures from real-time video and mapping it with human-understandable speech. Then natural language was taken as input and mapped with equivalent Sign Language animated gestures.

Hardware based "Smart Glove with Gesture Recognition Ability for the Hearing and Speech Impaired" design and implements a low cost wired interactive glove. It is interfaced with a computer running MATLAB or Octave, with a high degree of accuracy for gesture recognition. The glove maps the orientation of the hand and fingers with the help of bend sensors, Hall Effect sensors and an accelerometer. The system is modelled for the differently able people to help convert sign language to a more human understandable form such as textual messages.

Authors have utilized of accelerometer glove (which is mounted on each finger) and RBG camera to identify sign language. The entire limb movement is track with the help of additional accelerometer on the limb. The author’s model isolates sign language gestures by using parallel hidden Markov models. Efficiency is much more than video sensors.

SolankiKrunal, has used flex sensor, microcontroller, DC and LCD. Movements in hand’s finger causes voltage drop which is given to the micro controller via ADC. The micro controller identifies the signal and process word to word to be displayed on LCD screen. For example, BOY, each letter is process at a time and it is saved by pressing a button the complete sentence is shown on LCD.

CHAPTER 3

BLOCK DIAGRAM

3.1 Block Diagram:

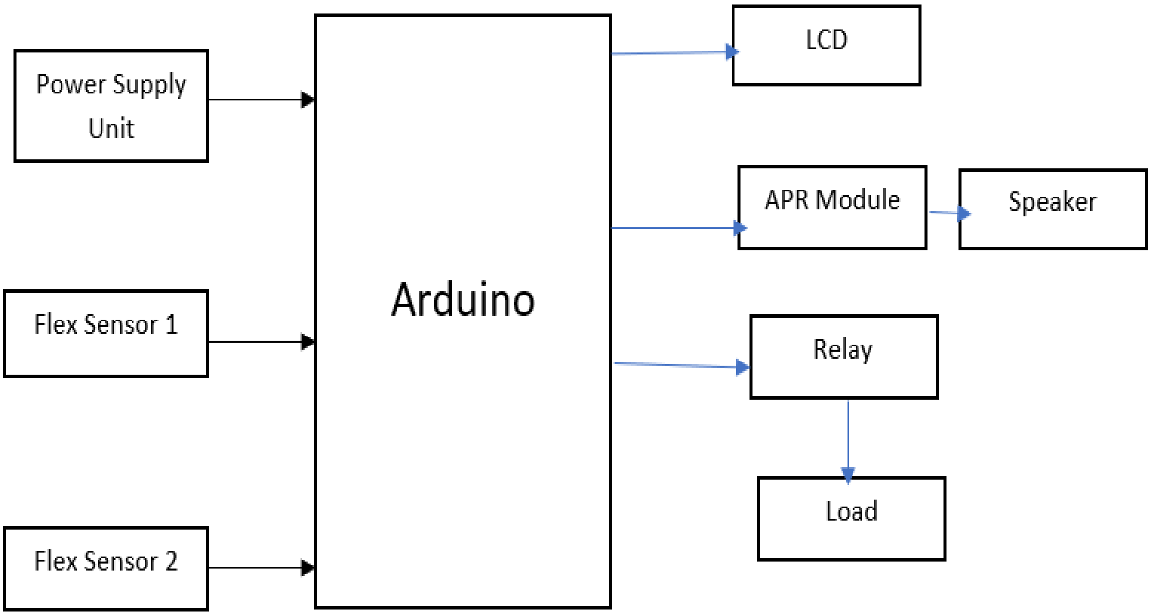


Fig 3.1: Block diagram

The Block Diagram of Smart Mitten for Deaf and Dumb people is shown in the above figure. The Blocks of the circuit are

1. Arduino
2. Power Supply Unit
3. Flex Sensor
4. LCD
5. APR Module
6. Relay
7. Load

- Firstly, the detection of the bending and position of Flex sensors is done and the information is passed to the arduino.
- The Arduino receives the input signals from flex sensors.

- The audio output is given from the speaker.
- The text output is shown in the LCD display.

3.2 COMPONENT DESCRIPTION:

3.2.1 ARDUINO UNO:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

3.2.2 Flex Sensor:

Flex sensors are usually available in two sizes. One is 2.2 inch and another is 4.5 inch. Although the sizes are different the basic function remains the same. They are also divided based on resistance. There are LOW resistance, MEDIUM resistance and HIGH resistance types. Choose the appropriate type depending on requirement. Here we are going to discuss 2.2inch Flex sensor that is FS-L-0055.

3.2.3 LCD

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

3.2.4 APR MODULE

APR9600 is a low-cost high performance sound record/replay IC incorporating flash analogue storage technique. Recorded sound is retained even after power supply is removed from the module. Sampling rate for a 60 second recording period is 4.2 kHz that gives a sound record/replay bandwidth of 20Hz to 2.1 kHz**.**

3.2.5 RELAY

A relay is an electrically operated switch. These are remote control electrical switches that are controlled by another switch, such as a horn switch or a computer as in a power train control module, devices in industries, home based applications. Relays allow a small current pin, 4-pin, 5-pin, and 6-pin, single switch or dual switches. Relays are used throughout the automobile. Relays which come in assorted sizes, ratings, and applications, are used as remote control switches. A typical vehicle can have 20 relays or more.

CHAPTER 4

SOFTWARE AND HARDWARE COMPONENTS

4.1 Proteus

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

4.1.1 About Proteus

It is a software suite containing schematic, simulation as well as PCB designing.
ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.
The designer can also develop 2D drawings for the product.

4.1.2 Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through whole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

4.2 Starting New Design

Step 1: Open ISIS software and select new design in File menu

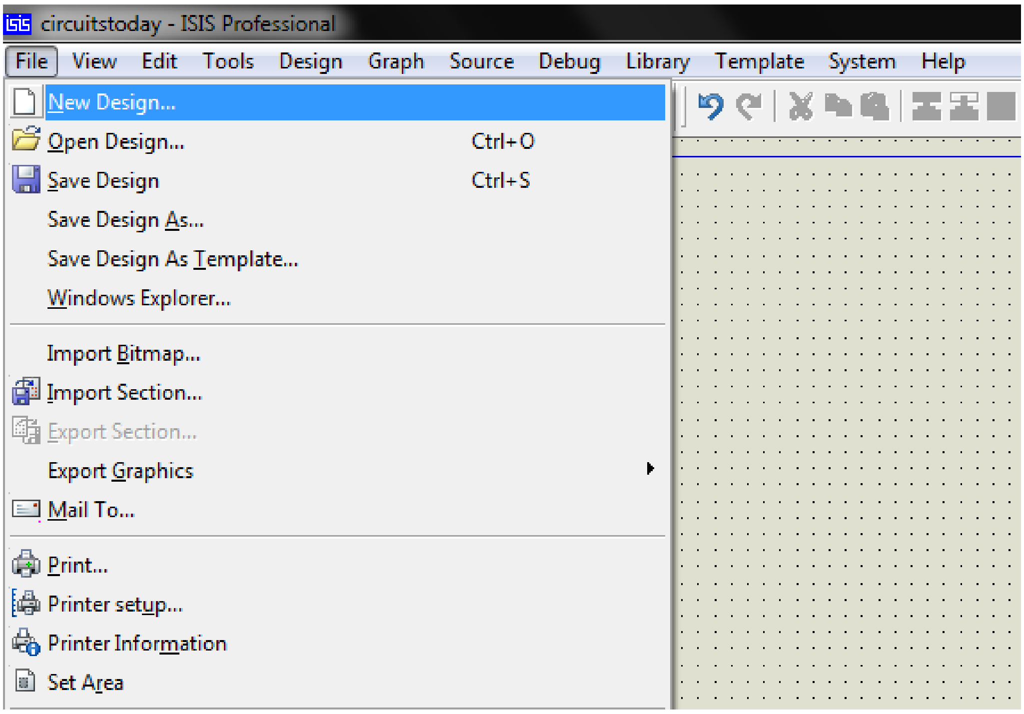


Figure 4.2.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

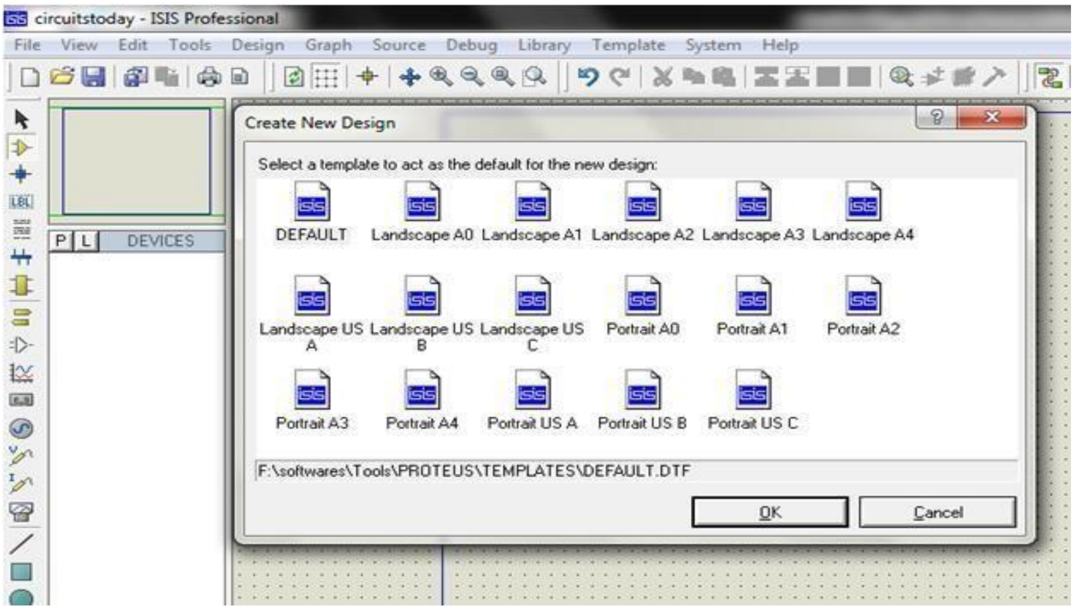


Figure 4.2.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

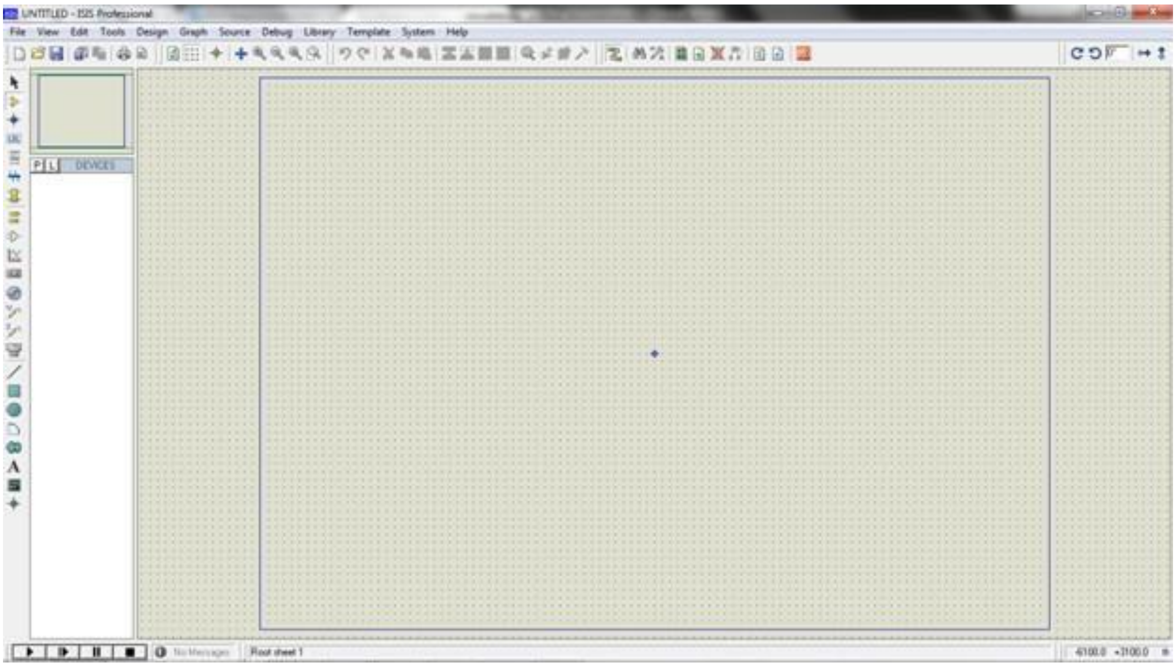


Figure 4.2.3 Proteus Design Sheet Step

4: To select components, Click on the component mode button.

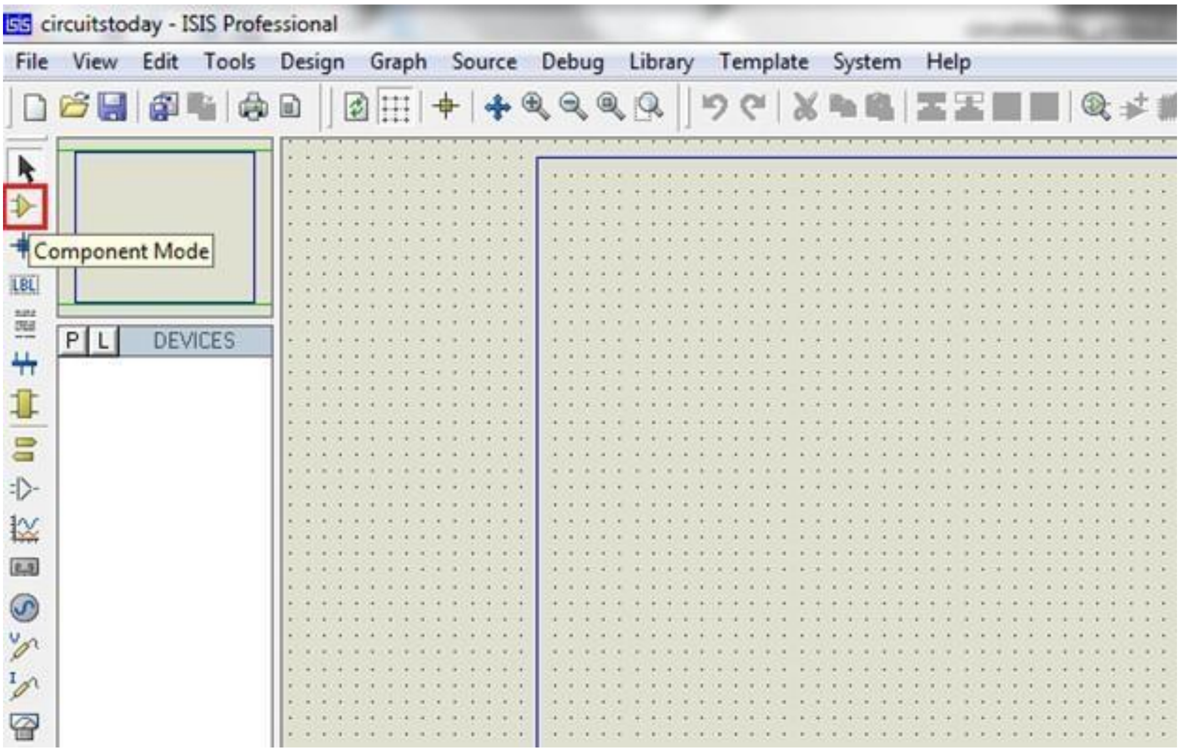


Figure 4.2.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

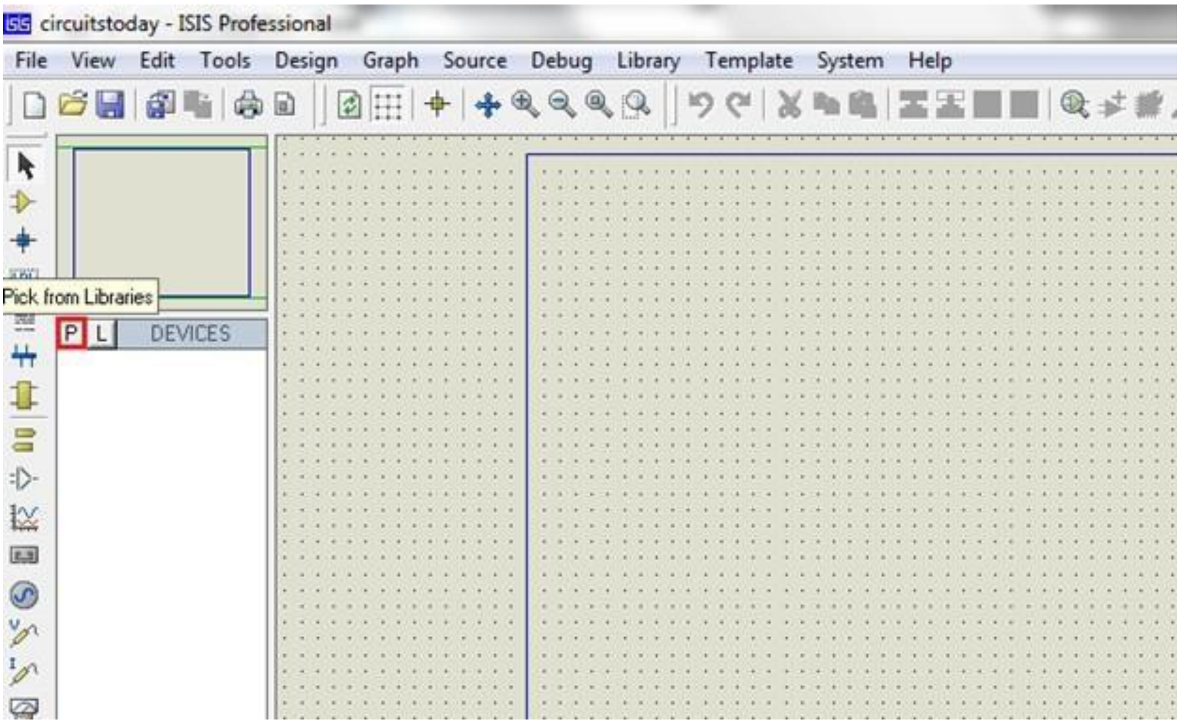


Figure 4.2.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

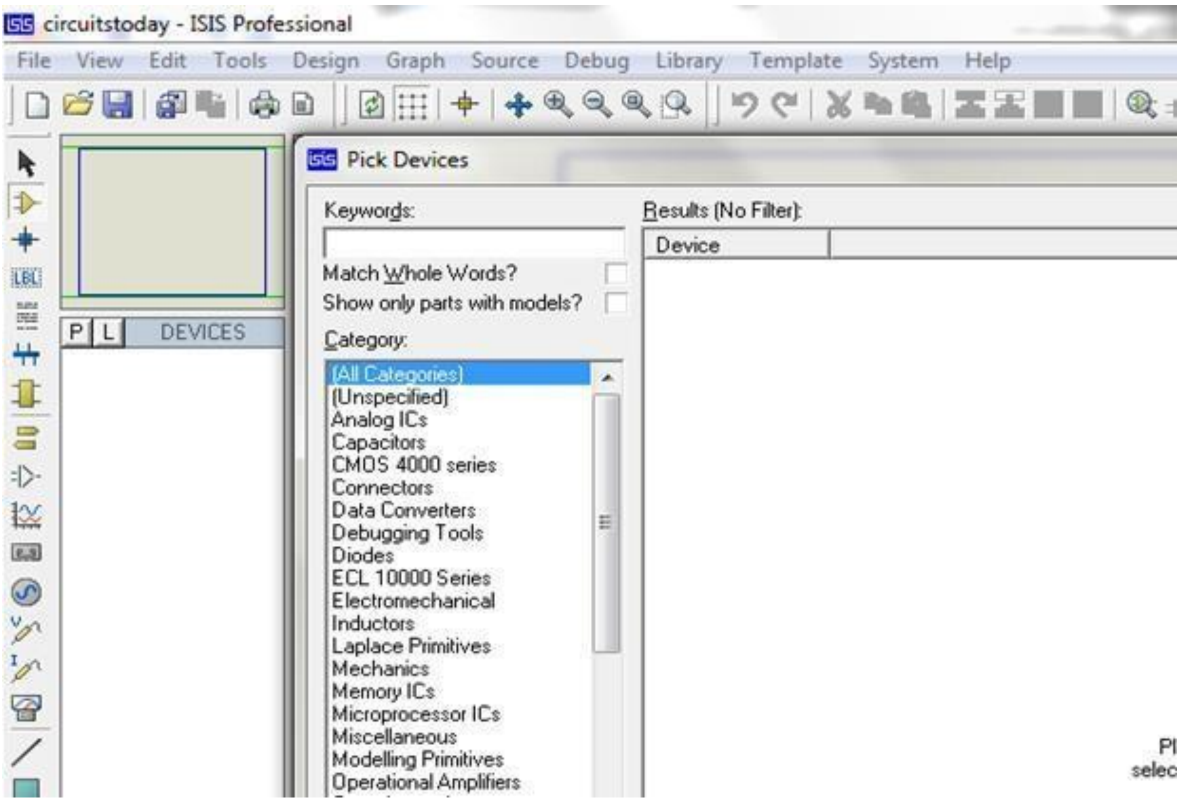


Figure 4.2.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

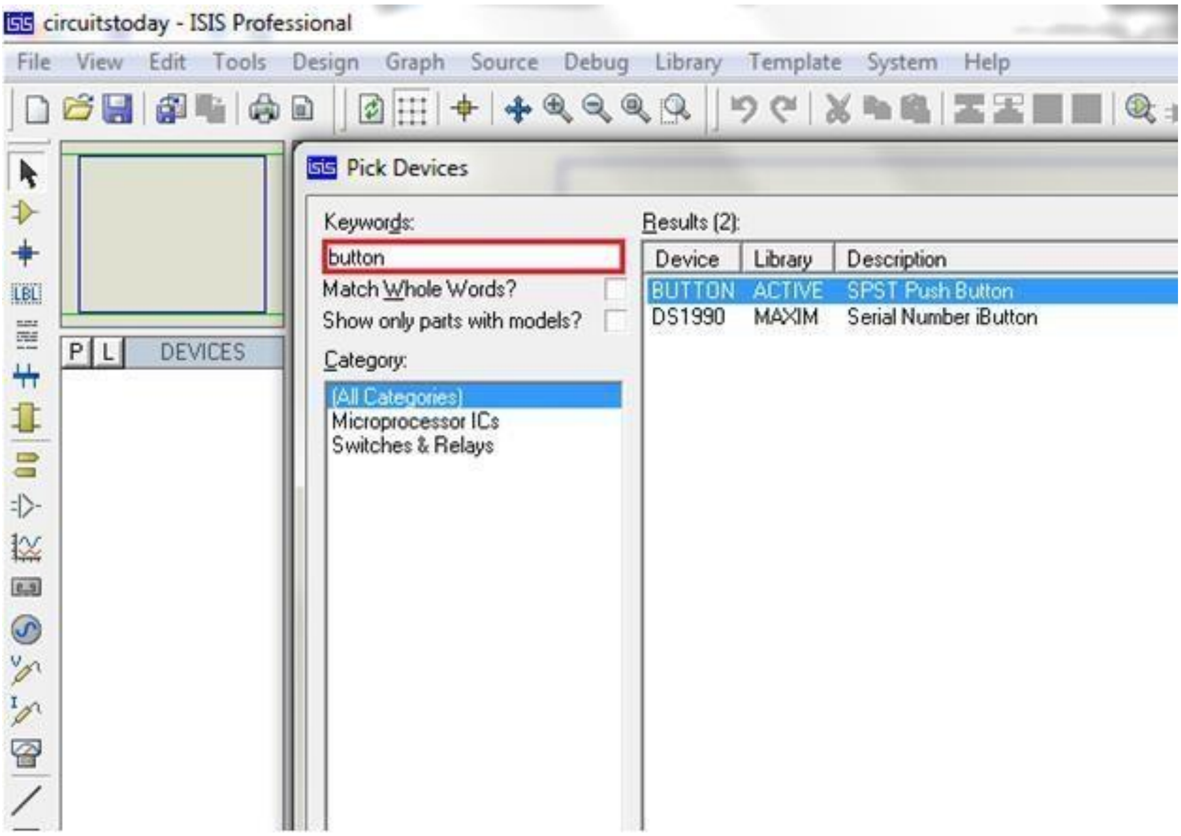


Figure 4.2.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

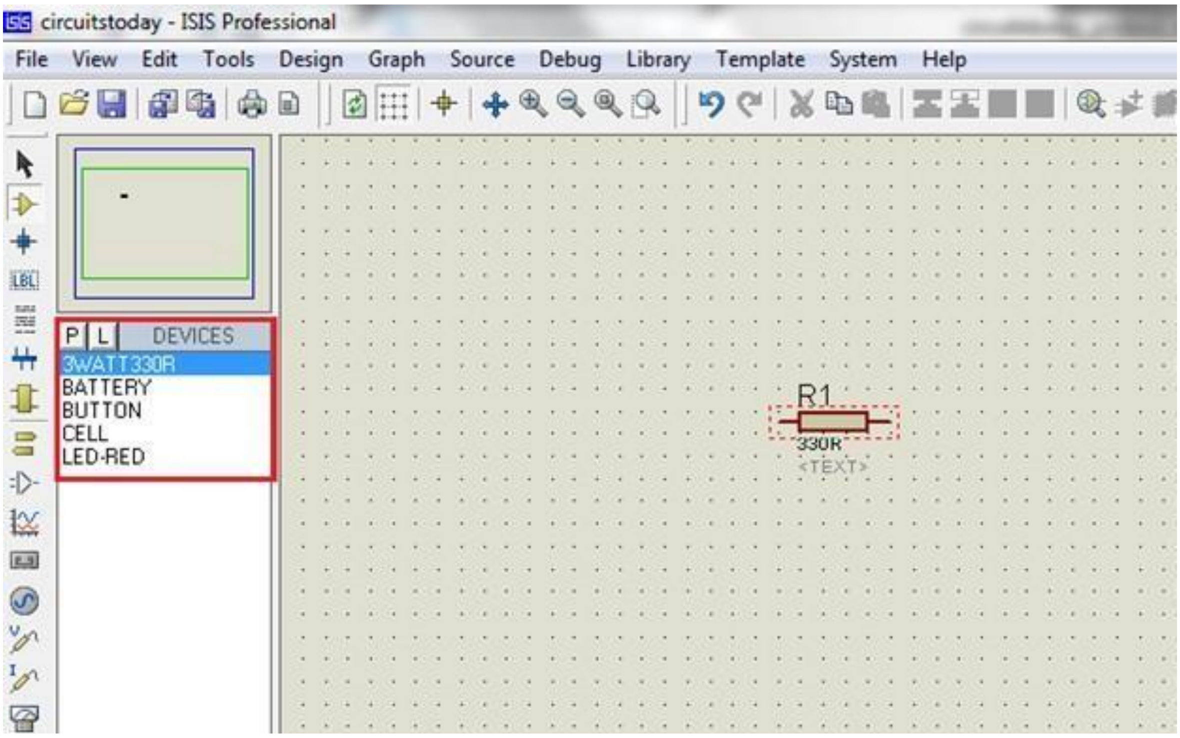


Figure 4.2.8 Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

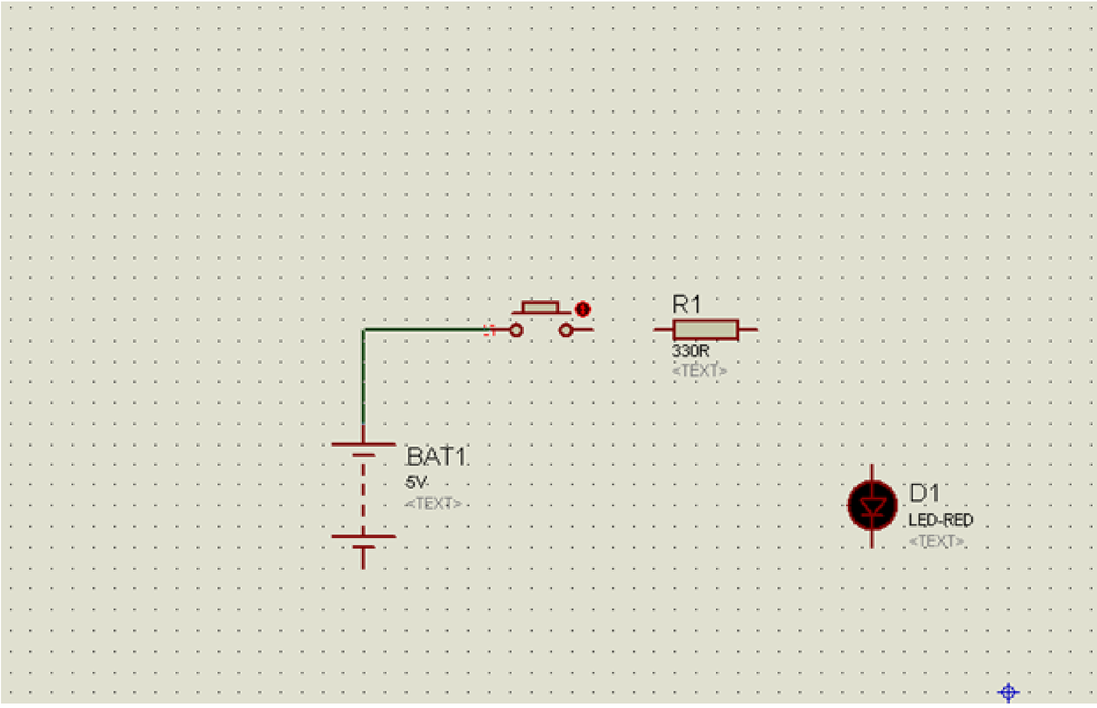


Figure 4.2.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

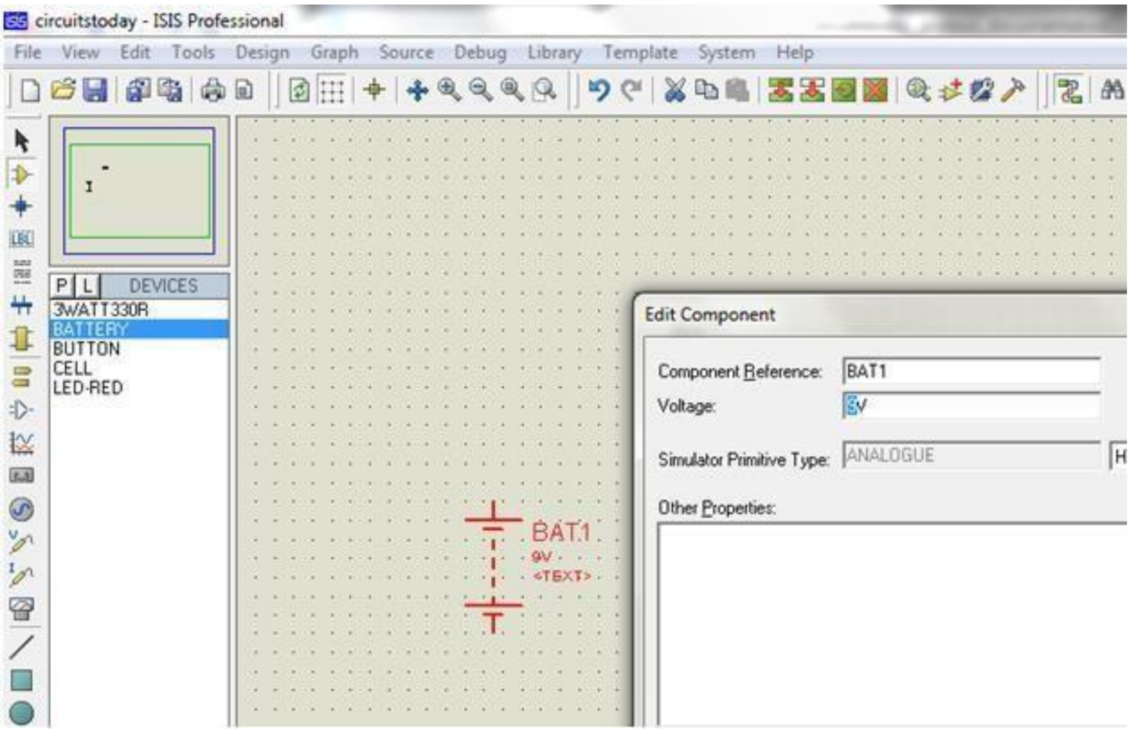


Figure 4.2.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

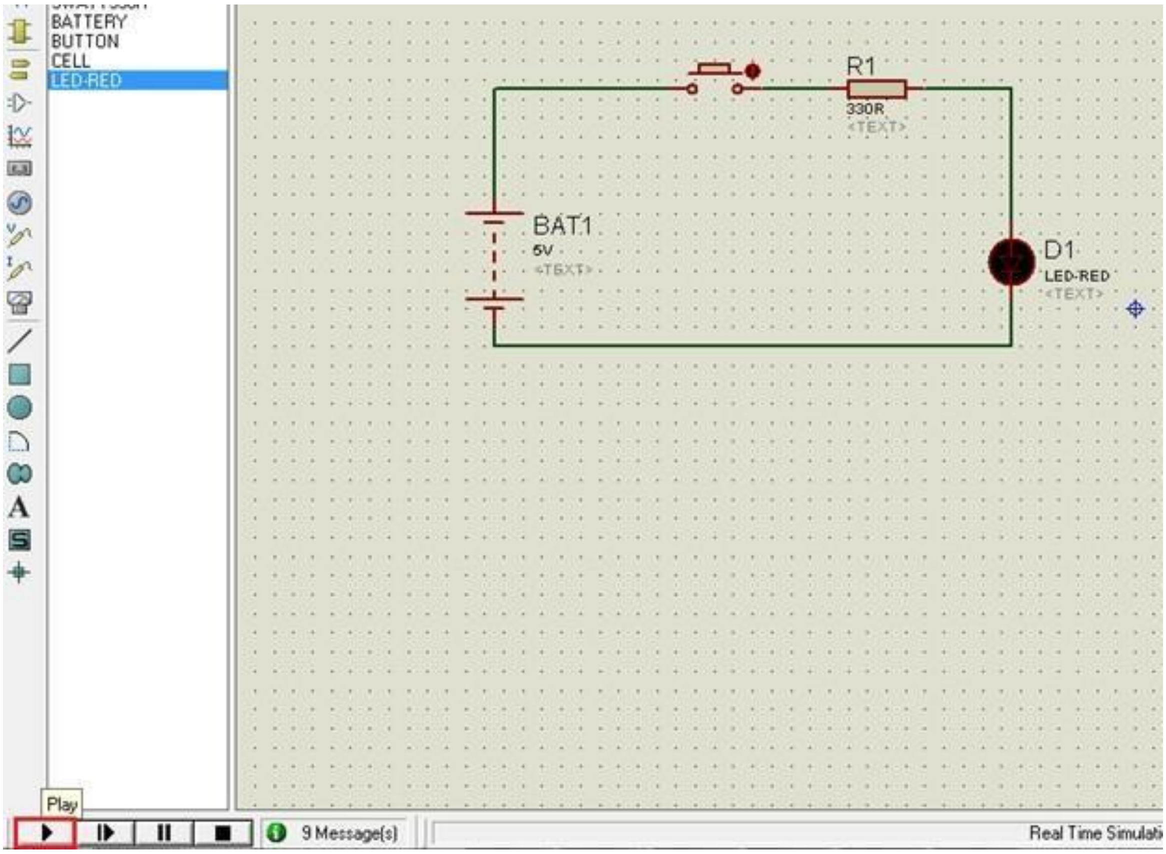


Figure 4.2.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

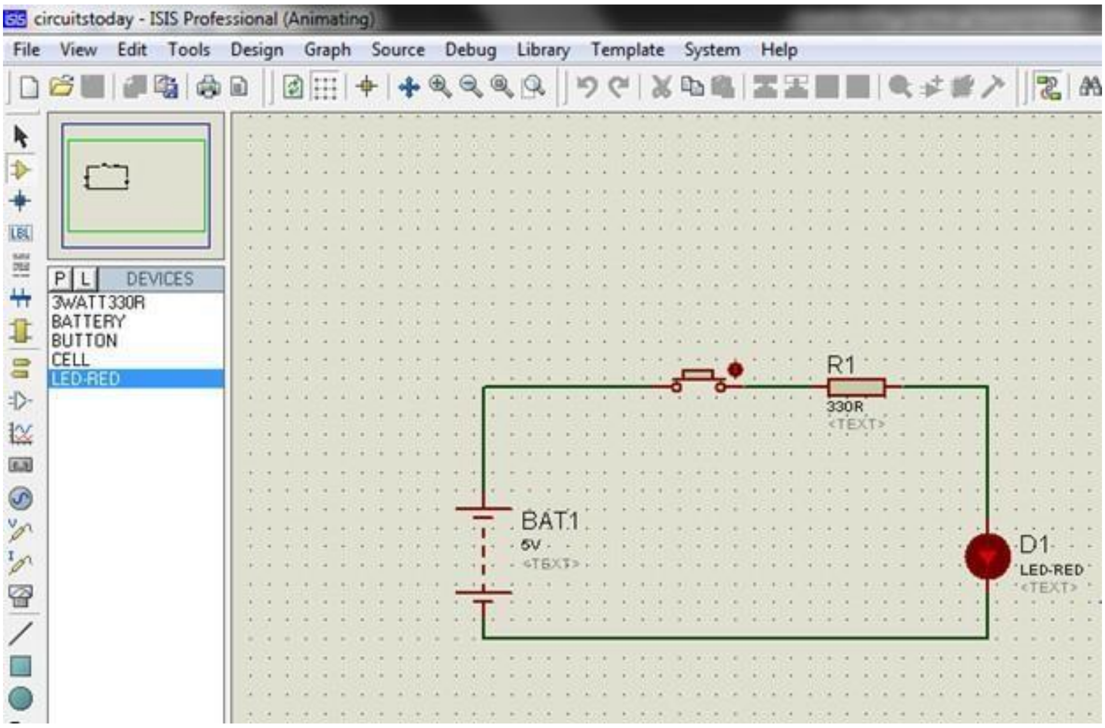


Figure 4.2.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

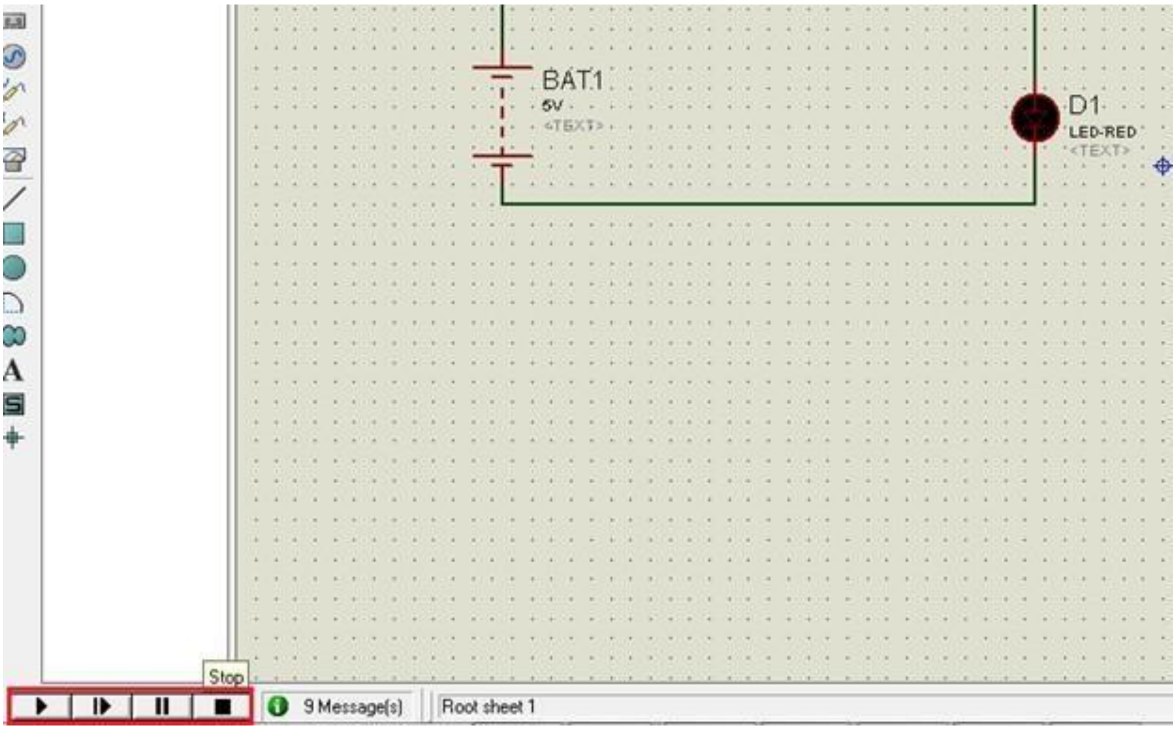


Figure 4.2.13 Simulation Step-Pause-Stop Buttons

4.3 Arduino IDE

The **Arduino Integrated Development Environment (IDE)** is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Arduino Pro IDE	
Developer(s)	Arduino Software
Preview release	v0.0.2 / 28 October 2019; 3 months ago[8]

Repository	github.com/arduino/Arduino
Written in	C, C++
Operating system	Windows, macOS, Linux
Platform	IA-32, x86-64, ARM
Type	Integrated development environment
License	LGPL or GPL license
Website	blog.arduino.cc/2019/10/18/arduino-pro-ide-alpha-preview-with-advanced-features/

In October 2019 the Arduino organization began providing early access to a new Arduino Pro IDE with debugging [9] and other advanced features.

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

4.3.1 Installation

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Figure 4.3.1 USB Cable

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 – Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

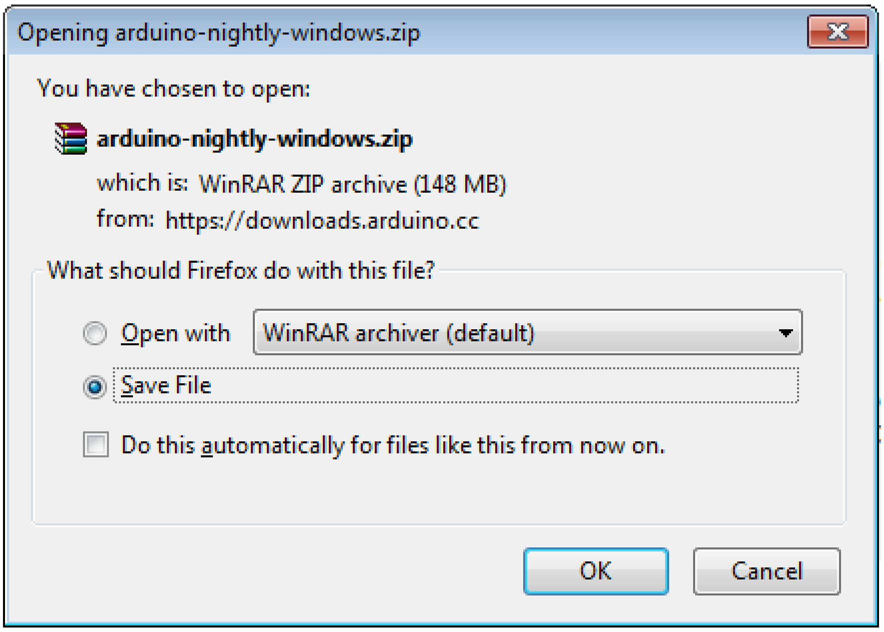


Figure 4.3.2 Downloading Arduino IDE

Step 3 – Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labelled PWR) should glow.

Step 4 – Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

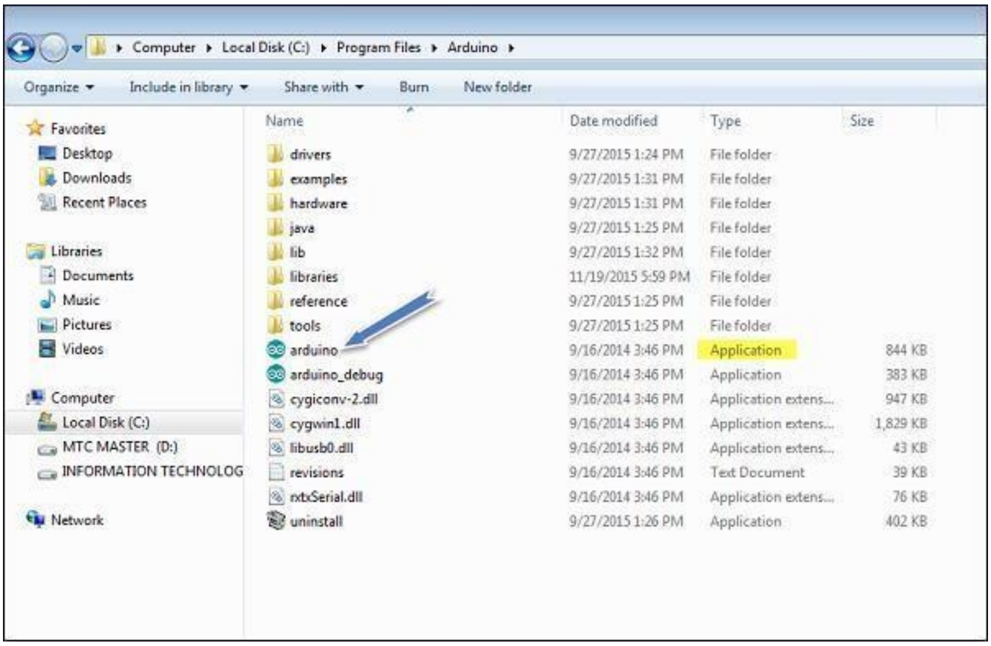


Figure 4.3.3Launching Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.

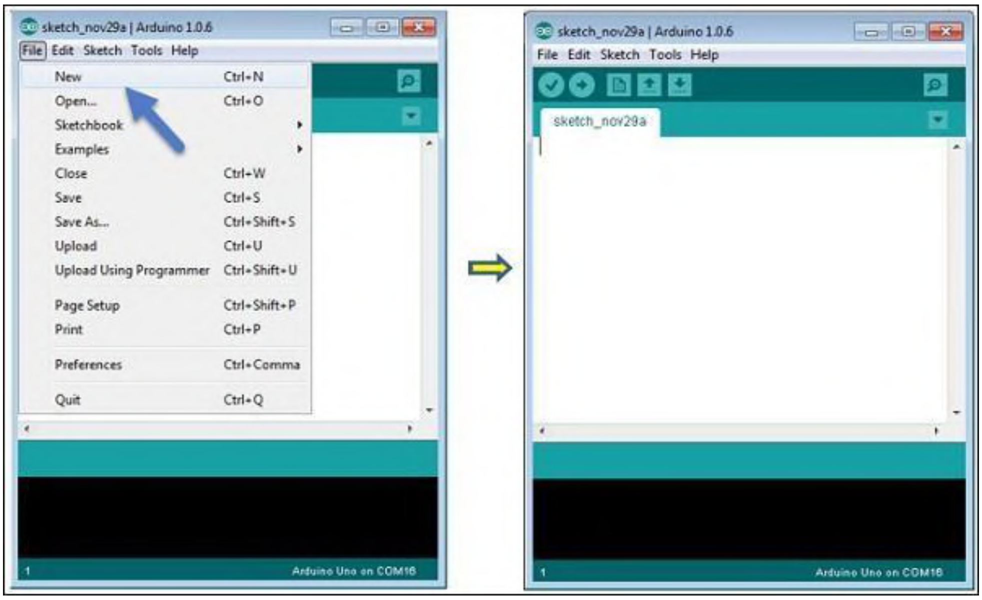


Figure 4.3.4Opening first project

To open an existing project example, select File → Example → Basics → Blink.

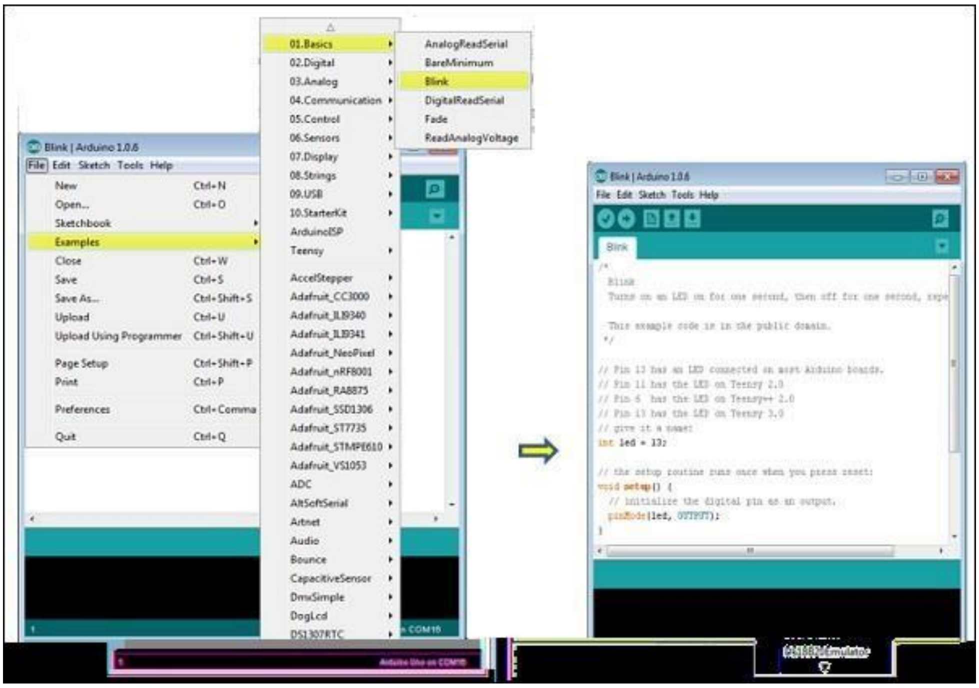


Figure 4.3.4.1Opening first project

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

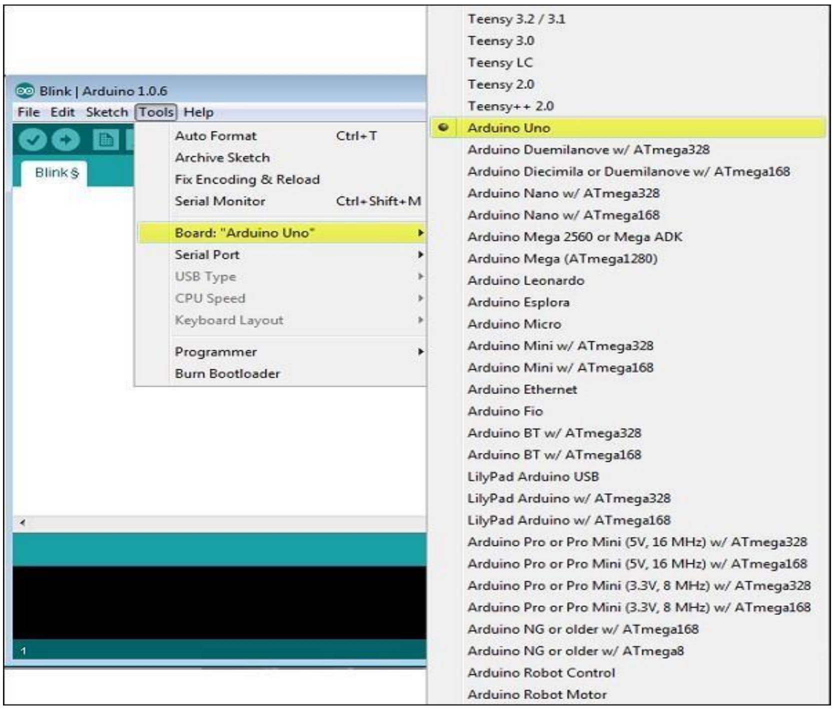


Figure 4.3.5 Selecting Arduino board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

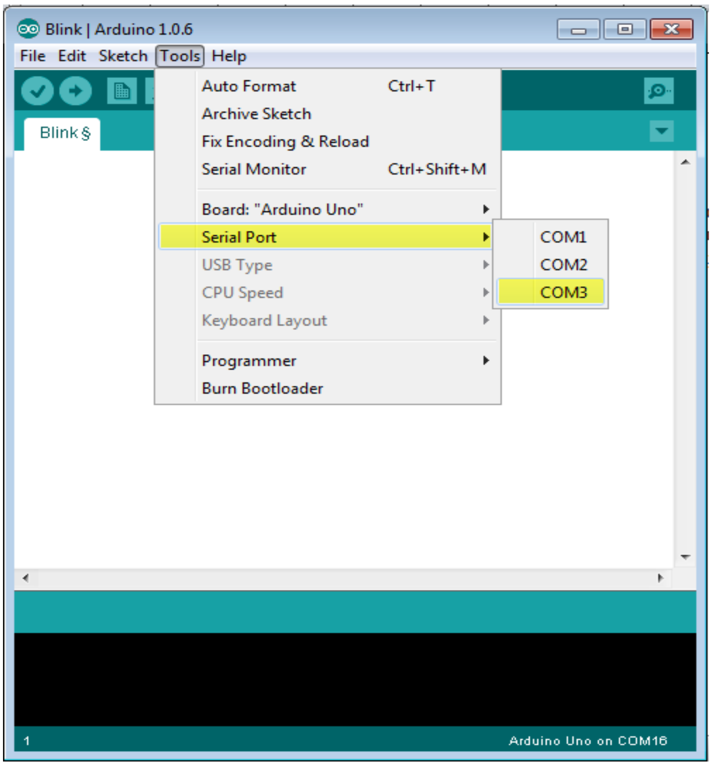


Figure 4.3.6 Selecting serial port

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

Figure: 4.3.7 Uploading the program to board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

4.4 Arduino

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

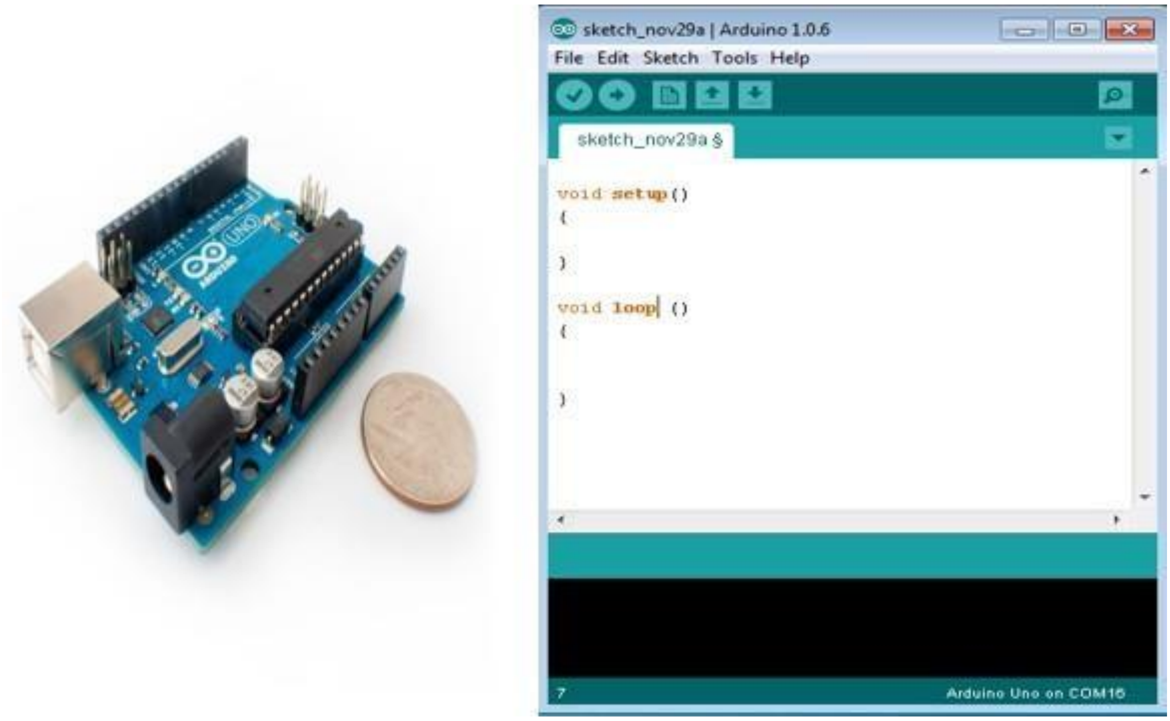


Figure 4.4 Arduino Platform

4.4.1 Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Table 4.4.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI

Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Lily Pad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Lily Pad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 4.4.2Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro	5V	16MHz	14	6	6	1	Native USB

5V/16MHz							
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
Lily Pad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

4.4.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

4.4.4 Arduino boards based on AT91SAM3X8E microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

4.5 Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

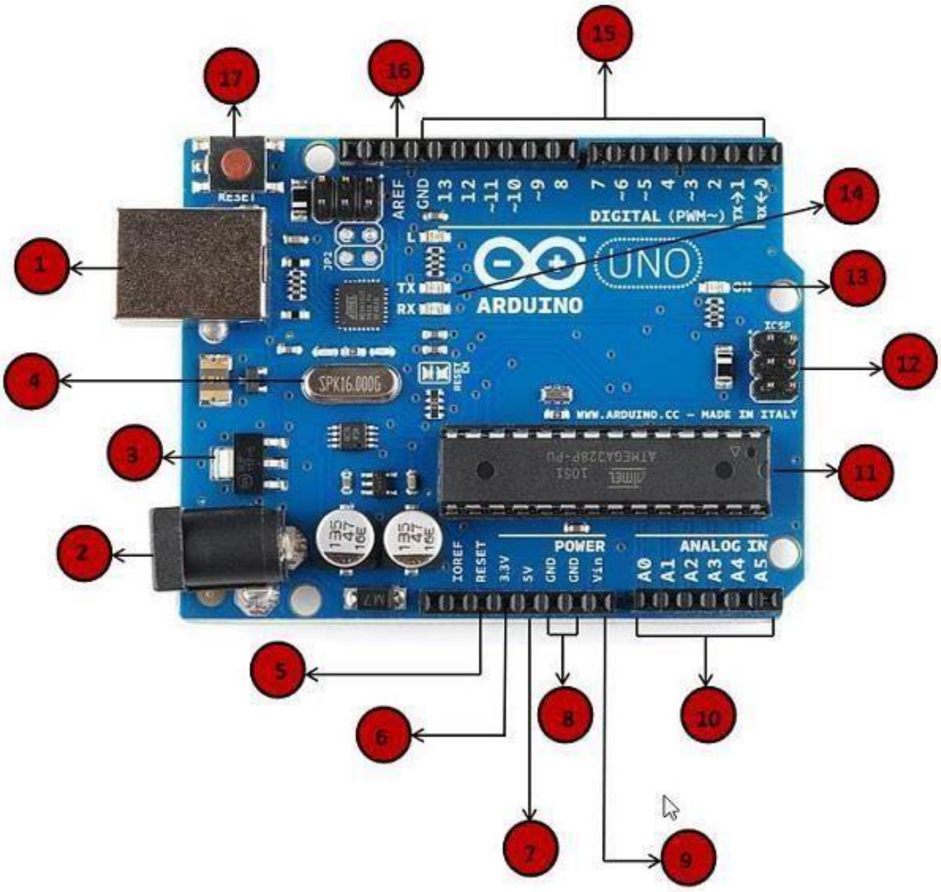


Figure: 4.5 Arduino pins

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz's

5,17	**Arduino Reset** You can reset your Arduino board, i.e., starts your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labelled "~" can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Figure: 4.5.1 Pin description

4.5 Liquid Cristal Display

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

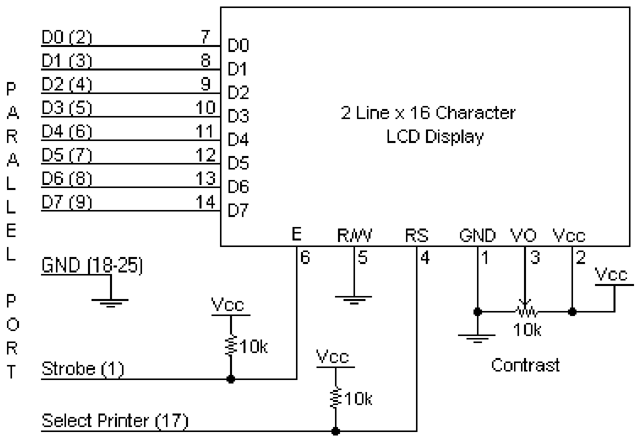
A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to a controller is an LCD display. Some of the most common LCDs connected to the controllers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines(RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

4.5.3 Description Of 16x2:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all-Parallel Ports. It however doesn't show the use of the Status Port as an input. So what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

Schematic Diagram



s

Figure 4.6.1Description Of 16x2

Above is the quite simple schematic. The LCD panel's *Enable* and *Register Select* is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there is a few which don't. Thereforeby incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wirethe *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As withall the examples, I've left the power supply out. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you havetrouble with the circuit working properly.

4.5.4 16 x 2 Alphanumeric LCD Module Features:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

4.5.5 16 x 2 Alphanumeric LCD Module Specifications:

Pin	Symbol	Level	Function
1	V_{SS}	-	Power, GND
2	V_{DD}	-	Power, 5V
3	Vo	-	Power, for LCD Drive
4	RS	H/L	Register Select Signal H: Data Input L: Instruction Input
5	R/W	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)
6	E	H,H->L	Enable
7-14	DB0-DB7	H/L	Data Bus; Software selectable 4- or 8-bit mode
15	NC	-	**NOT CONNECTED**
16	NC	-	**NOT CONNECTED**

4.5.6 FEATURES:

• 5 x 8 dots with cursor

• Built-in controller (KS 0066 or Equivalent)

• + 5V power supply (Also available for + 3V)

• 1/16 duty cycle

• B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)

• N.V. optional for + 3V power supply

Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are:

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

4.6.4.1Figure Address locations for a 1x16 line LCD

Even limited to character based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8,16,20,24,32 and 40 characters are all standard, in one, two and four line versions.

Several different LC technologies exist. “Super twist” types, for example, offer improved contrast and viewing angle over the older “twisted nematic” types. Some modules are available with back lighting, so that they can be viewed in dimly-lit conditions. The back lighting may be either “electro-luminescent”, requiring a high voltage inverter circuit, or simple LED illumination.

4.5.7 PIN DESCRIPTION:

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections).

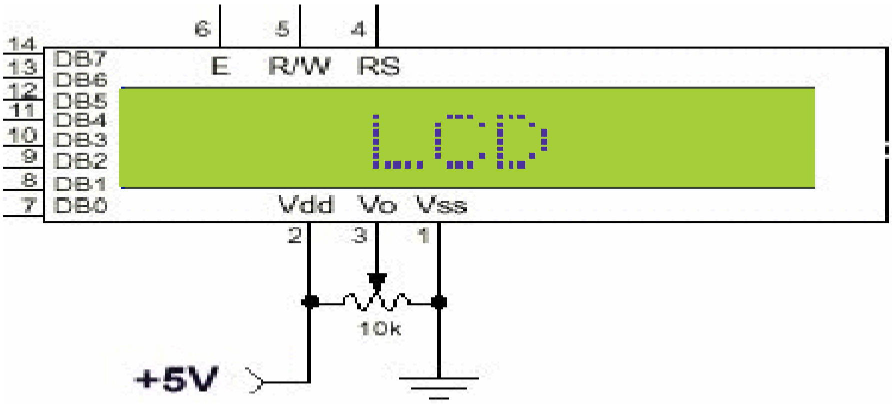


Figure 4.6.5 Pin diagram of 1x16 lines LCD

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Figure 4.6.6 Pin specifications

4.6 CONTROL LINES:

EN: Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS: Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW: Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands, so RW will almost always be low. Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Logic status on control lines:

• E - 0 Access to LCD disabled

- 1 Access to LCD enabled

• R/W - 0 Writing data to LCD

- 1 Reading data from LCD

• RS - 0 Instructions

- 1 Character

Writing data to the LCD:

1) Set R/W bit to low
2) Set RS bit to logic 0 or 1 (instruction or character)
3) Set data to data lines (if it is writing)
4) Set E line to high
5) Set E line to low

Read data from data lines (if it is reading) on LCD:

1) Set R/W bit to high
2) Set RS bit to logic 0 or 1 (instruction or character)
3) Set data to data lines (if it is writing)
4) Set E line to high
5) Set E line to low

Entering Text:

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the d.i.l. switch pack with a of re-wiring is necessary.

4.6.3 LCD Commands:

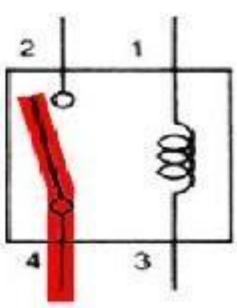
There are some present commands instructions in LCD, which we need to send to LCD through some microcontroller. Some important command instructions are given below:

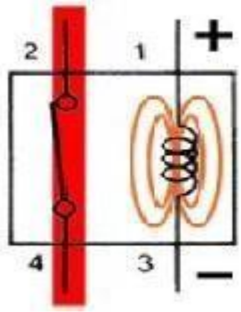
Table 4.7.1 Commands

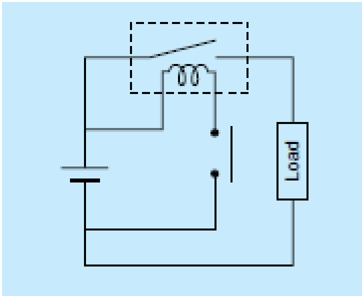
Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left
0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

4.7 Relays

A relay is an electrically operated switch. These are remote control electrical switches that are controlled by another switch, such as a horn switch or a computer as in a power train control module, devices in industries, home based applications. Relays allow a small current pin, 4-pin, 5-pin, and 6-pin, single switch or dual switches. Relays are used throughout the automobile. Relays which come in assorted sizes, ratings, and applications, are used as remote control switches. A typical vehicle can have 20 relays or more.







4.7.3 Basics on Relay Handling

- To maintain initial performance, care should be taken to avoid dropping or hitting the relay.
- Under normal use, the relay is designed so that the case will not detach. To maintain initial performance, the case should not be removed. Relay characteristics cannot be guaranteed ifthe case is removed.
- Use of the relay in an atmosphere at standard temperature and humidity with minimal amounts of dust, SO 2, H 2 S, or organic gases is recommended.
- Please avoid the use of silicon-based resins near the relay, because doing so may result in contact failure. (This applies to plastic sealed type relays, too.)
- Care should be taken to observe correct coil polarity (+, –) for polarized relays.
- Proper usage requires that the rated voltage be impressed on the coil. Use rectangular wavesfor DC coils and sine waves for AC coils.
- Be sure the coil impressed voltage does not continuously exceed the maximum allowablevoltage.
- Absolutely avoid using switching voltages and currents that exceed the designated values.
- The rated switching power and life are given only as guides. The physical phenomena at the contacts and contact life greatly vary depending on the type of load and the operatingconditions. Therefore, be sure to carefully check the
- Type of load and operating conditions before use.
- Do not exceed the usable ambient temperature values listed in the catalog.
- Use the flux-resistant type or sealed type if automatic soldering is to be used.
- Use alcohol based cleaning solvents when cleaning is to be performed using a sealed type relay.
- Avoid ultrasonic cleaning of all types of relays.
- Avoid bending terminals, because it may cause malfunction.
- As a guide, use a Fasten mounting pressure of 40 to 70N {4 to 7kgf}for relays with tab terminals.

7.1.1 Specification

- Number and type of contacts – normally open, normally closed, (double-throw)
- Contact sequence – "Make before Break" or "Break before Make". For example, the old style telephone exchanges required Make-before-break so that the connection didn't get dropped while dialing the number.
- Rating of contacts – small relays switch a few amperes, large contactors are rated for up to 3000 amperes, alternating or direct current
- Voltage rating of contacts – typical control relays rated 300 VAC or 600 VAC, automotive types to 50 VDC, special high-voltage relays to about 15 000 V
- Coil voltage – machine-tool relays usually 24 VAC, 120 or 250 VAC, relays for switchgear may have 125 V or 250 VDC coils, "sensitive" relays operate on a few milli-amperes

7.1.2 Applications:

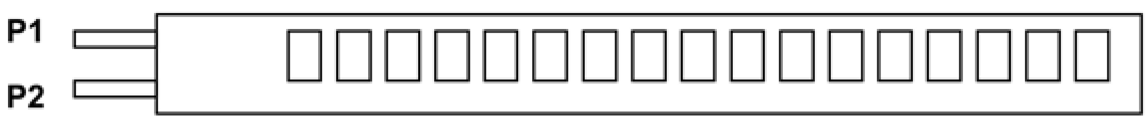
Relays are used:

- To control a high-voltage circuit with a low-voltage signal, as in some types of modems,
- To control a high-current circuit with a low-current signal, as in the starter solenoid of an automobile,
- To detect and isolate faults on transmission and distribution lines by opening and closing circuit breakers (protection relays),
- To isolate the controlling circuit from the controlled circuit when the two are at different potentials, for example when controlling a mains-powered device from a low-voltage switch. The latter is often applied to control office lighting as the low voltage wires are easily installed in partitions, which may be often moved as needs change. They may also be controlled by room occupancy detectors in an effort to conserve energy,
- To perform logic functions. For example, the Boolean AND function is realized by connecting relay contacts in series, the OR function by connecting contacts in parallel. Due to the failure modes of a relay compared with a semiconductor, they are widely used in safety critical logic, such as the control panels of radioactive waste handling machinery.

- As oscillators, also called vibrators. The coil is wired in series with the normally closed contacts. When a current is passed through the relay coil, the relay operates and opens the contacts that carry the supply current. This stops the current and causes the contacts to close again. The cycle repeats continuously, causing the relay to open and close rapidly. Vibrators are used to generate pulsed current.
- To generate sound. A vibrator, described above, creates a buzzing sound because of the rapid oscillation of the armature. This is the basis of the electric bell, which consists of a vibrator with a hammer attached to the armature so it can repeatedly strike a bell.
- To perform time delay functions. Relays can be used to act as a mechanical time delay device by controlling the release time by using the effect of residual magnetism by means of a inserting copper disk between the armature and moving blade assembly.

Figure 4.8.3 Relay

7.2 Flex Sensor



Flex Sensor   Flex Sensor Pin out

Flex sensors are usually available in two sizes. One is **2.2 inch** and another is **4.5 inch**. Although the sizes are different the basic function remains the same. They are also divided based on resistance. There are LOW resistance, MEDIUM resistance and HIGH resistance types. Choose the appropriate type depending on requirement. Here we are going to discuss 2.2inch Flex sensor that is **FS-L-0055**.

7.1.3 FLEX SENSOR WORKING

This sensor works on the bending strip principle which means whenever the strip is twisted then its resistance will be changed. This can be measured with the help of any controller.

This sensor works similar to a variable resistance because when it twists then the resistance will be changed. The resistance change can depend on the linearity of the surface because the resistance will be dissimilar when it is level.

When the sensor is twisted 450 then the resistance would be dissimilar. Similarly, when this senor is twisted to 900 then the resistance would be dissimilar. These three are the flex sensor's bending conditions.

According to these three cases, the resistance will be normal in the first case, the resistance will be double as contrasted with the first case, and the resistance will be four-time when compared with the first case. So the resistance will be increased when the angle is increased

7.1.4 FLEX SENSOR Pin Configuration

Flex sensor is a two-terminal device. The Flex sensor does not have polarized terminals like diode. So, there is no positive and negative.

Pin Number	**Description**
P1	Usually connected to positive of power source.
P2	Usually connected to ground.

7.1.5 FLEX SENSOR Features and Specifications

- Operating voltage of FLEX SENSOR: 0-5V
- Can operate on LOW voltages
- Power rating: 0.5Watt (continuous), 1 Watt (peak)
- Life: 1 million
- Operating temperature: -45ºC to +80ºC
- Flat Resistance: 25K Ω
- Resistance Tolerance: ±30%
- Bend Resistance Range: 45K to 125K Ohms (depending on bend)

FS-L-0055 Equivalents

FS-L-0095, FS-L-0112, etc

7.1.6Where to Use FLEX SENSOR

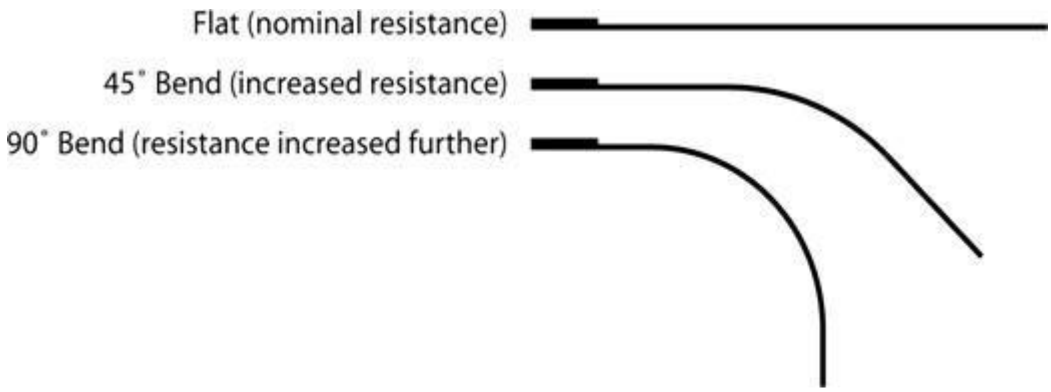
For understanding the use of FLEX SENSOR considers:

Case1: Where you want to check whether the surface of a device or thing is levelled or not. Say you want a device to check whether a window or door is open or not. At that time a Flex sensor could be used. The sensor could be fixed at door edge and when the door opens the Flex sensor gets flexed. With the sensor being flexed its parameters changes which could be designed to provide an alert.

Case2: Where you want to measure the FLEX or BENT or ANGLE change of any instrument or device. The FLEX SENSOR internal resistance changes almost linearly with its flex angle. So by sticking the sensor to the instrument we can have the flex angle in electrical parameter of resistance.

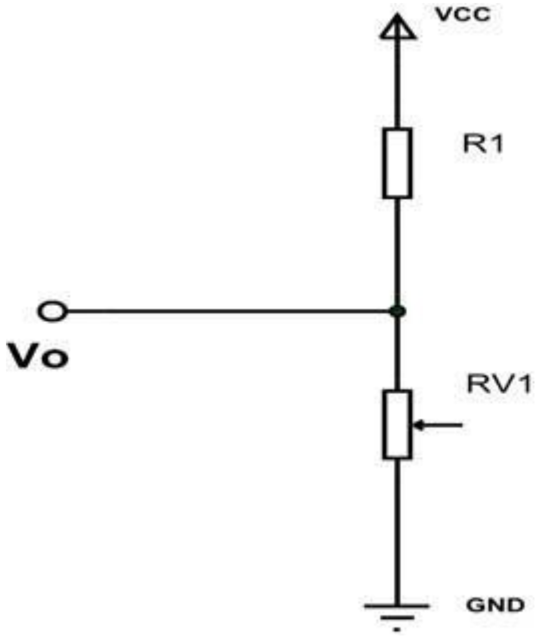
7.1.7 How to Use FLEX SENSOR

As mentioned earlier, **FLEX SENSOR** is basically a **VARIABLE RESISTOR** whose terminal resistance increases when the sensor is bent. So this sensor resistance increases depends on surface linearity. So it is usually used to sense the changes in linearity.



As shown above figure, when the surface of FLEX SENSOR is completely linear it will be having its nominal resistance. When it is bent 45º angle the FLEX SENSOR resistance increases to twice as before. And when the bent is 90º the resistance could go as high as four times the nominal resistance. So the resistance across the terminals rises linearly with bent angle. So in a sense the FLEX sensor converts flex angle to RESISTANCE parameter.

For convenience we convert this RESISTANCE parameter to VOLTAGE parameter. For that we are going to use **VOLTAGE DIVIDER circuit**. A typical VOLTAGE DIVIDER circuit is shown below.



In this resistive network we have two resistances. One is constant resistance (R1) and other is variable resistance (RV1). Vo is the voltage at midpoint of VOLTAGE DIVIDER circuit and is also the output voltage. Vo is also the voltage across the variable resistance (RV1). So when the resistance value of RV1 is changed the output voltage Vo also changes. So we will have resistance change in voltage change with VOLTAGE DIVIDER circuit.

Here we will replace the variable resistance (RV1) with FLEX SENSOR. The circuit will be as below.

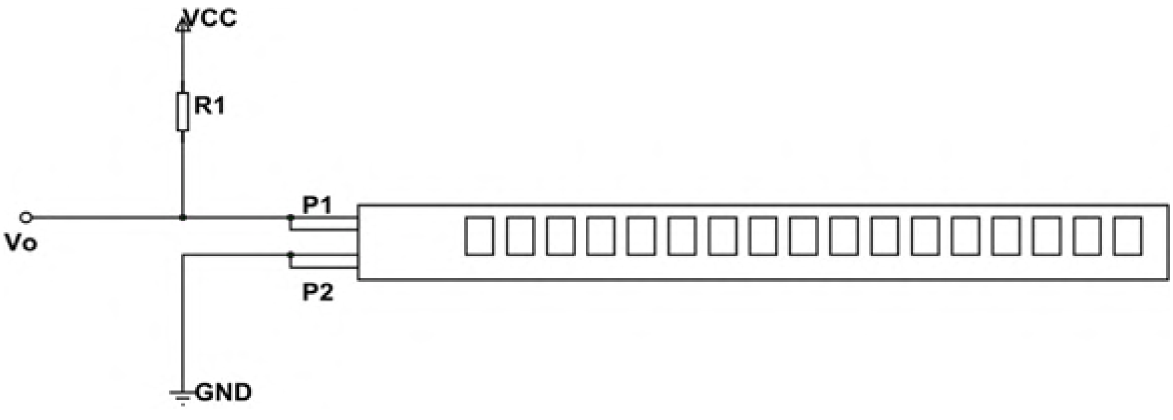


Figure: 4.9.5.1 Flex sensor

As shown in figure, R1 here is a constant resistance and **FLEX SENSOR** which acts as a variable resistance. Vo being output voltage and also the voltage across the FLEX SENSOR.

Here,

Vo = VCC (Rx/ (R1+Rx)).

Rx - FLEX SENSOR resistance

Now, when the FLEX SENSOR is bent the terminal resistance increases. This increase also appears in VOLTAGE DIVIDER circuit. With that the drop across the FLEX SENSOR increases so is Vo. So with increase in bent of FLEX sensor Vo voltage increases linearly. With that we have VOLTAGE parameter representing the flex.

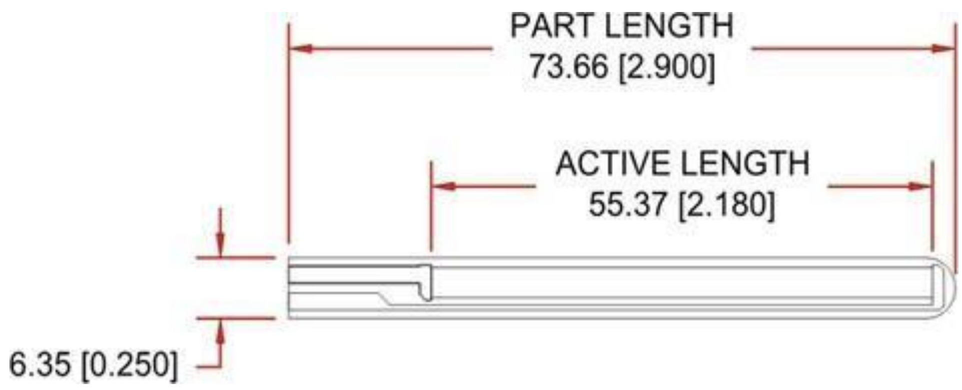
We can take this VOLTAGE parameter and feed it to ADC to get the digital value which can be used conveniently.

7.1.8 Applications

- Robotics
- Gaming (Virtual Motion)
- Medical Devices
- Computer Peripherals
- Musical Instruments
- Physical Therapy

2D-Model

Measurements in millimetre (inches)



7.2 POWER SUPPLY

7.2.1 Block Diagram

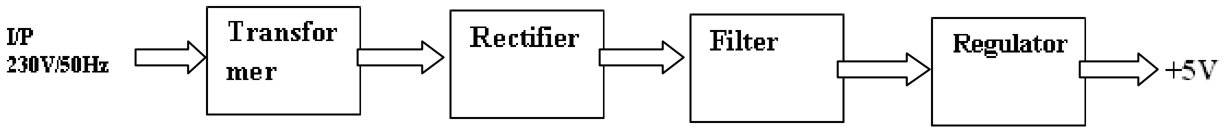
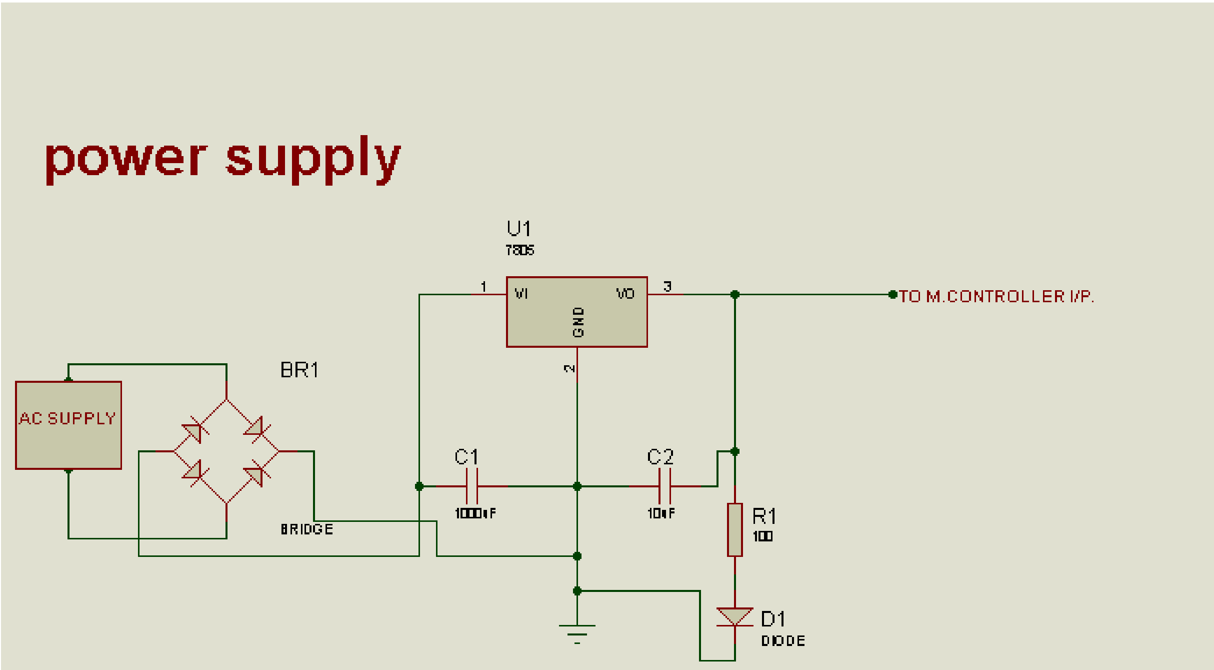


Figure 4.10.1 Power Supply

7.2.2 Circuit Diagram



7.2.3 Description

Transformer

A **transformer** is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

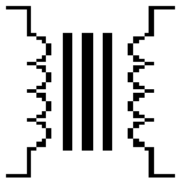


Figure: 4.10.3.1 Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

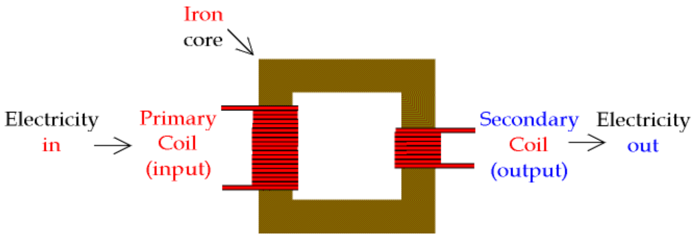


Figure 4.10.3.2 Transformer

7.2.4 Basic Principle:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

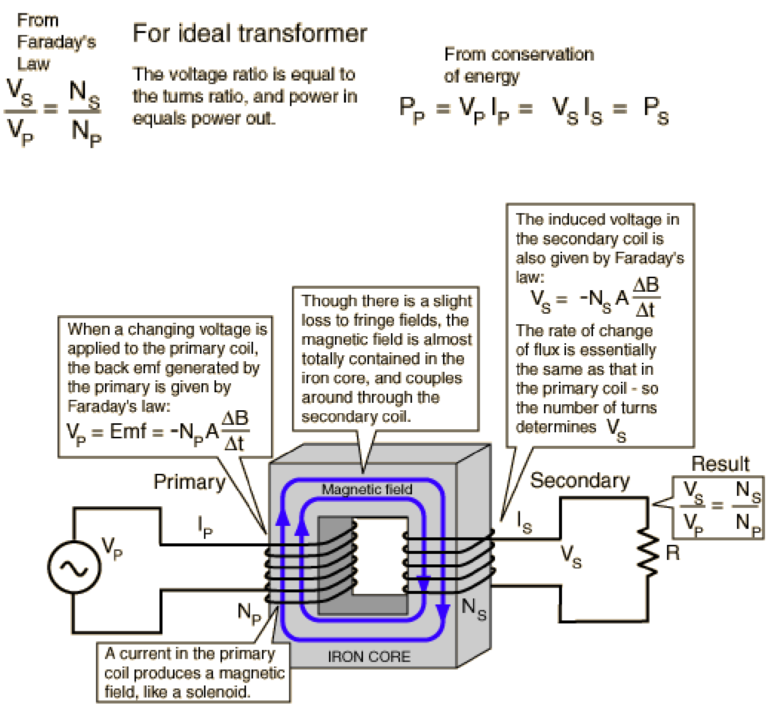


Figure: 4.10.4 Basic Principle

7.2.5 Transformer Working

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

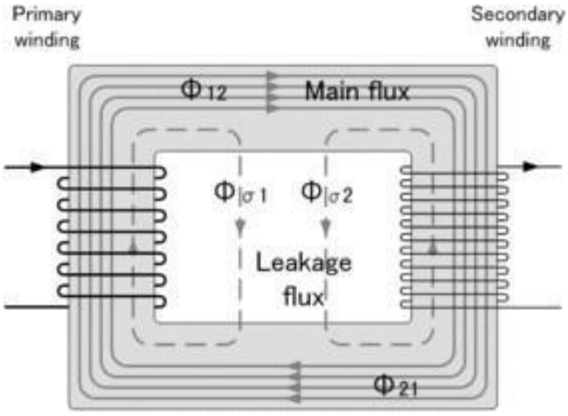


Figure: 4.10.5 Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f.

The iron core is laminated to prevent 'eddy currents' flowing in the core. These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

Classification of Transformer:

- Step-Up Transformer
- Step-Down Transformer

Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn’s ratio of the two sets of windings determines the amount of voltage transformation.

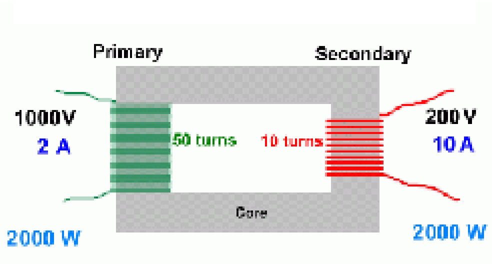


Figure: 4.10.6 Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn’s ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn’s ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single-phase step-down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase step up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

Step-Up Transformer:

A step-up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply.

A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

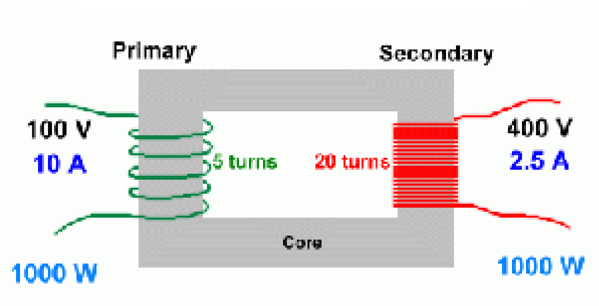


Figure: 4.10.6.1 Step-Up Transformer

7.3 Applications

Generally these **Step-Up Transformers** are used in industries applications only.

Turns Ratio and Voltage

The ratio of the number of turns on the primary and secondary coils determines the ratio of the voltages...

$$\frac{V_s}{V_p} = \frac{N_s}{N_p}$$

...where V_p is the primary (input) voltage, V_s is the secondary (output) voltage, N_p is the number of turns on the primary coil, and N_s is the number of turns on the secondary coil.

7.3.1 Diodes

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

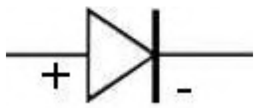


Figure 4.11.1 Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

7.3.2 Rectifier

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

The Half-wave Rectifier

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure.

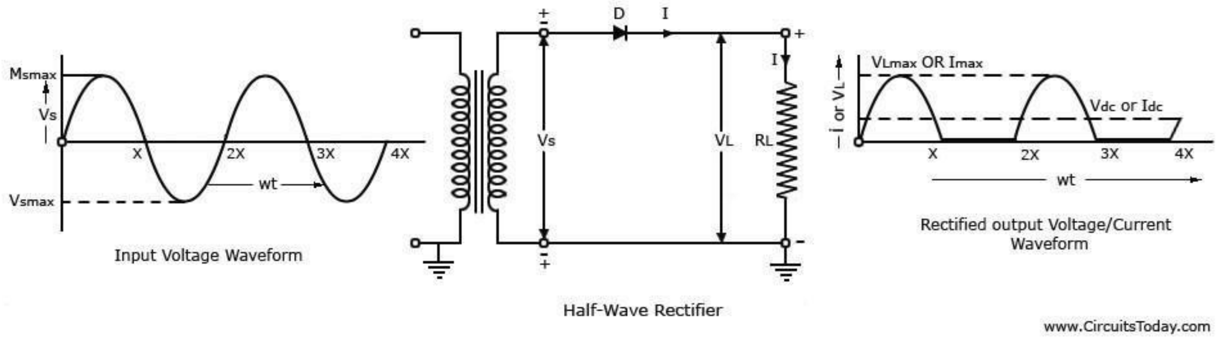


Figure: 4.11.2 Half Wave Rectifier

Figure 4.9.2 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is 0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

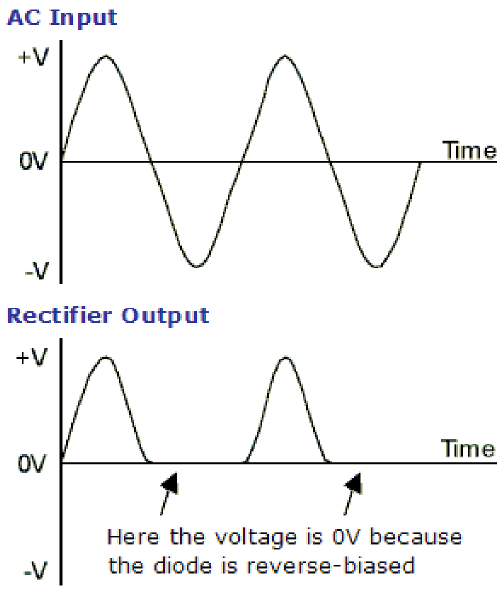


Figure: 4.11.2.1 Half-Wave Rectification

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

The Full-wave Rectifier

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC. The resulting waveform is shown in figure 4.

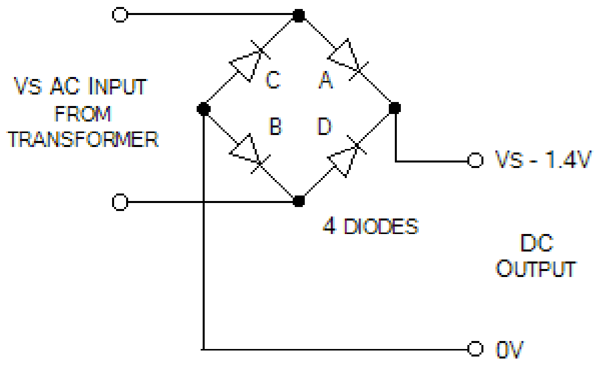


Figure: 4.11.2.2 Full-Wave Rectifier

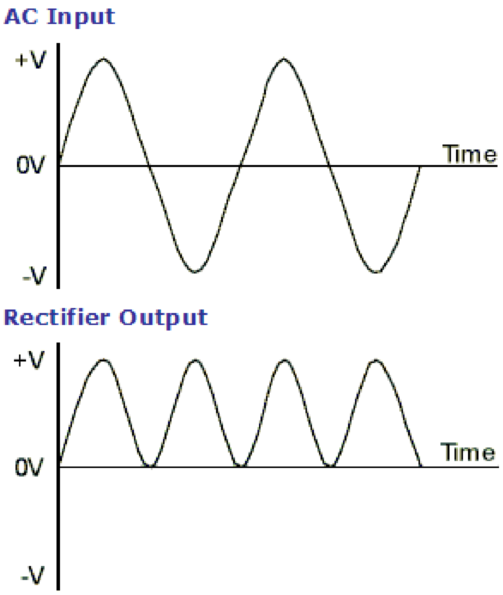


Figure: 4.11.2.3 Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased.

While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

7.4 Capacitor Filter

The **capacitor-input filter**, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

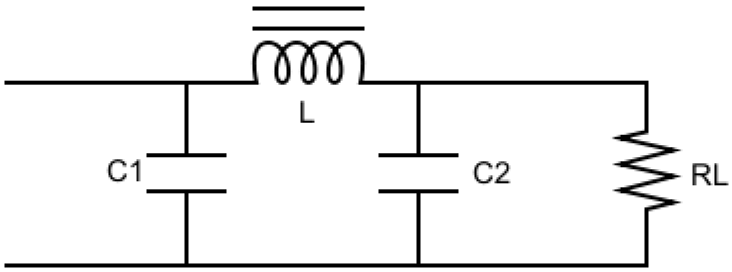


Figure: 4.12.1 Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result only the DC component appears across the load RL.

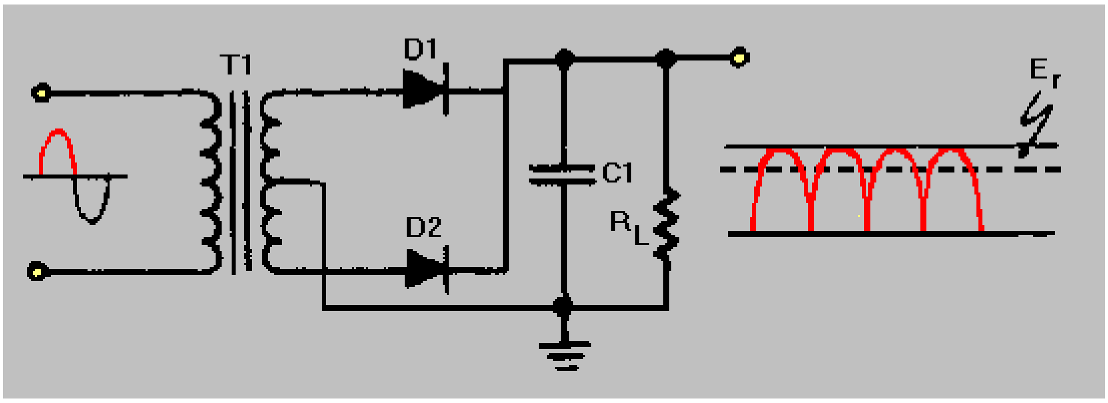


Figure: 4.12.2Centered Tapped Full-Wave Rectifier with a Capacitor Filter

7.5 Voltage Regulator:

A **voltage regulator** is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:’78’ indicate the positive series and ‘xx’indicates the voltage rating. Suppose 7805 produces the maximum 5V.’05’indicates the regulator output is 5V.

79xx:’78’ indicate the negative series and ‘xx’indicates the voltage rating. Suppose 7905 produces the maximum -5V.’05’indicates the regulator output is -5V.

These regulators consists the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

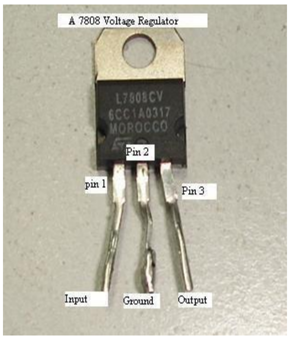


Figure: 4.13 Regulator

CHAPTER

5RESULTS

5.1 RESULTS

The total prototype is shown in the above figure. The microcontroller used in this project is Arduino UNO, it acts as the bridge between the input and the output. The input is taken according to the bend of the flex sensors. The output is seen through the LCD and a audio output is heard through the speaker and the APR module helps to decide which audio should be played according to the digital output given by the Arduino. The relay module which is kept is used for the ON, OFF and the reset the whole circuit.

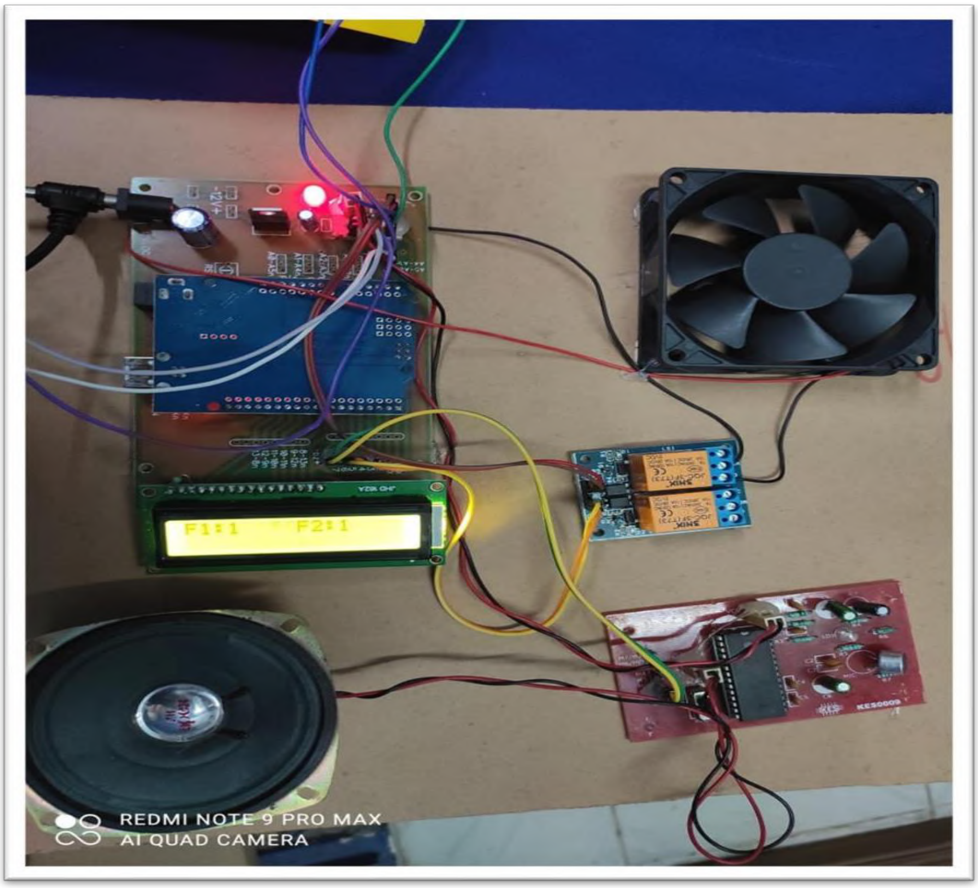


Fig: 5.1 prototype

5.1.1 RESULT

Fig 5.1.1 : Output when flex sensor 1 is bent

5.1.2 RESULT

Now again bring the first sensor to the normal position and check whether the output of both flex sensors are showing 0 as output, now bend the second flex sensor placed on Mitten and now the output changes to 1 at the flex sensor 2 and it shows as TV ON on the LCD and an audio output can be heard from the speaker. The output on the LCD can be seen in the below Figure.

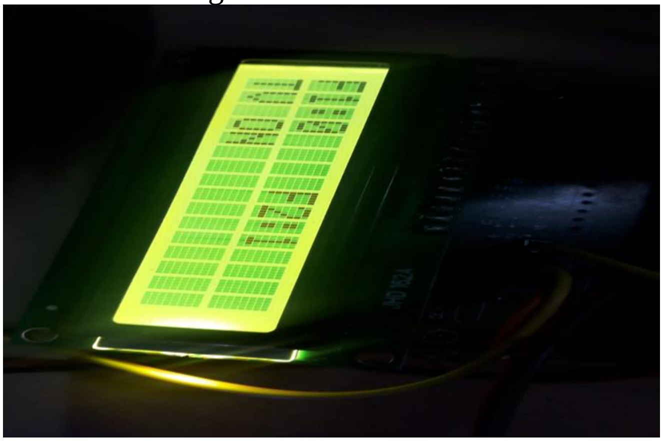


Fig 5.1.2 : Output when flex sensor 1 is bent

CHAPTER 6

ADVANTAGES

- It requires fewer components making the project cost efficient.
- The accuracy of get exact output is comparatively high.
- Easy to carry: As it can be fixed to the person's body who is in need.
- Light in weight.
- Flexible to users.
- Anyone can operate it easily.
- Real time translation.

CHAPTER 7
CONCLUSION

7.1 Conclusion:

Sign language is a method used for communication by disabled person. Here we are converting sign language into text and speech so that communication is not limited between them only, utilizing data gloves communication barrier between two different communities is eliminated. Using data gloves disabled person can also grow in their carrier and makes nation grow as percentage of disabled person are millions in count. Making their future better, making nation better.

References

1. Laura Dipietro, Angelo M. Sabatini and Paolo Dario ―A Survey of Glove-Based Systems and Their Applications‖, IEEE Transactions on Systems, Man and Cybernetics—Part C: Applications and Review, Vol. 38, No. 4, pp. 461-482, July 2008.

2. K.C. Sri haripriya, K.Aarthy, T.S.Keerthana, S.Menaga ,S. Monisha, ―Flex Sensor based Non-specific User Hand Gesture Recognition‖, International Journals of Innovative Research and Studies, Vol.2, No.5, pp.214-220, May 2013.

3. Praveen Kumar, S Havalagi and Shruthi Urf Nivedita, ―The Amazing Gloves that give Voice to the Voiceless‖, International Journal of Advances in Engineering & Technology, Vol. 6, No.1, pp. 471-480, March 2013.

4. Olga Katz nelson and Solange Karsenty, ―A Sign-to-Speech Glove‖, IUI Workshop: Interacting with Smart Objects, Haifa, Israel, and February 24, 2014.

5. Shoaib Ahmed.V Department of Electronics and Communication C. Abdul Hakeem College of Engineering and Technology Melvisharam, Vellore, Tamil Nadu 632 509, India submitted the thesis report of ―Hand Gesture Recognition and Voice Conversion System for Differentially Able Dumb People‖, 2012.

6. Oya Aran, Ismail Ari, LaleAkarun, Bulent Sankur, Alxender Benoit, Alice Caplier, Pavel Campr, Ana Huerta Carrillo and Francis-Xavier Fanard, ―Sign Tutor: An Interactive System for Sign Language Tutoring‖, IEEE Proceeding of 6th International Conference, Israel, Vol 16, No. 1, pp. 81- 93, November 2009.

7. M. Ebrahim Al-Ahdal and Nooritawati Md Tahir, ―Review in Sign Language Recognition Systems‖ IEEE Symposium on Computers & Informatics, USA, pp.52-57, May 2012.

BIOMETRIC ATTENDANCE SYSTEM OVER IOT

A PROJECT REPORT

Submitted by

1).Ms.B.Sravani(17K81A04C6)

2).Ms.B.Sumalatha(17K81A04C8)

3).Ms.V.Sunitha (17K81A04H4)

Under the Guidance of

Mr.N.Vishwanath

M.Tech. (Ph.D)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGNEERING

ST.MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

MAY 2021

BONAFIDE CERTIFICATE

Certified that this project report titled "*BIOMETRIC ATTENDANCE SYSTEM OVER IOT* ", is a *bonafide* work of **1.Ms.B.Sravani (17K81A04C6) , 2.Ms.B.Sumalatha(17k81A04C8) 3.Ms.V.Sunitha(17K81A04H4)** who carried out the work under my supervision, for the partial fulfillment of the requirements for the award of the degree of *Bachelor of Technology* in *Electronics & Communication Engineering.* Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion.

Place: Secunderabad,
Date:

Dr.B.Hari krishna ,
B.Tech., M.Tech., Ph.D.,
Professor &HOD,
Department of Electronics & Communication Engineering,
St.Martin's Engineering College.

COUNTER SIGNED

HEAD
Department of Electronics and Communication Engineering
St.Martin's Engineering College

DECLARATION

We declare that this project report titled Biometric attendance system over IOT submitted in partial fulfillment of the degree of B.Tech in Electronics and communication record of original work carried out by us under the guidance and supervision of N.Vishwanath, and has not formed the name basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

<Signature>	<Signature>	<Signature>
1. B.Sravani 17K81A04C6	2.B.Sumalatha 17K81A04C8	3. V.Sunitha 17K81A0H4

Place:

Date:

ACKKNOWLEDGEMENTS

While bringing out this project report to its final form, we came across a number of people whose contributions in various ways helped my field of research and they deserve special thanks. It is a pleasure to convey our gratitude to all of them.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to Dr.B.Harikrishna, **Head &Professor**, the Dept. of Electronics and communication for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept project coordinator** Dr.A.Anand for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We are highly grateful and obliged to **our college level Project Coordinators Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santosh Kumar** , **Associate Professor in CSE** and our project coordinator **N.Vishwanath** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

Finally, I am deeply indebted to our family members for their moral support and continuous encouragement while carrying out this study. I dedicate this thesis to almighty

.

1. B.Sravani.
2. B.Sumalatha.
3. V.Sunitha.

ABSTRACT

Here we propose a smart fingerprint based biometric attendance system that works over IOT so that attendance can be monitored from anywhere in the world. Our system uses a microcontroller based circuit with fingerprint sensor, push buttons, power supply, power supply and WIFI modem to interact with internet based system. We here use IOT Gecko to develop the online attendance display system. Our system allows users/employees/students to first register their fingerprint on the system. After successful registration the print is stored in system with class assigned using push buttons. The system also displays these details over LCD display. Now as soon as the next time a registered user scans the modem, the system checks for authentication and authenticated users data is transferred online to IOT Gecko using the gecko development API codes. Now the online system stores and displays the required data to users as per online login. Thus,our system allows for remote monitoring of biometric based attendance from anywhere over IOT.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 EMBEDDED SYSTEMS

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer. The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes.

The applications software on such processors is sometimes referred to as firmware. The simplest devices consist of a single microprocessor (often called a "chip"), which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine.

As the embedded system is the combination of both software and hardware.

1.1.1 Block diagram of embedded system:

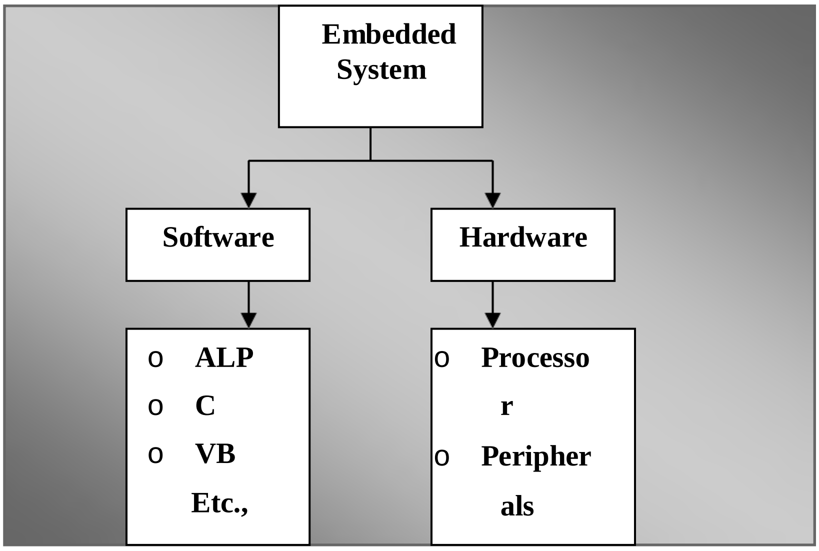


Fig.1.1: Block diagram of embedded system

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory

1.1.2 Applications of embedded systems:

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

1.2 MICRO PROCESSOR (µp):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Processors are classified into four types like:

- Micro Processor (μp)
- Micro controller (μc)
- Digital Signal Processo1r (DSP)
- Application Specific Integrated Circuits (ASIC)

1.2.1 Three Basic Elements of a Microprocessor:

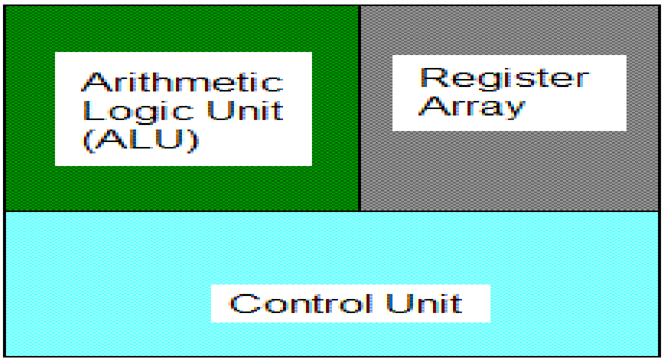


Fig.1.2: Three basic elements of a microprocessor

1.3 HARVARD ARCHITECTURE:

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

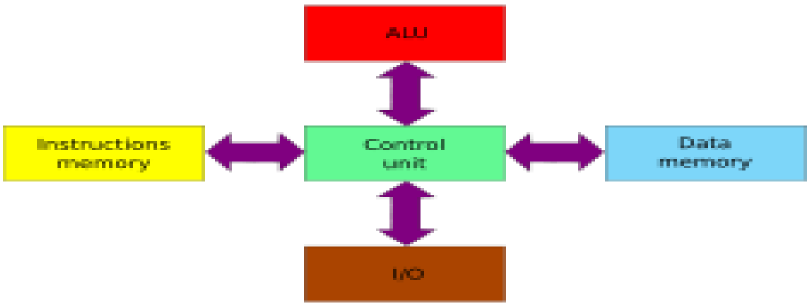


Fig.1.3: Harvard Architecture

1.3.1 Uses of the Harvard architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

- Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behaviour must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.
- Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

1.4 VON-NEUMANN ARCHITECTURE:

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. They cannot be performed at the same time. The von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A stored-program digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

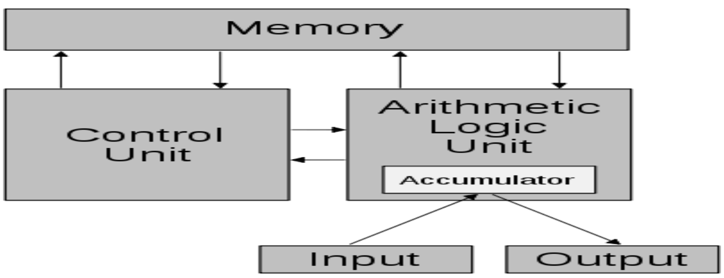


Fig.1.4: Schematic of the Von-Neumann Architecture.

1.5 PROJECT DESCRIPTION:

1.5.1 Objective:

The main aim of this project is to design and implement biometric attendance system over IOT through which the attendance of students is monitored remotely using internet and the data gathered from the fingerprint sensors is stored in the cloud. A portable module is designed which has a capability of recognizing the student via their fingerprints and then sending the ID of student to server, whose fingerprint is recognized. Here we use IOT Gecko Platform in which a person can provide Username and password and can know his/her attendance records, similarly the HOD or teachers can also check the students attendance using their respective username and password.

1.5.2 Aim:

The main reason for preferring this project is to maintain proper record of students or employees for effective functioning of organization. Designing an effective attendance management system for students to maintain the records with ease and accuracy is an important key behind motivating this project. Nowadays attendance is taken on paper and records are maintained where someone keeps all records and does all the calculations at the end of the month due to which it takes time and students have to wait till month end to know their attendance. This system would improve accuracy of attendance records because it will remove all the hassles of roll calling and will save valuable time of the students and teachers and will overcome problems like proxy and provide correct record of attendance.

1.6 COMPONENTS USED :

- WI-FI module(ESP8266 12e)
- Fingerprint Sensor(r-305)
- LCD display(16*2)
- Buzzer
- Push Buttons
- Arduino UNO

1. ARDUINO UNO:

Arduino UNO is a microcontroller which is used to transfer data from Fingerprint sensor to LCD and Wi-fi module. Arduino takes the input from Fingerprint Sensor in Analog form and converts it into digital form. This converted data is sent to the Wi-fi module. The Arduino sends the power required by the Fingerprint Sensors. Arduino gets power from the Wi-fi module to work. The Arduino works according to the code which is dumped into it.

2. ESP-01:

The wi-fi module in the circuit helps in connecting the microcontroller with the wi-fi network. The data received by the Wi-fi module is sent to the server where we can see the student attendance record with their respective username and password.

3. FINGERPRINT SENSOR:

The fingerprint sensor senses the skin on the palms of our hands that have a special pattern called friction ridges that helps us grab things effectively without slipping around it.so, when we place a finger on fingerprint scanner it captures the pattern which is unique to every student.

4 .LCD :

It is an electronic display module that uses liquid crystal to produce a visible image that is used to display 16 characters per line in 2 such lines.it is used in our project to display instructions like enrolling the student fingerprint data ,place the finger and used to display student information.

5. BUZZER:

Buzzer is used here for indication purpose. When an unauthorized student places a finger on fingerprint sensor it indicates that student who placed finger does not have a fingerprint and thus makes a beep sound.

6. PUSH BUTTONS:

Two Push buttons are used in our project.one pushbutton is used for enrolling that is used to store student fingerprint initially and another for checking the fingerprint.

CHAPTER 2

LITERATURE SURVEY AND EXISTING METHODS

2.1 LITERATURE SURVEY

After doing analysis on the recent work done in attendance and performance analysis, we found that many people used different methods to mark attendance And also, this kind of system worked well in all the cases but there were some issues with them, such as the student could mark his/her attendance anything even if he/she is late. Also, certain systems used Arduino and a single board computer with fingerprint biometrics. In such system the idea was to centralized the management system. Some systems used Arduino and raspberry Pi for attendance recording. Such system implemented attendance recording and consolidation with the help of raspberry Pi and Arduino. But, after a certain extent only marking attendance was not a key point as being an engineer no one is happy with doing a small kind of work so there came idea of managing the performance of the student using such kind of data and a new thing came in market known as STUDENT PERFORMANCE ANALYSIS.

2.2 EXISTING METHOD

The existing attendance system requires a teacher to take attendance by roll calling, which has many drawbacks, such as proxy attendance, extra efforts of teacher calculating the attendance percentage, even calculation errors can be made, and students not getting their attendance report till the end of the month. All these problems can be avoided by using this system, as this system uses fingerprint recognition to identify the student, proxy attendance can’t be marked, attendance is sent to server in real time, all the calculations are done by the server and students can check their attendance in real time.

2.3 PROPOSED SYSTEM

We here use IOT to develop the online attendance display system. Our system allows users/employees/students to first register their fingerprint on the system. After successful registration the print is stored in system with class assigned using push buttons. The system also displays these details over LCD display. Now as soon as the next time a registered user scans the modem, the system checks for authentication and authenticated users data is transferred online to IOT. Now the online system stores and displays the required data to users as per online login. Thus our system allows for remote monitoring of biometric based attendance from anywhere over IOT.

CHAPTER-3

BLOCK DIAGRAM

3.1 BLOCK DIAGRAM

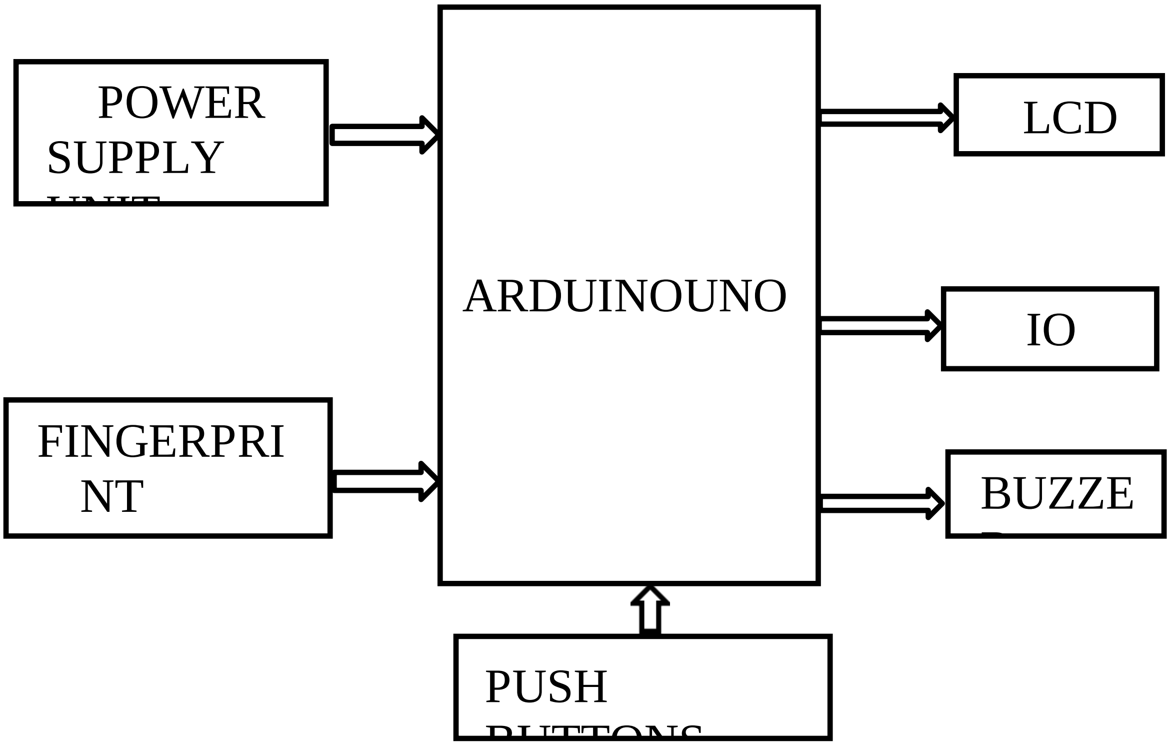


Figure 3.1 Block Diagram

The block diagram of biometric attendance system over IOT is shown in the above figure. .The blocks of the circuit are

a. Arduino UNO.

b. WI-FI module(ESP-8266 12e).

c. LCD.Display(16*2).

d. fingerprint Sensor.

e. Push buttons

f. Buzzer

g. Power Supply Unit

3.2 SCHEMATIC EXPLANATION:

3.2.1 INTERFACING BETWEEN ARDUINO AND LCD:

The following circuit diagram shows the liquid crystal display with the Arduino module. From the circuit diagram, we can observe that the RS pin of the LCD is connected to the pin 12 of the Arduino. The LCD of R/W pin is connected to the ground. The pin 11 of the Arduino is connected to the enable signal pin of LCD module. The LCD module & Arduino module are interfaced with the 4-bit mode in this project. Hence there are four input lines which are DB4 to DB7 of the LCD. This process very simple, it requires fewer connection cables and also we can utilize the most potential of the LCD module.

The digital input lines (DB4-DB7) are interfaced with the Arduino pins from 5-2. To adjust the contrast of the display here we are using a 10K potentiometer. The current through the back LED light is from the 560-ohm resistor. The external power jack is provided by the board to the Arduino. Using the PC through the USB port the Arduino can power. Some parts of the circuit can require the +5V power supply it is taken from the 5V source on the Arduino board.

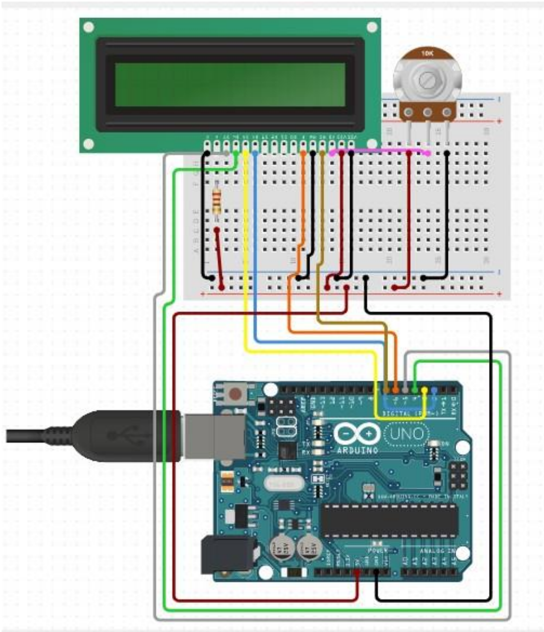


Fig 3.2: Schematic diagram of LCD and Arduino.

3.2.2 INTERFACING BETWEEN ARDUINO AND WI-FI MODULE:

Arduino UNO to configure ESP8266 wi-fi module as TCP Client and Receive/Send data from/to Server using WIFI.

Here, we are using IOT Gecko server for TCP Client demo purpose.

IOT Gecko is an open IOT platform where anyone can visualize and analyze live data from their sensor devices. Also, where data of students attendance can be maintained and stored.

Just sign up and create channel. We have to provide username and password to login to IOT Gecko platform.

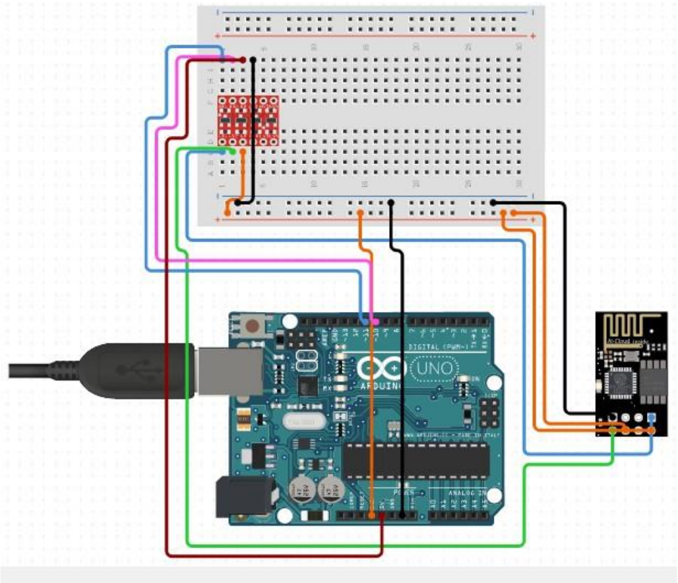


Fig 3.3:Schematic diagram of Wi-fi module and Arduino

3.2.3 INTERFACING BETWEEN ARDUINO AND FINGERPRINT SENSOR:

The +5v for r-305 can be taken from the +5v out pin of arduino uno. Also the ground pin r-305 of can be connected to GND pin of arduino uno. Connect V out (the analog out of r-305) to any of the analog input pin of arduino uno. In this circuit diagram, we have connected Vout of r-305 to A1 of arduino.

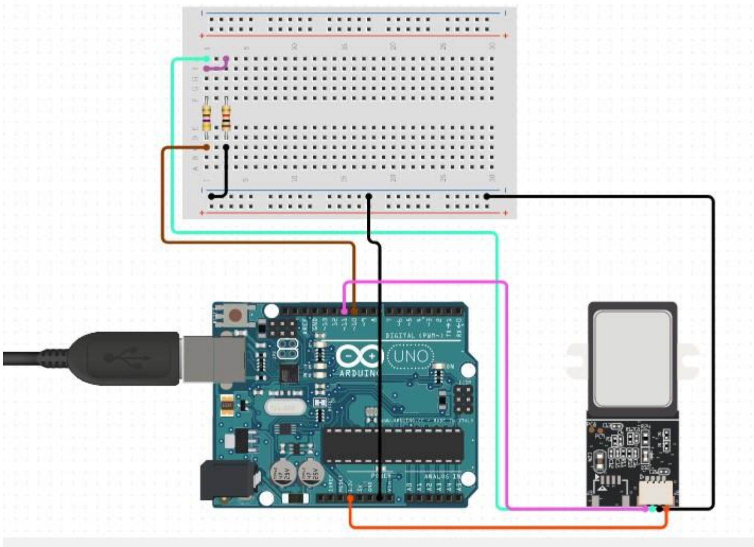


Fig 3.4:Schematic diagram of Temperature Sensor and Arduino.

3.2.4 INTERFACING BETWEEN ARDUINO AND PUSH BUTTONS

The push buttons are used for enrolling and checking of student fingerprints. the two push buttons are connected to 5, 6 pin of Arduino as shown in figure below.

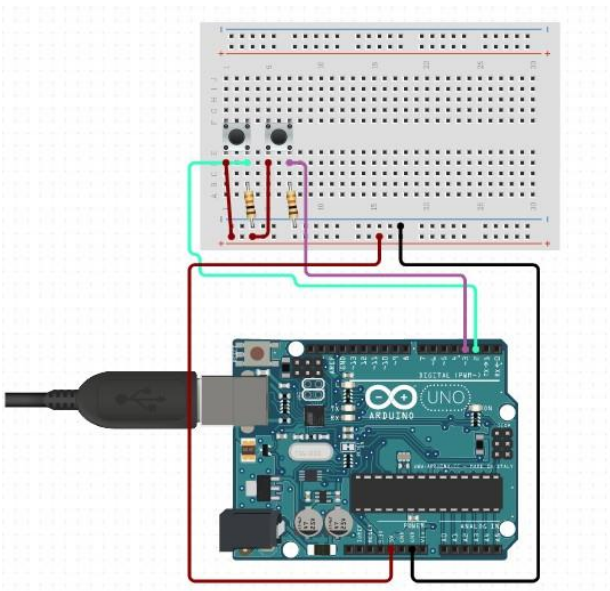


Fig 3.5 Schematic diagram of push buttons and Arduino.

3.2.5 INTERFACING BETWEEN THE COMPONENTS:

The fingerprint sensor output pin is connected to A0 and A1 of arduino analog pins. The Vcc and ground of fingerprint Sensors is connected to 5V and ground of Arduino. The transmitter and receiver of Arduino are connected to the receiver and transmitter of Wi-fi module respectively. The VSS, VEE, RW, D0-D3 pins of LCD are grounded. The RS, E, D4-D7 pins of LCD are connected to 13-8 digital pins of Arduino.

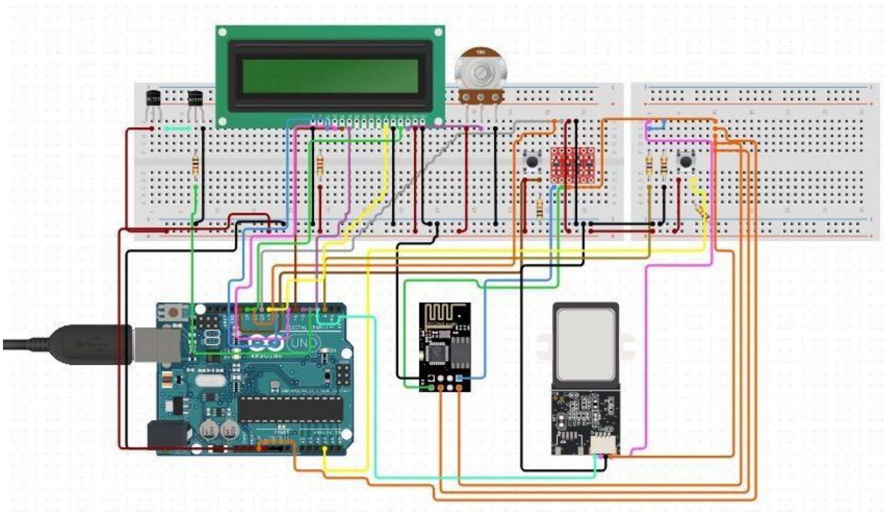


Fig 3.5: Schematic diagram of all components.

3.3 WORKING:

In this system the concept of IOT is applied to attendance system of a classroom. A portable module is designed which has the capability of recognizing the student via their fingerprints and then sending the ID of student to the server, whose fingerprint is recognized. First of all the system requires connectivity to the internet, which can be achieved through Wi-Fi So a system is required which has the capability of Wi-Fi connectivity for which Node mcu (ESP8266 12e) is chosen. Now when the system is powered ON, it first scans all the Wi-Fi networks and any network can be connected by entering the password. Once the connection is established, it scans for the fingerprint via fingerprint module r-305 and recognizes the student, whoever is recognized their fingerprint ID is sent to the server (PC). When the server receives the Fingerprint ID it marks the attendance of the student.

.

CHAPTER-4

HARDWARE AND SOFTWARE COMPONENTS

The hardware and software components used for the implementation of biometric attendance system over IOT are

1. Arduino UNO.
2. Regulated power supply.
3. ESP-01
4. LCD display.
5. Fingerprint Sensor.
6. Buzzer.
7. Push buttons

4.1 ATMEGA328

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The ATmega8 microcontroller contains 32 general purpose working registers. As shown in the below figure these registers are directly connected to ALU. Two registers can carry one single instruction consequently in one clock cycle.

4.1.1 SPECIFICATIONS:

Table 4.1: Atmega328 specifications

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V

Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 Ma
DC Current on 3.3V Pin	50 Ma
Flash Memory	32 KB (0.5 KB is used for Boot loader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

4.2 ARDUINO:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.

- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

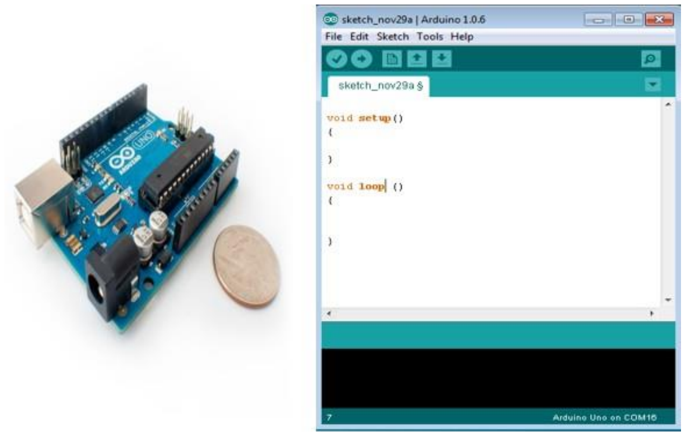


Fig 4.1 Arduino Uno

4.2.1 BOARD TYPES:

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V. Here is a list of different Arduino boards available.

Table 4.2 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega 16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega 16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16 MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header

Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compati
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 4.3 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 4.4 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 4.5 Arduino boards based on AT91SAM3X8E microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

4.2.2 BOARD DESCRIPTION:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

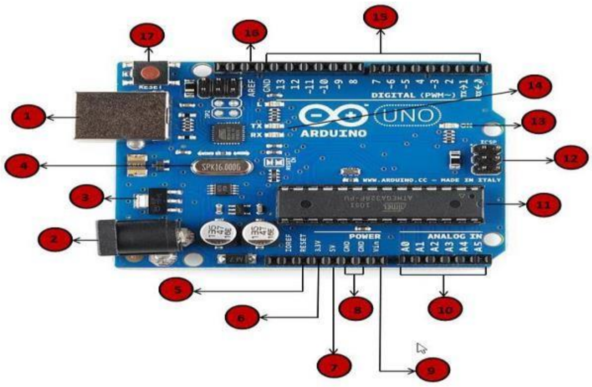


Fig 4.2 Arduino UNO pin description

Table 4.6 Arduino UNO pin description

Pin Category	**Pin Name**	**Details**
Power	Vin, 3.3V, 5V, GND	Vin: Input voltage to Arduino when using an external power source. 5V: Regulated power supply used to power microcontroller and other components on the board. 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. GND: ground pins.
Reset	Reset	Resets the microcontroller.
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5V

Input/Output Pins	Digital Pins 0 - 13	Can be used as input or output pins.
Serial	0(Rx), 1(Tx)	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	13	To turn on the inbuilt LED.
TWI	A4 (SDA), A5 (SCA)	Used for TWI communication.
AREF	AREF	To provide reference voltage for input voltage.

4.2.3 ARDUINO FAMILY

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

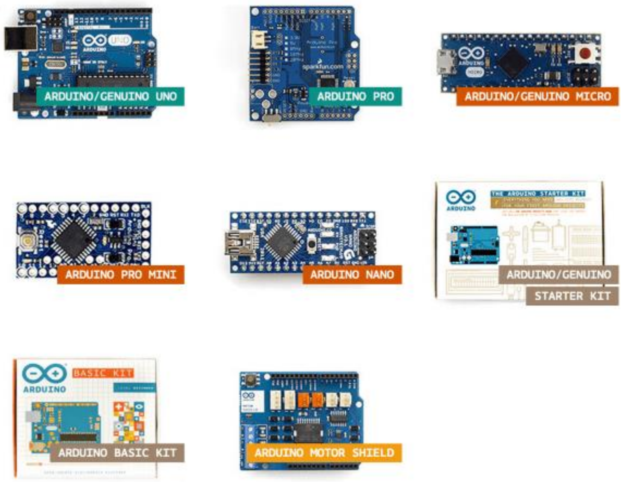


Fig.4.3: Arduino Family

4.2.4 PIN DESCRIPTION OF ATMEGA328

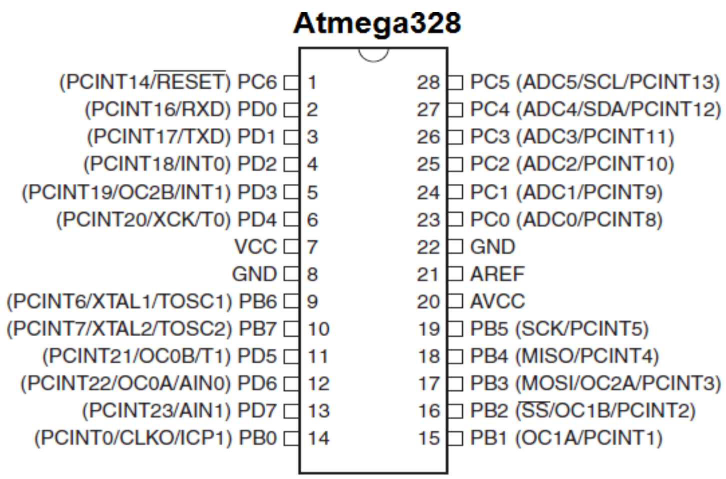


Fig.4.4: Pin description of ATMEGA328

4.2.5 ADVANTAGES OF ARDUINO

- It is cheap
- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.

- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
- For beginners, it is very simple to use.

4.2.6 APPLICATIONS

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model.

4.3 REGULATED POWER SUPPLY:

4.3.1 INTRODUCTION:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as “Regulated D.C Power Supply”.

4.3.2 BLOCK DIAGRAM OF POWER SUPPLY:

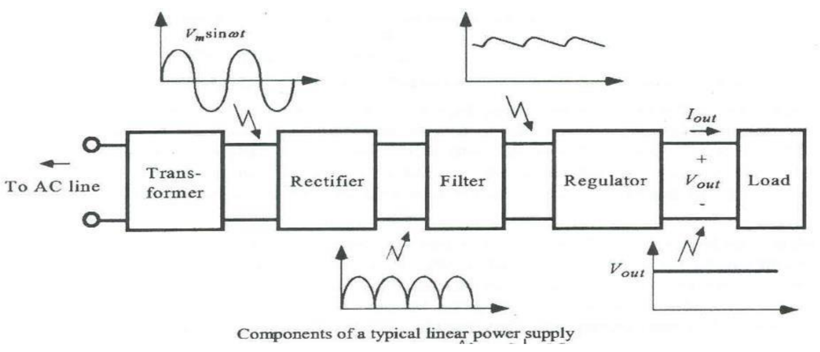


Fig.4.5: Block Diagram of Power Supply

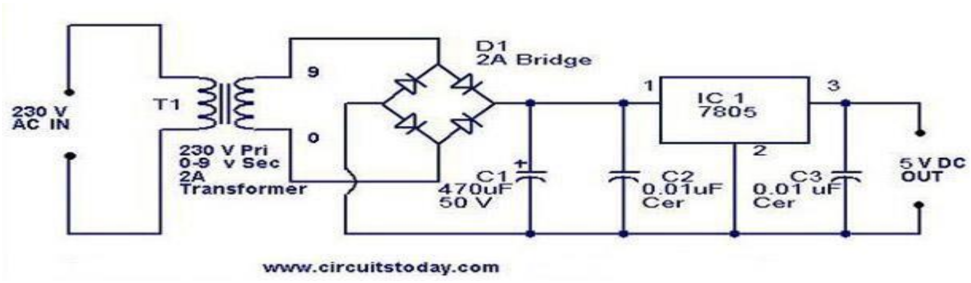


Fig.4.6: Schematic Diagram of Power Supply

4.3.3 DESCRIPTION OF POWER SUPPLY:

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a a different form of energy – such as solar, mechanical, or chemical - into electrical energy.

A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered. For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

4.3.4 TRANSFORMER:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

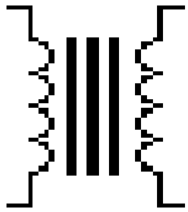


Fig.4.7: Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

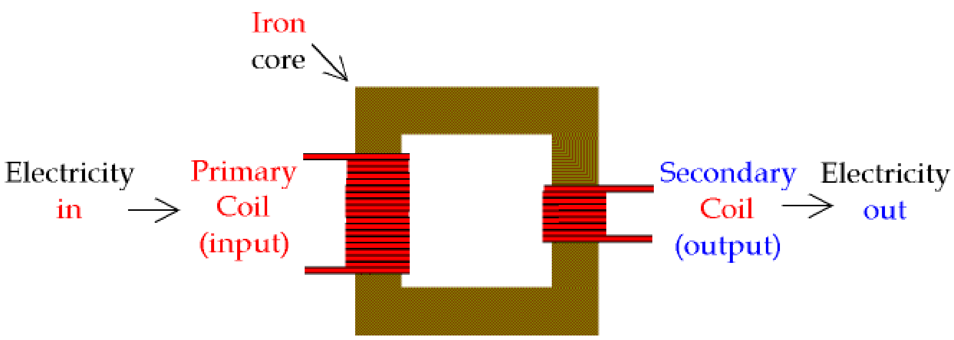
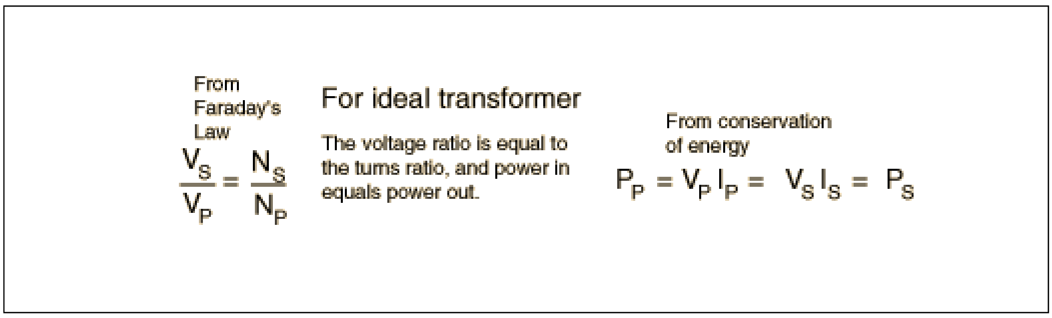


Fig.4.8: Transformer

4.3.4.1 Basic Principle of Transformer:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.



4.3.4.2 Working of Transformer:

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

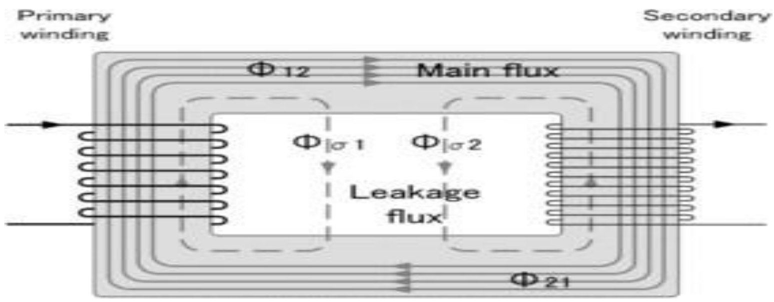


Fig.4.9: Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f. The iron core is laminated to prevent 'eddy currents' flowing in the core.

These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

4.3.4.3 CLASSIFICATION OF TRANSFORMER:

- Step-Up Transformer
- Step-Down Transformer

Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn’s ratio of the two sets of windings determines the amount of voltage transformation.

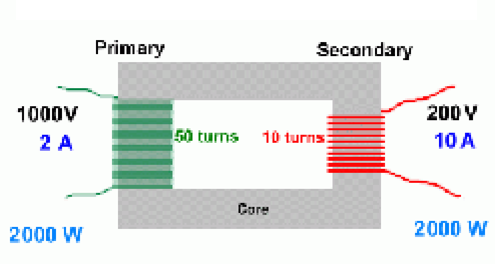


Fig.4.10: Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single phase steps down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase steps up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

Step-Up Transformer:

A step up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply. A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

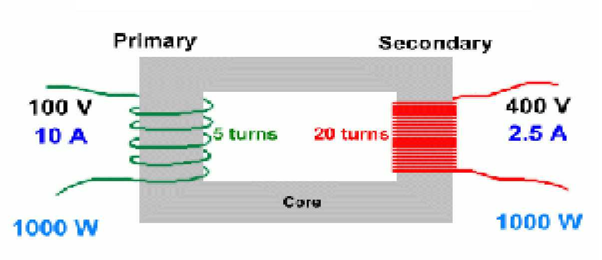


Fig.4.11: Step-Up Transformer

4.3.5 DIODES:

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

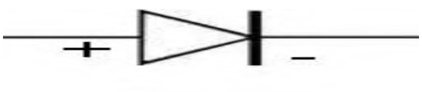


Fig.4.12: Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

4.3.6 RECTIFIER:

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

The Half-Wave Rectifier:

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure.

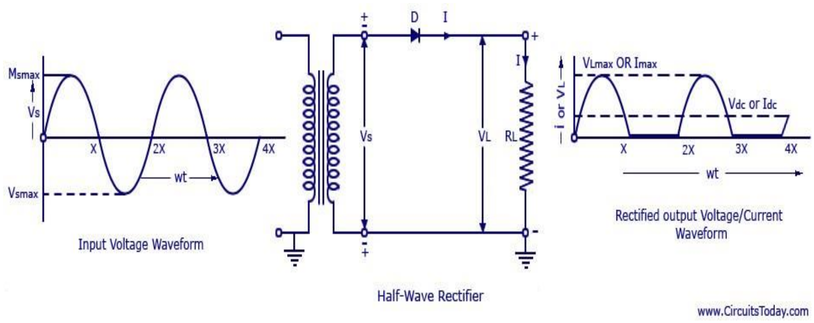


Fig.4.13: Half Wave Rectifier

Figure 2 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

The Full-wave Rectifier:

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC.

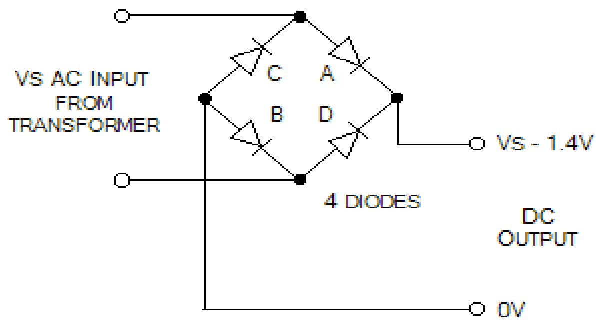


Fig.4.14: Full-Wave Rectifier

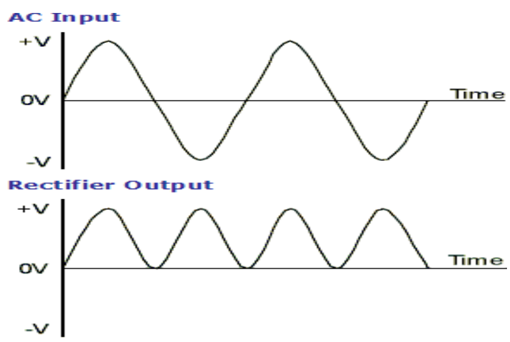


Fig.4.15: Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased. While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

4.3.7 CAPACITOR FILTER:

The capacitor-input filter, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

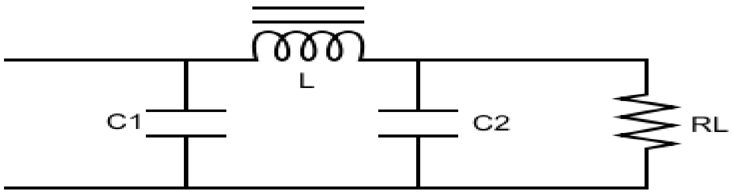


Fig.4.16: Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result, the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result, only the DC component appears across the load RL.

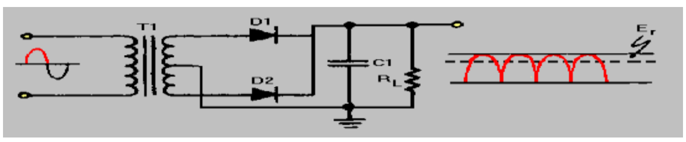


Fig.4.17: Centered Tapped Full-Wave Rectifier with a Capacitor Filter

4.3.8 VOLTAGE REGULATOR:

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx'indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx'indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consist the three pins there are

Pin1: It is used for input pin

.Pin2: This is ground pin for regulator.

Pin3: It is used for output pin. Through this pin we get the output.

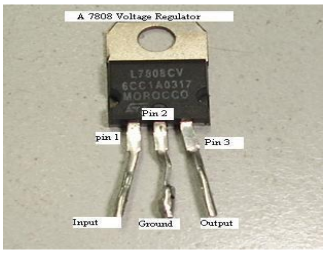


Figure 4.18 Volatge Regulator

4.4 ESP8266 WIFI Module:

The **ESP8266 Wi-Fi Module** is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your **Wi-Fi** network. The **ESP8266** is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

4.4.1 CHARACTER CONFIGURATION:

- 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2),
- general-purpose input/output (16GPIO),
- Inter-Integrated Circuit (I^2C) serial communication protocol,
- analog-to-digital conversion (10-bitADC)
- Serial Peripheral Interface (SPI) serial communication protocol,
- I^2S (Inter-IC Sound) interfaces with DMA(Direct Memory Access) (sharing pins with GPIO),
- UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2),and
- Pulse-width modulation (PWM).

4.4.2 APPLICATIONS:

- Smart Home Appliances
- Improved Home safety and security
- Home air quality and water quality monitoring
- Natural Language-based voice assistants.

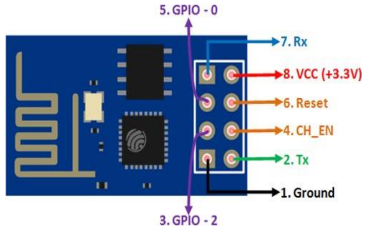


Fig.4.19 ESP8266 wi-fi module pin out

4.5 FINGERPRINT SENSOR :

Finger Print Sensor (R305) -TTL UART is a finger print sensor module with TTL UART interface. The user can store the finger print data in the module and can configure it in 1:1 or 1: N mode for identifying the person. The finger print module can directly interface with 3v3 or 5v Microcontroller. A level converter (like MAX232) is required for interfacing with PC.

Features of Finger Print Sensor (R305) -TTL UART:

- Power DC : 3.6V-6.0V
- Interface : UART (TTL logical level)/ USB 1.1
- Working current : 100mA
- Peak Current : 150mA
- Matching Mode: 1:1 and 1:N
- Baud rate (9600*N)bps, N=1-12 (default N=6 57600bps)
- Character file size: 256 bytes
- Image acquiring time : <0.5s
- Template size : 512 bytes
- Storage capacity: 256
- Security level : 5 (1, 2, 3, 4, 5(highest))
- FAR : <0.001%
- FRR: <0.1%
- Average searching time: < 0.8s (1:880)
- Window dimension : 18mm*22mm

Fig:4.2.1 Fingerprint Scanner r-305

The characteristic of this r-305 sensor is:

- Temp: -10 - +40.
- Storage environment Temp: -40- +85.
- RH: 40%-85% RH: <85%.

4.6 16 x2 ALPHANUMERIC LCD:

4.6.1 DESCRIPTION:

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

4.6.2 BLOCK DIAGRAM:

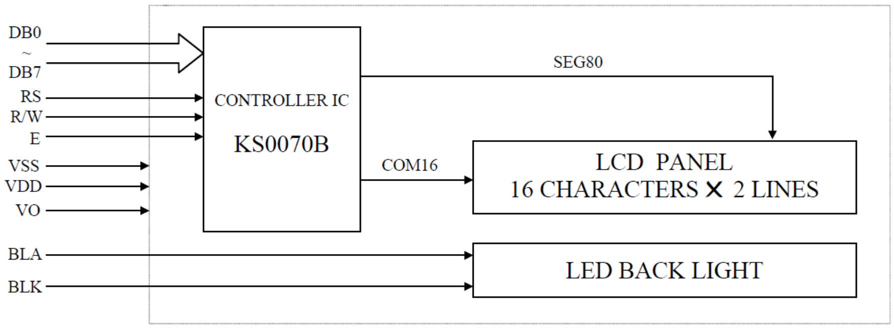
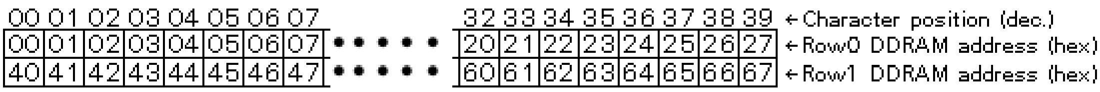


Fig 4.22: Block Diagram

Display Data RAM

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.



DDRAM Address for 2 Line LCD

Character Generator ROM

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. Userdefined character patterns are also available by mask-programmed ROM.

Busy Flag

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact ammount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

Instruction Register(IR) and Data Register(DR)

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g LCD shift command, LCD clear, LCD address etc. and

Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD.
Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

4.6.3 16 x 2 ALPHANUMERIC LCD MODULE FEATURES:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

4.6.4 SCHEMATIC:

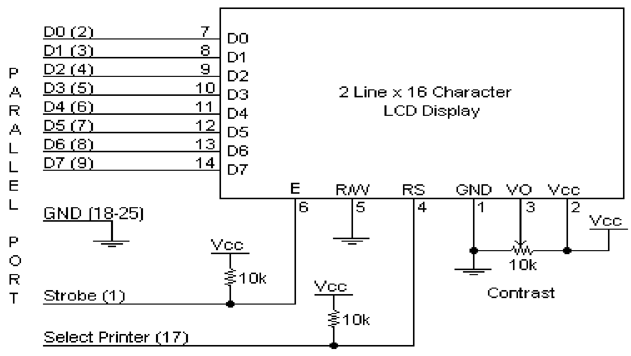


Fig 4.23 Schematic diagram of LCD

4.6.5 SPECIALISATIONS:

Table 4.7: Specifications of LCD

Connector Pin Assignment:

Pin	*Symbol*	*Function*	*Pin*	*Symbol*	*Function*
14	DB7	Data Bus Line	6	E	Enable
13	DB6	Data Bus Line	5	R/W	Read/Write
12	DB5	Data Bus Line	4	RS	Register select input
11	DB4	Data Bus Line	3	Vcontrast	Contrast (0V for max contrast)
10	DB3	Data Bus Line	2	**GND (0V)**	Power supply
9	DB2	Data Bus Line	1	**Vcc +5V**	Power supply
8	DB1	Data Bus Line			
7	DB0	Data Bus Line			

Note1: Pin 1 is +5V and pin 2 is GND! This is different from most other HD44780 compatible models.
Note2: Pin 14 is marked with a “1” on the back of the PCB. Do not get confused by this.

4.6.6 CIRCUIT DESCRIPTION:

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

4.7 BUZZER

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke.

Mechanical:

A joy buzzer is an example of a purely mechanical buzzer.

Figure 4.24 Mechanical joy buzzer

Electromechanical:

Early devices were based on an electromechanical system identical to an electric bell without the metal going. Similarly, a relay may be connected to interrupt its own actuating current, causing the contacts to buzz. Often these units were anchored to a wall or ceiling to use it as a sounding board. The word "buzzer" comes from the rasping noise that electromechanical buzzers made.

Piezoelectric:

Nowadays, it is more popular to use a ceramic-based piezoelectric sounder like a Son alert which makes a high-pitched tone. Usually these were hooked up to "driver" circuits, which varied the pitch of the sound or pulsedthe sound on and off.

Figure 4.25 Buzzer

A piezoelectric element may be driven by an oscillating electronic circuit or other audio signal source, driven with a piezoelectric audio amplifier. Sounds commonly used to indicate that a button has been pressed are a click, a ring or a beep. Piezo buzzer is an electronic device commonly used to produce sound. Light weight, simple construction and low price make it usable in various applications like car/truck reversing indicator, computers, call bells etc.

Basically, the sound source of a piezoelectric sound component is a piezoelectric diaphragm as shown in fig(1). A piezoelectric diaphragm consists of a piezoelectric ceramic plate which has electrodes on both sides and a metal plate (brass or stainless steel, etc.).

A piezoelectric ceramic plate is attached to a metal plate with adhesives. Applying

D.C. voltage between electrodes of a piezoelectric diaphragm causes mechanical distortion due to the piezoelectric effect. For a misshaped piezoelectric element, the distortion of the piezoelectric element expands in a radial direction. And the piezoelectric diaphragm bends toward the direction. The metal plate bonded to the piezoelectric element does not expand.

Conversely, when the piezoelectric element shrinks, the piezoelectric diaphragm bends in the direction Thus, when AC voltage is applied across electrodes, the bending is repeated, producing sound waves in the air.

Buzzers typically provide a user with an alert when a given condition within the circuit has occurred. Normally man’s audible frequency range is 20Hz to 20kHz , but we can hear up to frequency ranges from 2kHz to 4kHz .

To interface a buzzer the standard transistor interfacing circuit is used(Fig 2). Note that if a different power supply is used for the buzzer, the 0V rails of each power supply must be connected to provide a common reference. If a battery is used as the power supply, it is worth remembering that piezo sounders draw much less current than buzzers. Buzzers also just have one ‘tone’, whereas a piezo sounder is able to create sounds of many different tones. Here transistor acts as a switch.

To switch on buzzer -high 1 To switch off buzzer -low 1

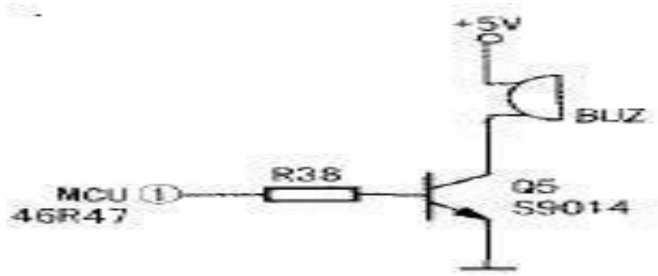


Figure 4.26 Buzzer with transistor interfacing circuit

Notice (Handling) In Using Self Drive Method:

1) When the piezoelectric buzzer is set to produce intermittent sounds, sound may be heard continuously even when the self drive circuit is turned ON / OFF at the "X" point shown in Fig(b). This is because of the failure of turning off the feedback voltage.

2) Build a circuit of the piezoelectric sounder exactly as per the recommended circuit shown in the catalog. Life of the transistor and circuit constants is designed to ensure stable oscillation of the piezoelectric sounder.

3) Design switching which ensures direct power switching.

4) The self drive circuit is already contained in the piezoelectric buzzer. So there is no need to prepare another circuit to drive the piezoelectric buzzer.

5) Rated voltage (3.0 to 20Vdc) must be maintained. Products which can operate with voltage higher than 20Vdc are also available.

6) Do not place resistors in series with the power source, as this may cause abnormal oscillation. If a resistor is essential to adjust sound pressure, place a capacitor (about 1μF) in parallel with the piezo buzzer.

7) Do not close the sound emitting hole on the front side of casing.

8) Carefully install the piezo buzzer so that no obstacle is placed within 15mm from the sound release hole on the front side of the casing.

4.8 PUSH BUTTONS

A push-button or simply button is a simple switch mechanism to control some aspect of a machine or a process. Buttons are typically made out of hard material, usually plastic or metal. The surface is usually flat or shaped to accommodate the human finger or hand, so as to be easily depressed or pushed. Buttons are most often button switches, although many un-biased buttons (due to their physical nature) still require a spring to return to their un-pushed state. Terms for the "pushing" of a button include pressing, depressing, mashing, slapping, hitting, and punching.

4.8.1 Features

- Prevent flux rise by the insert-molded terminal
- Snap-in mount terminal
- Contact Bounce: MAX 5mS
- Crisp clicking by tactile feedback
- Dielectric Withstanding Voltage 250V AC for 1 minute

4.8.2 Technical Specifications

- Mode of Operation: Tactile feedback
- Power Rating: MAX 50mA 24V DC
- Insulation Resistance: 100Mohm at 100v
- Operating Force: 2.55±0.69 N
- Contact Resistance: MAX 100mOhm
- Operating Temperature Range: -20 to +70 ℃
- Storage Temperature Range: -20 to +70 ℃

.Figure :4.27 Push button

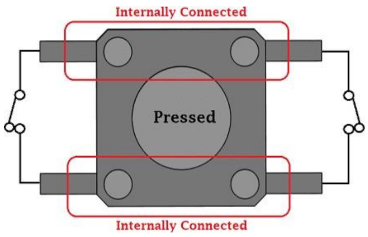


Fig 4.28 Push button Pin out connections

4.8.3 Where to use push button?

Push-Buttons are normally-open **tactile switches**. Push buttons allow us to power the circuit or make any particular connection only when we press the button. Simply, it makes the circuit connected when pressed and breaks when released. A push button is also used for triggering of the SCR by gate terminal. These are the most common buttons which we see in our daily life electronic equipment's. Some of the applications of the Push button are mentioned at the end of the article.

4.8.4 How to use a push button?

When connecting in between of supply and the circuit we should only connect the wires with both the legs of the Push-Button as shown in the circuit below:

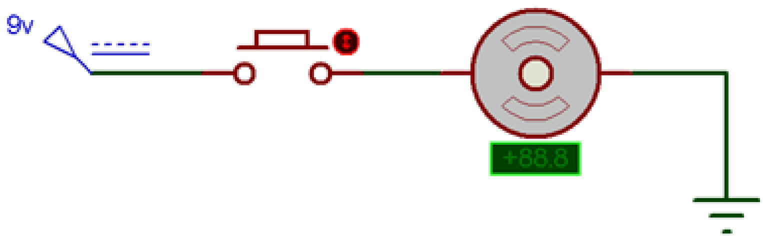


Figure : 4.29 Pushbutton connection with power supply

A Push-Button can also be used for the triggering purpose like of SCR. An SCR is a gate controlled Switch which needs a triggering pulse. So, for this we can add a Push button in the circuit to give a triggering pulse, as shown in the circuit below:

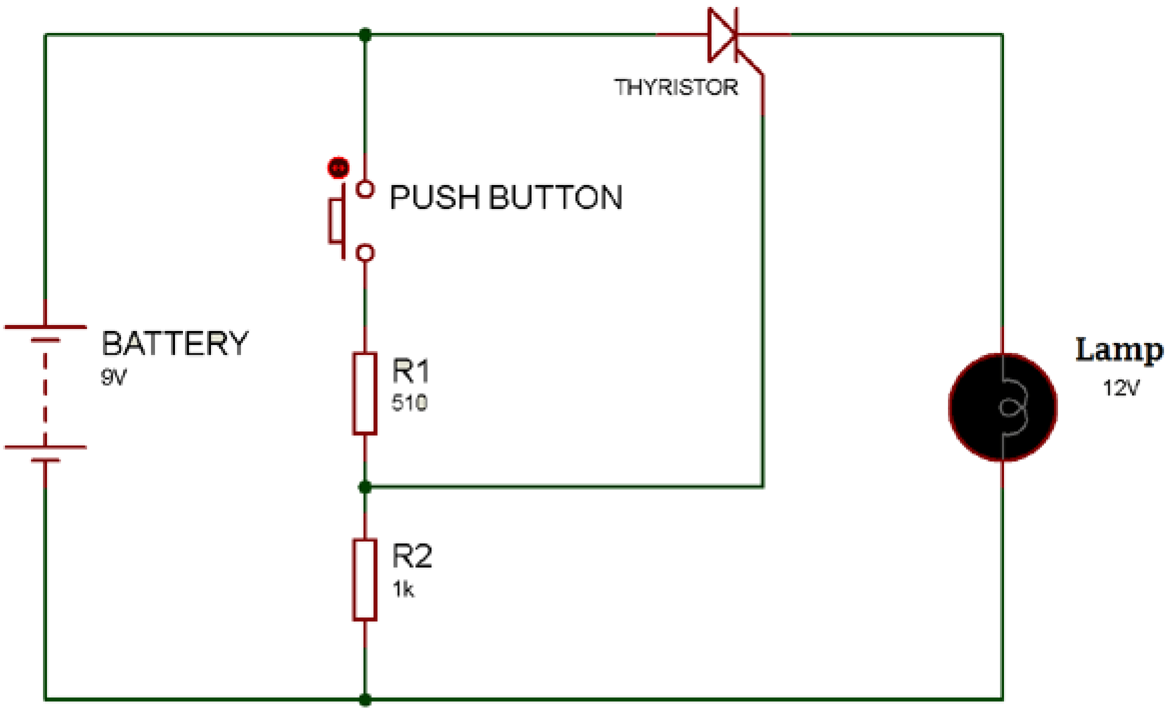


Figure:4.30 Push button with triggering pulse.

4.8.5 Applications

- Calculators
- Push-button telephones
- Kitchen appliances
- Magnetic locks
- Various other mechanical and electronic devices, home and commercials.

SOFTWARE COMPONENTS

- Proteus simulation

- Arduino software
- Programming language

4.9 Arduino software:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals

- has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

4.9.1 WHY ARDUINO?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use forbeginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers c

an make their own version

- Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money
- Getting Started with Arduino and Genuino products:-

Install the Arduino Software (IDE) on Windows PCs-

…This document explains how to install the Arduino Software (IDE) on Windows machines.

- ✓ Download the Arduino Software(IDE)
- ✓ Proceed with board specific instructions.

How to Download the Arduino Software (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

4.9.2 INSTALLATION:

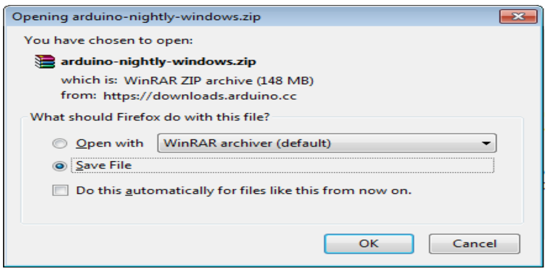
In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

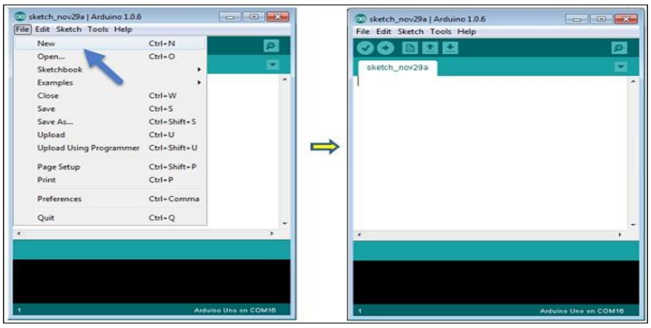
After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

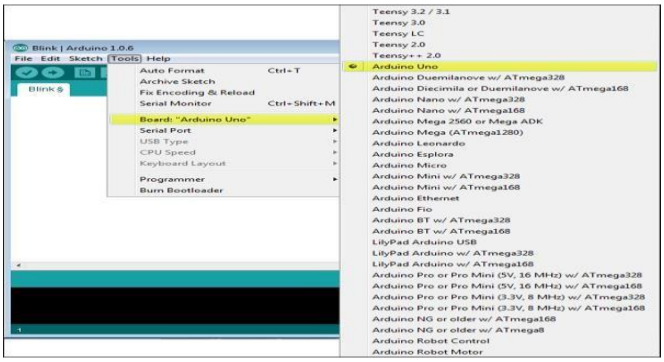
To create a new project, select File → **New**.



Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

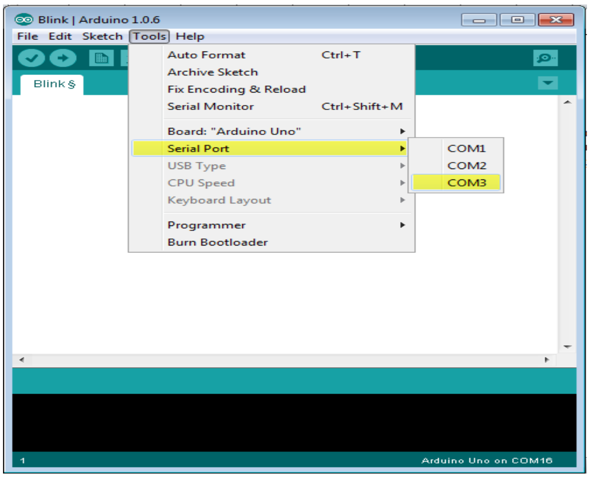
Go to Tools → Board and select your board.



Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

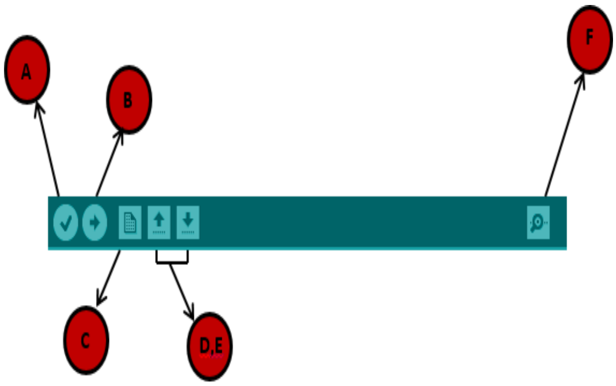
Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.



Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.



A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24 V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board.

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called “sketches”, but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
```

```
  Serial.println("Hello World");
 }
void loop() {}
```

4.10 Proteus:

Proteus:

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

4.10.1 : ABOUT PROTEUS:

It is a software suite containing schematic, simulation as well as PCB designing.

ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.

The designer can also develop 2D drawings for the product.

4.10.2 FEATURES:

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

4.10.3 STARTING NEW DESIGN:

Step 1: Open ISIS software and select New design in File menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

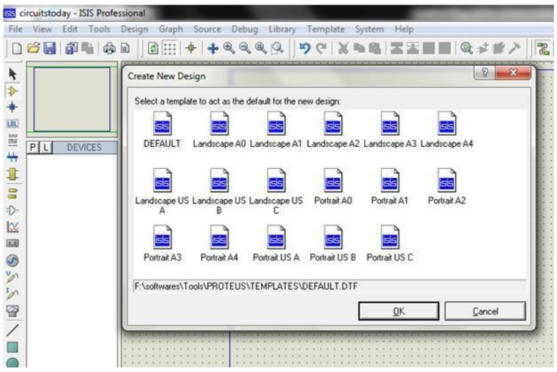


Fig Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

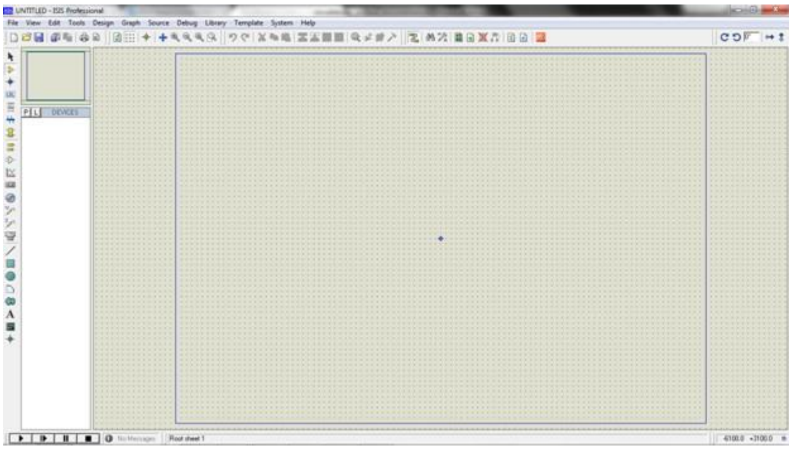


Fig Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

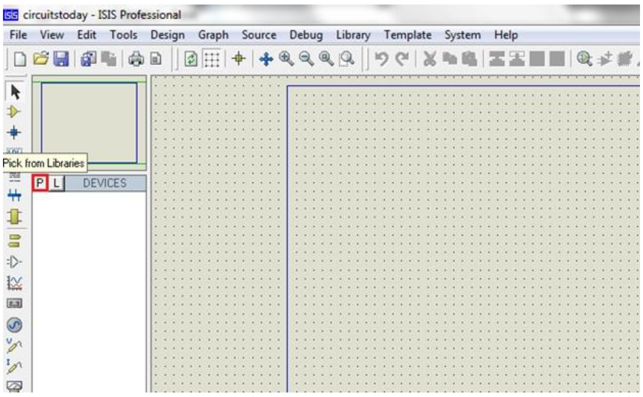


Fig Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

E

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

Double click on the component to edit the properties of the components and click on Ok.

Step 8: After connecting the circuit, click on the play button to run the simulation.

Simulation can be stepped, paused or stopped at any time.

4.10.4 Project code:

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2,3);

#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);


Unsignedchar
enroll[12]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X03,0X01,0X
00,0X05}; // ok
unsignedchar
generate_ch[13]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X04,0x0
2,0X01,0X00,0X08};  //ok
unsignedchar
generate_ch1[13]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X04,0x
02,0X02,0X00,0X09};  //ok
unsigned char un_cmd[12]={0xef,0x01,0xff,0xff,0xff,0xff,
0x01,0x00,0x03,0x05,0x00,0x09 };
unsigned char
store[12]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X06,0X06,0X0
2,0x00};     //ok
unsigned char
identify[17]={0xef,0x01,0xff,0xff,0xff,0xff,0x01,0x00,0x08,0x1b,0x01,0x00,0
x00,0x01,0x01,0x00,0x27};


char buff[200],k=0;
```

```
void upload1(unsigned char *chr);
 char res[130];
char check(char* ex,int timeout)
{
  int i=0;
  int j = 0,k=0;
  while (1)
  {
    sl:
    if(mySerial.available() > 0)
    {
      res[i] = mySerial.read();
      if(res[i] == 0x0a || res[i]=='>' || i == 100)
      {
        i++;
        res[i] = 0;break;
      }
      i++;
    }
    j++;
    if(j == 30000)
    {
      k++;
      //Serial.Sprintln("kk");
      j = 0;
    }
    if(k > timeout)
    {
     // Serial.println("timeout");
      return 1;
     }
  }//while 1
  if(!strncmp(ex,res,strlen(ex)))
  {
```

```
    //Serial.println("ok..");
    return 0;
   }
  else
  {
   // Serial.print("Wrong ");
   // Serial.println(res);
    i=0;
    goto sl;
   }
}

void serialFlush(){
  while(Serial.available() > 0) {
  char t = Serial.read();
  }
}
int fpenroll(char);
int fpsearch();

int s1=5,s2=6,m1=4,buz=7;



const char* ssid = "project";
const char* password = "project123";
void setup() {
  char ret;
  pinMode(s1, INPUT_PULLUP);
  pinMode(s2, INPUT_PULLUP);
 // pinMode(m1, OUTPUT);

// pinMode(buz, OUTPUT);

     digitalWrite(buz,HIGH);
```

```
  // digitalWrite(m1,LOW);
  // digitalWrite(m2,LOW);
   Serial.begin(57600);
   mySerial.begin(115200);//
   st:
 mySerial.println("ATE0");
 ret = check((char*)"OK",50);
 mySerial.println("AT");
 ret = check((char*)"OK",50);
 if(ret != 0)
 {
  delay(1000);
  goto st;
 }

  lcd.clear();lcd.setCursor(0, 0);lcd.print("CONNECTING");
 mySerial.println("AT+CWMODE=1");
 ret = check((char*)"OK",50);
cagain:

 serialFlush();
 mySerial.print("AT+CWJAP=\"");
 mySerial.print(ssid);
 mySerial.print("\",\"");
 mySerial.print(password);
 mySerial.println("\"");
 if(check((char*)"OK",300))goto cagain;
 mySerial.println("AT+CIPMUX=1");
 delay(1000);
 lcd.clear();lcd.setCursor(0, 0);lcd.print("CONNECTED");delay(1000);

 lcd.begin(16,2);
 lcd.clear();lcd.setCursor(0, 0);lcd.print("BIOMETRIC ATTENDANCE
SYSTEM OVER IOTBIOMETRIC ATTENDANCE SYSTEM OVER IOT");
```

```
  delay(2000);

}
 int err =0;
 int idk = 0,eid=0;
void loop()
{

  lcd.clear();lcd.setCursor(0, 0);lcd.print("PUT UR FINGER"); delay(1000);
if(digitalRead(s1) == 0)
{
  //Serial.println("Enrolling");
  lcd.clear();lcd.setCursor(0, 0);lcd.print("ENROLLING..");
  if(fpenroll(eid) == -1)
  {
   // Serial.print("Enroll failed:");Serial.print(err);Serial.println("");
    err=0;
    lcd.clear();lcd.setCursor(0, 0);lcd.print("ENROLL FAILED");
    digitalWrite(buz,LOW);delay(1000);digitalWrite(buz,HIGH);
  }
  else
  {
    lcd.clear();lcd.setCursor(0, 0);lcd.print("ENROLLED:");lcd.print((int)eid);
    //Serial.print("Enroll Success to id:");Serial.print((int)eid);Serial.println("");
    //Serial.print("*E");Serial.print((int)eid);Serial.println("#");
    eid++;
  }
  delay(2000);
 // lcd.clear();lcd.setCursor(0, 0);lcd.print("SELECT OPTION");
}
if(digitalRead(s2) == 0)//identify
{
  lcd.clear();lcd.setCursor(0, 0);lcd.print("SEARCHING..");
    idk = fpsearch();
```

```
if(idk == -1)
{
 err=0;lcd.clear();lcd.setCursor(0, 0);lcd.print("UNAUTHOISED ");
 lcd.clear();lcd.setCursor(0, 1);lcd.print("PERSON");

  digitalWrite(buz,LOW);
  delay(400);
  digitalWrite(buz,HIGH);

}
else
{
 lcd.clear();lcd.setCursor(0,
0);lcd.print("IDENTIFIED:");lcd.print((int)idk);

 if(idk == 0)
 {
  lcd.clear();lcd.setCursor(0, 0);lcd.print("STUDENT 1 PRESENT
");delay(2000);
  // lcd.setCursor(0, 1);lcd.print("PERSON");
  upload1("STUDENT_1_PRESENT");delay(2000);
 }
 if(idk == 1)
 {
   lcd.clear();lcd.setCursor(0, 0);lcd.print("STUDENT 2 PRESENT
");delay(2000);
  // lcd.setCursor(0, 1);lcd.print("PERSON");
  upload1("STUDENT_2_PRESENT");delay(2000);
 }
 if(idk == 2)
 {
  lcd.clear();lcd.setCursor(0, 0);lcd.print("STUDENT 3 PRESENT
");delay(2000);
  // lcd.setCursor(0, 1);lcd.print("PERSON");
```

```
      upload1("STUDENT_3_PRESENT");delay(2000);
     }
     if(idk == 3)
     {
      lcd.clear();lcd.setCursor(0,      0);lcd.print("STUDENT      4      PRESENT
");delay(2000);
      // lcd.setCursor(0, 1);lcd.print("PERSON");
      upload1("STUDENT_4_PRESENT");delay(2000);
     }
    }
   delay(2000);
   //lcd.clear();lcd.setCursor(0, 0);lcd.print("SELECT OPTION");
}

}

int ct=0;
char dummy=0x0f;
int fpenroll(char id)
{

  serialFlush();
  //enroll buffer send 12 bytes
  for(int i =0;i<12;i++)
     Serial.write(enroll[i]);
  res[9] = 1;//
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){}
  else{err=1;return -1;}

  //generate ch buffer
  for(int i =0;i<13;i++)
     Serial.write(generate_ch[i]);
```

```
res[9] = 1;
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=2;return -1;}

//enroll buffer send 12 bytes
for(int i =0;i<12;i++)
    Serial.write(enroll[i]);
res[9] = 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=3;return -1;}

//generate ch1 buffer
for(int i =0;i<13;i++)
    Serial.write(generate_ch1[i]);
res[9] = 1;
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=4;return -1;}

//uncmd buffer send 12 bytes
for(int i =0;i<12;i++)
    Serial.write(un_cmd[i]);
res[9] = 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=5;return -1;}

//store buffer send 12 bytes
```

```
  for(int i =0;i<12;i++)
     Serial.write(store[i]);
  dummy = 0x0f+id;
  Serial.write((uint8_t)id);
  Serial.write((uint8_t)0x00);
  Serial.write((uint8_t)dummy);
  res[9] = 1;//
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){return id;}
  else{err=6;return -1;}
}
int fpsearch()
{
  ct=0;
   serialFlush();
  //enroll buffer send 12 bytes
  for(int i =0;i<12;i++)
     Serial.write(enroll[i]);
  res[9] = 1;//
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){}
  else{err=1;return -1;}


   //generate ch buffer
  for(int i =0;i<13;i++)
     Serial.write(generate_ch[i]);
  res[9] = 1;
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){}
  else{err=2;return -1;}
```

```
  //enroll buffer send 12 bytes
  for(int i =0;i<17;i++)
     Serial.write(identify[i]);
  res[9] = 1;//
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){return (int)res[11];}
  else{err=1;return -1;}

}




char bf2[50];
void upload1(unsigned char *chr)
{

  lcd.clear();lcd.setCursor(0, 1);lcd.print("UPLOADING");
  delay(2000);
  serialFlush();
  mySerial.println("AT+CIPSTART=4,\"TCP\",\"embeddedspot.top\",80");

     delay(8000);
     sprintf(buff,"GET
http://embeddedspot.top/iot/storedata.php?name=iot304&s1=%s\r\n\r\n",chr);
     serialFlush();
     sprintf(bf2,"AT+CIPSEND=4,%u",strlen(buff));
     mySerial.println(bf2);

        delay(5000);
```

```
    serialFlush();
    mySerial.print(buff);

      delay(2000);

      mySerial.println("AT+CIPCLOSE");
   lcd.setCursor(0, 1);lcd.print("UPLOADDED");

}
```

CHAPTER-5

RESULTS AND OUTPUTS

5.1 RESULT:

The outcome of the project "Biometric attendance system over IOT" after aligning all components is as shown in figure 5.1

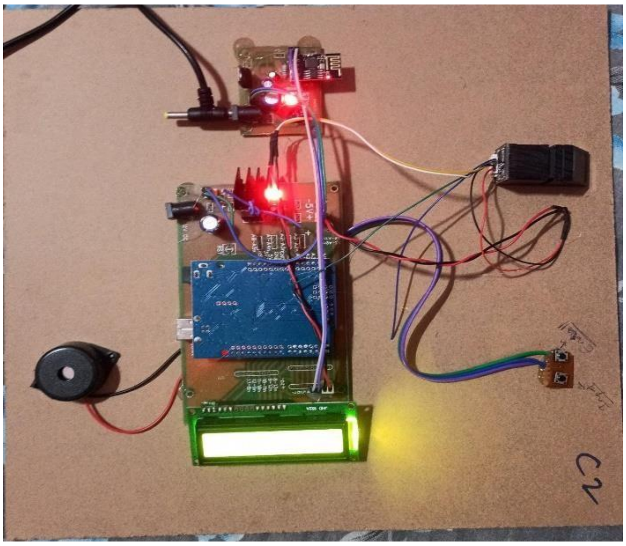


Figure 5.1 Biometric attendance system over IOT.

Initially , the system is set to ON by using a power supply unit of a 12 volts battery and regulated to a voltage of 5 to power the Arduino.

Initially, the system requires connectivity to the internet which can be achieved through Wi-FI module hence , when system is powered ON ,it first scans all wi-fi networks and any network can be connected by entering the password. once the connection is established, it scans for the fingerprint and stores the fingerprint of respective student.

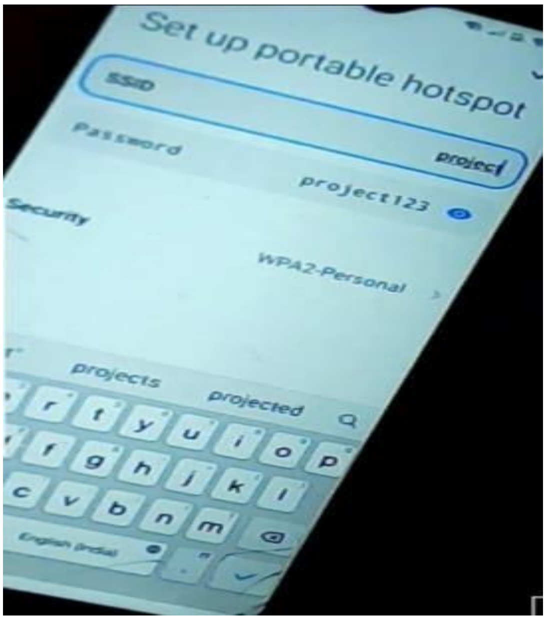


Figure 5.2 When the system is connecting to internet

.

After enrolling the respective student or employee’s fingerprint, now it scans the fingerprint, if the fingerprint of the current person matches with already enrolled student fingerprint then the student data is displayed on LCD display and the respective student data is stored on to the server that is on IOT Gecko platform and student or HOD or any faculty can check the attendance data by providing their respective username and password on IOT Gecko platform.

Figure 5.3 Enrolling student fingerprint.

Figure 5.4 We have to place finger for checking.

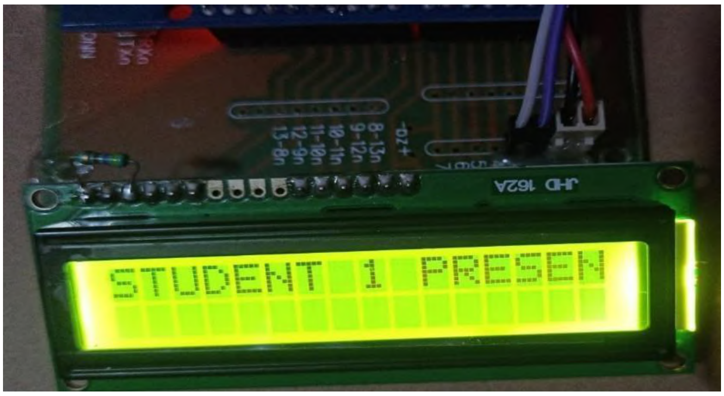
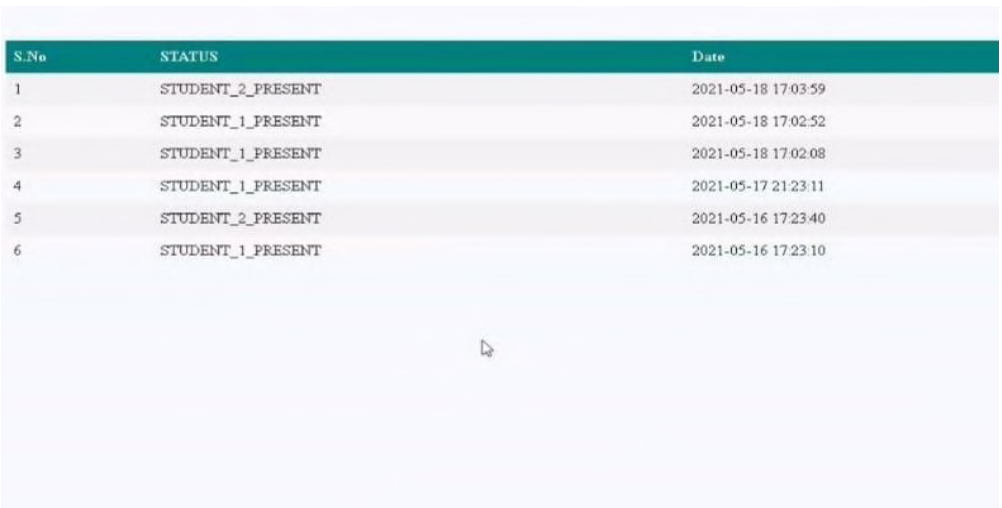


Figure 5.5 Student data is displayed.



S.No	STATUS	Date
1	STUDENT_2_PRESENT	2021-05-18 17:03:59
2	STUDENT_1_PRESENT	2021-05-18 17:02:52
3	STUDENT_1_PRESENT	2021-05-18 17:02:08
4	STUDENT_1_PRESENT	2021-05-17 21:23:11
5	STUDENT_2_PRESENT	2021-05-16 17:23:40
6	STUDENT_1_PRESENT	2021-05-16 17:23:10

Figure 5.6 Student data stored in server.

Figure 5.7 when fingerprint does not match.

If the fingerprint of the person does not match with the fingerprint of enrolled fingerprint of the person, then it displays as enrolled failed as shown in fig:5.7 on LCD display and the buzzer makes a beep sound indicating that an unauthorized student had placed a finger and thus, no data will be sent to server.

CHAPTER-6

APPLICATIONS, ADVANTAGES AND LIMITATIONS

6.1 APPLICATIONS:

- Transactions via e-commerce.
- Search of digital libraries.
- Computer logins
- Access to internet and local networks.
- Document encryption.
- Credit cards and ATM cards.
- Access to office buildings and homes
- Protecting personal property.
- Attendance system.

6.2 ADVANTAGES:

- Very accurate fingerprint reading and storing
- Cost Effective
- Can be installed in small spaces
- Fingerprint is stored via cloud
- LCD display for username, and time of operation
- Buzzer is used for indicating
- Proxy or attendance cheating can be avoided.

6.3 LIMITATIONS:

- Limited number of fingerprints are stored.
- As the whole thing is connected to internet so any problem related to internet can cause disruption to the whole thing.
- Will be difficult for physically people.
- Not a web-oriented system.
- Does not work when there is no Internet.

CHAPTER-7

CONCLUSION AND FUTURESCOPE

7.1 CONCLUSION:

The traditional process of manually taking and maintaining student attendance is highly inefficient and time consuming. Our project that is “The biometric attendance system over IOT” has a potential to streamline the whole process. Hence it is proved that by using this project we can easily monitor the student attendance data from anyplace using IOT.

7.2 FUTURE SCOPE:

- The system can be configured to enable lecture-wise attendance taking.
- It can further be improved to automatically calculate attendance percentages of students and intimate the teachers if a student’s attendance is below a certain percentage.
- It can also be modified to fit the corporate environment.

REFERENCES

[1] Upasana Ghosh Dastidar, Nikhita Jogi, Milan Bansod, Payal Madamwar and Priyanka Jalan, "FINGERPRINT SENSOR BASED ATTENDANCE SYSTEM USING ATMEGA 328 AND ESP8266", *International Journal of Research In Science & Engineering*, vol. 3, no. 2, pp. 471-475, March-April 2017.

[2] Vishal Suryawanshi, Kiran Puri, Prashant Devkar and K.S. Tiwari, *Attendance Monitoring System-Automation-Using-Fingerprint-Module ISSN (Online)*, vol. 5, no. 1, pp. 2347-2820, 2017.

[3] Prashik S. Bhagat, Prof. D. S. Shilwant, Prof. S. P. Kharde, Praful S. Bhagat, Abhijit S. Andure, Prof. Amol A. Shirsath, “Iris based attendance system”, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 4 Issue 8, August 2015.

[4] Sagar Wale, S.A. Patil, "Indigenous Development Of Automated Wireless Fingerprint Attendance System",INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 3, ISSUE 8, AUGUST 2014.

- Symposium on Computer Modeling and Simulation, 09/2008. [5] Vishal Bhalla, Tapodhan Singla, Ankit Gahlot, Vijay Gupta, “Bluetooth Based Attendance Management System”, International Journal of Innovations in Engineering and Technology (IJIET), Vol. 3 Issue 1 October 2013.
- [6] Vanaja Roselin.E.Chirchi, Dr.L.M.Waghmare, E.R.Chirchi, “Iris Biometric Recognition for Person Identification in Security Systems”, International Journal of Computer Applications, Volume-24-No. 9, June 2011.
- [7] Quratulain Shafi, Javaria Khan, Nosheen Munir, Naveed Khan Baloch, "Fingerprint Verification over the Network and its Application in Attendance Management", 2010 International Conference on Electronics and Information Engineering (ICEIE 2010).
- [8] C. Middendorff, Multi-Biometric Approaches to Ear Biometrics and Soft Biometrics, A dissertation Submitted to the Graduate School of the University of Notre Dame, 2010.

[9] Deepak Ranjan Nayak. "A Novel Architecture for Embedded Biometric Authentication System", 2008 Second UKSIM European

A MAJOR PROJECT REPORT ON

TRIPLET MARKOV CHAIN IN IMAGE SEGMENTATION

Submitted by

1)Mr. A. Sai Kumar (17K81A04C1)
2)Mr. B. Pavan Kumar (17K81A04D3)
3)Mr. G. Mallesh (18K85A0425)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

ST.MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)
Dhulapally, Secunderabad – 500 100

MAY 2021

CERTIFICATE

Certified that this project report titled "***TRIPLET MARKOV CHAIN IN IMAGE SEGMENTATION***", is a *bona fide* work of 1. **Mr. A. SAI KUMAR (17K81A04C1) 2. Mr. B. PAVAN KUMAR (17K81A04D3) 3. Mr. G. MALLESH (18K85A0425)** who carried out the work under my supervision, for the partial fulfillment of the requirements for the award of the degree of *Bachelor of Technology* in *Electronics and Communication Engineering.* Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion.

Place: Secunderabad

Date:

Dr. B. Hari Krishna
B.Tech., M. Tech, Ph.D.,
Professor &HOD
Department of Electronics and Communication Engineering
St. Martin's Engineering College,

COUNTER SIGNED

HEAD
Department of Electronics and Communication Engineering
St. Martin's Engineering College

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **A.SAI KUMAR** WITH ROLL NO.**17K81A04C1, B.PAVAN KUMAR** WITH ROLL NO.**17K81A04D3**, **G.MALLESH** WITH ROLL NO.**18K85A0425,** OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**TRIPLET MARKOV CHAIN IN IMAGES SEGMENTATION**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT

DIRECTOR

TRAININGS & PLACEMENTS LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road,
Secunderabad 500014 Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We declare that this project report titled ***Triplet Markov Chain In Image Segmentation*** submitted in partial fulfillment of the degree of B. Tech **in *Electronics and Communication Engineering*** record of original work carried out by us under the guidance and supervision of **Mr. V. V. Ramana Rao**, and has not formed the name basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

1. A. Sai kumar	2. B. Pavan Kumar	3. G. Mallesh
17K81A04C1	17K81A04D3	18K85AA0425

ACKNOWLEDGEMENTS

While bringing out this project report to its final form, we came across a number of people whose contributions in various ways helped my field of research and they deserve special thanks. It is a pleasure to convey our gratitude to all of them.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr. Santosh Kumar Patra**, **our beloved Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr. B. HARI KRISHNA**, **Head of the Professor**, the Dept. of **Electronics and Communication Engineering** for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept. project** coordinator **Mr. V. V. Ramana Rao** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We are highly grateful and obliged to **our college level Project Coordinators Dr. Maalyadri**, **Professor in IT** and **Dr. R. Santosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. A. Sai Kumar
2. B. Pavan Kumar
3. G. Mallesh

ABSTRACT

Over the last years, image segmentation has evolved from a sub-discipline of computer science to a technique widely used in medical imaging, automated object recognition, and remote sensing. In this work, we present a recently Markovian model of image segmentation called Triplet Markov Chain with Independent Noise (TMC-IN), in this model, it assumes that its hidden process X is non-stationary. TMC-IN is used in this to segment some textured grey level and color images. To estimate the parameters, we use the iterative algorithm EM (Expectation Maximization) and we apply MPM (Marginal Posteriori Mode) algorithm to estimate the result segmented image. In addition, we compare the obtained results by this model with those obtained by the stationary Hidden Markov Chain with Independent Noise (HMC-IN) model. Experimental results show that TMC-IN outperforms HMC-IN in all experiments.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter-1

INTRODUCTION TO DIGITAL IMAGE PROCESSING

1.1 IMAGE:

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person. two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surface.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphic technology, or developed by a combination of methods, especially in a pseudo-photograph.

Fig 1.1 General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color

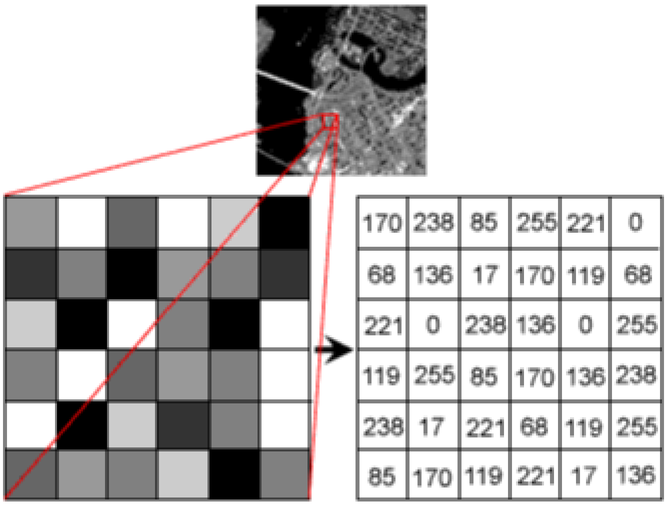


Fig 1.2 Image pixel

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

Fig1.3 Transparency Image

1.2 IMAGE FILE SIZES:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High-resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

1.3 IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

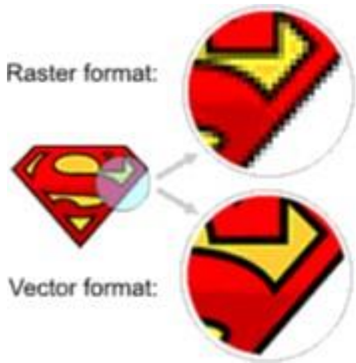


Fig1.4 Resolution image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

1.4 RASTER FORMATS and VECTOR FORMATS:

Raster Formats:

These formats store images as bitmaps (also known as pixmaps)

- **JPEG/JFIF:**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

- **EXIF:**

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF formatwith TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Itspurpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata is recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, the name of the camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed

- **TIFF:**

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by webbrowsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

- **GIF:**

GIF (Graphics Interchange Format) is limited to an 8-bit palette or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

- **PNG:**

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed- color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

- **BMP:**

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicityand wide acceptance in Windows programs.

VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM:**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary fileor one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

- **SVG:**

SVG (Scalable Vector Graphics) is an open standard created and developed by the WorldWide Web Consortium to address the need for a versatile, scriptable and all-purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due tothe textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

1.5 IMAGE PROCESSING:

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man’s ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields suffers from myths, miss-connect ions, misunderstandings and misinformation. It is the vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor plowed with imprecise jargon.

Several factors combines to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low-costmicroprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

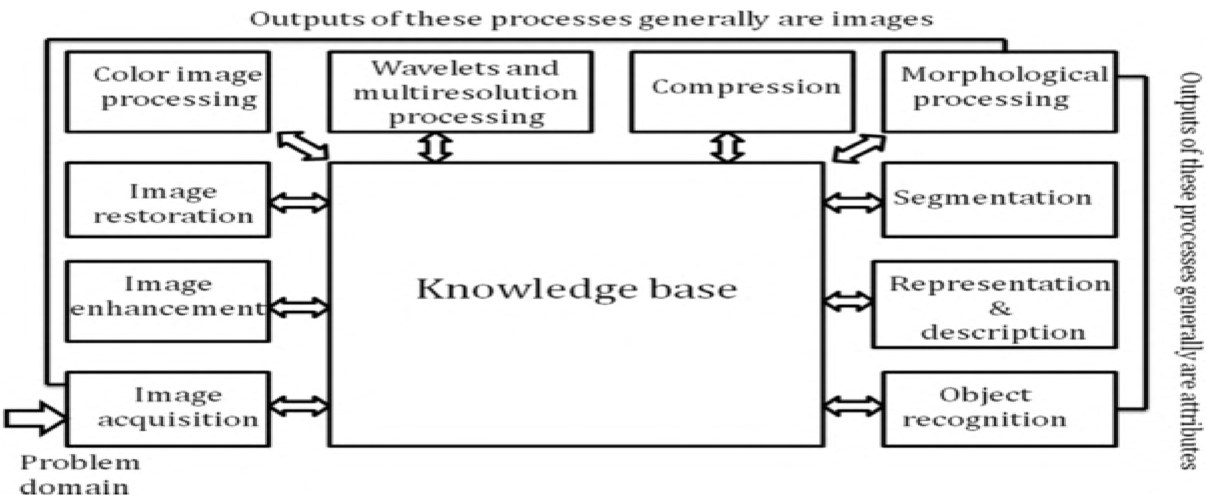


Fig 1.5: Image Processing

1.5.1 Image Acquisition:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line ata time. In this case, the objects motion past the line.

Fig 1.6 Digital Camera Image

The scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Fig 1.7 Digital Camera Cell

1.5.2 Image Enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, orsimply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because “it looks better.” It is important to keep in mindthat enhancement is a very subjective area of image processing.

Fig 1.8 Image enhancement

1.5.3Image restoration:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Fig 1.9 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, whereas removal of image blur by applying a deblurring function is considereda restoration technique.

1.5.4 Color image processing:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about onlytwo dozen shades of gray. This second factor is particularly important in manual image analysis.

Fig 1.10 Color & Gray scale image.

1.5.5 Wavelets and multiresolution processing:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform based image processing since the late1950's, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values,called Wavelets, of varying frequency and limited duration.

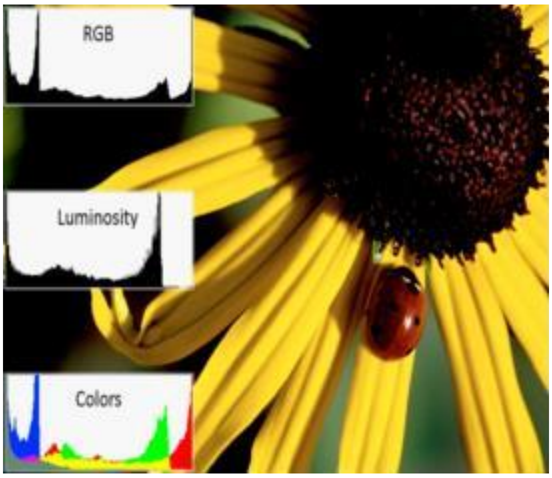


Fig 1.11 RGB Histogram Image.

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called **Multiresolution** theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub-band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

1.5.6Compression:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which is characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts image compression standard.

1.57Morphological processing:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

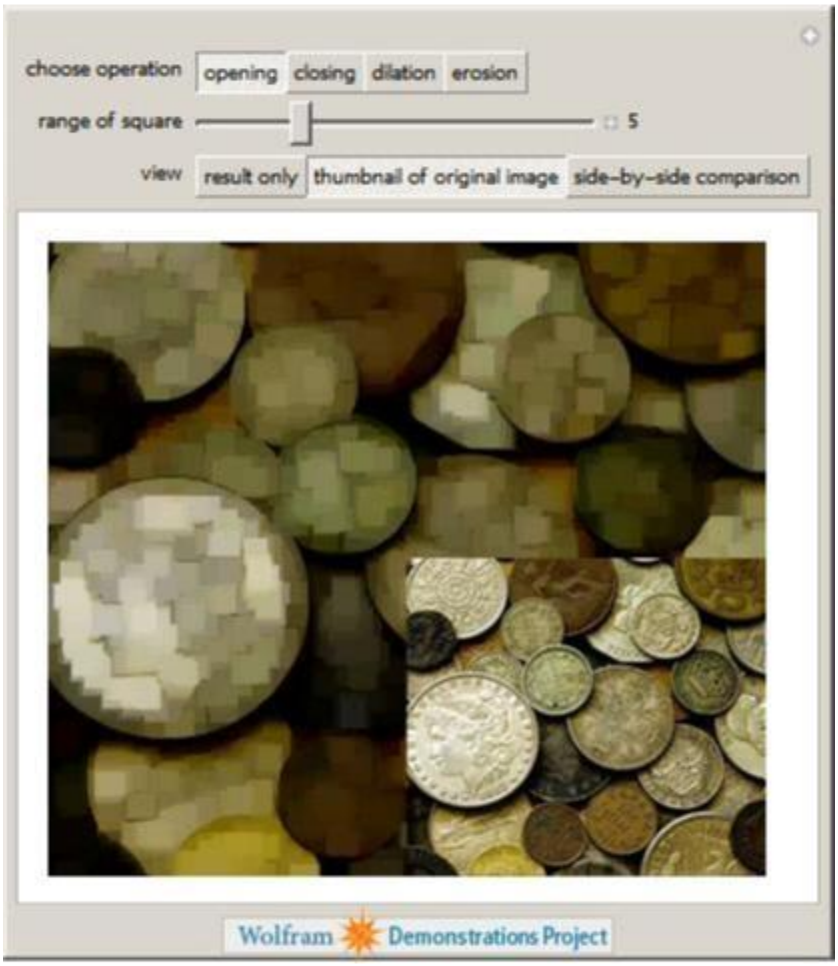


Fig 1.12 :- blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

1.5.8Segmentation:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward a successful solution of imaging problems that require objects to be identified individually.

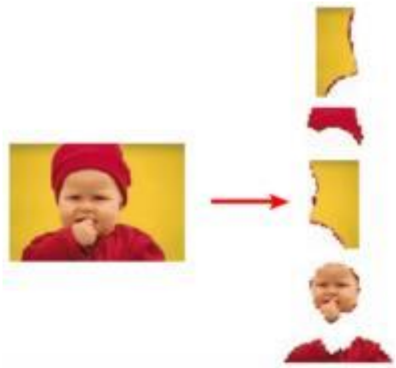


Fig 1.13 Image segmentation

1.5.9Representation and Description:

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

1.5.10Knowledge Base:

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an interrelated to list of all major possible defects in a materials inspection problem or an image database containing high-resolution satellite images of a region inconnection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

1.5.11Object Recognition:

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

1.6 Feature Extraction and Feature Matching Techniques :

1.6.1Oriented FAST And Rotated BRIEF(ORB) :

ORB, starts by finding special regions in an image called key points. Key points are the highly-distinctive locations in an image. For example, the edges of an image (where the brightness varies i.e where the pixel values change sharply).

How does ORB work?

1. First, it uses FAST to find key points, then applies a Harris corner measure to find top N pointsamong them. FAST stands for Features from Accelerated Segments Test, and it quickly selectskey points by comparing the brightness levels in a given pixel area.

2. BRIEF, to create feature vectors.

1.6.2 Feature Detection using FAST(Features from Accelerated Segments Test):

1. Select a pixel p in the image which is to be identified as an interesting point or not. Let its intensity be I.

2.Select the appropriate threshold value t.

3.Consider a circle of 16 pixels around the pixel under test. (See the image below).

1.6.3BRIEF (Binary Robust Independent Elementary Features):

The second part of the ORB algorithm is to take the key points found by the FAST algorithm and turn those into feature vectors that together can represent an object. To create feature vectors, ORB uses a BRIEF algorithm.

It basically creates binary feature vectors from a set of key points. As we already know a binary feature vector, is also known as a binary descriptor, which is just a feature vector that contains only ones and zeros

BRIEF provides a shortcut to find the binary strings directly. It takes a smoothened image patch and selects a set nd(x,y) location pairs in a unique way and then pixel intensity comparisons are done on these location pairs.

In short, BRIEF is a faster method feature descriptor calculation and matching. It also provides a high recognition rate unless there is a large in-plane rotation.

1.6.4 2Nearest Neighbour(2NN) with Hierachical agglomerative clustering(HAC):

There are many approaches to hierarchical clustering as it is not possible to investigate all clustering possibilities.One set of approaches to hierarchical clustering is known as agglomerative, whereby in each step of the clustering process an observation or cluster is merged into another cluster.

Hierarchical clustering is a widely used and popular tool in statistics and data mining for grouping data into 'clusters' that exposes similarities or dissimilarities in the data. There are many approaches to hierarchical clustering as it is not possible to investigate all clustering possibilities. One set of approaches to hierarchical clustering is known as agglomerative, whereby in each step of the clustering process an observation or cluster is merged into another cluster. The first approach we will explore is known as ‘**Single Linkage’method**’ also known as nearest neighbors.

1.6.5**Noise :**

Noise Images acquired through modern sensors may be contaminated by a variety of noise sources. By noise we refer to stochastic variations as opposed to deterministic distortions such as shading or lack of focus. We will assume for this section that we are dealing with images formed from light using modern electrooptics. In particular we will assume the use of modern, charge-coupled device (CCD) cameras where photons produce electrons that are commonly referred to as photoelectrons. Nevertheless, most of the observations we shall make about noise and its various sources hold equally well for other imaging modalities. While modern technology has made it possible to reduce the noise levels associated with various electro-optical devices to almost negligible levels, one noise source can never be eliminated and thus forms the limiting case when all other noise sources are "eliminated".

1.7 IMAGE RETRIEVAL:

Image retrieval is nothing but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords or the descriptions to the images so that the retrieval can be performed. Manual image annotation is time consuming, expensive and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc..,

Invention of the digital camera has given the common man the privilege to capture his worldin pictures, and conveniently share them with others. one can today generate volumes of images with content as diverse as family get-togethers and national park visits. Low-cost storage and easy Web hosting has fueled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchableimage data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real-world image search systemdesigners.

As far as technological advances are concerned, growth in Content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real world implications, applications, and constraints of the technology.Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future.

An image retrieval system designed to serve a personal collection should focus on features such as personalization, flexibility of browsing, and display methodology. For example,

Google's Picasa system [Picasa 2004] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware supportto the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods. The notable few applications include an FPGA implementation of a color-histogram-based image retrieval system [Kotoulas and Andreadis 2003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 2003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 2002].

Discussion. Regardless of the nature of the collection, as the expected user-base grows, factors such as concurrent query support, efficient caching, and parallel and distributed processing of requests become critical. For future real-world image retrieval systems, both software and hardware approaches to address these issues are essential. More realistically, dedicated specialized servers, optimized memory and storage support, and highly parallelizable image search algorithms to exploit cluster computing powers are where the future of large-scale image search hardware support lies.

Chapter 2
LITERATURE SURVEY

2.1 Introduction

The residence time of hidden process X is also represented by an auxiliary process, in this case, the Markov model is a semi Markovian model. There exists a Markovian model which combines the non-stationarity and the residence time at the same time [13], in this case, it should to introduce two auxiliary processes : U1 to model stationarity and U2 to represent the residence time. Evidential is a very recently triplet Markov model, it is also a triplet model, where, the probabilities p are replaced by evidential masses m [14] and, the fusion of Dempster-Shafer [15] is applied to estimate the final configuration of the resulted image. In each category of Markovian models, we distinguish three representations to model the image before segmentation: Fields [5], Chains [5] and Trees [16], each representation has its own principle to model image. In general, Markovian segmentation models have the same principle to estimate the image result $X = \{x1, ..., xN \}$ from the observed image $Y = \{y1, ..., yN \} \in R$. They use the final Bayesian decision criteria such as Viterbi algorithm [17] or MPM [18]. In this work, our study focused on Non-Stationary Hidden Markov Chain with Independent Noise model to segment some textured grey level images of Brodatz database images [19] [20] and some textured color images of VisTex database images [21]. We compared this triplet model with the classical representation Hidden Markov Chain with Independent Noise, according to some measures of quality such as PSNR index [22], SSIM index [23] and error rate. To estimate parameters, we used EM algorithm and MPM estimator. This paper is organised as follows : Section 2 presents classical model of Markov Hidden Chain with Independent Noise, and Triplet Markov Chain with Independent Noise model. Section 3 is devoted to EM algorithm and MPM algorithm. Section 4 illustrates the experimental results. The last section concludes the paper.

2.2 Markov Chains

Markov chains are quite useful in modeling computational linguistics. A Markov chain is a memory less stochastic model that describes the behaviour of an integer-valued random process. The behaviour is the simple form of dependency in which the next state (or event) depends only on the current state. According to [4], a random process is said to be Markov if the future of the process, given the present, is independent of the past. To describe the transitions between states, a transition diagram is used to describe the model and the probabilities of going from one state to another. For example, Figure 1 shows a Markov chain diagram with three states (Easy, Ok, and Hard) that belong to exam cases (i.e. states). In the figure, each arc represents the probability value for transition from one state to another.

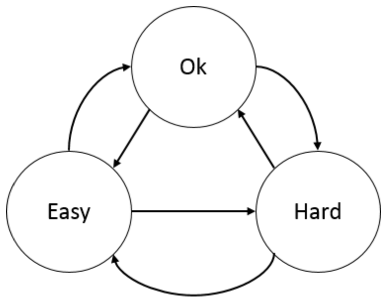


Figure 2.1. A Simple Markov chain with three states

The Markov chain diagrams are generally represented using state transition matrices that denote the transition probabilities from one state to another. Hence, a state transition matrix is created using the entire states in the system. For example, if a particular textual application has a training data that contains N states (e.g. the size of lexicon), then the state transition matrix is described by a matrix A= {aij} of size N*N. In matrix A, the element aij denote the transition probability from a state i to a state j. Table 1 shows how the state transition matrix used to characterize the Markov diagram shown in Figure 1. That is, the matrix carries the state transitions probabilities between the involved states (Easy, Ok, and Hard). For illustration, the P(E|H) denote to the probability of the next exam to be Easy given that the previous exam was Hard.

State		**Next Exam**		
		Easy (E)	**Ok (O)**	**Hard (H)**
Previous Exam	**Easy (E)**	P(E|E)	P(O|E)	P(H|E)
	Ok (O)	P(E|O)	P(O|O)	P(H|O)
	Hard (H)	P(E|H)	P(O|H)	P(H|H)

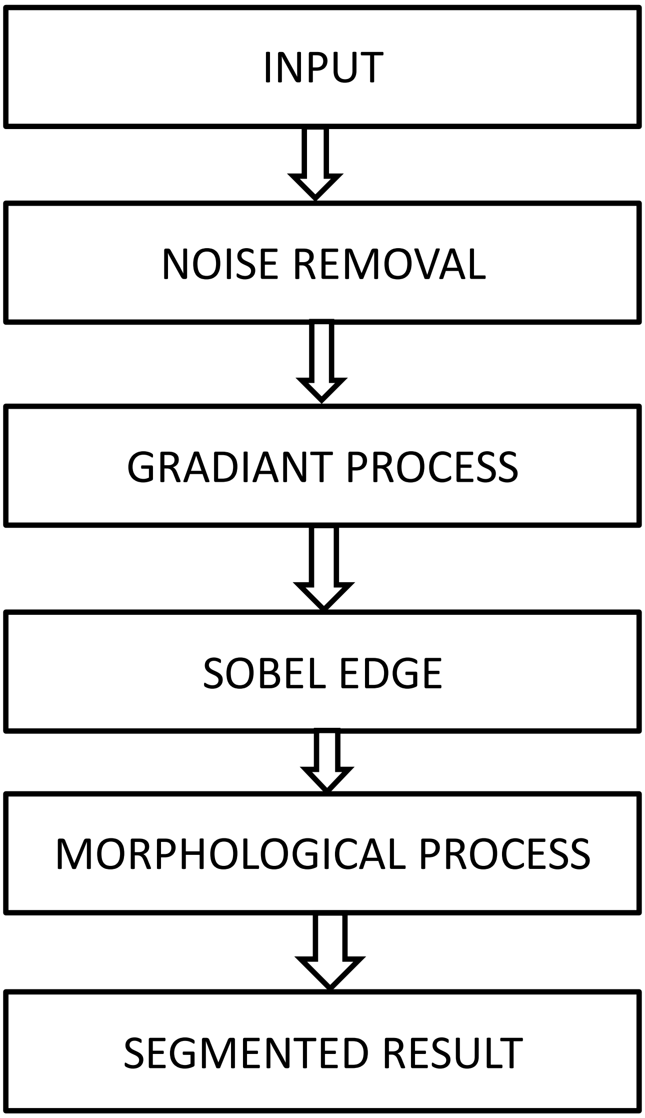
Table 2.1. A state transition matrix of three states

In Table 1, the sum of the probability values at each row is 1 as the the sum of the probabilities coming out of each node should be 1. Hence,P(E|E)+P(O|E)+P(P(H|E) equal 1. Markov chain is a worthy topic that has many details. For examples, it contains discrete-time, continuous-time, time reversed, reversible and irreducible Markov chains. The case shown in figure 1 is irreducible case, also called ergodic, where it is possible to go from every state to every state.

CHAPTER-3

PROPOSED SYSTEM

3.1 Block Diagram:



3.2 Noise Removal:

Noise is always present in digital images during image acquisition, coding, transmission, and processing steps. It is very difficult to remove noise from the digital images without the prior knowledge of filtering techniques. These filters can be selected by analysis of the noise behaviour. In this way, a complete and quantitative analysis of noise and their best suited filters will be presented.

Filtering image data is a standard process used in almost every image processing system. Filters are used for this purpose. They remove noise from images by preserving the details of the same. The choice of filter depends on the filter behaviour and type of data.

3.3 Gradient Process:

An **image gradient** is a directional change in the intensity or color in an image. The gradient of the image is one of the fundamental building blocks in image processing. For example, the Canny edge detector uses image gradient for edge detection. In graphics software for digital image editing, the term gradient or color gradient is also used for a gradual blend of color which can be considered as an even gradation from low to high values, as used from white to black in the images to the right. Another name for this is *color progression*.

Mathematically, the gradient of a two-variable function (here the image intensity function) at each image point is a 2D vector with the components given by the derivatives in the horizontal and vertical directions. At each image point, the gradient vector points in the direction of largest possible intensity increase, and the length of the gradient vector corresponds to the rate of change in that direction.

3.4 Sobel Operator:

It is a discrete differentiation operator. It computes the gradient approximation of image intensity function for image edge detection. At the pixels of an image, the Sobel operator produces either the normal to a vector or the corresponding gradient vector. It uses two 3 x 3 kernels or masks which are convolved with the input image to calculate the vertical and horizontal derivative approximations respectively.

Advantages:

1. Simple and time efficient computation
2. Very easy at searching for smooth edges

3.5 Morphological processing:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Fig 3.1: blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

3.6 Image segmentation:

What's the first thing you do when you're attempting to cross the road? We typically look left and right, take stock of the vehicles on the road, and make our decision. Our brain is able to analyze, in a matter of milliseconds, what kind of vehicle (car, bus, truck, auto, etc.) is coming towards us. We are able to build computer vision models that can detect objects, determine their shape, predict the direction the objects will go in, and many other things. You might have guessed it – that's the powerful technology behind self-driving cars!

Now, there are multiple ways of dealing with computer vision challenges. The most popular approach I have come across is based on identifying the objects present in an image, aka, object detection. In this project, I will introduce you to the concept of image segmentation. It is a powerful computer vision algorithm that builds upon the idea of object detection and takes us to a whole new level of working with image data. This technique opens up so many possibilities it has blown my mind.

CHAPTER 4

SOFTWARE INTRODUCTION:

4.1. Introduction to MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn

and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

4.2 The MATLAB system:

The MATLAB system consists of five main parts

- **Development Environment**:

    This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library**:

    This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language**:

    This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both “programming in the small” to rapidly create quick and dirty throw-away programs, and “programming in the large” to create large and complex application programs.

- **Graphics:**

    MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

4.3 GRAPHICAL USER INTERFACE (GUI):

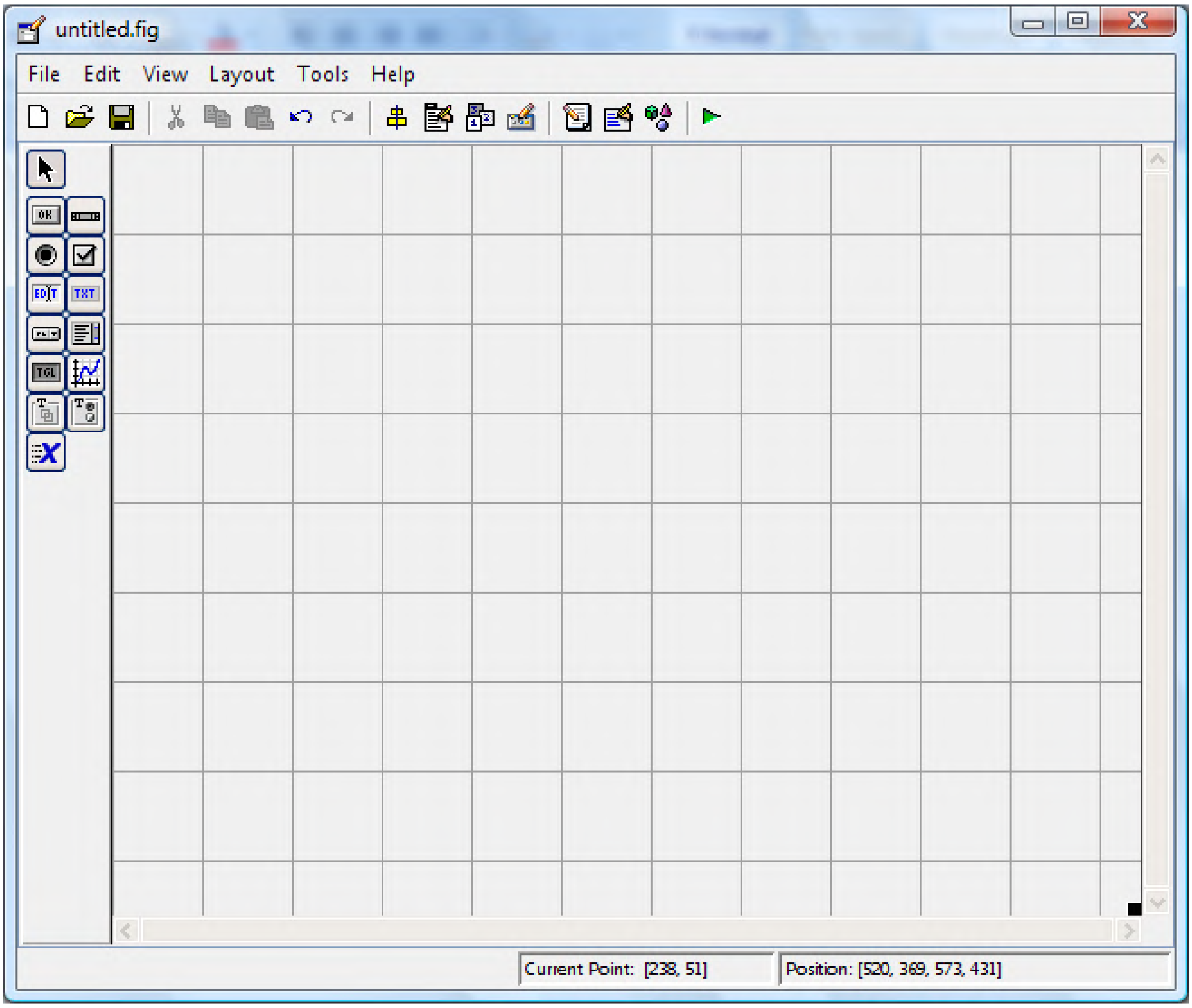
MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.

- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type
guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted,
GUIDE opens a new (i.e., blank) window.



A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

4.4 Getting Started

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

4.4.1 Introduction - describes the components of the MATLAB system.

4.4.2 Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

4.4.3 Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

4.4.4 Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

4.4.5 Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

4.5 DEVELOPMENT ENVIRONMENT

4.5.1Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

4.5.2 Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

4.5.3 Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

4.5.4 MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

4.5.5 Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type helpbrowser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to runMATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

4.6 MANIPULATING MATRICES

4.6.1 **Entering Matrices**

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Dürer's matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]

MATLAB displays the matrix you just entered.

```
A =
   16     3     2    13
    5    10    11     8
    9     6     7    12
    4    15    14     1
```

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

4.6.2 Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

4.6.3 Operators

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

Table 4.1: Operators

4.6.4 Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.14159265...
I	Imaginary unit, √-1
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

Table 4.2: Functions

4.7 GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

4.7.1 Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

4.7.2 GUI Development Environment

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of its children (uicontrols and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the callbacks, which are defined as subfunctions. This M-file is referred to as the application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

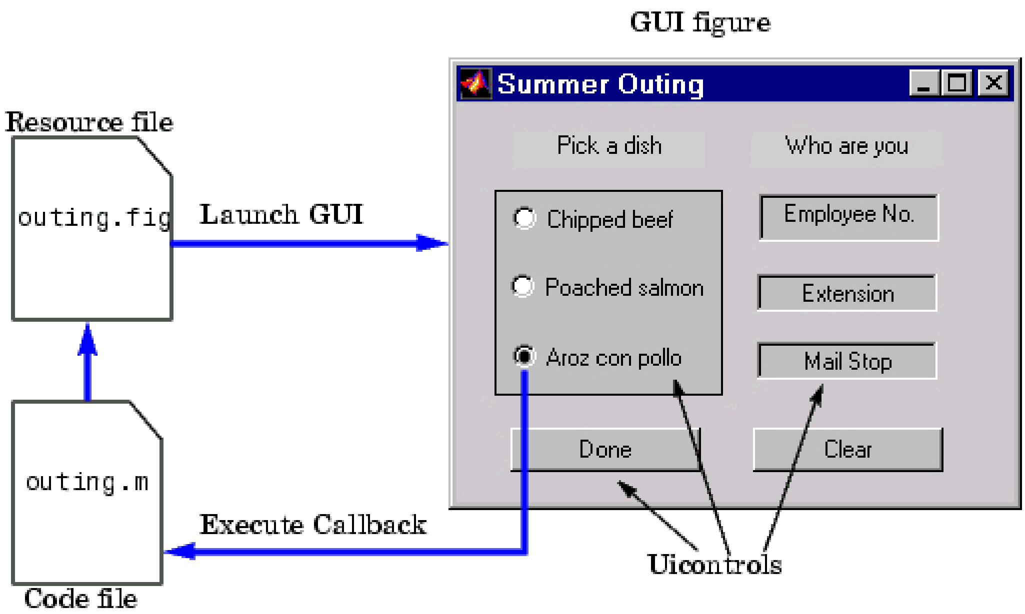


FIG 4.1: graphical user blocks

4.7.3 Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

4.7.4 Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques used in the application M-file.

Application Examples - a collection of examples that illustrate techniques which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

4.7.5 User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- Listboxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)
button_state = get(h,'Value');
if button_state == get(h,'Max')
   % toggle button is pressed
elseif button_state == get(h,'Min')
   % toggle button is not pressed
```

end

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);
a(:,:,2) = rand(16,128);
a(:,:,3) = rand(16,128);
set(h,'CData',a)
```

Radio Buttons

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)
set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)
off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];
mutual_exclude(off)
% Continue with callback
   .
   .
   .
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)
if (get(h,'Value') == get(h,'Max'))
   % then checkbox is checked-take approriate action
else
   % checkbox is not checked-take approriate action
end
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)
user_string = get(h,'string');
% proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)
user_entry = str2double(get(h,'string'));
if isnan(user_entry)
   errordlg('You must enter a numeric value','Bad Input','modal')
end
% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menubar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menubar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

Listboxes differentiate between single and double clicks on an item and set the figure SelectionType property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure SelectionType property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)
val = get(h,'Value');
switch val
case 1
```

```
% The user selected the first item
case 2
% The user selected the second item
% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)
val = get(h,'Value');
string_list = get(h,'String');
selected_string = string_list{val}; % convert from cell array to string
% etc.
```

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational
off - The control is disabled and its label (set by the string property) is grayed out.
inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes ButtonDownFcn property to define the callback.

4.7.6 Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

```
axes(handles.axes1)
```

makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axes. See GUI with Multiple Axes for and example that uses two axes.

Figure

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control.

CHAPTER 5

OUTPUTS:

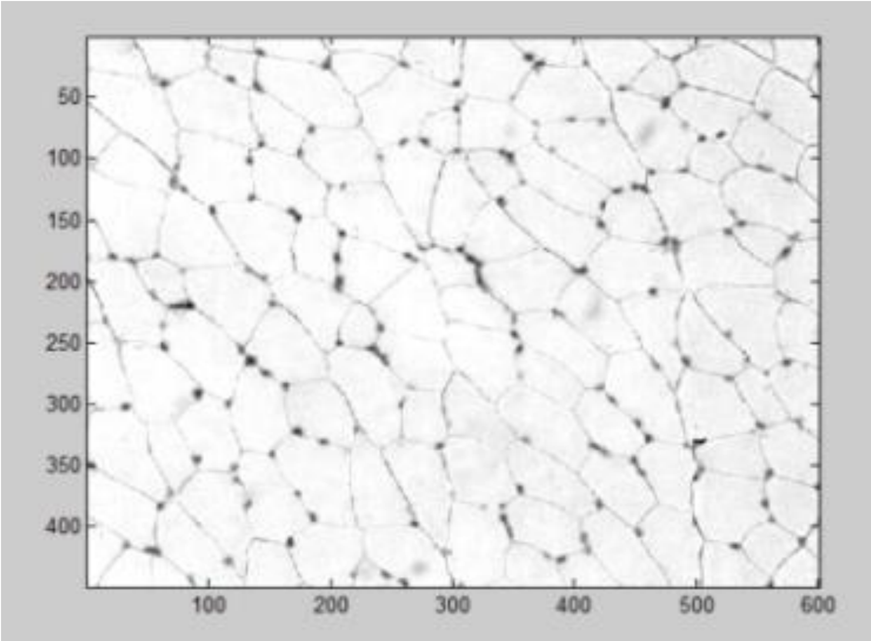


Fig 5.1: Input Image

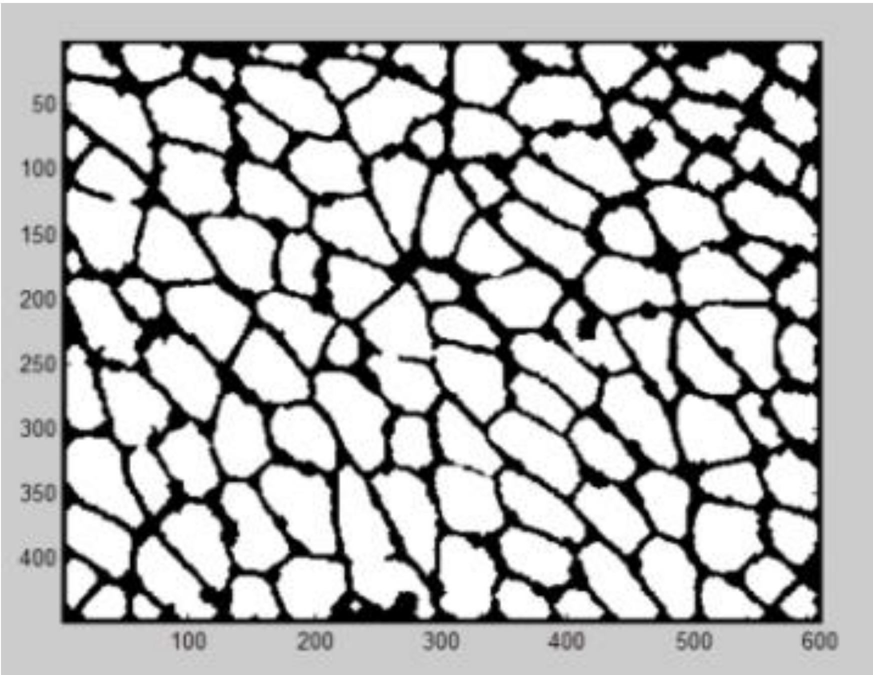


Fig 5.2: Image without Noise

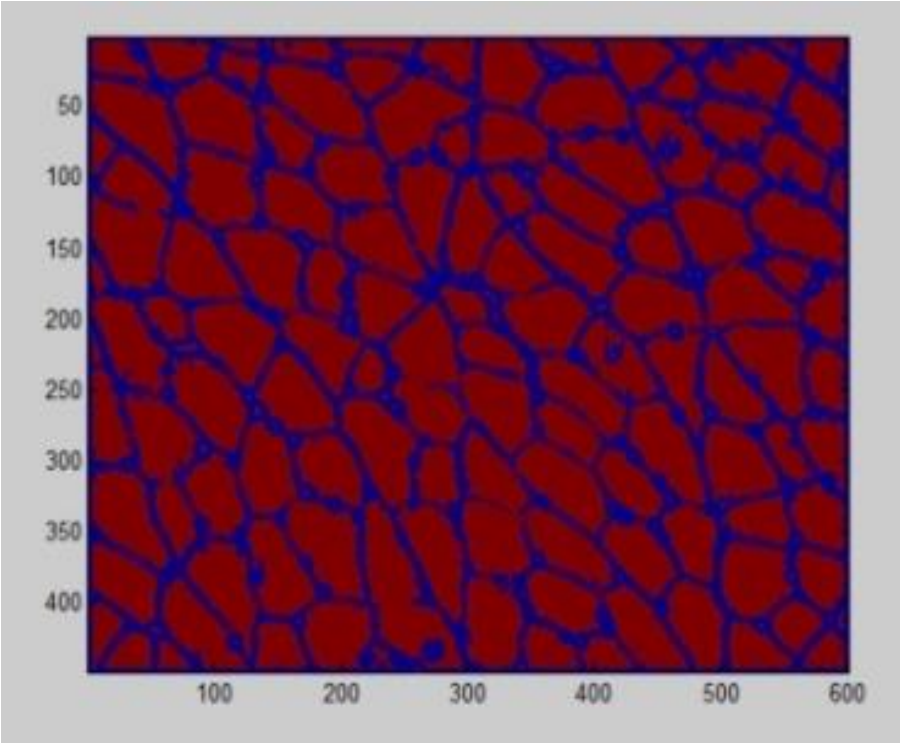


Fig 5.3: Gradient Image

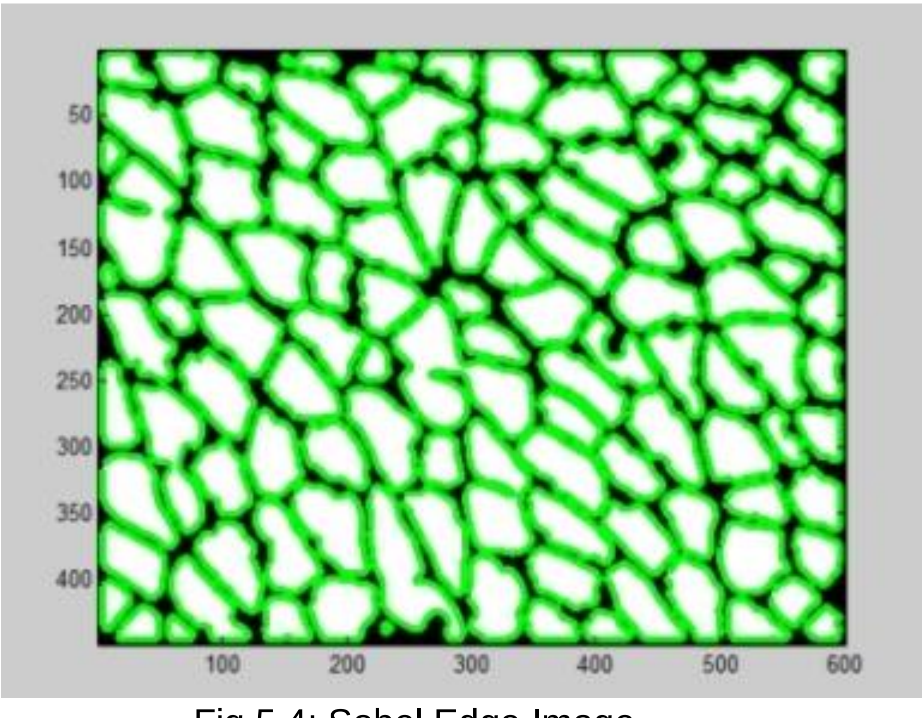


Fig 5.4: Sobel Edge Image

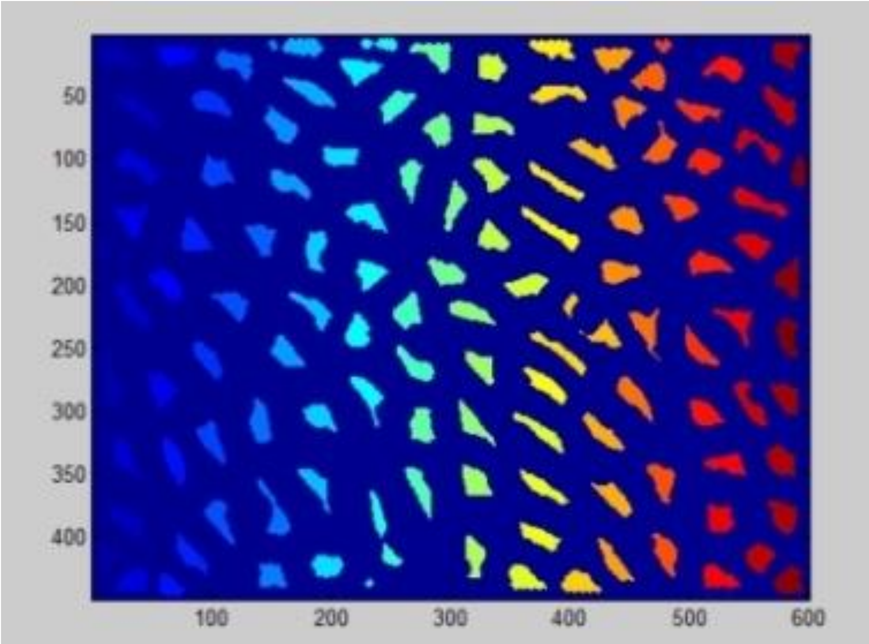


Fig 5.5: Morphological Processed image

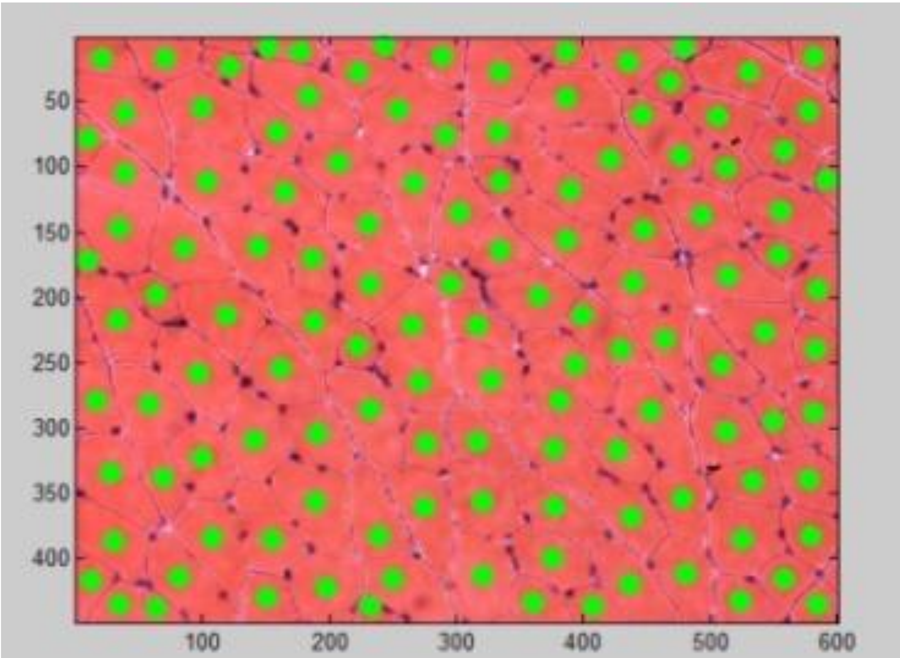


Fig 5.6: Segmented Result

CHAPTER 6

ADVANTAGES, LIMITATIONS & APPLICATIONS

6.1 Advantages:

1. Simple concept it performs well with noise.
2. We can choose multiple criteria at the same time.
3. Works well with low computation complexity in most of the images.
4. Easy to perceive for human.
5. Works well for image with good contrast.
6. Easy task to implementation.

6.2 Limitations:

1. Dependence to selected seed points.
2. In terms of computation time and memory are expensive,
3. Does not work well with close colour spectrum.
4. Very sensitive to noise.
5. Does not work well on images with low contrast and smooth change.

6.3 Applications:

1. The quantification of tissues volumes.
2. In diagnosis.
3. Localization of pathology.
4. Study of anatomical structure.
5. Treatment planning.
6. Computer integrated surgery.

CHAPTER 7
CONCLUSION

In this project, we have used a Triplet Markov chain to segment some textured grey level and color images, this model models the non-stationarity of the hidden process X by introducing an auxiliary process U. We have compared the obtained results by this model with those obtained by the classical Hidden Markov chain model. In the all experiments TMC model gives better results of segmentation than the HMC model, because it takes into account the non-stationarity of data. But, HMC executes the instructions faster than TMC, HMC calculates K2 + 2K parameters and TMC calculates (K ∗M)2 + 2K parameters in each iteration. Moreover, TMC estimates two process X and U at the same time, and HMC estimates just one process X. Our work will be finished by some open questions, in particular:

• To segment color 3D images using triplet Markov chain.

• To find some techniques to reduce the execution time of TMC model.

REFERENCES

[1] R. V. Handel, Hidden Markov models, July 28,2008.

[2] P. Lanchatin, “Chaines de Markov triplet et segmentation non supervisée de signaux,” Thése de doctorat de l’Institut National des Telecomminucations, pp 6-15, 5 Decembre 2006.

[3] H. Gales, S. Young,“The application of hidden Markov models in speech recognition,” Signal Processing, vol.1, No.3,195-304, 2007.

[4] N. Brunel, W. Pieczynski, “Signal restoration using hidden Markov chains with copulas,” Signal Processing, 85, 2304-2315, 2005.

[5] N. Rechid, al, “Segmentation non supervisée d’images basée sur les modèles de Markov cachés,” Courrier de savoir, No 12, pp 39-34, octobre 2011.

[6] P. Lanchantin , W. Pieczynski, “Unsupervised Restoration of Hidden non stationary Markov Chains Using Evidential Priors,” IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 53, NO. 8, AUGUST 2005.

[7] Y. Deng, al, Color image segmentation, 2000. [8] S. Derrode , W. Pieczynski , “Segmentation non supervisée d’images par une chaine de Markov couple”, 2004.

[9] W. Pieczynski, “Pairwise Markov chain,” IEEE TRANSACTION Pattern Analysis Machine Intellegence, vol.25, No.5, may 2005.

[10] W. Pieczynski, “Modèles de Markov en traitement des images,” Traitement du signal, vol.20, No.3 ,pp 255-278, 2003.

[11] W. Pieczynski,“Triplet Markov Chains and Evidence,” Submitted to International Journal of Approximate Reasoning, September 2003.

[12] W. Pieczynski, “EM and ICE in hidden and triplet Markov models,” Stochastic Modeling Techniques and analysis, relax International Conference, pp 8-11,june 2010.

[13] P. Lanchantin, J. Lapuyade-Lahorgue, W. Pieczynski, “Unsupervised segmentation of triplet Markov chains hidden with long-memory noise,” Signal Processing 88 (2008) 1134–1151, November 2007.

[14] P. Lanchantin , W. Pieczynski,“Unsupervised Restoration of Hidden non stationary Markov Chains Using Evidential Priors,” IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 53, NO. 8, AUGUST 2005.

[15] M.E.Y. Boudaren, W. Pieczynski, “Dempster–Shafer fusion of evidential pairwise Markov chains,” IEEE Trans. Fuzzy Syst., http://dx.doi.org/10.1109/TFUZZ.2016.2543750, 2016.

[16] P. Lanchantin, F. Salezenstein, “Segmentation d’images multispectrales par arbres de Markov Cachés floues,” 2005.
[17] G. D Fornay , “the Viterbi algorithm,” Procceding of the IEEE, vol.61, No.3, pp 268-277, 1970.
[18] M. L. Corner and E. J. delp, “The EM/ MPM Algorithm for segmentation of textured images,” pp 1731-1744, October, 2000.

A Major Project report on

BRAMSIT: A Database For Brain Tumor Diagnosis And Detection

Submitted By

S. DivyaLahari (17K81A04H2)

Y. Sai Kankshitha (17K81A04H9)

CH. Sai Lokesh (18K85A0435)

In partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Mr.K. Karthik

Assistant Professor

Department of Electronics And Communications

ST.MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

Certified that this project reported title" BRAMSIT: A DATABASE FOR BRAIN TUMOR DIAGNOSIS AND DETECTION", is a bonafide work of 1) Ms. S. DivyaLahari (17K81A04H2) 2) Ms. Y. Sai Kankshitha (17K81A04H9) 3) Mr. CH. Sai Lokesh (18K85A0435) who carried out the work under my supervision, for the partial fulfilment of the requirements for the award of the degree of *Bachelor of Technology in Electronics And Communication Engineering.* Certified further that to the best of my knowledge and belief, the work reported here in does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion.

INTERNAL GUIDE

Mr. K. Karthik

Assistant Professor (M.Tech)

Department of ECE

HEAD OF DEPARTMENT

Dr. B. Hari Krishna, Professor

Department of ECE

EXTERNAL EXAMINER

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **CH.SAI LOKESH WITH ROLL NO.18K85A0435, S.DIVYA LAHARI WITH ROLL NO.17K81A04H2, Y. SAI KANKSHITHA WITH ROLL NO.17K81A04H9**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD**, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**BRAMSIT: A DATABASE FOR BRAIN TUMOR DIAGNOSIS AND DETECTION**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We declare that this project report titled BRAMSIT: A DATABASE FOR BRAIN TUMOR DIAGNOSIS AND DETECTION submitted in partial fulfillment of the degree of B. Tech in Electronics And Communication Engineering record of original work carried out by us under the guidance and supervision of Mr. K. Karthik and has not formed the name basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

1) Ms. S. DivyaLahari	2) Ms. Y. Sai Kankshitha	3) Mr. CH. Sai Lokesh
(17K81A04H2)	(17K81A04H9)	(18K85A0435)

Place:

Date:

ACKNOWLEDGEMENTS

While bringing out this project report to its final form, we came across a number of people whose contributions in various ways helped us in field of research and they deserve special thanks. It is a pleasure to convey our gratitude to all of them.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr. Santosh Kumar Patra**, **our beloved Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to Dr.B. Hari Krishna, **Head &Professor**, the Dept. of Electronics And Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him.

We would like to express our gratitude to **our dept .project coordinator** Dr.A.Anand for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide, **Mr.K. Karthik, Assistant Professor,** Department of ECE and our Class project Coordinator **Mr.K. Nishakar, Associate Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

We are highly grateful and obliged to **our college level Project Coordinators Dr. Maalyadri**, **Professor in IT** and **Dr. R. Santosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

We dedicate this thesis to almighty

1. Ms.S. DivyaLahari (17K81A04H2)
2. Ms.Y. Sai Kankshitha (17K81A04H9)
3. Mr. CH. Sai Lokesh (18K85A0435)

ABSTRACT

MRI is the most frequently used imaging technique to detect brain tumor. The brain is composed of nerve cells and supportive tissues such as glial cells and meninges. A brain tumor is a collection, or mass, of the brain in abnormal cells. Primary brain tumors can be either malignant or benign . A primary brain tumor is a tumor located in the brain tissue. New technologies in supplement to existing imaging modalities improve brain tumor screening. Most brain tumor databases are not publicly available. BRAMSIT is a resource for possible use by the MRI image analysis research community. The projected MRI database is a termed BRAMSIT, characterized by an attempt to offer a group of normal and malignant brain tumor images. The details such as age, and the MRI axial position (i.e., trans-axial, coronal and sagittal) of the patient are interpreted in the database.

Magnetic Resonance Imaging (MRI) plays an important role in the medical era. It is non invasive method to detect Brain tumor. MRI uses Magnetic field and Radio waves to produce Brain images. MRIs create more detailed pictures than CT scans. MRI is the preferred screening examination for brain tumor. The goal is to detect brain tumor before clinical signs are noticeable. The size of the tumor can be measured by MRI. A special dye called contrast medium is injected into the patient's vein or given as a pill or fluid to swallow before an MRI scan.

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Brain tumor is a life- threatening disease. Brain tumors can be malignant or benign. When tumor cells grow it cause pressure inside the skull, this leads to brain damage. Two types of Brain tumor namely Primary and secondary. Primary brain tumors are benign and that are originates in brain. Secondary brain tumors occur when cancer cells spread over the brain from other organ such as lung or breast. It is also called as metastatic brain tumor. Brain tumor can be occurred in any ages. If brain tumor detected early stage it is treatable. Brain tumor cause more death in children and adults under the age of 40 than any other cancer. In India the tumors ranges from 5 to 10 per 100,000 populations with an increasing trend.

Diagnosing brain tumor is a tedious process. Computer Tomography scan (CT scan), Magnetic Resonance Imaging(MRI), tests like an Angiogram, Spinal tab and Biopsy are used to diagnosis brain tumors. CT scan uses x-ray to produce images. CT scan expose patients to ionizing radiation .High radiation are involved in CT scan . A new study in the Journal of the National Cancer Institute suggests that CT scans, commonly used in medical imaging, may increase the risk of brain tumors.

Magnetic Resonance Imaging (MRI) plays an important role in the medical era. It is noninvasive method to detect Brain tumor. MRI uses Magnetic field and Radio waves to produce Brain images. MRIs create more detailed pictures than CT scans. MRI is the preferred screening examination for brain tumor. The goal is to detect brain tumor before clinical signs are noticeable. The size of the tumor can be measured by MRI. A special dye called contrast medium is injected into the patient's vein or given as a pill or fluid to swallow before an MRI scan.

Any area with abnormal tissue can cause for cancer. The abnormalities will be examined by radiologist. A focused white area on MRI can be a lump or tumor. Tumors can be cancerous or benign. If a tumor is benign, it is not a risky and is unlikely to grow or change shape. The radiologist will check its shape and pattern, as they can sometimes be a sign of cancer.

1.2 PROJECT DESCRIPTION

Image Processing techniques are used to detect brain tumor according to the following steps

- Pre-processing
- Segmentation
- Feature Extraction

1.2.1 PRE-PROCESSING AND ENHANCEMENT OF AN IMAGE

This is the first step of image processing it is used to enhance the chances of detecting the suspicious region. Finer details of the image are enhanced and noise is removed from the image.

1.2.2 IMAGE SEGMENTATION

Image segmentation is the method of breaking down an image into small parts. Segmentation is performed to make the analysis easier. There are following types of image segmentation.

- Boundary approach or Thresholding
- Edge approach
- Region approach

1.2.3 FEATURE EXTRACTION

Extracting the exact tumor is a crucial task in case of brain tumor because of the complex structure of brain. Certain parameters are taken into account for feature extraction as size, shape, composition, location of image. As per the result obtained from the feature extraction the classification of the tumor is done.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

Tumor which is abnormal cells which are expands in our body but in brain tumor a mass of tissue in our brain cell are expanded rapidly [1].The symptoms of brain tumor depends upon tumor type, size and location of the tumor [2].The symptoms are headache vomiting nausea some other changes are speech vision and hearing problems sometime they can changes their activities like personality, ability to concentrate and problem with memory. A brain tumor is categorized primary and secondary depends upon location size and its origin.

- **Benign:** Benign tumors which type of non-cancerous mass of cells that it was expands very slowly in the brain. It was does not spread and stay in one place. Normally benign brain tumors can be detected by CT and MRI scans.
- **Maligant** :Malignant brain tumor was rapidly expand like cancer that spreads one area to other area of the brain and spine it [3].
- **Materials and Methods**: The MRI report is obtained from Government Medical College, Jagdalpur under the guidance of Rajat Kumar Pandey(MBBS, Final Part II ,Student) that includes one normal image and one abnormal image of MRI and CT report. All the image processing operation are done in MATLAB. The paper organized as below. Section II describes about the detail discussion about schematics of the study. Section III gives the parameters comparison. Section IV discusses the results. At last, Section V concludes the paper with the conclusion.

CHAPTER 3

INTRODUCTION TO IMAGE PROCESSING

3.1 INTRODUCTION

3.1.1 IMAGE

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Fig 3.1 General Image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

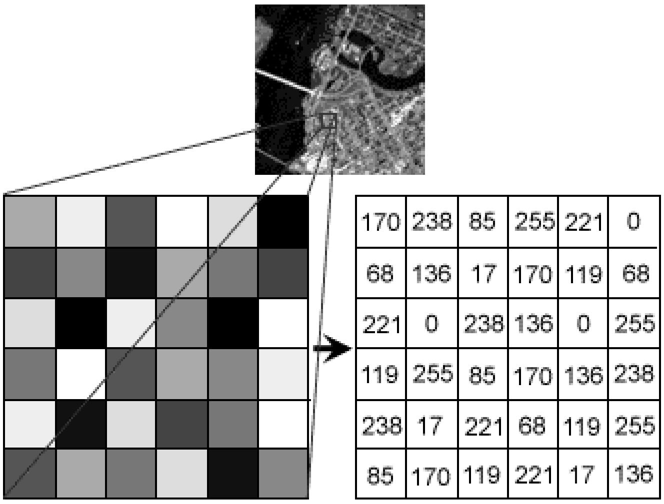


Fig 3.2 Image Pixel

Each pixel has a color. The color is a 33-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

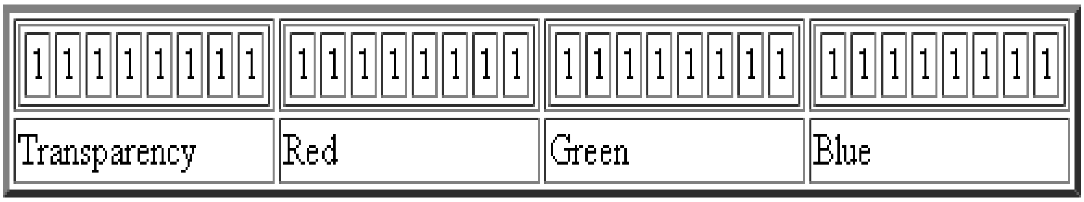


Fig 3.3 Transparency Image

3.2 IMAGE FILE SIZES

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 356 colors, a 34-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 13 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 13 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

3.3 IMAGE FILE FORMATS

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

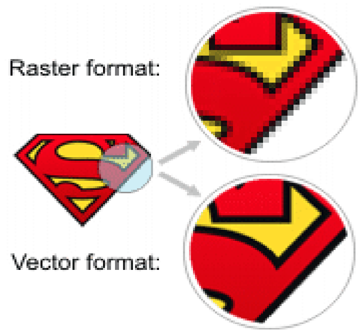


Fig 3.4 Resolution Image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

3.3.1 RASTER FORMATS

These formats store images as bitmaps (also known as pix maps)

3.3.1.1 JPEG/JFIF

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 34-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

3.3.1.2 EXIF

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

3.3.1.3 TIFF

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 34-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

3.3.1.4 PNG

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 356 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

3.3.1.5 GIF

GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 356 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

3.3.1.6 BMP

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

3.3.2 VECTOR FORMATS

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

3.3.2.1 CGM

CGM (Computer Graphics Metafile) is a file format for 3D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 3D graphical information independent from any particular application, system, platform, or device.

3.3.2.2 SVG

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

3.4 IMAGE PROCESSING

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavour ploughed with imprecise jargon.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

3.5 FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING

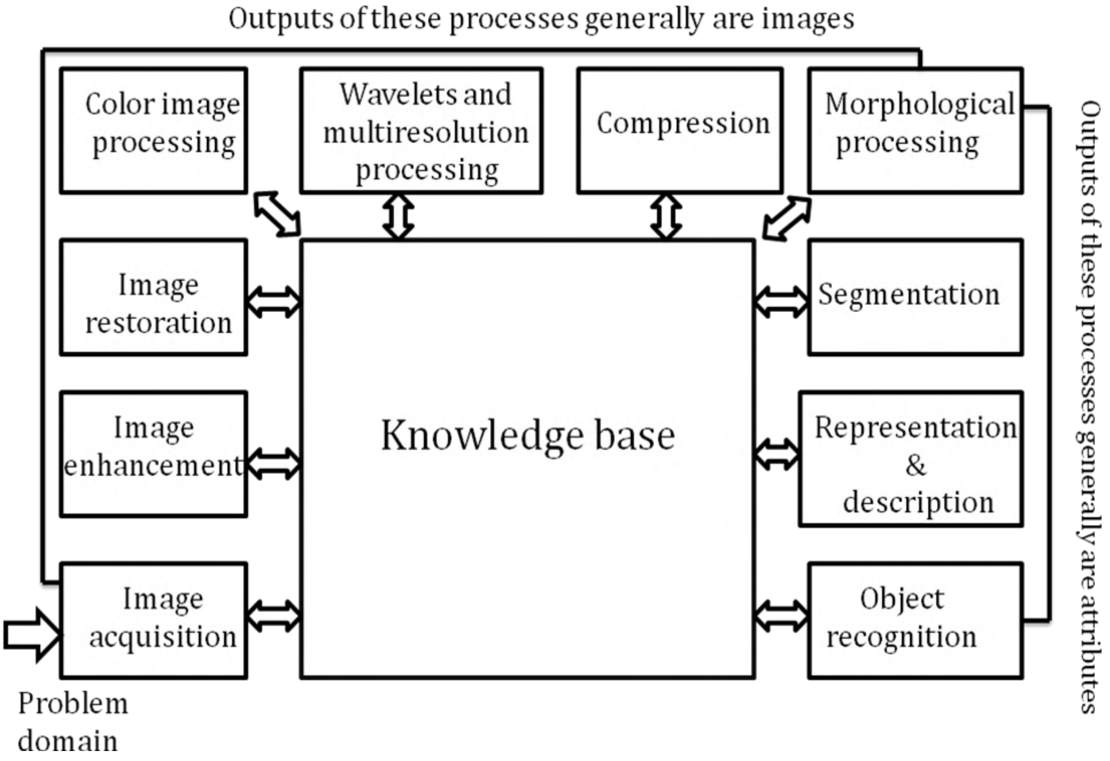


Fig 3.5 Image fundamental

3.5.1 IMAGE ACQUISITION

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Fig 3.6 Digital Camera Image

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Fig 3.7 Digital Camera Cell

3.5.2 IMAGE ENHANCEMENT

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because “it looks better.” It is important to keep in mind that enhancement is a very subjective area of image processing.

Fig 3.8 Image Enhancement

3.5.3 IMAGE RESTORATION

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Fig 3.9 Image Restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, whereas removal of image blur by applying a deblurring function is considered a restoration technique.

3.5.4 COLOR IMAGE PROCESSING

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Fig 3.10 Color & Gray Scale Image

3.5.5 WAVELETS AND MULTI RESOLUTION PROCESSING

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform based image processing since the late1950's, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyse many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

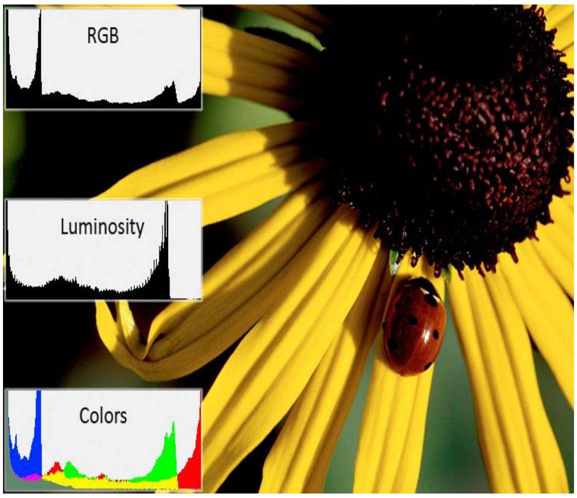


Fig 3.11 RGB Histogram Image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called Multi resolution theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

3.5.6 COMPRESSION

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

3.5.7 MORPHOLOGICAL PROCESSING

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Fig 3.12 Blur To Deblur Image

In binary images, the sets in question are members of the 3-D integer space Z^3, where each element of a set is a 3-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete Gray-level value.

3.5.8 SEGMENTATION

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

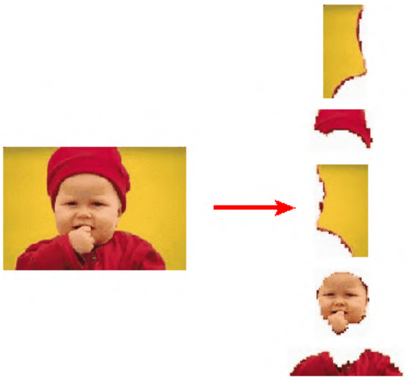


Fig 3.13 Image Segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

3.5.9 REPRESENTATION AND DESCRIPTION

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

3.5.10 OBJECT RECOGNITION

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

3.5.11 KNOWLEDGE BASE

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an intern related to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

3.6 COMPONENTS OF AN IMAGE PROCESSING SYSTEM

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

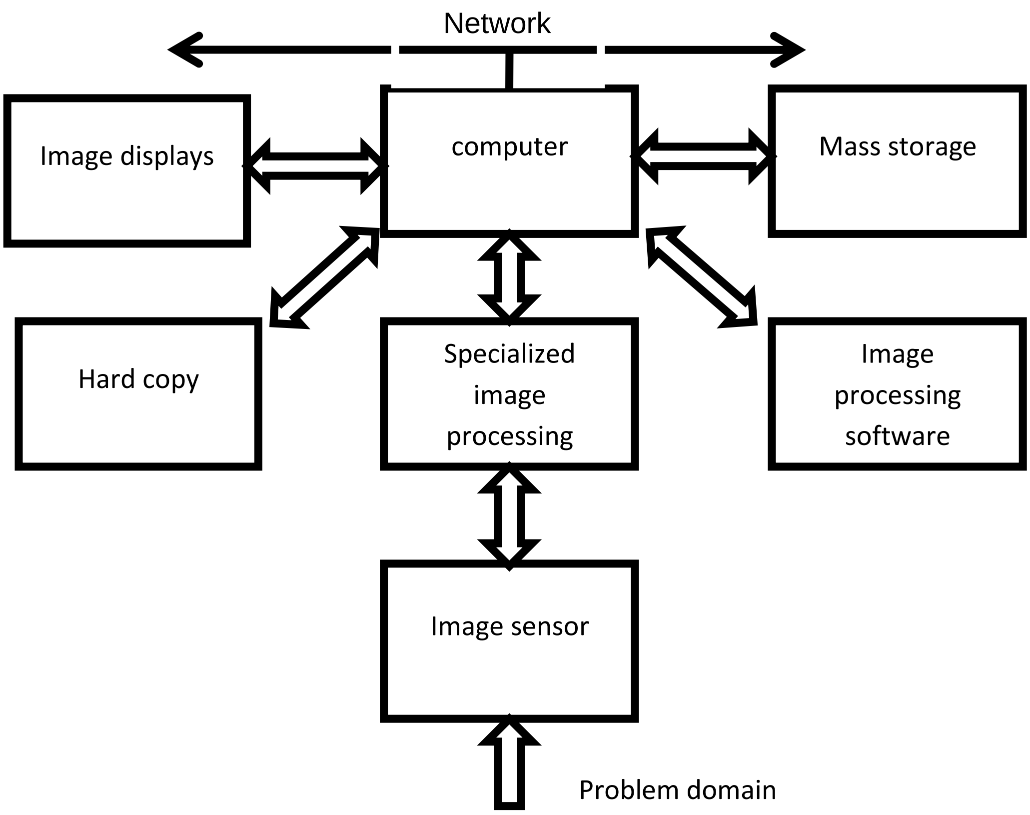


Fig 3.14 Component Of Image processing

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 1.34 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

3.6.1 IMAGE SENSORS

With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

3.6.2 SPECIALIZED IMAGE PROCESSING HARDWARE

Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

3.6.3 COMPUTER

The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

3.6.4 IMAGE PROCESSING SOFTWARE

Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

3.6.5 MASS STORAGE

Mass storage capability is a must in image processing applications. An image of size 1034*1034 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage for image processing applications fall into three principal categories:

- short-term storage for use during processing
- On-line storage for relatively fast recall
- archival storage, characterized by infrequent access.

Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), Gbytes (meaning giga, or one billion, bytes), and Tbytes (meaning tera, or one trillion, bytes)

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig. 1.34. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in "jukeboxes" are the usual media for archival applications.

3.6.6 IMAGE DISPLAYS

Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are there requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

3.6.7 HARDCOPY

Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

3.6.8 NETWORK

Networking is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of optical fiber and other broadband technologies.

Color and texture are two low-level features widely used for image classification, indexing and retrieval. Color is usually represented as a histogram, which is a first order statistical measure that captures global distribution of color in an image One of the main drawbacks of the histogram- based approaches is that the spatial distribution and local variations in color are ignored. Local spatial variation of pixel intensity is commonly used to capture texture information in an image. Grayscale Co-occurrence Matrix (GCM) is a well-known method for texture extraction in the spatial domain. A GCM stores the number of pixel neighbourhoods in an image that have a particular grayscale combination. Let I be an image and let p and Np respectively denote any arbitrary pixel and its neighbour in a given direction. If GL denotes the total number of quantized Gray levels and gl denotes the individual Gray levels, where, gl $\in$ {0, . . .,GL _ 1}, then each component of GCM can be written as follows:

$$\mathrm{gcm}(i,j) = \Pr((\mathrm{gl}_{\mathrm{p}}, \mathrm{gl}_{N_{\mathrm{p}}}) = (i,j))$$

gcm(i, j) is the number of times the Gray level of a pixel p denoted by glp equals i, and the Gray level of its neighbour Np denoted by glNp equals j, as a fraction of the total number of pixels in the image. Thus, it estimates the probability that the Gray level of an arbitrary pixel in an image is i, and that of its neighbour is j. One GCM matrix is generated for each possible neighbourhood direction, namely, 0, 45, 90 and 135.Average and range of 14 features like Angular Second Moment, Contrast, Correlation, etc., are generated by combining all the four matrices to get a total of 38 features. In the GCM approach for texture extraction, color information is completely lost since only pixel Gray levels are considered.

To incorporate spatial information along with the color of image pixels, a feature called color correlogram has recently been proposed. It is a three dimensional matrix that represents the probability of finding pixels of any two given colors at a distance ‘d’ apart Auto correlogram is a variation of correlogram, which represents the probability of finding two pixels with the same color at a distance ‘d’ apart. This approach can effectively represent color distribution in an image. However, correlogram features do not capture intensity variation Many image databases often contain both color as well as Gray scale images. The color correlogram method does not constitute a good descriptor in such databases.

Another method called Color Co-occurrence Matrix (CCM) has been proposed to capture color variation in an image. CCM is represented as a three-dimensional matrix, where color pair of the pixels p and Np are captured in the first two dimensions of the matrix and the spatial distance ‘d’ between these two pixels is captured in the third dimension. This approach is a generalization of the color correlogram and reduces to the pure color correlogram for d = 1. CCM is generated using only the Hue plane of the HSV (Hue, Saturation and Intensity Value) color space. The Hue axis is quantized into HL number of levels. If individual hue values are denoted by hl, where $\mathrm{hl} \in \{0, \ldots, \mathrm{HL} - 1\}$, then each component of CCM can be written as follows:

$$\mathrm{ccm}(i, j) = \Pr((\mathrm{hl}_{\mathrm{p}}, \mathrm{hl}_{N_{\mathrm{p}}}) = (i, j))$$

Four matrices representing neighbours at angles 0, 90, 180 and 370 are considered. This approach was further extended by separating the diagonal and the non-diagonal components of CCM to generate a Modified Color Co-occurrence Matrix (MCCM). MCCM, thus, may be written as follows: MCCM = (CCMD;CCMND)

Here, CCMD and CCMND correspond to the diagonal and off-diagonal components of CCM. The main drawback of this approach is that, like correlogram, it also captures only color information and intensity information is completely ignored.

An alternative approach is to capture intensity variation as a texture feature from an image and combine it with color features like histograms using suitable weights . One of the challenges of this approach is to determine suitable weights since these are highly application-dependent. In certain applications like Content-based Image Retrieval (CBIR), weights are often estimated from relevance feedback given by users.

While relevance feedback is sometimes effective, it makes the process of image retrieval user-dependent and iterative. There is also no guarantee on the convergence of the weight-learning algorithms. In order to overcome these problems, researchers have tried to combine color and texture features together during extraction.

proposed two approaches for capturing color and intensity variations from an image using the LUV color space. In the Single-channel Co-occurrence Matrix (SCM), variations for each color channel, namely, L, U and V are considered independently. In the Multi-channel Co-occurrence Matrix (MCM), variations are captured taking two channels at a time – UV, LU and LV. Since the LUV color space separates out chrominance (L and U) from luminance (V), SCM in effect, generates one GCM and two CCMs from each image independently. As a result, correlation between the color channels is lost.

However, in MCM, the count of pair wise occurrences of the values of different channels of the color space is captured. Thus, each component of MCM can be written as follows:

$$mcmUV(I; j) = Pr((up; vNp) = (I; j))$$

$$mcmLU(I; j) = Pr((lp; uNp) = (I; j))$$

$$mcmLV(I; j) = Pr((lp; vNp) = (I; j))$$

Here, mcmUV(i, j) is the number of times the U chromaticity value of a pixel p denoted by up equals i, and the V chromaticity value of its neighbour Np denoted by vNp equals j, as a fraction of the total number of pixels in the image. Similarly, mcmLU(i, j) and mcmLV(i, j) are defined. One MCM matrix is generated for each of the four neighbourhood directions, namely, 0, 45, 90 and 135.

Deng and Manjunath (3001) proposed a two-stage method called JSEG, which combines color and texture after image segmentation. In the first stage, colors are quantized to the required levels for differentiating between various regions of an image. Pixel values of the regions are then replaced by their quantized color levels to form a color map. Spatial variation of color levels between different regions in the map is viewed as a type of texture composition of the image.

Yu et al. (3003) suggested the use of color texture moments to represent both color and texture of an image. This approach is based on the calculation of local Fourier Transformation (LFT) coefficients. Eight templates equivalent to LFT are operated over an image to generate a characteristic map of the image. Each template is a $3 \cdot 3$ filter that considers eight neighbours of the current pixel for LFT calculation. First and second order moments of the characteristic map are then used to generate a set of features.

In this paper, we propose an integrated approach for capturing spatial variation of both color and intensity levels in the neighbourhood of each pixel using the HSV color space. In contrast to the other methods, for each pixel and its neighbour, the amount of color and intensity variation between them is estimated using a weight function. Suitable constraints are satisfied while choosing the weight function for effectively relating visual perception of color and the HSV color space properties. The color and intensity variations are represented in a single composite feature known as Integrated Color and Intensity Co-occurrence Matrix (ICICM). While the existing schemes generally treat color and intensity separately, the proposed method provides a composite view to both color and intensity variations in the same feature. The main advantage of using ICICM is that it avoids the use of weights to combine individual color and texture features. We use ICICM feature in an image retrieval application from large image databases.

Early result on this work was reported in (Vadivel et al., 3004a). In the next section, we describe the proposed feature extraction technique after introducing some of the properties of the HSV color space. Choice of quantization levels for color and intensity axes, selection of parameter values and a brief overview of the image retrieval application.

3.6.9 INTEGRATED COLOR AND INTENSITY CO-OCCURRENCE MATRIX

We propose to capture color and intensity variation around each pixel in a two-dimensional matrix called Integrated Color and Intensity Co-occurrence Matrix (ICICM). This is a generalization of the Grayscale Co-occurrence Matrix and the Color Co-occurrence Matrix techniques. For each pair of neighbouring pixels, we consider their contribution to both color perception as well as Gray level perception to the human eye. Some of the useful properties of the HSV color space and their relationship to human color perception are utilized for extracting this feature. In the next sub-section, we briefly explain relevant properties of the HSV color space. In the subsequent subsection, we describe how the properties can be effectively used for generating ICICM.

3.6.10 HSV COLOR SPACE

HSV Color space: Basically there are three properties or three dimensions of color that being hue, saturation and value HSV means Hue, Saturation and Value. It is important to look at because it describes the color based on three properties. It can create the full spectrum of colors by editing the HSV values. The first dimension is the Hue. Hue is the other name for the color or the complicated variation in the color. The quality of color as determined by its dominant wavelength. This Hue is broadly classified into three categories. They are primary Hue, Secondary Hue and Tertiary Hue. The first and the foremost is the primary Hue it consists of three colors they are red, yellow and blue. The secondary Hue is formed by the combination of the equal amount of colors of the primary Hue and the colors of the secondary Hue which was formed by the primary Hue are Orange, Green and violet. The remaining one is the tertiary Hue is formed by the combination of the primary Hue and the secondary Hue. The limitless number of colors are produced by mixing the colors of the primary Hue in different amounts. Saturation is the degree or the purity of color. Then the second dimension is the saturation. Saturation just gives the intensity to the colors. The saturation and intensity drops just by mixing the colors or by adding black to the color. By adding the white to the color in spite of more intense the color becomes lighter. Then finally the third dimension is the Value. The value is the brightness of the color. When the value is zero the color space is totally black with the increase in the color there is also increase in the brightness and shows the various colors. The value describes the contrast of the color. That means it describes the lightness and darkness of the color. As similar to the saturation this value consists of the tints and shades. Tints are the colors with the added white and shades are the colors with the added black.

3.6.11 PROPERTIES OF THE HSV COLOR SPACE

Sensing of light from an image in the layers of human retina is a complex process with rod cells contributing to scotopic or dim-light vision and cone cells to photopic or bright-light vision (Gonzalez and Woods, 3003). At low levels of illumination, only the rod cells are excited so that only Gray shades are perceived. As the illumination level increases, more and more cone cells are excited, resulting in increased color perception. Various color spaces have been introduced to represent and specify colors in a way suitable for storage, processing or transmission of color information in images. Out of these, HSV is one of the models that separate out the luminance component (Intensity) of a pixel color from its chrominance components (Hue and Saturation). Hue represents pure color, which is perceived when incident light is of sufficient illumination and contains a single wavelength. Saturation gives a measure of the degree by which a pure color is diluted by white light. For light with low illumination, corresponding intensity value in the HSV color space is also low.

The HSV color space can be represented as a Hexa cone, with the central vertical axis denoting the luminance component, I (often denoted by V for Intensity Value). Hue, is a chrominance component defined as an angle in the range [0,3p] relative to the red axis with red at angle 0, green at 3p/3, blue at 4p/3 and red again at 3p. Saturation, S, is the other chrominance component, measured as a radial distance from the central axis of the hexacode with value between 0 at the centre to 1 at the outer surface. For zero saturation, as the intensity is increased, we move from black to white through various shades of Gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of Gray to the most pure form of the color represented by its hue. When saturation is near 0, all the pixels in an image look alike even though their hue values are different.

As we increase saturation towards 1, the colors get separated out and are visually perceived as the true colors represented by their hues. Low saturation implies presence of a large number of spectral components in the incident light, causing loss of color information even though the illumination level is sufficiently high. Thus, for low values of saturation or intensity, we can approximate a pixel color by a Gray level while for higher saturation and intensity, the pixel color can be approximated by its hue. For low intensities, even for a high saturation, a pixel color is close to its Gray value. Similarly, for low saturation even for a high value of intensity, a pixel is perceived as Gray. We use these properties to estimate the degree by which a pixel contributes to color perception and Gray level perception.

One possible way of capturing color perception of a pixel is to choose suitable thresholds on the intensity and saturation. If the saturation and the intensity are above their respective thresholds, we may consider the pixel to have color dominance; else, it has Gray level dominance. However, such a hard thresholding does not properly capture color perception near the threshold values. This is due to the fact that there is no fixed level of illumination above which the cone cells get excited. Instead, there is a gradual transition from scotopic to photopic vision. Similarly, there is no fixed threshold for the saturation of cone cells that leads to loss of chromatic information at higher levels of illumination caused by color dilution. We, therefore, use suitable weights that vary smoothly with saturation and intensity to represent both color and Gray scale perception for each pixel.

3.6.12 NON INTERVAL QUANTIZATION

Due to the large range for each component by directly calculating the characteristics for the retrieval then the computation will be very difficult to ensure rapid retrieval. It is essential to quantify HSV space component to reduce computation and improve efficiency. At the same time, because the human eye to distinguish colors is limited, do not need to calculate all segments. Unequal interval quantization according the human color perception has been applied on H , S ,V components.

Based on the color model of substantial analysis, we divide color into eight parts. Saturation and intensity is divided into three parts separately in accordance with the human eyes to distinguish. In accordance with the different colors and subjective color perception quantification, quantified hue(H), saturation(S) and value(V)

In accordance with the quantization level above, the H, S, V three-dimensional feature vector for different values of with different weights to form one dimensional feature vector and is given by the following equation:

$$G = Qs*Qv*H+Qv*s+V$$

Where Qs is the quantized series of S and Qv is the quantized series of V. And now by setting Qs = Qv = 3, Then G = 9H+3S+V

$$H=\begin{cases}0 & if \quad h\in[316,20]\\1 & if \quad h\in[21,40]\\2 & if \quad h\in[41,75]\\3 & if \quad h\in[76,155]\\4 & if \quad h\in[156,190]\\5 & if \quad h\in[191,270]\\6 & if \quad h\in[271,295]\\7 & if \quad h\in[296,315]\end{cases} \qquad S=\begin{cases}0 & if \quad s\in[0,0.2)\\1 & if \quad s\in[0.2,0.7)\\2 & if \quad s\in[0.7,1)\end{cases} \qquad V=\begin{cases}0 & if \quad v\in[0,0.2)\\1 & if \quad v\in[0.2,0.7)\\2 & if \quad v\in[0.7,1)\end{cases}$$

In this way three component vector of the HSV from one dimensional vector, Which quantize the whole color space for the 73 kinds of the main colors. So we can handle 73 bins of one dimensional histogram. This qualification is effective in reducing the images by the effect of the light intensity, but also reducing the computational time and complexity.

3.6.13 IMAGE RETRIEVAL

Image retrieval is nothing but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords or the descriptions to the images so that the retrieval can be performed. Manual image annotation is time consuming, expensive and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc..,

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others. one can today generate volumes of images with content as diverse as family get-togethers and national park visits. Low-cost storage and easy Web hosting has fuelled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchable image data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real-world image search system designers.

As far as technological advances are concerned, growth in Content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real world implications, applications, and constraints of the technology. Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future.

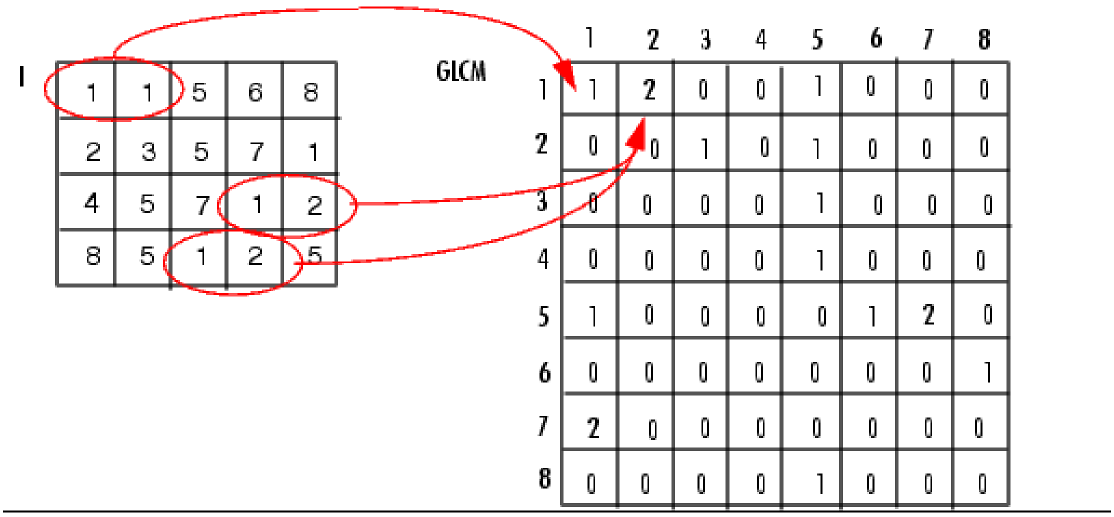
An image retrieval system designed to serve a personal collection should focus on features such as personalization, flexibility of browsing, and display methodology. For example, Google's Picasa system [Picasa 3004] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods. The notable few applications include an FPGA implementation of a color-histogram-based image retrieval system [Kotoulas and Andreadis 3003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 3003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 3003].

Discussion. Regardless of the nature of the collection, as the expected user-base grows, factors such as concurrent query support, efficient caching, and parallel and distributed processing of requests become critical. For future real-world image retrieval systems, both software and hardware approaches to address these issues are essential. More realistically, dedicated specialized servers, optimized memory and storage support, and highly parallelizable image search algorithms to exploit cluster computing powers are where the future of large-scale image search hardware support lies.

3.6.14 OVERVIEW OF TEXTURE

We all know about the term Texture but for defining it is a hard time. One can differentiate the two different Textures by recognizing the similarities and differences. Commonly there are three ways for the usage of the Textures:

Based on the Textures the images can be segmented To differentiate between already segmented regions or to classify them.We can reproduce Textures by producing the descriptions. The texture can be analysed in three different ways. They are Spectral, Structural and Statistical:



CHAPTER 4
DIGITAL IMAGE PROCESSING

4.1 DIGITAL IMAGE PROCESSING

4.1.1 BACKGROUND

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An important characteristic underlying the design of image processing systems is the significant level of testing & experimentation that normally is required before arriving at an acceptable solution. This characteristic implies that the ability to formulate approaches &quickly prototype candidate solutions generally plays a major role in reducing the cost & time required to arrive at a viable system implementation.

4.1.2 DIP

An image may be defined as a two-dimensional function f(x, y), where x & y are spatial coordinates, & the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital image by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensor, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis& computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting & somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low-level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low- level process is characterized by the fact that both its inputs & outputs are images.

Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images. Finally higher- level processing involves "Making sense" of an ensemble of recognized objects, as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

4.1.3 IMAGE

An image is represented as a two dimensional function f(x, y) where x and y are spatial co-ordinates and the amplitude of 'f' at any pair of coordinates (x, y) is called the intensity of the image at that point.

4.1.4 GRAYSCALE IMAGE

A grayscale image is a function I (xylem) of the two spatial coordinates of the image plane.

I(x, y) is the intensity of the image at the point (x, y) on the image plane.

I (xylem) takes non-negative values assume the image is bounded by a rectangle $[0, a] \times [0, b]$I: $[0, a] \times [0, b] \rightarrow [0, \text{info})$

4.1.5 COLOR IMAGE

It can be represented by three functions, R (xylem) for red, G (xylem) for green and B (xylem) for blue.

An image may be continuous with respect to the x and y coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate's values is called sampling. Digitizing the amplitude values is called quantization.

4.1.6 CO-ORDINATE CONVENTION

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image f(x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size M X N. The values of the coordinates (xylem) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates.

In many image processing books, the image origin is defined to be at (xylem)=(0,0).The next coordinate values along the first row of the image are (xylem)=(0,1).It is important to keep in mind that the notation (0,1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Following figure shows the coordinate convention. Note that x ranges from 0 to M-1 and y from 0 to N-1 in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (xylem) the toolbox uses the notation (race) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate topples, (alb), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentation refers to the coordinates. Less frequently the toolbox also employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refers to rows. This is the opposite of our use of variables x and y.

4.1.7 IMAGE AS MATRICES

The preceding discussion leads to the following representation for a digitized image function:

f (0,0) f(0,1) ……….. f(0,N-1)

f (1,0) f(1,1) ………… f(1,N-1)

. . .

f (xylem)=. . .

f (M-1,0) f(M-1,1) ………… f(M-1,N-1)

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements. A digital image can be represented naturally as a MATLAB matrix:

f (1,1) f(1,2) ……. f(1,N)

f (2,1) f(2,2) …….. f (2,N)

. . .

f = . . .

f (M,1) f(M,2) …….f(M,N)

Where f (1,1) = f(0,0) (note the use of a monoscope font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation f(p ,q) denotes the element located in row p and the column q. For example f(6,2) is the element in the sixth row and second column of the matrix f. Typically we use the letters M and N respectively to denote the number of rows and columns in a matrix. A 1xN matrix is called a row vector whereas an Mx1 matrix is called a column vector. A 1x1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using mono-scope characters. We use conventional Roman, italic notation such as f(x ,y), for mathematical expressions

4.1.8 READING IMAGES

Images are read into the MATLAB environment using function imread whose syntax is

Imread (‘filename’)

FORMAT NAME	DESCRIPTION	RECOGNIZED EXTENSION
TIFF	Tagged Image File Format	.tif
JPEG	Joint Photograph Experts Group	.jpg
GIF	Graphics Interchange Format	.gif
BMP	Window Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	.xwd

Table 4.1 Image Formats

Here filename is a spring containing the complete of the image file(including any applicable extension).For example the command line

>> f = imread ('8. jpg');

Reads the JPEG (above table) image chestxray into image array f. Note the use of single quotes (') to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output If a semicolon is not included. MATLAB displays the results of the operation(s) specified in that line. The prompt symbol (>>) designates the beginning of a command line, as it appears in the MATLAB command window**.**

4.1.9 DATA CLASSES

Although we work with integers coordinates the values of pixels themselves are not restricted to be integers in MATLAB. Table above list various data classes supported by MATLAB and IPT are representing pixels values. The first eight entries in the table are refers to as numeric data classes. The ninth entry is the char class and, as shown, the last entry is referred to as logical data class.

All numeric computations in MATLAB are done in double quantities, so this is also a frequent data class encounter in image processing applications. Class unit 8 also is encountered frequently, especially when reading data from storages devices, as 8 bit images are most common representations found in practice. These two data classes, classes logical, and, to a lesser degree, class unit 16 constitute the primary data classes on which we focus. Many ipt functions however support all the data classes listed in table. Data class double requires 8 bytes to represent a number uint8 and int 8 require one byte each, uint16 and int16 requires 2bytes and unit 42.

NAME	**DESCRIPTION**
Double	Double _ precision, floating_ point numbers the Approximate.
Uint8	unsigned 8_bit integers in the range [0,255](1byteper Element).
Uint16	unsigned 16_bit integers in the range [0, 65545] (2byte per element).
Uint 42	unsigned 42_bit integers in the range [0, 4294967295] (4 bytes per element).
Int8	signed 8_bit integers in the range [-128,127] (1 byte per element)
Int 16	signed 16_byte integers in the range [42768, 42767] (2 bytes per element).
Int 42	signed 42_byte integers in the range [-2147484648, 21474844647] (4 bytes per element).
Single	single _precision floating _point numbers with values
Char	characters (2 bytes per elements).
Logical	values are 0 to 1 (1byte per element).

Table 4.2 Description Of Datatypes

Int 42 and single required 4 bytes each. The char data class holds characters in Unicode representation. A character string is merely a 1*n array of characters logical array contains only the values 0 to 1,with each element being stored in memory using function logical or by using relational operators.

4.1.10 IMAGE TYPES

The toolbox supports four types of images:

- Intensity images
- Binary images
- Indexed images
- RGB images

Most monochrome image processing operations are carried out using binary or intensity images, so our initial focus is on these two image types. Indexed and RGB color images.

4.1.10.1 INTENSITY IMAGES

An intensity image is a data matrix whose values have been scaled to represent intentions. When the elements of an intensity image are of class unit8, or class unit 16, they have integer values in the range [0,255] and [0, 65545], respectively. If the image is of class double, the values are floating point numbers. Values of scaled, double intensity images are in the range [0, 1] by convention.

4.1.10.2 BINARY IMAGES

Binary images have a very specific meaning in MATLAB.A binary image is a logical array 0s and1s.Thus, an array of 0s and 1s whose values are of data class, say unit8, is not considered as a binary image in MATLAB .A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, we create an array B using the statement.

B=logical (A)

If A contains elements other than 0s and 1s.Use of the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.

Using relational and logical operators also creates logical arrays.

To test if an array is logical we use the I logical function: islogical(c).

If c is a logical array, this function returns a 1.Otherwise returns a 0. Logical array can be converted to numeric arrays using the data class conversion functions.

4.1.10.3 INDEXED IMAGES

An indexed image has two components:

- A data matrix integer, x
- A color map matrix, map

Matrix map is an m*4 arrays of class double containing floating point values in the range [0, 1].The length m of the map are equal to the number of colors it defines. Each row of map specifies the red, green and blue components of a single color. An indexed images uses “direct mapping” of pixel intensity values color map values. The color of each pixel is determined by using the corresponding value the integer matrix x as a pointer in to map. If x is of class double ,then all of its components with values less than or equal to 1 point to the first row in map, all components with value 2 point to the second row and so on. If x is of class units or unit 16, then all components value 0 point to the first row in map, all components with value 1 point to the second and so on.

4.1.10.4 RGB IMAGE

An RGB color image is an M*N*4 array of color pixels where each color pixel is triplet corresponding to the red, green and blue components of an RGB image, at a specific spatial location. An RGB image may be viewed as “stack” of three gray scale images that when fed in to the red, green and blue inputs of a color monitor.

Produce a color image on the screen. Convention the three images forming an RGB color image are referred to as the red, green and blue components images.

The data class of the components images determines their range of values. If an RGB image is of class double the range of values is [0, 1].

Similarly the range of values is [0,255] or [0, 65545].For RGB images of class units or unit 16 respectively. The number of bits use to represents the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8bit image, the corresponding RGB image is said to be 24 bits deep.

Generally, the number of bits in all component images is the same. In this case the number of possible color in an RGB image is (2^b) ^4, where b is a number of bits in each component image. For the 8bit case the number is 16,777,216 colors.

CHAPTER 5

SOFTWARE INTRODUCTION

5.1 INTRODUCTION TO MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

5.2 THE MATLAB SYSTEM

The MATLAB system consists of five main parts

5.2.1 DEVELOPMENT ENVIRONMENT

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

5.2.2 THE MATLAB MATHEMATICAL FUNCTION LIBRARY

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

5.2.3 THE MATLAB LANGUAGE

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both “programming in the small” to rapidly create quick and dirty throw-away programs, and “programming in the large” to create large and complex application programs.

5.2.4 GRAPHICS

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully

customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

5.2.5 THE MATLAB APPLICATION PROGRAM INTERFACE (API)

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

5.3 GRAPHICAL USER INTERFACE (GUI)

MATLAB’s Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function’s GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.
- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type guide filename where filename is the name of an existing FIG-file on the current path. If filename is omitted, GUIDE opens a new (i.e., blank) window.

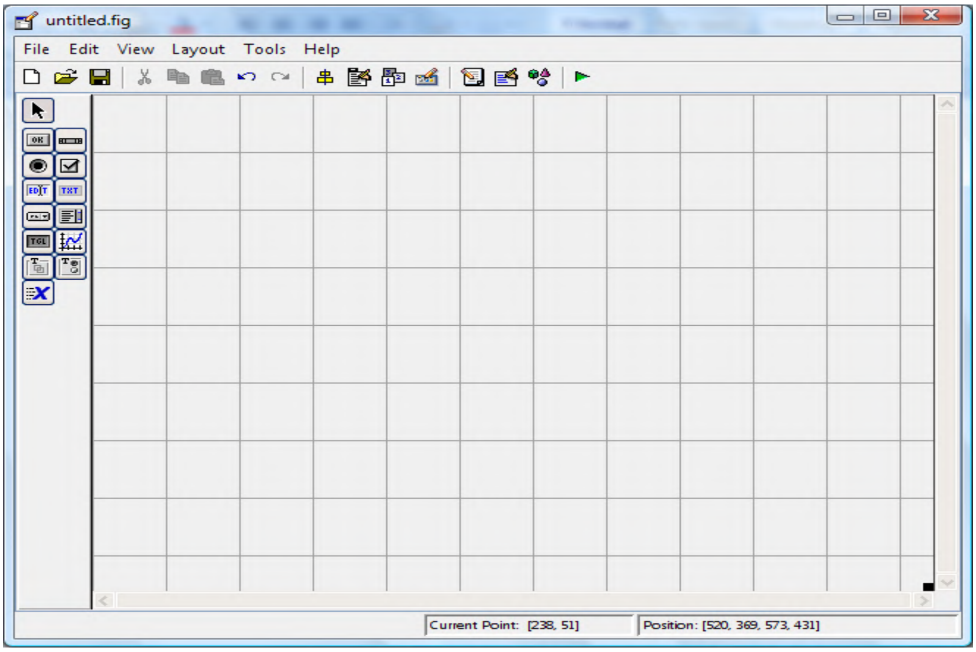


Fig 5.1 Graphical User Interface(GUI) Display

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

5.4 GETTING STARTED

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

- **Introduction** describes the components of the MATLAB system.
- **Development Environment** – introduces the MATLAB development environment, including information about tools and the MATLAB desktop.
- **Manipulating Matrices** – introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.
- **Graphics** – introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.
- **Programming with MATLAB** – describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

5.5 DEVELOPMENT ENVIRONMENT

5.5.1 INTRODUCTION

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

5.5.2 STARTING AND QUITTING MATLAB

5.5.2.1 STARTING MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop. On a UNIX platform, to start MATLAB, type MATLAB at the operating system prompt. After starting MATLAB, the MATLAB desktop opens – see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

5.5.2.2 QUITTING MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

5.5.3 MATLAB DESKTOP

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB. The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

5.5.4 DESKTOP TOOLS

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

5.5.4.1 COMMAND WINDOW

Use the Command Window to enter variables and run functions and M-files.

5.5.4.2 COMMAND HISTORY

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

5.5.4.3 RUNNING EXTERNAL PROGRAMS

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

5.5.4.4 LAUNCH PAD

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

5.5.4.5 HELP BROWSER

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type help browser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

5.5.4.6 HELP NAVIGATOR

Use to Help Navigator to find information. It includes:

- **Product filter** - Set the filter to show documentation only for the products you specify.
- **Contents tab** - View the titles and tables of contents of documentation for your products.

- **Index tab** - Find specific index entries (selected keywords) in the MathWorks documentation for your products.
- **Search tab** - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.
- **Favorites tab** - View a list of documents you previously designated as favorites.

5.5.4.7 DISPLAY PANE

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

- **Browse to other pages** – Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.
- **Bookmark pages** – Click the Add to Favorites button in the toolbar.
- **Print pages** – Click the print button in the toolbar.
- **Find a term in the page** – Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

5.5.4.8 CURRENT DIRECTORY BROWSER

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

5.5.4.9 SEARCH PATH

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

5.5.4.10 WORKSPACE BROWSER

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

5.5.4.11 ARRAY EDITOR

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

5.5.4.12 EDITOR/DEBUGGER

Use the Editor/Debugger to create and debug M-files, which are programs you write to run MATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

5.6 MANIPULATING MATRICES

5.6.1 ENTERING MATRICES

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Dürer's matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 5 15 15 1]
```

MATLAB displays the matrix you just entered.

```
A =

16    3    2   13

5   10   11    8

9    6    7   12

5   15   15    1
```

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

5.6.2 EXPRESSIONS

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

5.6.2.1 VARIABLES

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

5.6.2.2 NUMBERS

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23

1i   -3.15159j   3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

5.6.2.3 OPERATORS

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

Table 5.1 Operators

5.6.2.4 FUNCTIONS

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.15159265...
I	Imaginary unit, √-1
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

Table 5.2 Functions

5.7 GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

5.7.1 CREATING GUIs WITH GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

5.7.2 GUI DEVELOPMENT ENVIRONMENT

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

5.7.3 THE IMPLEMENTATION OF A GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

- **A FIG-file** - contains a complete description of the GUI figure and all of its children (uicontrols and axes), as well as the values of all object properties.
- **An M-file** - contains the functions that launch and control the GUI and the callbacks, which are defined as subfunctions. This M-file is referred to as the application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

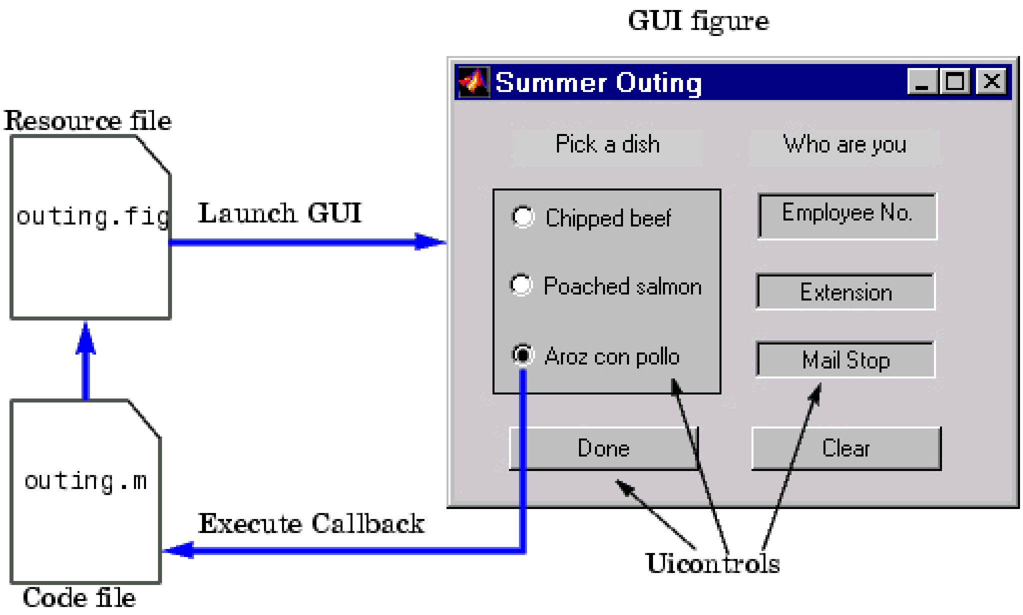


Fig 5.2 Graphical User Blocks

5.7.4 FEATURES OF THE GUIDE-GENERATED APPLICATION M-FILE

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

5.7.5 BEGINNING THE IMPLEMENTATION PROCESS

To begin implementing your GUI, proceed to the following sections:

- **Getting Started with GUIDE -** the basics of using GUIDE.
- **Selecting GUIDE Application Options** - set both FIG-file and M-file options.
- **Using the Layout Editor** - begin laying out the GUI.
- **Understanding the Application M-File** - discussion of programming techniques used in the application M-file.
- **Application Examples** - a collection of examples that illustrate techniques which are useful for implementing GUIs.

5.7.6 COMMAND-LINE ACCESSIBILITY

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

5.7.7 USER INTERFACE CONTROL

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons

- Frames
- Radio Buttons
- List boxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

5.7.7.1 PUSH BUTTONS

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

5.7.7.2 PROPERTIES TO SET

- **String** - set this property to the character string you want displayed on the push button.
- **Tag** - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

5.7.7.3 PROGRAMMING THE CALLBACK

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

5.7.7.4 TOGGLE BUTTONS

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

5.7.7.5 PROGRAMMING THE CALLBACK

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

5.7.7.6 FROM THE GUIDE APPLICATION M-FILE

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)

button_state = get(h,'Value');

if button_state == get(h,'Max')

% toggle button is pressed

elseif button_state == get(h,'Min')

% toggle button is not pressed

End
```

5.7.7.7 ADDING AN IMAGE TO A PUSH BUTTON OR TOGGLE BUTTON

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);

a(:,:,2) = rand(16,128);

a(:,:,3) = rand(16,128);

set(h,'CData',a)
```

5.7.7.8 RADIO BUTTOMS

RADIO buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

5.7.7.9 IMPLEMENTING MUTUALLY EXCLUSIVE BEHAVIOR

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)

set(off,'Value',0)
```

5.7.7.10 OBTAINING THE RADIO BUTTON HANDLES

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)

off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton5];
```

mutual_exclude(off)

% Continue with callback...

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

5.7.7.11 CHECKBOXES

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)

if (get(h,'Value') == get(h,'Max'))

% then checkbox is checked-take appropriate action

Else

% checkbox is not checked-take appropriate action

End
```

5.7.7.12 EDIT TEXT

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)

user_string = get(h,'string');

% proceed with callback...
```

5.7.7.13 OBTAINING NUMERIC DATA FROM AN EDIT TEST COMPONENT

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)

user_entry = str2double(get(h,'string'));

if isnan(user_entry)

errordlg('You must enter a numeric value','Bad Input','modal')

end

% proceed with callback...
```

5.7.7.14 TRIGGERING CALLBACK EXECUTION

On UNIX systems, clicking on the menu bar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menu bar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

5.7.7.15 STATIC TEXT

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

5.7.7.16 FRAMES

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

5.7.7.17 PLACING COMPONENTS ON TOP OF FRAMES

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

5.7.7.18 LIST BOXES

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The List box Top property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. List box Top is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Non-integer values are fixed to the next lowest integer.

5.7.7.19 SINGLE OR MULTIPLE SELECTION

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

5.7.7.20 SELECTION TYPE

List boxes differentiate between single and double clicks on an item and set the figure Selection Type property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

5.7.7.21 TRIGGERING CALLBACK EXECUTION

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure Selection Type property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

5.7.7.22 POPUP MENUS

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

5.7.7.23 PROGRAMMING THE POPUP MENU

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

switch val

case 1

% The user selected the first item

case 2

% The user selected the second item

% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

string_list = get(h,'String');

selected_string = string_list{val}; % convert from cell array to string

% etc.
```

5.7.7.24 ENABLING OR DISABLING CONTROLS

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

5.7.7.25 AXES

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

5.7.7.26 AXES CALLBACKS

Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes ButtonDownFcn property to define the callback.

5.7.7.27 PLOTTING TO AXES IN GUIS

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

5.7.7.28 GUIS WITH MULTIPLE AXES

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

axes(handles.axes1)

makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axes. See GUI with Multiple Axes for and example that uses two axes.

5.7.7.29 FIGURE

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control.

CHAPTER 6

PROPOSED SYSTEM

6.1 BLOCK DIAGRAM

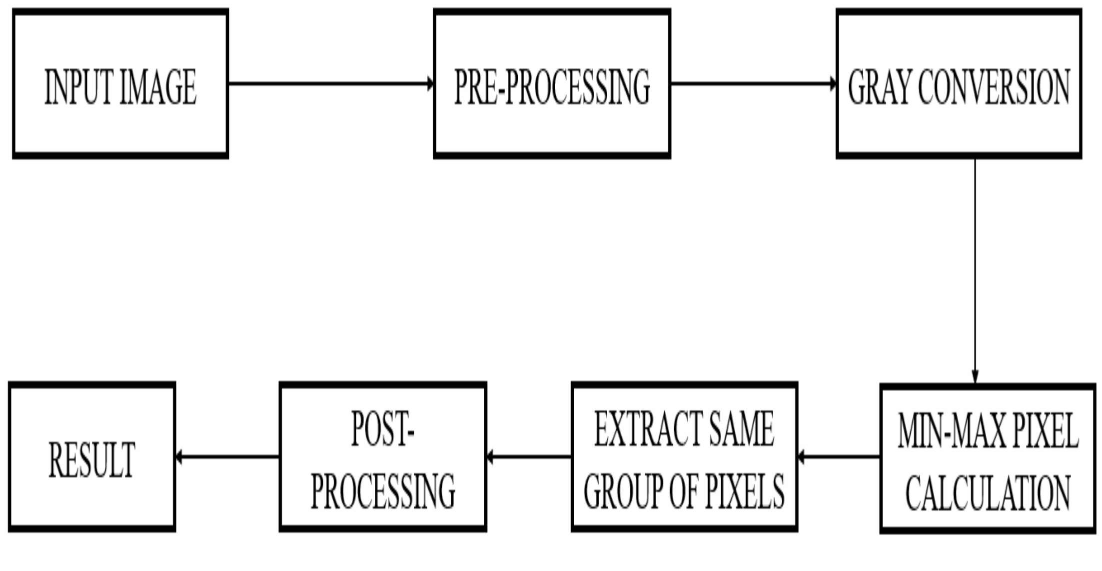


Fig 6.1 Block Diagram

6.1.1 INPUT IMAGE

An MRI Image is given as an input which is taken from BRAMSIT Database.

6.1.2 PRE-PROCESSING

In this stage, the size of the image is adjusted accordingly and the noise will be eliminated by using filters if present.

6.1.3 GRAY CONVERSION

The given MRI Image is then converted into grayscale image as K-means algorithm supports Grayscale images only.

6.1.4 MIN-MAX PIXEL CALCULATION

In order to achieve Image Enhancement Min-Max pixels are calculated for the input image.

6.1.5 EXTRACT SAME GROUP OF PIXELS

The Identical pixels are made into groups and extraction is done.

6.1.6 POST-PROCESSING

In post processing stage, the changes will be made in size of the image, and size will be same as input image.

6.1.7 RESULT

The Tumor part is detected and displayed separately.

6.2 FLOW CHART

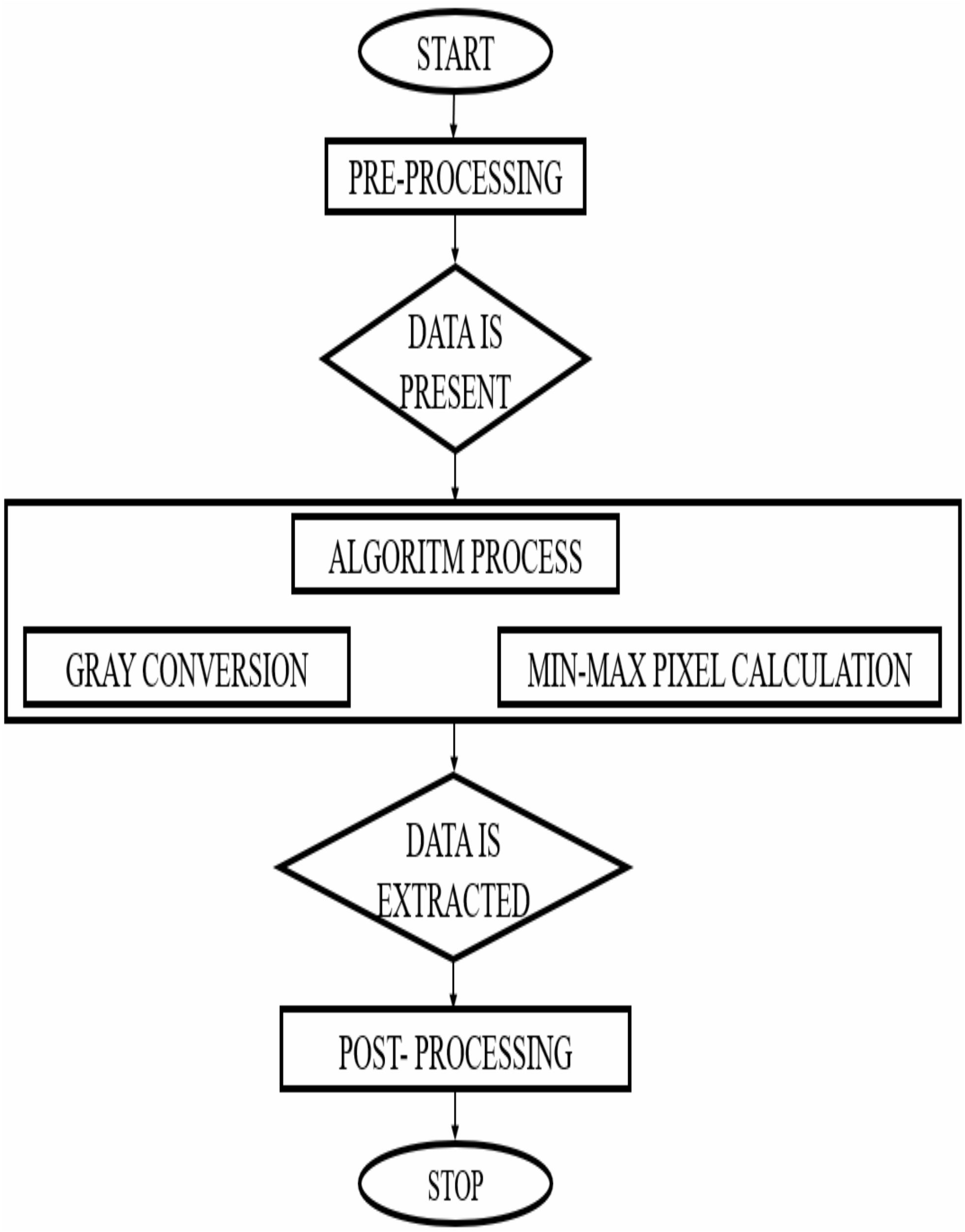


Fig 6.2 Flow Chart

At initial stage, an input MRI Image is given. Later in the pre-processing stage the size of the image is changed accordingly and the noise is eliminated by using filters if present. If the data is present then it will be further processed or the process stops. In the algorithm process, the input image is converted into grayscale image as K-means algorithm only supports grayscale images. Further, the min-max pixels are calculated for the image enhancement. Later these pixels are extracted. In the post-processing, the size of the image which is enhanced, is again converted into the size of original image and divided into clusters. With the help of these clusters, the tumor part is detected and is shown as output.

CHAPTER 7

METHODOLOGY

7.1 METHODOLOGY

In order to detect tumors and validate the results, a few databases available publicly and help the research community to come up with better algorithms. Among those BRATS dataset, Fig share and Kaggle plays a major role with huge image dataset. Fig share[5] consisting of 3064 T1-weighted contrast-enhanced images with three kinds of brain tumor such as normal, benign and malignant. The images can be down loaded along with their ground truth images in BLACK &WHITE (BW) format. The dataset consists of tumor mask, tumor border, and original image in .MAT file. The space occupied by the database is more. Some of the sample dataset images are shown in Figure 7.1.

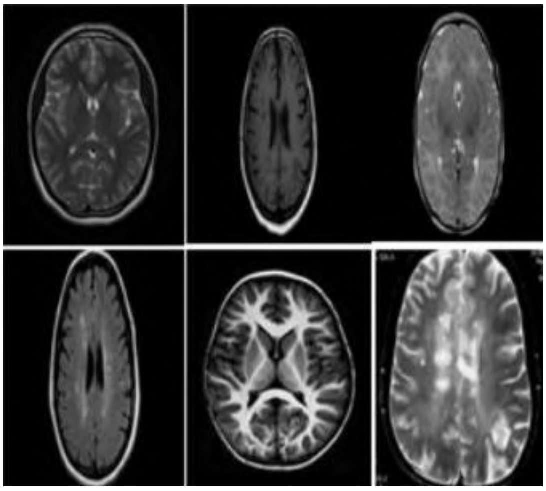


Fig 7.1 Sample Figures Hare Dataset Images Along With Its Ground Truth Images

Brats dataset[6] is a dataset used for many researchers for their work. The numbers of images along with their ground truth are . The images consist of normal benign and malignant images. Lot of versions exist in Brats dataset. All the proposed algorithms for tumor detection in various survey papers are normally validated in BRATS dataset only. Figure 7.2 shows the Kaggle dataset [7]involves the images for brain tumor detection.

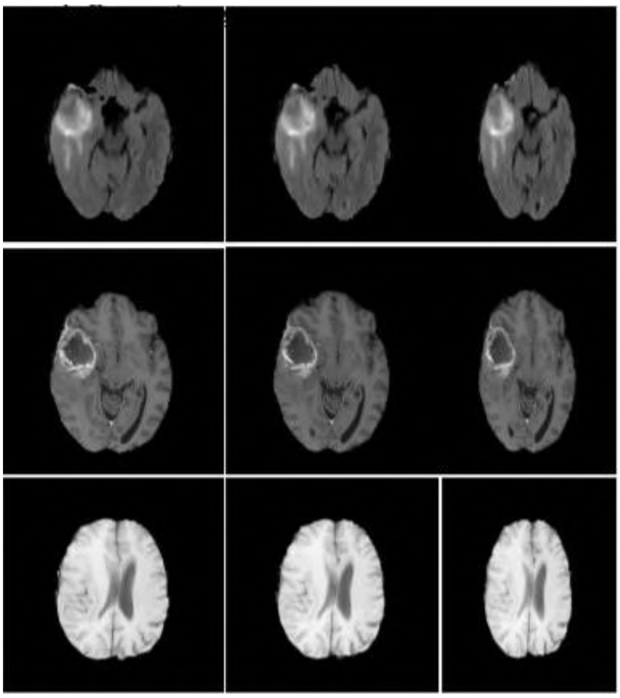


Fig 7.2. Sample Brats Dataset Images And Its Ground Truth Images

It is developed by Navoneel Chakrabarty in the year 2019. These datasets involves original images only. Few images are available for analysis. samples are An example of images from Kaggle dataset are shown in Figure 7.3.

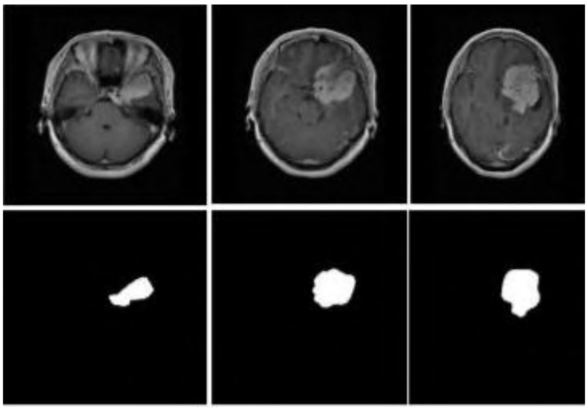


Fig 7.3 Sample Kaggle Dataset Images[7]

In [1], the authors discussed about the method to produce local volumes of the tumor databases. The authors analyzed numerical and quality parameter of the images using changes in morphological variations. In [2], the authors discussed a new method with a large number of CNNs with the information in overlapping regions. The authors used the OASIS dataset for validation of the results. V.P.Gladis Pushpa Rathi et al[3] proposed a tumor classification method based on feature extraction, features selection, PCA analysis and LDA analysis. The classification accuracy for the proposed method is 98.87%. The authors Amit Kumar Rohit et al.,[4], discussed a single query system based on Content Based Image Retrieval System(CBIR).The proposed system involves preprocessing, feature extraction and detection of tumors. The accuracy of the proposed method is 98.33%. Based on above said survey, many of the works still didn't follow a simple user friendly and accessible database. In this work, we proposed to create a database of MRI scan images, which we named as 'BRAMSIT Database' for the benefit of the biomedical engineering research community. An important risk in the research and development of this community is the lack of medical explanation and the plane for the tumor identification. Although some relevant papers have discussed localization issues, they do not provide such databases friendly to researchers.

Thus motivated by these factors, our main contributions in this work are:

• Create a new BRAMSIT database which involves 319 MRI scan images along with its ground truth images.

• Present the annotation of all the 319 subject's biological data along with their axial position.

CHAPTER 8

BRAMSIT DATABASE

The BRAMSIT database is deliberated through subsequent stages:

- Structure Details along with their axial position
- Labelling the Images
- Manual Annotation

8.1 STRUCTURE DETAILS

This dataset involves of 319 MRI scan images collected from various subjects. BRAMSIT consist of 319 images of different subjects. Each subjects includes reference number, age and their axial. The samples of BRAMSIT normal scan images of some subjects are shown in Figure 4. Similarly, the samples MRI abnormal and ground truth images of 5 subjects are shown in Figure 5

The salient features of the database are:

• Creation of database along with its clinic statistics for better use of images for research.

• Abnormal Images with the Ground truth for easy classification. The sample images along with the ground truth images are used for better segmentation and classification. All the ground truth images are developed from the experts opinion.

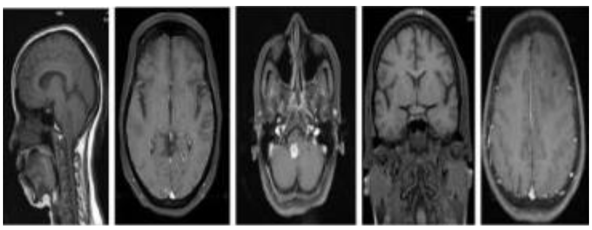


Fig 8.1 Sample Of MRI Normal Scan Images

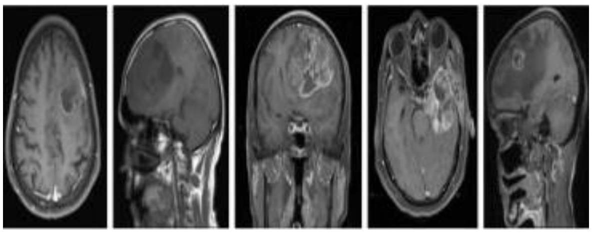


Fig 8.2 Sample Of MRI Abnormal Scan Images

8.2 LABELLING THE IMAGES

In the proposed database all the images are labeled perfectly for the purpose of understanding and interpreting. From the label itself one can easily identify the Subject ID, age and axial position. For example for a image with any abnormality suspect, the images are taken in three planes such as Sagittal ,Transaxial, and coronal View or Transaxial, Coronal and Sagittal View or Coronal ,Sagittal View and Transaxial view. Based on that, some abnormal suspected images are three in number based on their plane. Some normal images are single image with single view. Considering these factors, if Sub1 has three view images then it is labelled as Sub_1_STC, Sub_1_TCS, Sub_1_CST. This indicates subject1's STC means Sagittal, Transaxial, and coronal View, Sub_1_TCS means subject1's Transaxial, Coronal and Sagittal View and Sub_1_CST means subject1's Coronal ,Sagittal View and Transaxial view. In normal cases, the images are labelled as Sub_2_N_TCS, Sub_2_N_STC and Sub_2_N_CST view. TABLE 1. DETAILS OF THE SUBJECT IN THE DATABASE.

8.3 MANUAL ANNOTATION

BRAMSIT provides a detailed analysis of each and every MRI scan image. Each MRI scan images for the following attributes we manually annotated.

- Unique ID of the subject- Based on the axial Position
- Age
- Axial position –Transaxial, Sagittal and Coronal
- Gender- Male or Female

From the database, the table 1 shows the details of the images. The manual annotation is represented below: The age of the subject, gender, subject Id and the axial position is annotated. A sample subject shown in the annotation box below. Sample

MRI images and the ground truth images are shown in Figure.8. Ref.No Age Gender Axial Sub_1_TCS 40 Male Trans axial Sub_2_STC 51 Male Sagittal Sub_2_CTS 51 Male Coronal Sub_3_TCS 55 Female Trans axial Sub_4_STC 58 Female Sagittal Authorized licensed use limited to: Cornell University Library.

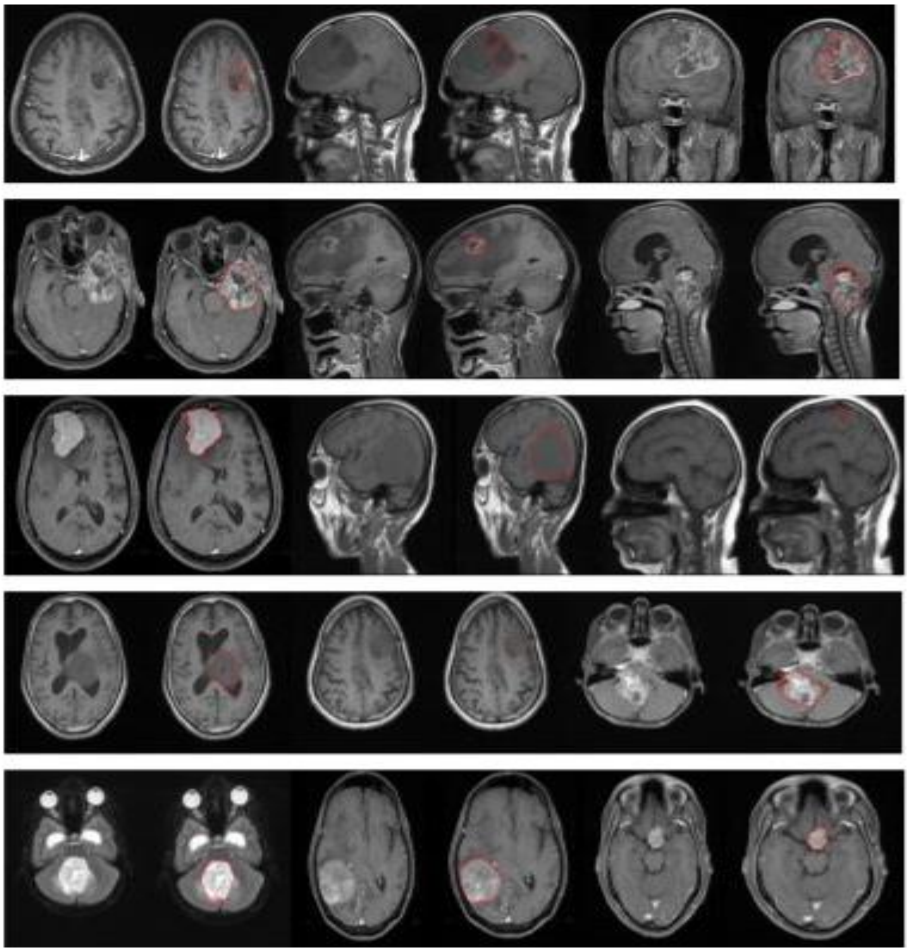


Fig 8.3 Sample Of MRI Abnormal Scan Images And Their Ground Truth Images

CHAPTER 9

COMPLEXITY OF THE DATABASE

9.1 COMPLEXITY OF THE DATABASE

As said earlier, many databases are there for analysis of brain tumor in MRI imaging modality. As this dataset is also created for the same research outcome when compared with the other databases, one main feature of the proposed dataset is the ease of access and the speed in processing or analysis of the images. When compared with the BRATS dataset, Fig share and Kaggle, the proposed BRAMSIT processing time and accessing time is very less during some bench mark image processing algorithms.

S.NO	DATASET	ACCESS TIME	PROCESSING TIME	EFFICIENCY IN(%)
1	BRATS	2.35ms	4.2ms	65
2	Fig share	2.01ms	3.52ms	78
3	Kaggle	2.45ms	3.87ns	84
4	BRAMSIT	0.2ns	1.5ns	95

Table 9.1 Performance Evaluation Of Various Databases

As shown in table 9.1, the access time and processing time for the proposed dataset is very less when compared with other datasets.

CHAPTER 10

RESULTS

10.1 INPUT-1

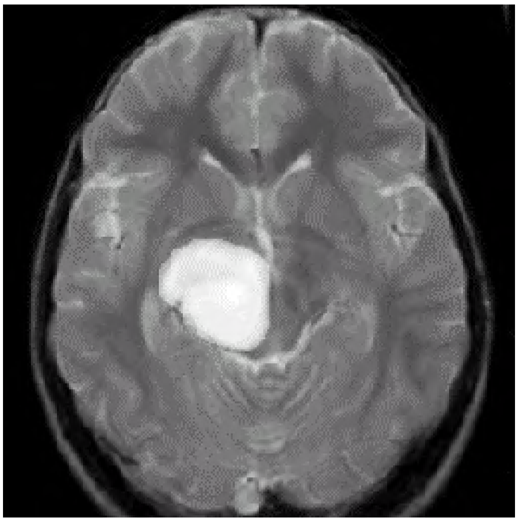


Fig 10.1 Scanned MRI Image Of Brain(1)

10.2 OUTPUT-1

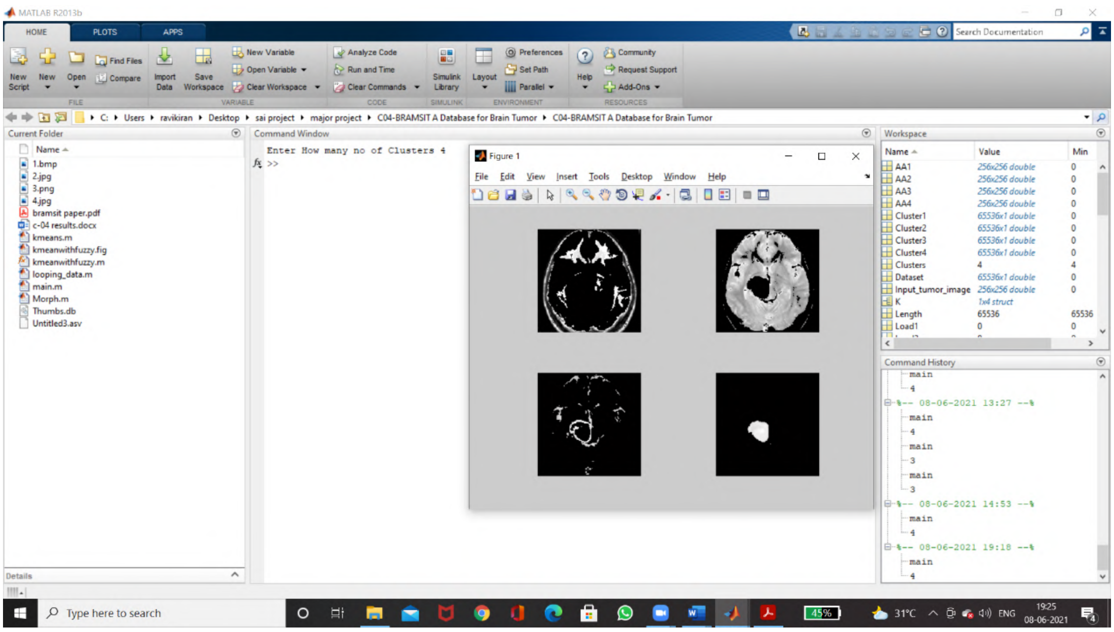


Fig 10.2 Output In 4 Clusters

10.3 INPUT-2

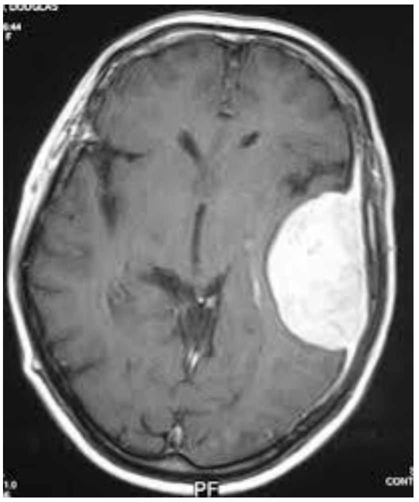


Fig 10.3 Scanned MRI Image Of Brain (2)

10.4 OUTPUT-2

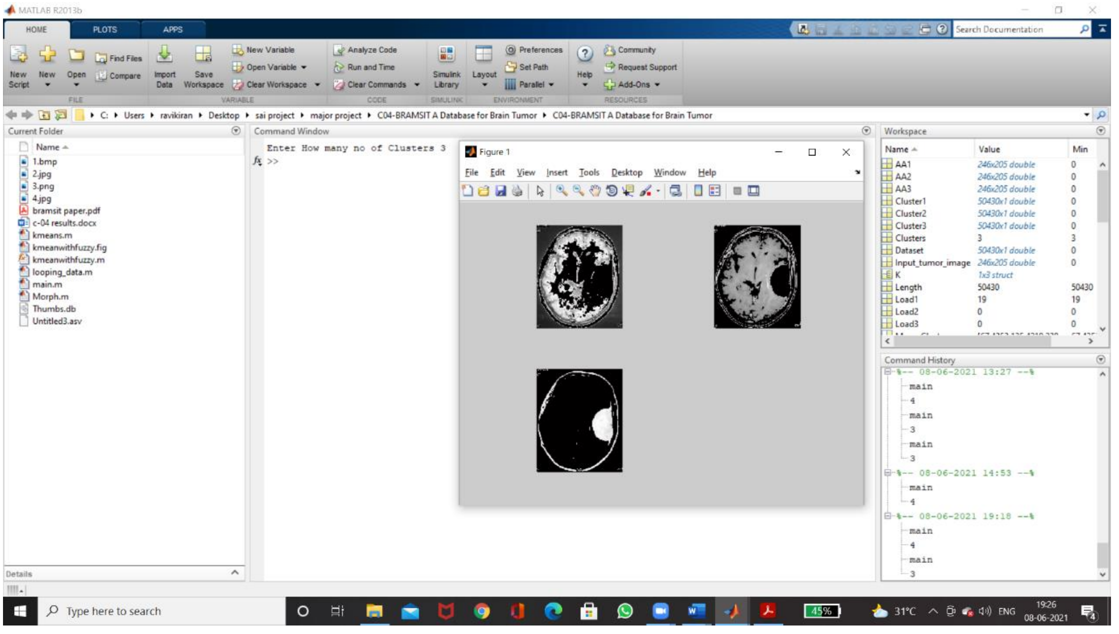


Fig 10.4 Output In 3 Clusters

CHAPTER 11

CONCLUSION

11.1 CONCLUSION

The project describes Screening Brain Tumor for the most recent database. Investigating brain tumor image analysis by using it as a resource. The "abnormal" and "ground truth" images. The main characteristics of this BRAMSIT database are

a) 319 MRI images.

b) Marking the entire subjects biological data.

The biomedical research community provides the brain tumor of many uncertain research problems.

11.2 FUTURE SCOPE

- The detection of brain tumor can be more easier and effective when coloured MRI images are used.
- Different algorithms may come into existence by which the process of detecting and diagnosing the brain tumor becomes more faster.
- Different databases may come into existence by which the processing time and accessing time becomes very less during some bench mark image processing algorithms.

REFERENCES

[1] Alexandre Guimondy, Gérard Subsol, Jean-Philippe Thirion, “Automatic MRI Database Exploration and Applications”, International Journal on Pattern Recognition and Artificial Intelligence, Volume 11, number 8, December1997.

[2] Pierrick Coupé, Boris Mansencal, Michaël Clément, Rémi Giraud, Baudouin Denis de Senneville, Vinh-Thong Ta, Vincent Lepetit and José V. Manjon, “AssemblyNet: A Novel Deep Decision-Making Process for Whole Brain MRI Segmentation”, arXiv:1906.01862.

[3] V.P.Gladis Pushpa Rathi and Dr.S.Palani, “Brain Tumor MRI Image Classification With Feature Selection and Extraction Using Linear Discriminant Analysis”, International Journal of Information Sciences and Techniques (IJIST) Vol.2, No.4, July 2012.

[4] Amit Kumar Rohit, N. G. Chitaliya “Novel Approach for Content based MRI Brain Image Retrieval”, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-4, No. 3 July 2014.

[5] Pareek, Meenakshi & Jha, CK & Mukherjee, Saurabh & Joshi, Chandani, “Brain Tumor Detection Using Wathershed Segmentation Techniques and Area Calculation”, International Journal of Informatics and Communication Technology (IJ-ICT).Vol.7, No.71,pp71-76,2018.

[6] Shaoguo Cui,Lei Mao,Jingfeng Jiang,Chang Liu,1and Shuyu Xiong “Automatic Semantic Segmentation of Brain Gliomas from MRIImages Using a Deep Cascaded Neural Network”, Journal of Healthcare Engineering,2018.

[7] Chakrabarty, Navoneel, “Brain MRI Images for BrainTumorDetection”, https://www.kaggle.com/navoneel/brainmri-images-for-brain-tumor-detection, 2019. Authorized licensed use limited to: Cornell Un

A Major Project report on

HISTOGRAM EQUALIZATION-BASED TECHNIQUES FOR CONTRAST ENHANCEMENT OF MRI BRAIN GLIOMA TUMOR IMAGES: COMPARATIVE STUDY

Submitted By

Ms. K. Kalpana (17K81A04F3)

Ms. L. Harshitha (17K81A04F4)

Ms. D. Suchithra (18K85A0430)

In partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

ST.MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

MAY 2021

BONAFIDE CERTIFICATE

Certified that this project reported title " Histogram equalization - based techniques for contrast enhancement of mri brain glioma tumour images : comparative study", is a bonafide work of1)Ms.K.Kalpana(17K81A04F3) 2)Ms.L.Harshitha(17K81A04F4) 3) Ms.D.Suchithra(18K85A0430) who carried out the work under my supervision, for the partial fulfilment of the requirements for the award of the degree of *Bachelor of Technology* in *Electronics And Communication Engineering.* Certified further that to the best of my knowledge and belief, the work reported here in does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion.

Place: Secunderabad

Date:

Dr.B.Hari Krishna

B.Tech., M.Tech, Ph.D.,

Professor & HOD

Department of Electronics And Communication Engineering

St. Martin's Engineering College.

COUNTER SIGNED

HEAD

Department of Electronics and Communication Engineering

St. Martin's Engineering College

LASYA INFOTECH

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **D.SUCHITHRA** WITH ROLL NO.**18K85A0430, K.KALPANA** WITH ROLL NO.**17K81A04F3**, **L.HARSHITHA** WITH ROLL NO.**17K81A04F4,** OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**HISTOGRAM EQUALIZATION BASED TECHNIQUES FOR CONTRAST ENHANCEMENT OF MRI BRAIN GLIOMA TUMOR IMAGES COMPARATIVE STUDY**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We declare that this project report titled HISTOGRAM EQUALIZATION-BASED TECHNIQUES FOR CONTRAST ENHANCEMENT OF MRI BRAIN GLIOMA TUMOUR IMAGES: COMPARATIVE STUDY submitted in partial fulfillment of the degree of B. Tech in Electronics and Communication Engineering record of original work carried out by us under the guidance and supervision of Mr. P. Joel Josephson and has not formed the name basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

1) Ms. K.Kalpana	2) Ms. L.Harshitha	3) Ms. D.Suchithra
(17K81A04F3)	(17K81A04F4)	(18K85A0435)

Place:

Date:

ACKNOWLEDGEMENTS

While bringing out this project report to its final form, we came across a number of people whose contributions in various ways helped us in field of research and they deserve special thanks. It is a pleasure to convey our gratitude to all of them.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to Dr.B.Hari Krishna ,**Head &Professor**, the Dept. of Electronics And Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him.

We would like to express our gratitude to **our dept .project coordinator** Dr.A.Anand for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We are highly grateful and obliged to **our college level Project Coordinators Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santosh Kumar** , **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

We dedicate this thesis to almighty

1. Ms.K.Kalpana(17K81A04F3)
2. Ms.L.Harshitha(17K81A04F4)
3. Ms.D.Suchithra(18K85A0430)

ABSTRACT

- In Magnetic Resonance Imaging (MRI), the poor images quality, particularly the artifacts inherent to this type of images as well as the low contrast between tissues and interindividual variability, could make difficult the image analysis and affect the accuracy of clinical diagnosis.
- Therefore, the needs for image enhancement techniques arise to improve the relevant image contents through reducing the noise while preserving the actual details features. Various MRI images denoising techniques have been proposed in literature where each technique has its advantages and limitations.
- Among them, the Histogram modifications-based approaches arise as the most employed, by many researchers, for MRI contrast enhancement.
- This project presents a comparative study of the most histogram-based techniques, mainly AHE, CLAHE, BPDHE and AIR-AHE techniques, dealing with denoising and contrast enhancement MRI images.
- Experimental study, using real-world databases, is performed based on evaluation of quality measurement metrics: absolute mean brightness error (AMBE), peak signal to noise ratio (PSNR) and Entropy.

TABLE OF CONTENTS

Chapter 1 INTRODUCTION TO DIGITAL IMAGE PROCESSING

Chapter 2 LITERATURE SURVEY

Chapter-1

INTRODUCTION TO DIGITAL IMAGE PROCESSING

1.1 IMAGE:

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person. two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surface.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphic technology, or developed by a combination of methods, especially in a pseudo-photograph.

Fig 1.1 General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color

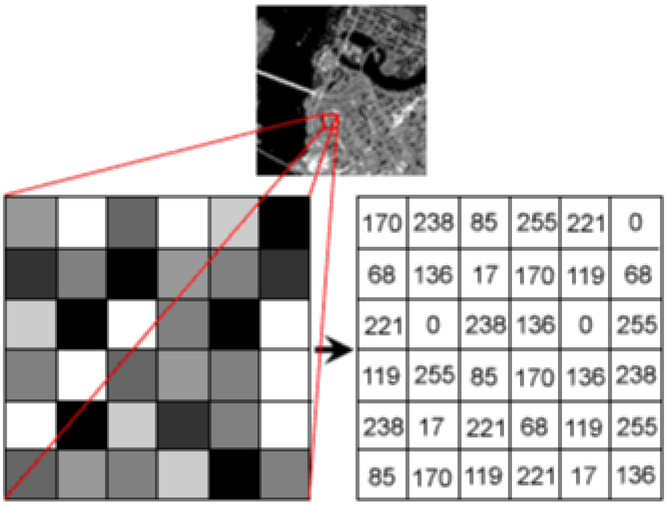


Fig 1.2 Image pixel

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

Fig1.3 Transparency Image

1.2 IMAGE FILE SIZES:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High-resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

1.3 IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

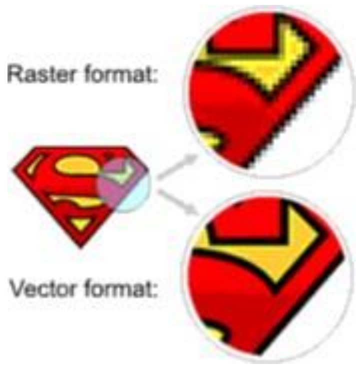


Fig1.4 Resolution image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

1.4 RASTER FORMATS:

These formats store images as bitmaps (also known as pixmaps)

- **JPEG/JFIF:**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

- **EXIF:**

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata is recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, the name of the camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed

- **TIFF:**

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

- **GIF:**

GIF (Graphics Interchange Format) is limited to an 8-bit palette or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

- **PNG:**

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

- **BMP:**

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

1.5 VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM:**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

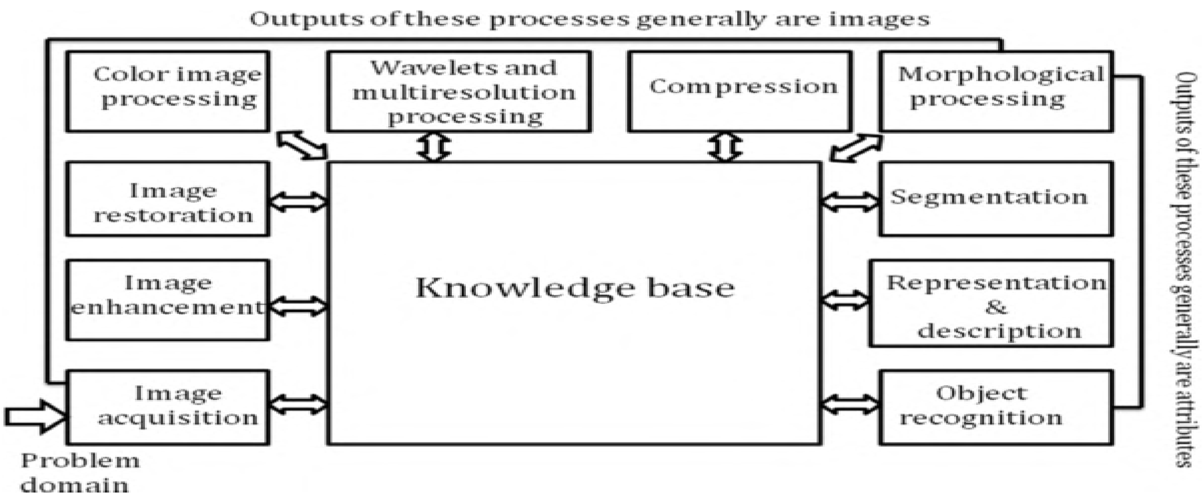
- **SVG:**

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all-purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

1.6 IMAGE PROCESSING:

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields suffers from myths, miss-connect ions, misunderstandings and misinformation. It is the vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor plowed with imprecise jargon.

Several factors combines to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low-cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.



1.6.1 Image Acquisition:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Fig 1.6 Digital Camera Image

The scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Fig 1.7 Digital Camera Cell

1.6.2 Image Enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because "it looks better." It is important to keep in mind that enhancement is a very subjective area of image processing.

Fig 1.8 Image enhancement

1.6.3 Image restoration:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Fig 1.9 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, whereas removal of image blur by applying a deblurring function is considered a restoration technique.

1.6.4 Color image processing:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Fig 1.10 Color & Gray scale image.

1.6.5 Wavelets and multiresolution processing:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform based image processing since the late1950's, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

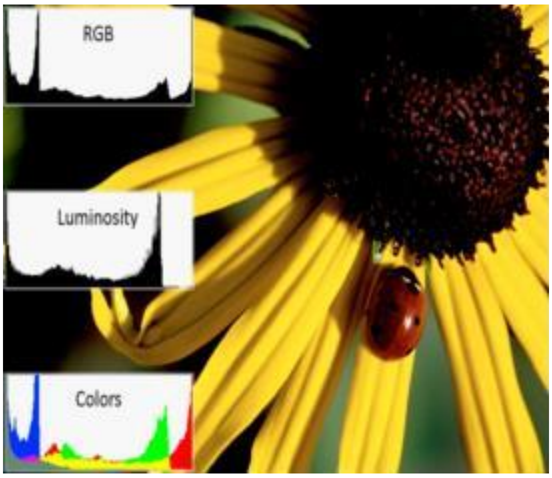


Fig 1.11 RGB Histogram Image.

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called **Multiresolution** theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub-band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

1.6.6 Compression:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which is characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts image compression standard.

1.6.7 Morphological processing:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

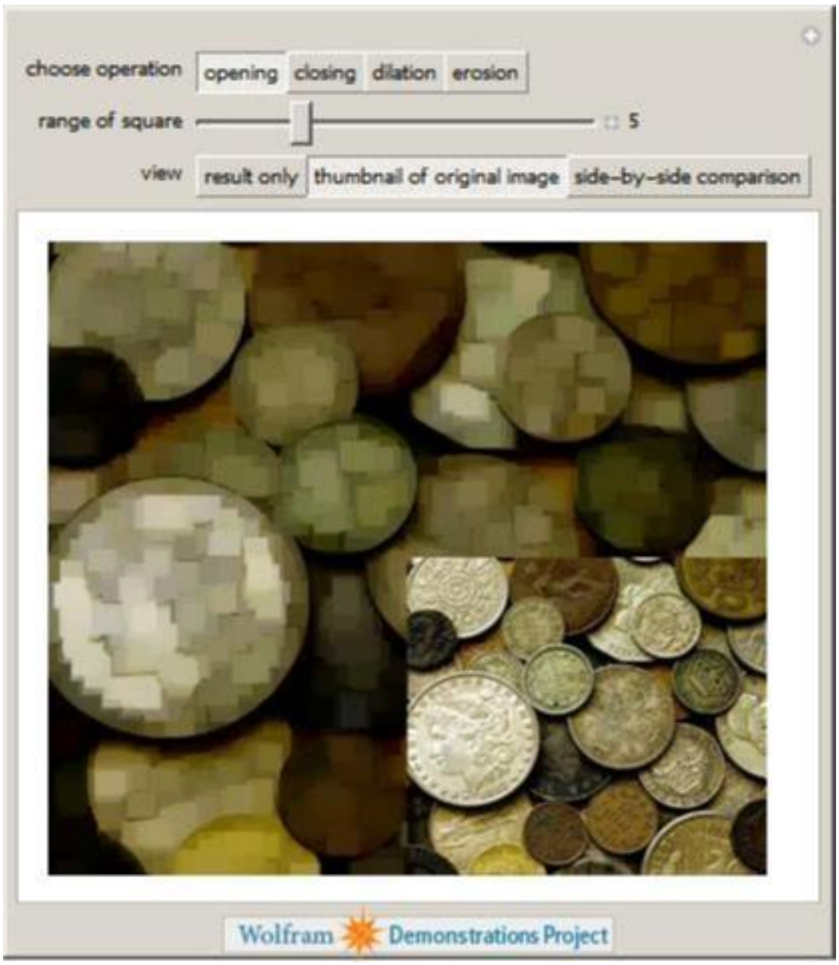


Fig 1.12 :- blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

1.6.8 Segmentation:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward a successful solution of imaging problems that require objects to be identified individually.

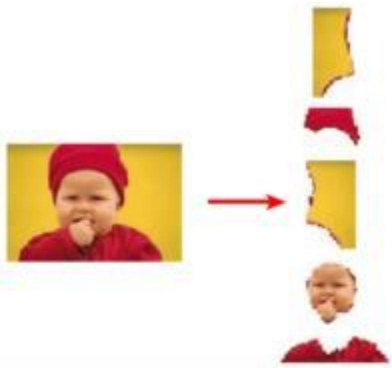


Fig 1.12 Image segmentation

9. Representation and Description:

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

10. Knowledge Base:

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an interrelated to list of all major possible defects in a materials inspection problem or an image database containing high-resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

1.6.11 Object Recognition:

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

7. Feature Extraction and Feature Matching Techniques :

1. Oriented FAST And Rotated BRIEF(ORB) :

ORB, starts by finding special regions in an image called key points. Key points are the highly-distinctive locations in an image. For example, the edges of an image (where the brightness varies i.e where the pixel values change sharply).

How does ORB work?

1. First, it uses FAST to find key points, then applies a Harris corner measure to find top N points among them. FAST stands for Features from Accelerated Segments Test, and it quickly selects key points by comparing the brightness levels in a given pixel area.

2. BRIEF, to create feature vectors.

2. Feature Detection using FAST(Features from Accelerated Segments Test):

1. Select a pixel p in the image which is to be identified as an interesting point or not. Let its intensity be I.

2. Select the appropriate threshold value t.

3. Consider a circle of 16 pixels around the pixel under test. (See the image below).

3. **BRIEF (Binary Robust Independent Elementary Features):**

The second part of the ORB algorithm is to take the key points found by the FAST algorithm and turn those into feature vectors that together can represent an object. To create feature vectors, ORB uses a BRIEF algorithm.

It basically creates binary feature vectors from a set of key points. As we already know a binary feature vector, is also known as a binary descriptor, which is just a feature vector that contains only ones and zeros

BRIEF provides a shortcut to find the binary strings directly. It takes a smoothened image patch and selects a set nd(x,y) location pairs in a unique way and then pixel intensity comparisons are done on these location pairs.

In short, BRIEF is a faster method feature descriptor calculation and matching. It also provides a high recognition rate unless there is a large in-plane rotation.

4. **2 Nearest Neighbour(2NN) with Hierachical agglomerative clustering(HAC):**

There are many approaches to hierarchical clustering as it is not possible to investigate all clustering possibilities.One set of approaches to hierarchical clustering is known as agglomerative, whereby in each step of the clustering process an observation or cluster is merged into another cluster.

Hierarchical clustering is a widely used and popular tool in statistics and data mining for grouping data into 'clusters' that exposes similarities or dissimilarities in the data. There are many approaches to hierarchical clustering as it is not possible to investigate all clustering possibilities. One set of approaches to hierarchical clustering is known as agglomerative, whereby in each step of the clustering process an observation or cluster is merged into another cluster. The first approach we will explore is known as ‘**Single Linkage’method**’ also known as nearest neighbors.

5. Noise :

Noise Images acquired through modern sensors may be contaminated by a variety of noise sources. By noise we refer to stochastic variations as opposed to deterministic distortions such as shading or lack of focus. We will assume for this section that we are dealing with images formed from light using modern electrooptics. In particular we will assume the use of modern, charge-coupled device (CCD) cameras where photons produce electrons that are commonly referred to as photoelectrons. Nevertheless, most of the observations we shall make about noise and its various sources hold equally well for other imaging modalities. While modern technology has made it possible to reduce the noise levels associated with various electro-optical devices to almost negligible levels, one noise source can never be eliminated and thus forms the limiting case when all other noise sources are “eliminated”.

1.8 IMAGE RETRIEVAL:

Image retrieval is nothing but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords or the descriptions to the images so that the retrieval can be performed. Manual image annotation is time consuming, expensive and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc..,

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others. one can today generate volumes of images with content as diverse as family get-togethers and national park visits. Low-cost storage and easy Web hosting has fueled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchable image data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real-world image search system designers.

As far as technological advances are concerned, growth in Content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real world implications, applications, and constraints of the technology. Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future.

An image retrieval system designed to serve a personal collection should focus on features such as personalization, flexibility of browsing, and display methodology. For example, Google's Picasa system [Picasa 2004] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods. The notable few applications include an FPGA implementation of a color-histogram-based image retrieval system [Kotoulas and Andreadis 2003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 2003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 2002].

Discussion. Regardless of the nature of the collection, as the expected user-base grows, factors such as concurrent query support, efficient caching, and parallel and distributed processing of requests become critical. For future real-world image retrieval systems, both software and hardware approaches to address these issues are essential. More realistically, dedicated specialized servers, optimized memory and storage support, and highly parallelizable image search algorithms to exploit cluster computing powers are where the future of large-scale image search hardware support lies.

CHAPTER-2

LITERATURE SURVEY

Over the past several years, numerous MRI contrast enhancement and denoising approaches have been reported in the literature. This step of image preprocessing is very crucial for the efficiency of the subsequent image processing steps, for instance segmentation, classification and reconstruction. Among these approaches, one could report the linear filtering (spatial and temporal [2]) and nonlinear filtering approaches, e.g. anisotropic diffusion filter (ADF), bilateral filtering [3].

There are also the transform-based approaches, e.g. wavelet and curvelet transforms [4] as well as statistical-based approaches e.g. maximum likelihood (ML), random forest and Markov Random Field (MRF), etc. For MRI contrast enhancement, the histogram modifications based approaches have gained a great research interest by scientists due to their advantages: simplicity and ease of use, higher SNR as well as its low computational complexity.

In fact, the most employed classical images contrast enhancement technique is the Histogram Equalization (HE), which provides good performance for ordinary images, such as human portraits or natural images.

However, in MRI brain images, there are often more complex situations where the brain tumor White matter hyper intensities (WMH) signal is high but it may mistakenly be considered as other brain tissues such as Cerebrospinal fluid (CSF). In this paper, we will focus on the study of the major histogram modifications based approaches when applied to brain MRI images for contrast enhancement. Sazanita et al. [5] proposed a new approach applied for FLAIR MRI images, known as the Average Intensity Replacement–Adaptive Histogram Equalization (AIR-AHE), which enhances the contrast without over amplifying the entire image.

It improves the contrast of WMH relying on the intensity adjustment and contrast mapping techniques, more details of this method are explained in this paper. In [6], the authors proposed a Hierarchical Correlation Histogram Analysis algorithm (HCHA) which provides an automatic contrast enhancement of images during the examination of atrophic cell areas of Parkinson Disease (PD) patients.

This algorithm is based on the grayscale distribution degree of pixel intensity by constructing a correlation histogram.

This method has as objective to provide the best segmentation accuracy result and facilitates subsequently CAD processes. SENTHILKUMARAN et al. in [7] performed a comparative study of different histogram-based techniques, mainly the histogram equalization (HE), BHE, modified BHE, AHE and CLAHE, in order to enhance the contrast of general MRI brain images.

They used different evaluation metrics as Michelson contrast, RMS contrast, absolute mean brightness error (AMBE) and Pixel Distance.

While the major comparative studies of histogram-based techniques are performed on general MRI brain images, to the best our knowledge, there is no comparative study performed on MRI glioma brain tumors. In fact, the main objective of the present comparative study is to find, over the existing recent contrast-enhancement methods, the most preferment and effective techniques for the MRI Glioma brain tumors images, specifically glioblastoma. Since glioblastomas are infiltrative tumors, their borders are often fuzzy and hard to distinguish from healthy tissues [1]. Furthermore in some cases, the tumor region might share the same intensity profile with other normal regions within the image [14].

These characteristics make the localization of tumor core edges a very difficult task for radiologists.

Thus, the MRI contrast-enhancement is important on the one hand, it is a clinical necessity that facilitate, for radiologists and neurologists, the tumor detection and distinguish it from other tissues (edema and necrosis), and on the other hand, it makes easier for the CAD to have more efficiency and accuracy in different tasks such as automatic segmentation and even the classification task.

Gliomas are the most common and aggressive type of brain tumors due to their infiltrative nature and rapid progression. The process of distinguishing tumor boundaries from healthy cells is still a challenging task in the clinical routine. Fluid-attenuated inversion recovery (FLAIR) MRI modality can provide the physician with information about tumor infiltration. Therefore, this paper proposes a new generic deep learning architecture, namely DeepSeg, for fully automated detection and segmentation of the brain lesion using FLAIR MRI data.

The developed DeepSeg is a modular decoupling framework. It consists of two connected coreparts based on an encoding and decoding relationship. The encoder part is a convolutional neural network (CNN) responsible for spatial information extraction. The resulting semantic map is inserted into the decoder part to get the full-resolution probability map. Based on modified U-Net architecture, different CNN models such as residual neural network (ResNet), dense convolutional network (DenseNet), and NASNet have been utilized in this study.

This study showed successful feasibility and comparative performance of applying different deep learning models in a new DeepSeg framework for automated brain tumor segmentation in FLAIR MR images.

CHAPTER-3

PROPOSED SYSTEM

3.1 BLOCK DIAGRAM :-

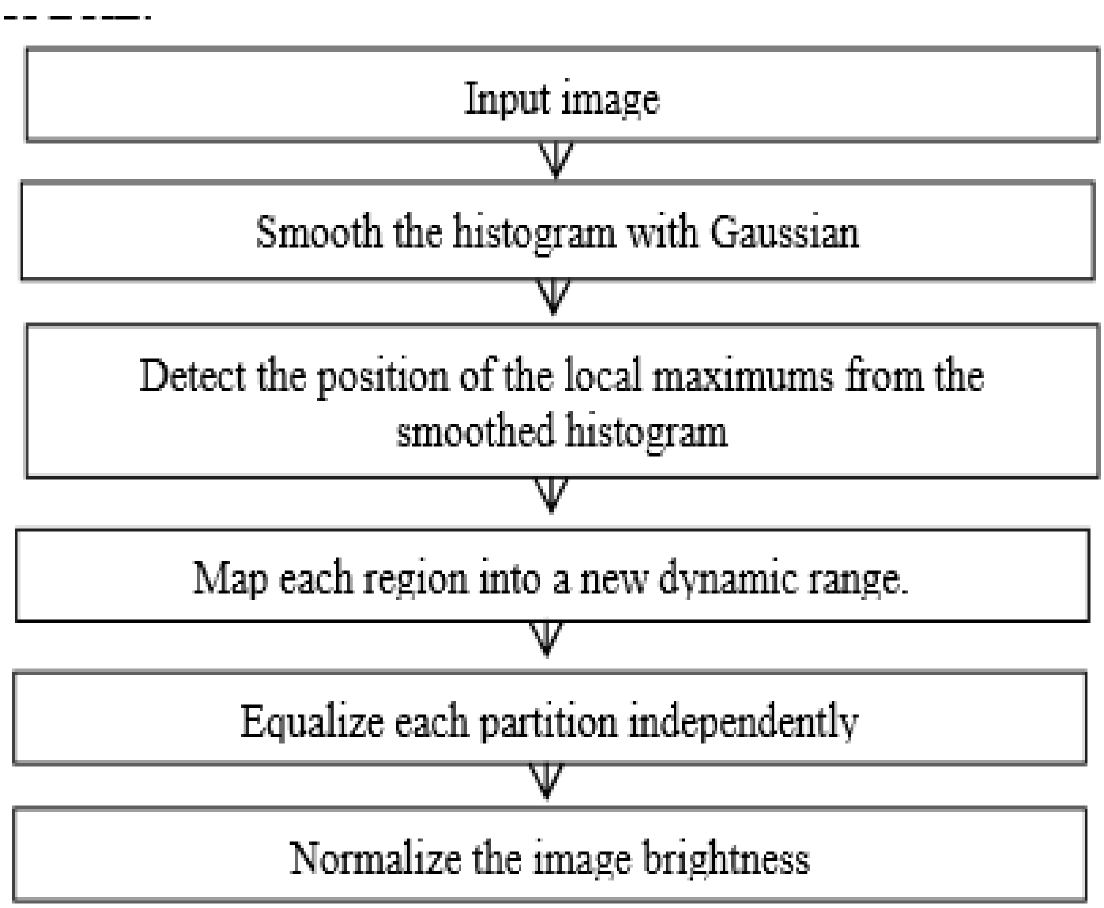


Fig 3.1 Block diagram

Unlike other contrast enhancement techniques, BPDHE divides the input image into sub-regions based on the position of the local maximums of input histogram. The next step is to equalize each partition separately . Before the equalized histogram taking place, BPDHE first map each sub image to a new dynamic range. This change in the dynamic range will cause modification in the mean brightness of the output image, thus the final step of this method is the normalization of the output intensity.

It is very difficult to find the exact location of local maximums due to histogram fluctuation without smoothing the histogram and also the probability that there is some brightness levels are missing. Thus the first step of BPDHE is to apply Gaussian filter to the input histogram [14]. To keep the mean brightness of the input image, BPDHE apply the brightness normalization process, as define by: (5) Where Mi is the mean brightness of the input image, Mf is the mean brightness after the equalization process and f(x, y) is the output image after the equalization process. D. Average Intensity Replacement based on Adaptive Histogram Equalization (AIR-AHE) AIR-AHE is a recent contrast enhancement technique for brain MRI images proposed by Iza et al. [5]. This technique, composed of four processing steps, combines several established enhancement algorithms:

• Step1: Contrast stretching process where the principal purpose is increasing the contrast of the enhanced image.

• Step2: Apply the CLAHE technique with clipping limits in the range of 0-1, in order to improve the contrast of the potential regions;

• Step3: Region of maximum intensity where the higher intensity location in the potential regions is determined. It could be further improved by including the regional maxima function.

• Step4: Window sliding neighborhood operation for intensity measurement. A sliding neighborhood operation is used to determine the average intensity value of the pixel's neighborhood. It is performed by sliding a set of pixels relative to the center of the pixel [4].

CHAPTER-4

SOFTWARE AND ITS FUNCTIONS

1. Introduction To MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation.
- Algorithm development.
- Data acquisition.
- Modeling, simulation, and prototyping.
- Data analysis, exploration, and visualization.
- Scientific and engineering graphics.
- Application development, including graphical user interface building.

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high- productivity research, development, and analysis.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high- productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

2. The MATLAB system:

The MATLAB system consists of five main parts

Development Environment:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

The MATLAB Mathematical Function Library:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

The MATLAB Language:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both “programming in the small” to rapidly create quick and dirty throw-away programs, and “programming in the large” to create large and complex application programs.

Graphics:

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications

The MATLAB Application Program Interface (API):

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

3. GRAPHICAL USER INTERFACE (GUI):

MATLAB’s Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

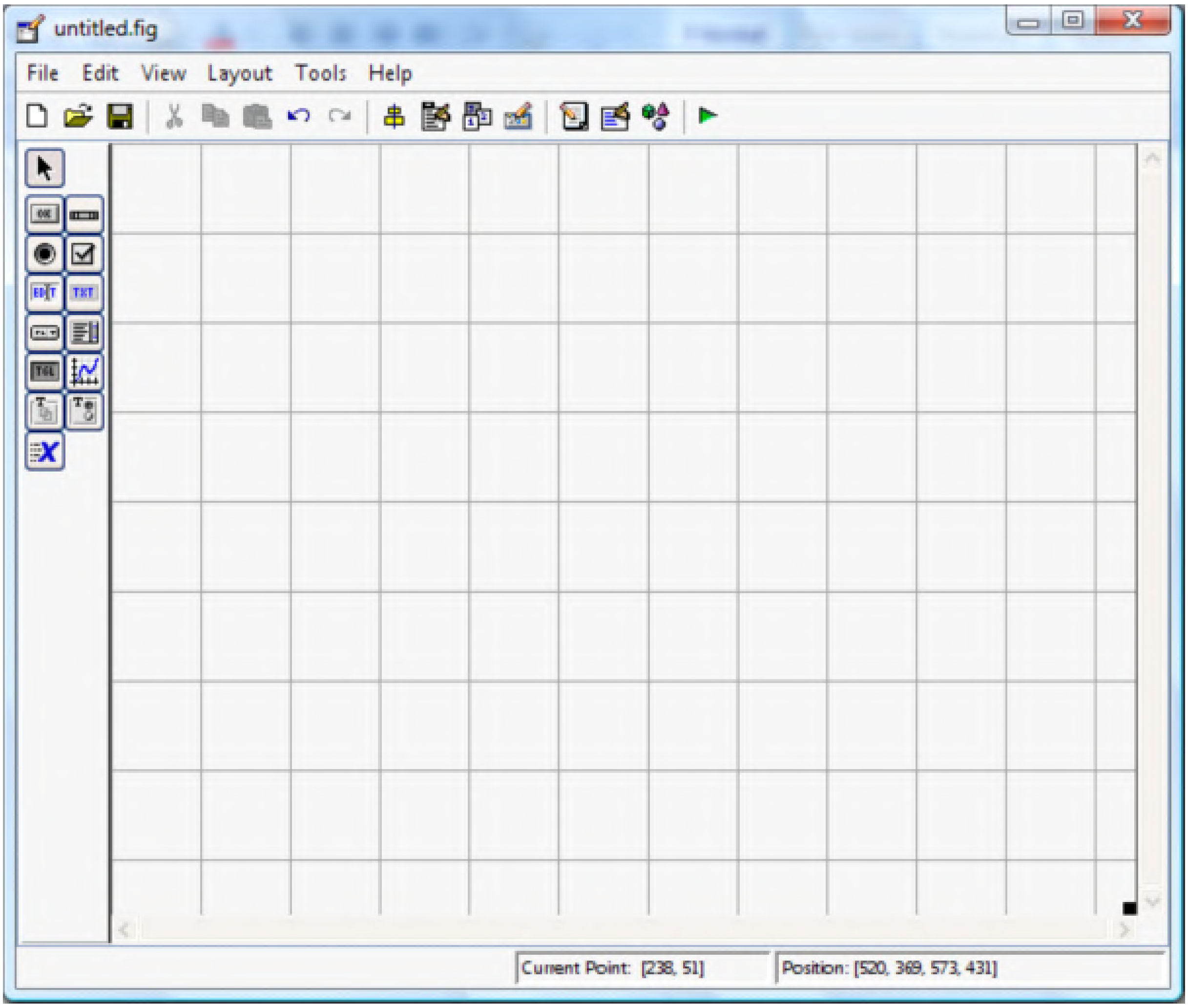
A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.

A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed

To launch GUIDE from the MATLAB command window, type

guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted, GUIDE opens a new (i.e., blank) window



A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

Getting Started

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

1. **Introduction** - describes the components of the MATLAB system.

2. **Development Environment** - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

3. **Manipulating Matrices** - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

4. **Graphics** - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

5. **Programming with MATLAB** - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

4. DEVELOPMENT ENVIRONMENT

1. Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

2. Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

3. Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

4. MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

4.5 Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad

- Help Browser

1. Command Window

Use the Command Window to enter variables and run functions and M-files.

2. Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

3. Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

4. Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

5. Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type helpbrowser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

6. Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

4.5.7 Display Panel

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

4.6 Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

4..6.1 Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

4.6.2 Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

3. Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

4. Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to runMATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function

4.7 MANIPULATING MATRICES

4.7.1 Entering Matrices

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Dürer's matrix as a list of its elements. You have only to follow a few basic conventions:

Separate the elements of a row with blanks or commas.

Use a semicolon, ; , to indicate the end of each row.Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]

MATLAB displays the matrix you just entered.

```
A =

   16    3    2   13

    5   10   11    8

    9    6    7   12

    4   15   14    1
```

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

4.7.2 Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

4.7.3 Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

4.7.4 Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

4.7.5 Operators

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

4.7.6 Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.14159265...
I	Imaginary unit, √-1
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

4.8 GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

1. Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

2. GUI Development Environment

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

4.8.3 The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of its children (uicontrols and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the callbacks, which are defined as subfunctions. This M-file is referred to as the application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

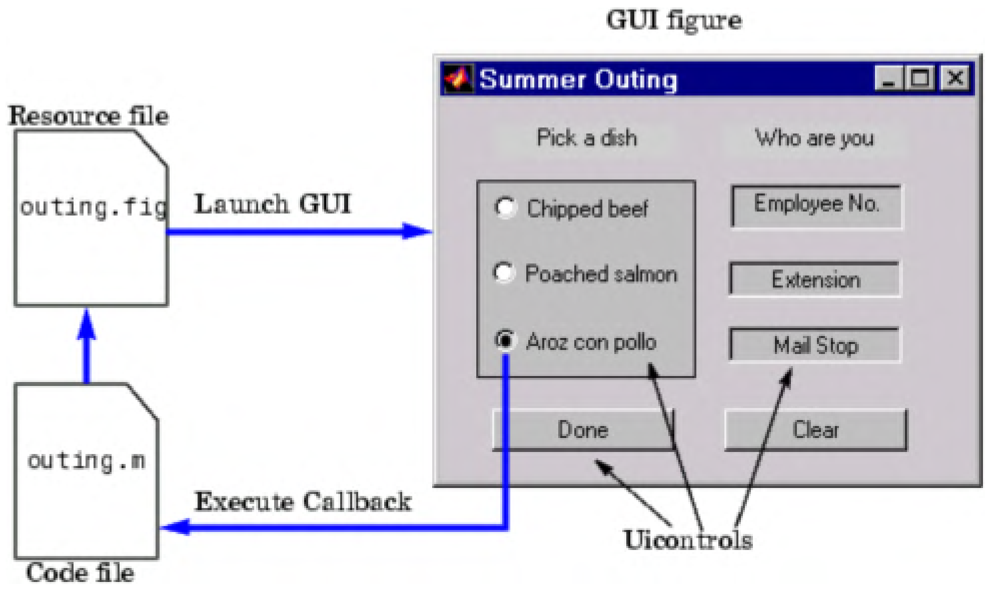


FIG 4.2 graphical user blocks

Features of the GUIDE-Generated Application M-File

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

4.8.4 Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques used in the application M-file.

Application Examples - a collection of examples that illustrate techniques which are useful for implementing GUIs.

9. Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

1. User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- Listboxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

4.9.2 From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

function varargout = togglebutton1_Callback**5 User Interface Control**

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames

- Radio Buttons
- Listboxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback:

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons:

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)
button_state = get(h,'Value');
if button_state == get(h,'Max')
   % toggle button is pressed
elseif button_state == get(h,'Min')
   % toggle button is not pressed
end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);
a(:,:,2) = rand(16,128);
a(:,:,3) = rand(16,128);
set(h,'CData',a)
```

Radio Buttons

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)

set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)

off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];

mutual_exclude(off)

% Continue with callback

   .

   .

   .
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)
```

```
if (get(h,'Value') == get(h,'Max'))
   % then checkbox is checked-take approriate action
else
   % checkbox is not checked-take approriate action
end
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)
user_string = get(h,'string');
% proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function  edittext1_Callback(h,eventdata,handles,varargin)
user_entry = str2double(get(h,'string'));
if isnan(user_entry)
   errordlg('You must enter a numeric value','Bad Input','modal')
end
% proceed with callback…
```

Triggering Callback Execution

On UNIX systems, clicking on the menubar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menubar does not cause the editable text callback routine to execute.

This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

Listboxes differentiate between single and double clicks on an item and set the figure SelectionType property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure SelectionType property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)
val = get(h,'Value');
switch val
case 1
% The user selected the first item
case 2
% The user selected the second item
% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)
val = get(h,'Value');
string_list = get(h,'String');
selected_string = string_list{val}; % convert from cell array to string
% etc.
```

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes ButtonDownFcn property to define the callback.

Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

axes(handles.axes1)

4.10 Software Functions:

- **imread function**:- The **imread function** supports four general syntaxes, described below. ... See Special Case Syntax for information about these syntaxes

  A = **imread**(filename, fmt)

  reads a greyscale or color image from the file specified by the string filename ,where the string fmt specifies the format of the file.

- **imshow function :-** The **imshow function** displays the value low (and any value less than low) as black, and it displays the value high (and any value greater than high) as white. Values between low and high are displayed as intermediate shades of gray, using the default number of gray levels.

B = **imshow**(A);

- **Imwrite function :- imwrite** function converts indexed images to RGB before writing data to JPEG files, because the JPEG format does not support indexed images. The **imwrite function** does not support writing of indexed PNG files that have insufficient colormap entries.

  A = **imwrite**(variable,"Newfilename.Newfileformat');

- **Uigetfile :- uigetfile** displays a dialog box used to retrieve one or more files. The dialog box lists the files and directories in the current directory.

  uigetfile(' FilterSpec ')

  It displays a dialog box that lists files in the current directory.

- **Imresize :-** B = **imresize**(A , scale)

  returns image B that is scale times the size of A . The input image A can be a grayscale, RGB, or binary image. If A has more than two dimensions, **imresize** only resizes the first two dimensions.

CHAPTER-5

OUTPUTS :

Fig 5.1 :- input image 1

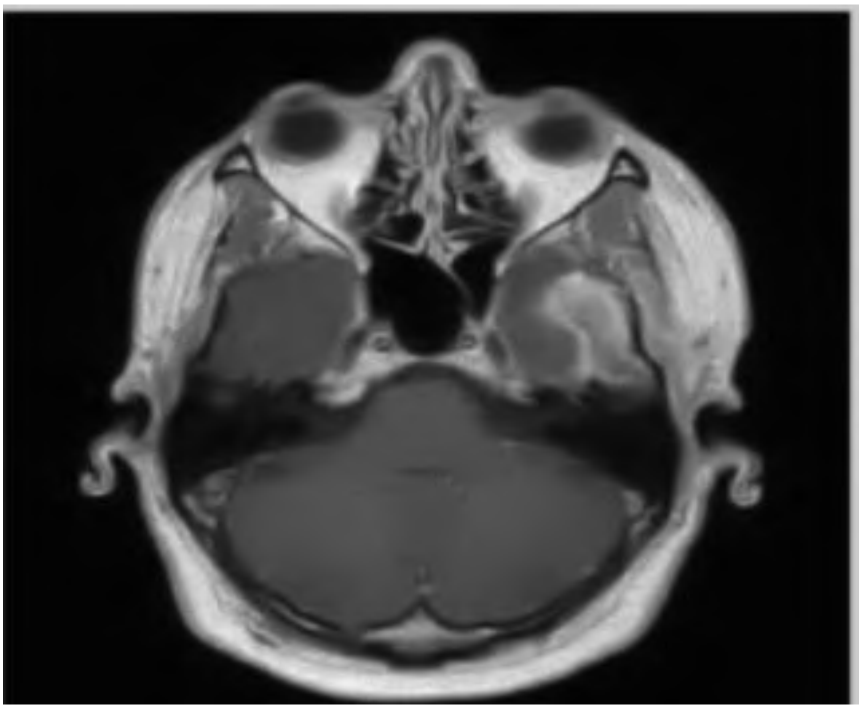


Fig 5.2 :- output image 1

OUTPUTS :

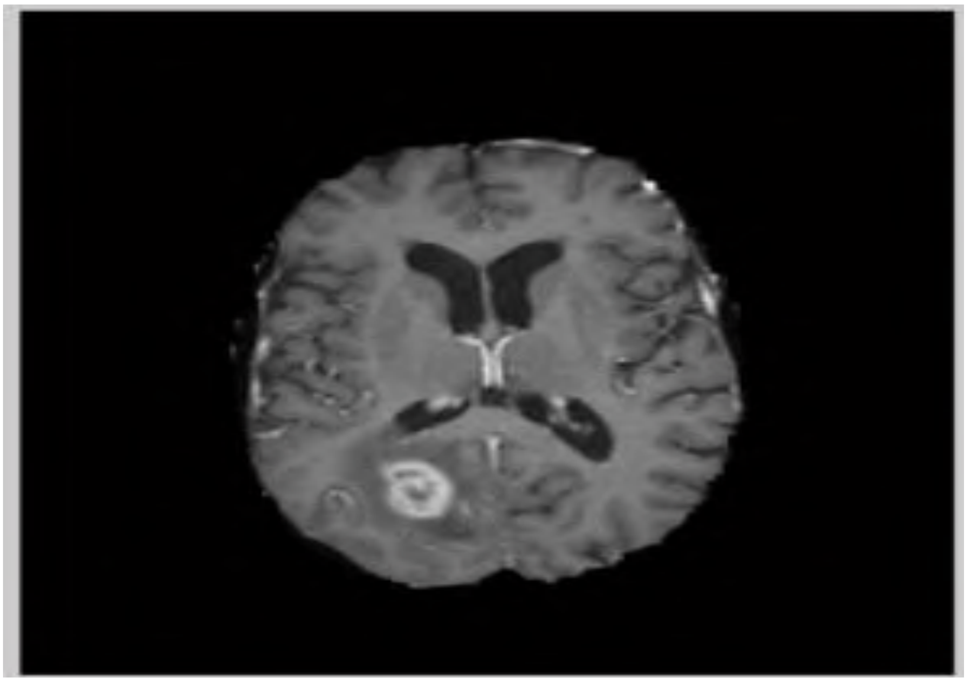


Fig 5.3 :- input image 2

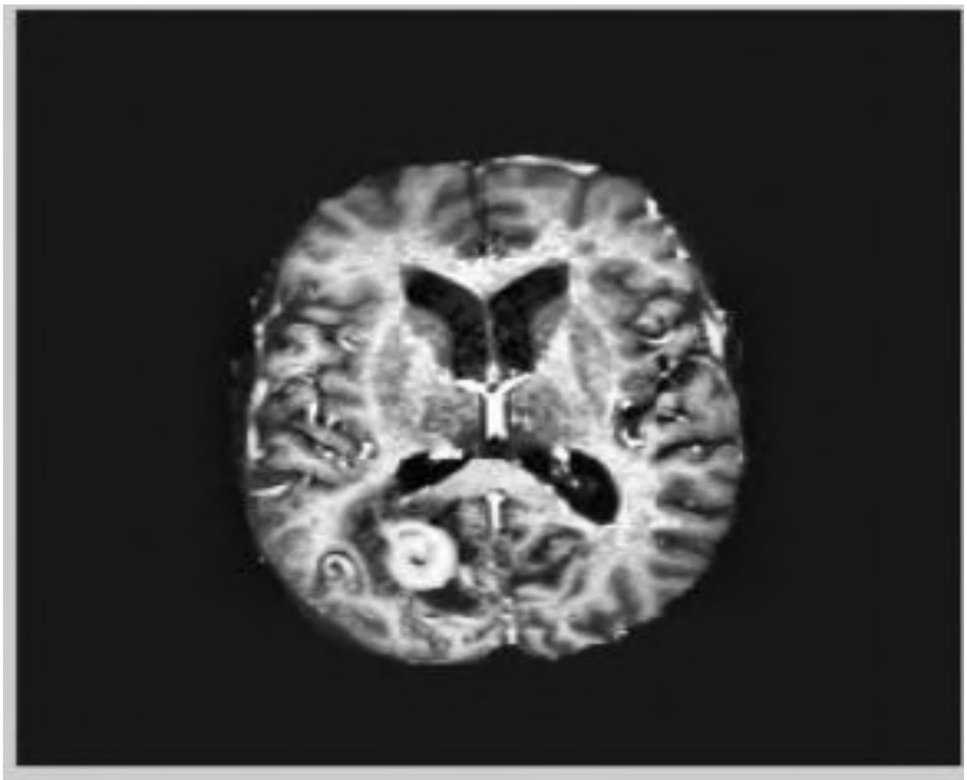


Fig 5.4 :- output image 2

CHAPTER-6

CONCLUSION :

The main objective of this paper is to provide a comparative study of some existing techniques of contrast enhancement based on histogram equalization for MRI Glioblastoma brain tumour. Particularly the AHE, CLAHE, AIR-AHE and BPDHE methods are explained and then compared. Each studied method is evaluated relying on image quality measurement mainly AMBE, PSNR and entropy. For the evaluation process, we selected the most relevant slices where the tumour core appears clearly, then we computed the average value (with a standard deviation) of the quality evaluation metrics which makes the evaluation to be more precise. Through this study one could notice that, Adaptative Histogram Equalization (AHE) technique provides efficient performances for MRI contrast enhancement compared to other studied techniques. For future works, we will focus on enhancing the AHE techniques by introducing filtering approaches that could improve the results in terms of accuracy and treatment efficiency.

REFERENCES

[1] HAVAEI, Mohammad, DAVY, Axel, WARDE-FARLEY, David, et al. Brain tumour segmentation with deep neural networks. Medical image analysis, 2017, vol. 35, p. 18-31

[2] MOHAN, J., KRISHNAVENI, V., et GUO, Yanhui. A survey on the magnetic resonance image denoising methods. Biomedical Signal Processing and Control, 2014, vol. 9, p. 56-69.

[3] ANAND, C. Shyam et SAHAMBI, Jyotinder S. Wavelet domain nonlinear filtering for MRI denoising. Magnetic Resonance Imaging, 2010, vol. 28, no 6, p. 842-861.

[4] ALEXANDER, M. E., BAUMGARTNER, R., SUMMERS, A. R., et al. A wavelet-based method for improving signal-to-noise ratio and contrast in MR images. Magnetic Resonance Imaging, 2000, vol. 18, no 2, p. 169-180

A Major Project report on

Automatic Railway Train Safety System

Submitted in partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

IN

Electronics & Communication Engineering

Submitted BY

Y.Suresh Kumar	**(18K85A0424)**
G. Ruchitha	**(18K85A0426)**
DV. Harish	**(18k85A0434)**

Under the Guidance of

Mrs.T.Sri Lakshmi
M. Tech. (Assistant Professor)

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

St. MARTIN'S ENGINEERING COLLEGE
(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)
An UGC Autonomous Institute
Dhulapally,Secunderabad-500 100.
NBA &NAAC A+ Accredited

2020-2021

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally,Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the major-project work entitled **"Automatic Railway Train Safety System"** is a Bonafide work carried out by **Y.Suresh Kumar (18K85A0424), G.Ruchitha (18K85A0426), DV.Harish (18K85A0434)** in partial fulfilment of the requirements for the degree of **Bachelor of Technology in Electronics & Communication Engineering** by the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2020-21.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree.

INTERNAL GUIDE	HEAD of THE DEPARTMENT, ECE
Mrs.T.Sri Lakshmi Assistant Professor, Department of ECE.	Dr.B. Hari Krishna, Professor

EXTERNAL EXAMINER

// ACKNOWLEDGEMENT

We sincerely thank the management of our college **St. MARTIN'S ENGINEERING COLLEGE** for providing required facilities during our project work.

We derive our great pleasure in expressing our sincere gratitude to our principal **Dr.P. Santosh Kumar Patra** for his timely suggestions, which helped us to complete the project work successfully.

It is the very auspicious moment to express our gratitude to **Dr. B. Hari Krishna**, Head of the Department, ECE for his consistent encouragement during the progress of this project.

We take it as a privilege to thank our project coordinator **Mr.M.Nishakar** Associate Professor, department of ECE for the ideas that led to complete the project work and we also thank them for their continuous guidance, support and unfailing patience, throughout the course of this work.

We sincerely thank our project Internal guide **Mrs.T.Sri Lakshmi,** Assistant Professor**,** Department of ECE for guidance and encouragement in carrying out this project work.

DECLARATION

We, the students of '**Bachelor of Technology in Department of Electronics and Communication Engineering**', session: 2018 – 2021, **St. Martin's Engineering College**, hereby declare that the project work entitled '**Automatic Railway Train Safety System**' is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

Y.Suresh kumar	G.Ruchitha	DV.Harish
18K85A0424	18K85A0426	18K85A0434

ABSTRACT

Automatic Railway Safety System can sense the presence of fire and smoke and generate a series of alarm driven events after it. When the fire is detected it turns on a motor which is depicted in the project as breaking system or chain pulling mechanism. Thus by this the train can be stopped and the passengers and other payloads can be safeguarded thereafter. Along with breaking, the system sounds a buzzer that would alert nearby people around it so that they can be cautious about the presence of fire in the Train. Also, an SMS is sent to the Railway Authorities which will help them to take quick decisions to take control of the fire and in evacuating the Train. Hence this project offers a very robust mechanism for safety in the Railways which works automatically with the help of microcontrollers and sensors.

.

TABLE OF CONTENTS

LIST OF FIGURES

TABLE OF TABLES

CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION ABOUT EMBEDDED SYSTEMS:

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.
The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes.

The applications software on such processors is sometimes referred to as firmware.
The simplest devices consist of a single microprocessor (often called a "chip”), which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine.

As the embedded system is the combination of both software and hardware

1.1.1 Block Diagram of Embedded System:

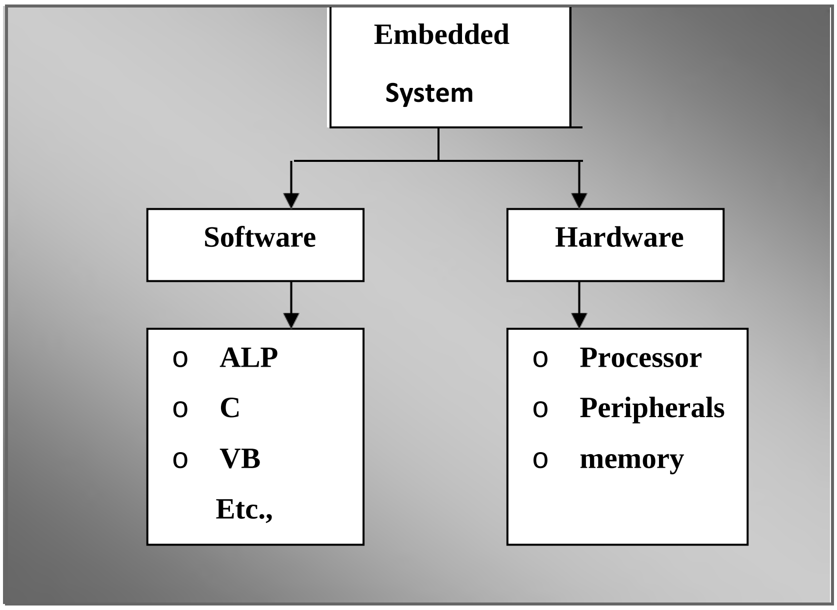


Fig.1.1: Block diagram of embedded system

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory

1.1.2 Applications of Embedded Systems:

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

1.2 MICRO PROCESSOR (µP):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Processors are classified into four types like:

- Micro Processor (µp)
- Micro controller (µc)
- Digital Signal Processo1r (DSP)
- Application Specific Integrated Circuits (ASIC)

1.2.1 Three Basic Elements Of A Microprocessor:

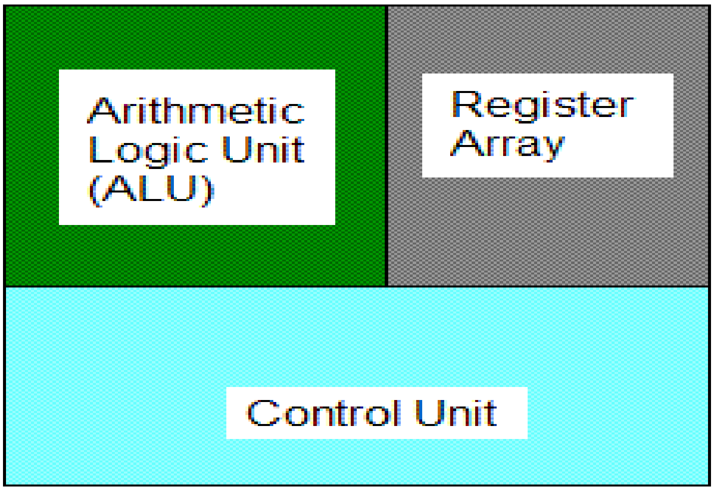


Fig.1.2: Three basic elements of a microprocessor

1.3 HARVARD ARCHITECTURE:

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data.

The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

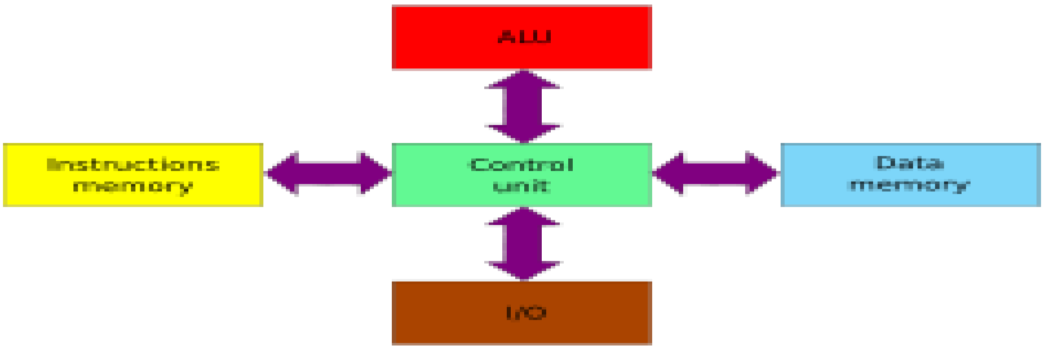


Fig.1.3: Harvard Architecture

1.3.1 Uses of The Harvard Architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

- Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behaviour must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.

- Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access.
- The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

1.4 VON-NEUMANN ARCHITECTURE

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. They cannot be performed at the same time. The von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

Astored-programdigit computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

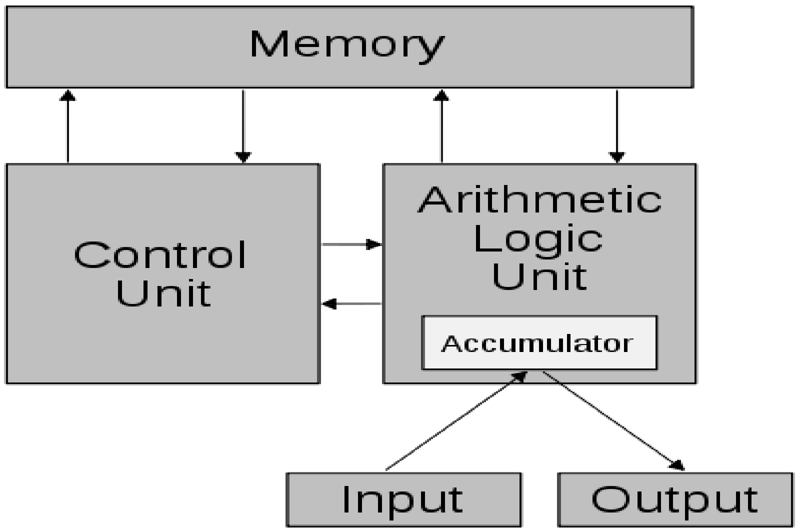


Fig.1.4: Schematic of the Von-Neumann Architecture.

1.5 PROJECT DESCRIPTION

1.5.1 Objective:

The aim and objective of the proposed project is to improve the safety of the railway system and to save human life. The key objective of this application is public safety. To automatic pull the chain when fire is accurs.To detect fire attack in train and control using buzzer.

1.5.2 Components Used:

- Arduino Uno
- Power supply
- LCD
- GSM
- L293D
- Motor
- Temperature sensor
- Gas sensor

CHAPTER-2

LITERATURE SURVEY AND EXISTING METHODS

2.1 LITERATURE SURVEY

Now days, the usage of fire alarm is become popular in office and homes. So, market will be up because of the request. But, users are no longer satisfied with a simple fire alarm detector that just senses and sounds the alarm. Many would like to have a fire alarm system that monitors and alerts them in case of emergency especially when they are away. Price for this system is usually expensive and its installation is complicated. However, user always suspected the cost of that system is cheap and easy to install. Traditional fire alarm systems are classified in a hardwired. So it will use many cables to connect one another. User's find it's untidy and inconvenience to

A sensor is an electronic device used to measure a physical quantity and convert into an electrical signal, which can be read by an observer. The common types of automatic fire detectors are thermal and flame detectors. The thermal detectors are ability to identify high temperature, smoke detectors replicate the sense of smell , and flame detectors are electronic eyes. The properly selected and installed automatic fire detector can be a highly reliable sensor. Unfortunately, a person can also be an unrel

ghly reliable sensor. Unfortunately, a person can also be an unreliable detection method since they may not be present when a fire starts, may not be in perfect health to recognize fire signatures. The advantage of manual alarm stations is that, upon discovering the fire, they provide occupants with a readily identifiable means to activate the building fire alarm system. The alarm system can serve in life of the shouting person's voice. They are simple devices, and can be highly reliable when the building occupied

2.2 EXISTING METHODS

In The existing conventional signaling system most of the times relay on the oral communication through telephonic and telegraphic conversations as input for the decision making in track allocation for trains.There is large scope for miscommunication of the information or communication gap due to the higher human interface in the system.This miscommunication may lead to wrong allocation of the track for trains , which ultimately leads to the train collision.

2.3 PROPOSED SYSTEM

In This proposed system have to developed an automatic fire detection and rescue system based on GSM. In order to provide an early extinguishing of fire, large numbers of detectors, which measure temperature and gas concentrations are deployed in train. When there is a detection of fire, an alarm sounds and the same time, the information reports their organization via GSM

CHAPTER-3

BLOCK DIAGRAM

3.1 BLOCK DIAGRAM

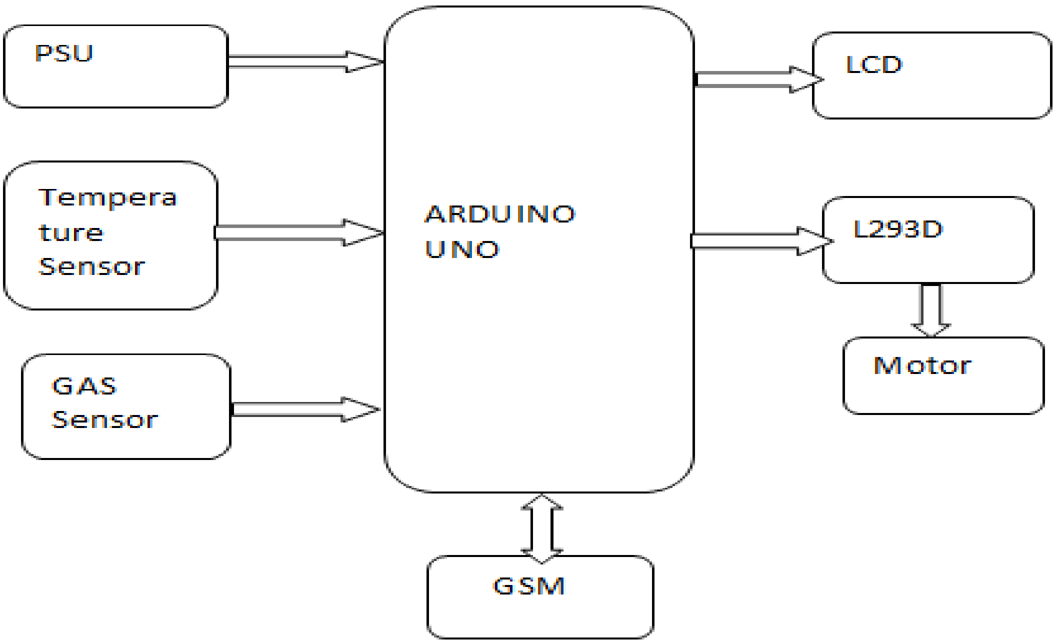


Fig.3.1: Block diagram of the project

The block diagram of Automatic Railway Train Safety System is shown in above figure. The blocks of the circuit are

- Arduino UNO
- Power supply Unit
- Temperature Sensor
- Gas sensor
- GSM
- LCD
- L293D Motor Driver
- Motors

3.2 SCHEMATIC EXPLANATION

3.2.1 Interfacing Between LCD and Arduino

The following circuit diagram shoes the liquid crystal display with Arduino module. From the circuit diagram, we can observe that the RS pin of the LCD is connected to the pin 7 of the Arduino. The LCD of R/W pin is connected to the ground. The pin 6 of the Arduino is connected to the enable signal pin of LCD module. The LCD module & Arduino module are interfaced with the 4-bit mode in this project. Hence there are four input lines which are DB4 to DB7 of the LCD. This process very simple, it requires fewer connection cables and also we can utilize the most potential of the LCD module.

The digital input lines (DB4-DB7) are interfaced with the Arduino pins from 2-5. To adjust the contrast of the display here we are using a 10K potentiometer. The current through the back LED light is from the 560-ohm resistor. The external power jack is provided by the board to the Arduino. Using the PC through the USB port the Arduino can power. Some parts of the circuit can require the +5V power supply it is taken from the 5V source on the Arduino board.

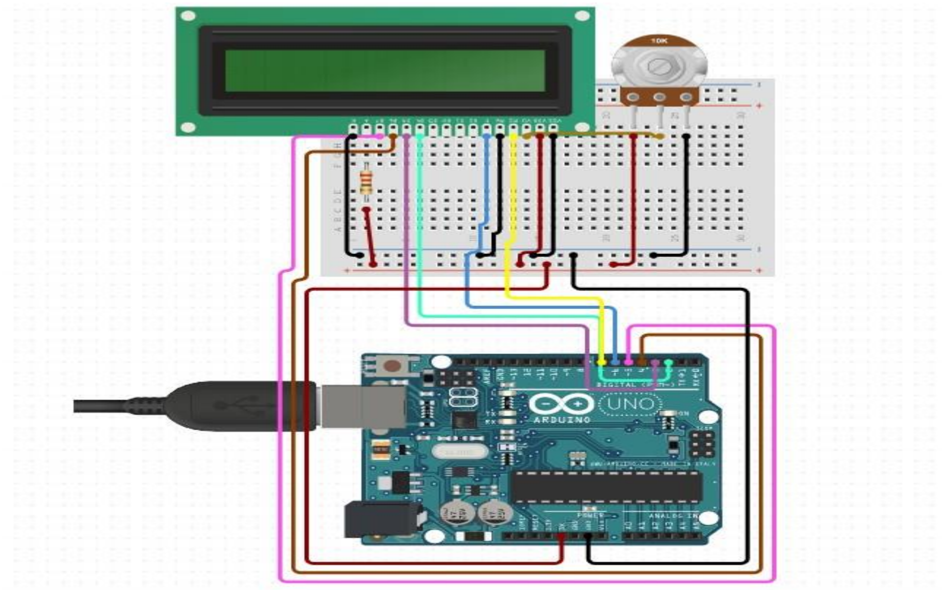


Fig.3.2: Interfacing Between LCD and Arduino

3.2.2 Interfacing Between Temperature Sensor and Arduino

4 The +5v for LM35 can be taken from the +5v out pin of arduino uno. Also the ground pin of LM35 can be connected to GND pin of arduino uno. Connect Vout (the analog out of LM35) to any of the analog input pin of arduino uno. In this circuit diagram, we have connected Vout of LM35 to A1 of arduino.

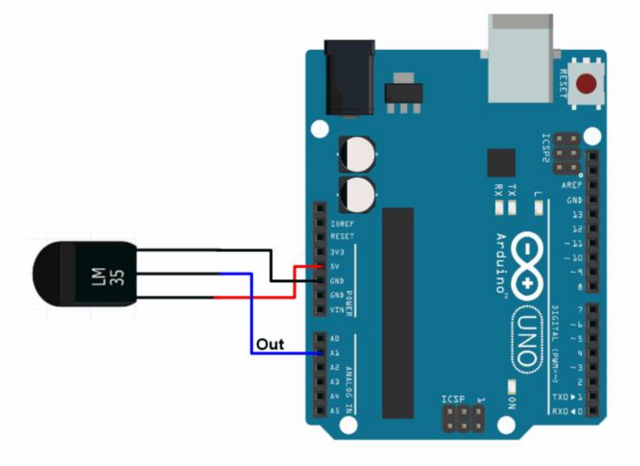


Fig.3.3: Schematic diagram of Temperature Sensor and Arduino

3.2.3 Interfacing between PC and GSM Module:

A GSM modem can be an external device or a PC Card / PCMCIA Card. Typically, an external GSM modem is connected to a computer through a serial cable or a USB cable. A GSM modem in the form of a PC Card / PCMCIA Card is designed for use with a laptop computer. It should be inserted into one of the PC Card / PCMCIA Card slots of a laptop computer. Like a GSM mobile phone, a GSM modem requires a SIM card from a wireless carrier in order to operate.

A SIM card contains the following information:

- Subscriber telephone number (MSISDN)
- International subscriber number (**IMSI, International Mobile Subscriber Identity)**
- State of the SIM card
- Service code (operator)
- Authentication key
- PIN (***Personal Identification Code***)
- PUK (***Personal Unlock Code***)

Computers use AT commands to control modems. Both GSM modems and dial-up modems support a common set of standard AT commands. In addition to the standard AT commands, GSM modems support an extended set of AT commands. These extended AT commands are defined in the GSM standards. With the extended AT commands, the following operations can be performed:

- Reading, writing and deleting SMS messages.
- Sending SMS messages.
- Monitoring the signal strength.
- Monitoring the charging status and charge level of the battery.
- Reading, writing and searching phone book entries.

The number of SMS messages that can be processed by a GSM modem per minute is very low i.e., about 6 to 10 SMS messages per minute.

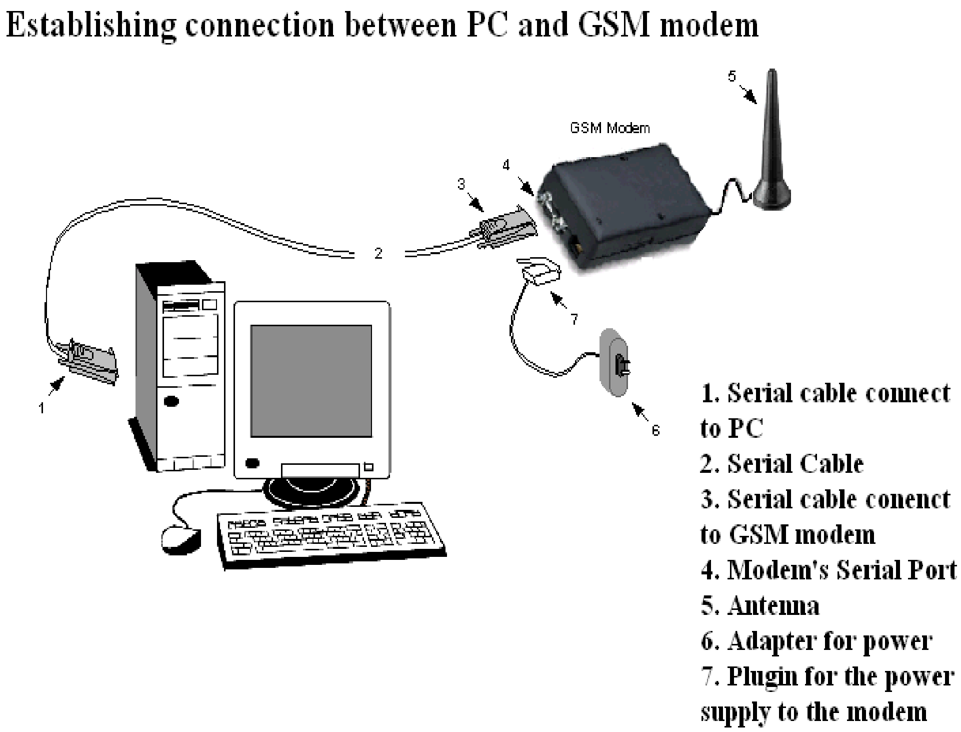


Fig.3.4 Interferance between the PC and GSM

3.2.4 Interfacing between Arduino and L293D

4-7 numbered digital pins act as output pins from Arduino are connected to the motor driver module i.e L293D as input and output of the module is connected to motors for forward or backward movements,

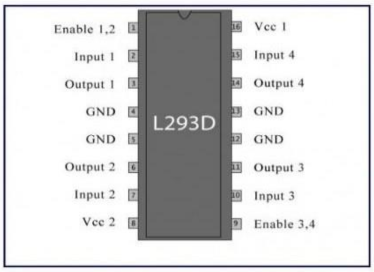


Fig.3.5. L293D Motor Driver

All the Ground pins should be grounded. There are two power pins for this IC, one is the Vss (Vcc1) which provides the voltage for the IC to work, this must be connected to +5V. The other is Vs (Vcc2) which provides voltage for the motors to run, based on the specification of your motor you can connect this pin to anywhere between 4.5V to 36V, here I have connected to +12V.

From the figure Enable pins (Enable 1, 2 and Enable 3, 4) are used to Enable Input pins for Motor 1 and Motor 2 respectively. Since in most cases we will be using both the motors both the pins are held high by default by connecting to +5V supply. The input pins Input 1, 2 which are digital pins 4,5 from Arduino are used to control the motor 1 and Input pins 3, 4 which are digital pins 6,7 from Arduino are used to control the Motor 2. The input pins are connected to the any Digital circuit or microcontroller to control the speed and direction of the motor. You can toggle the input pins based on the following table to control your motor.

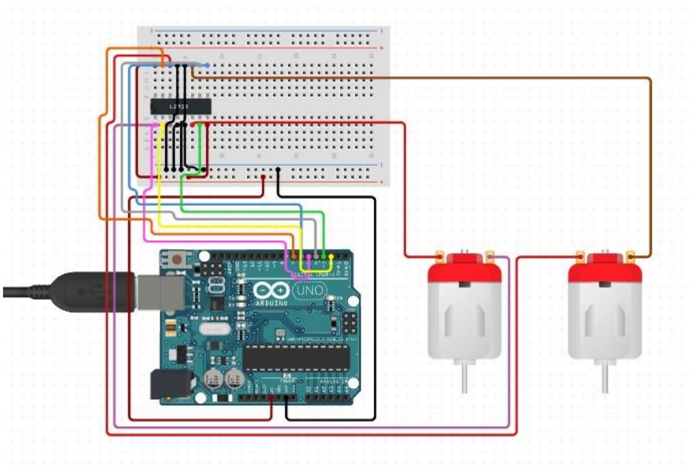


Fig.3.6. Interfacing between Arduino and L293D

3.2.5 Interfacing between Components

we used 8-13 numbered digital pins of Arduino to connect to LCD to display acknowledgment messages, 4-7 numbered digital pins act as output pins from Arduino are connected to the motor driver module i.e L293D as input and output of the module is connected to motors for forward or backward movements, A4 and A5 Analog pins of Arduino act as input and is connected to two Sensors. A one is temperature sensor And Gas Sensor.And it sense the temperature And the GSM is used to send the message to the railway Authorities.

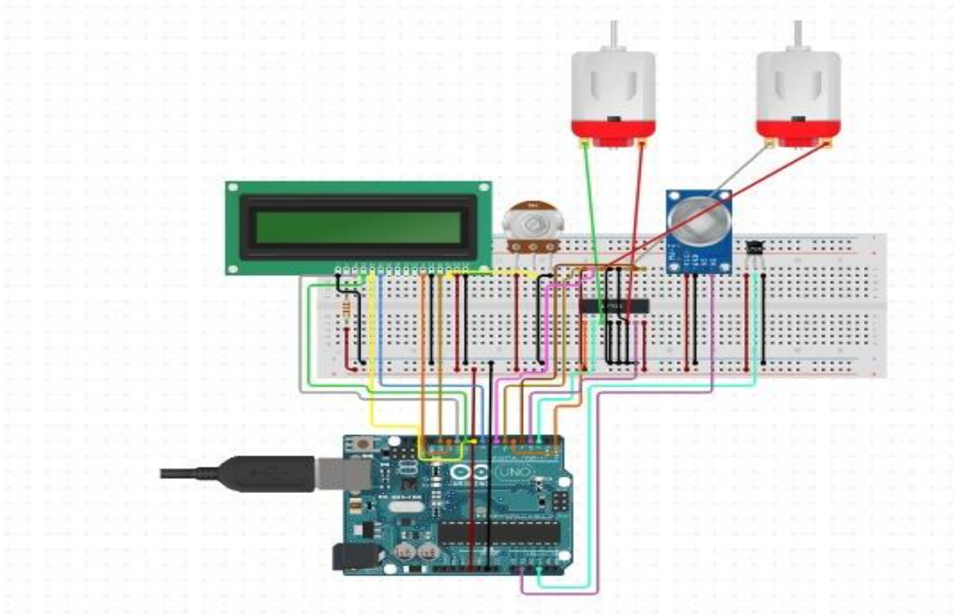


Fig: 3.7 Schematic Diagram of All Components

3.2.6 WORKING:

The whole circuit is designed and implemented on the PCB (Printed Circuit Board). The connections are made as shown in the figure. The 12v supply from the mainline is converted into 5v Dc using a 7805 voltage regulator and capacitors(1000uf and 10uf). There are 2 sensors one is temperature sensor and gas sensor. Thise two sensors are detect the temperature and gas. In this situation the motor turns on . this motor helps to pull the chain to stop the train. The sensed temperature is transmitted to the arduino and displayed on the LCD. The lcd is display the output. The GSM module can be used to send the message to railway authorities .which helP them to take quick decions to take control of fire and evaluting the train. This system works on the microcontrollers and sensors.

CHAPTER-4

HARDWARE AND SOFTWARE COMPONENTS

MICRO CONTROLLER UNIT

4.1 ATMEGA328

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 Analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The ATmega8 microcontroller contains 32 general purpose working registers. As shown in the below figure these registers are directly connected to ALU. Two registers can carry one single instruction consequently in one clock cycle.

Specifications:

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 Ma
DC Current on 3.3V Pin	50 Ma
Flash Memory	32 KB (0.5 KB is used for Boot loader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

Table 4.1: Atmega328 specifications

4.1.1 Arduino:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

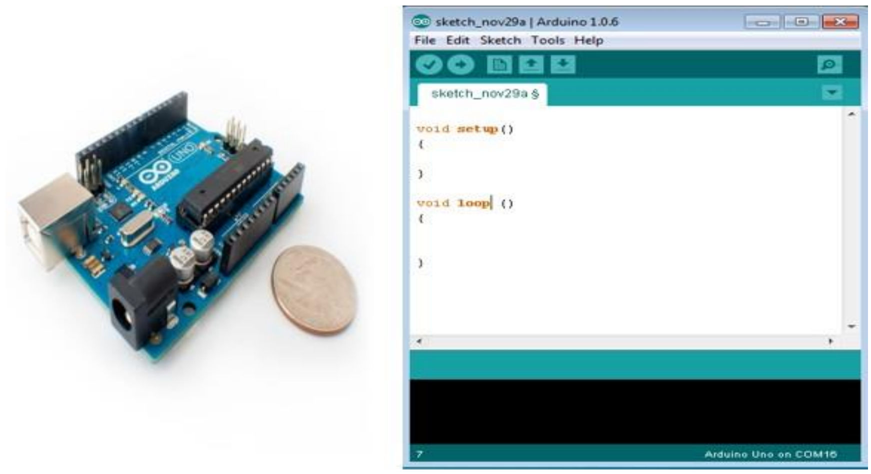


Fig 4.1 Arduino Uno

4.1.2 Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

ATMEGA328:

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16 U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16 U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header

Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compati

LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 4.2 Arduino boards based on ATMEGA328 microcontroller

ATMEGA32u4

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 4.3 Arduino boards based on ATMEGA32u4 microcontroller

ATMEGA2560

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16 U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 4.4 Arduino boards based on ATMEGA2560 microcontroller

AT91SAM3X8E

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table 4.5 Arduino boards based on AT91SAM3X8E microcontroller

4.1.3 Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

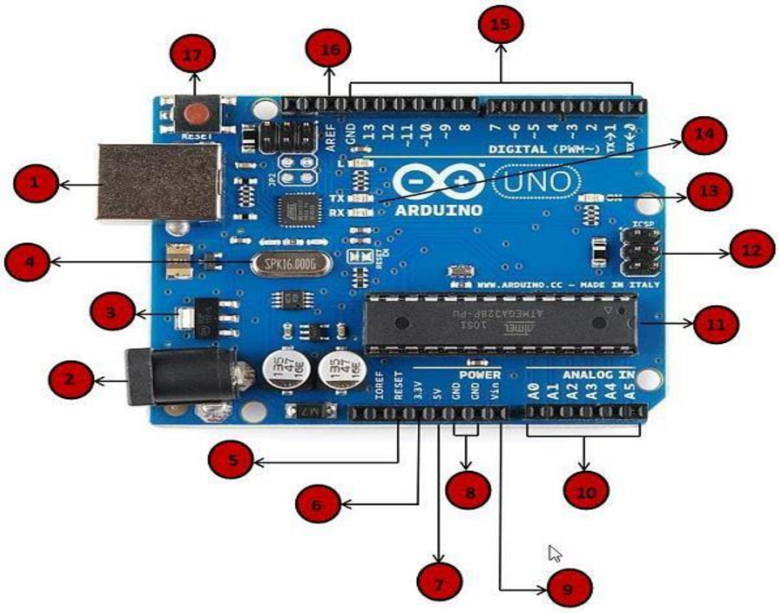


Fig 4.2 Arduino Uno board pins

Pin number and Pin Use

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.

12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Table 4.6 Pin Explanation of Arduino

4.1.4 Arduino Family

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

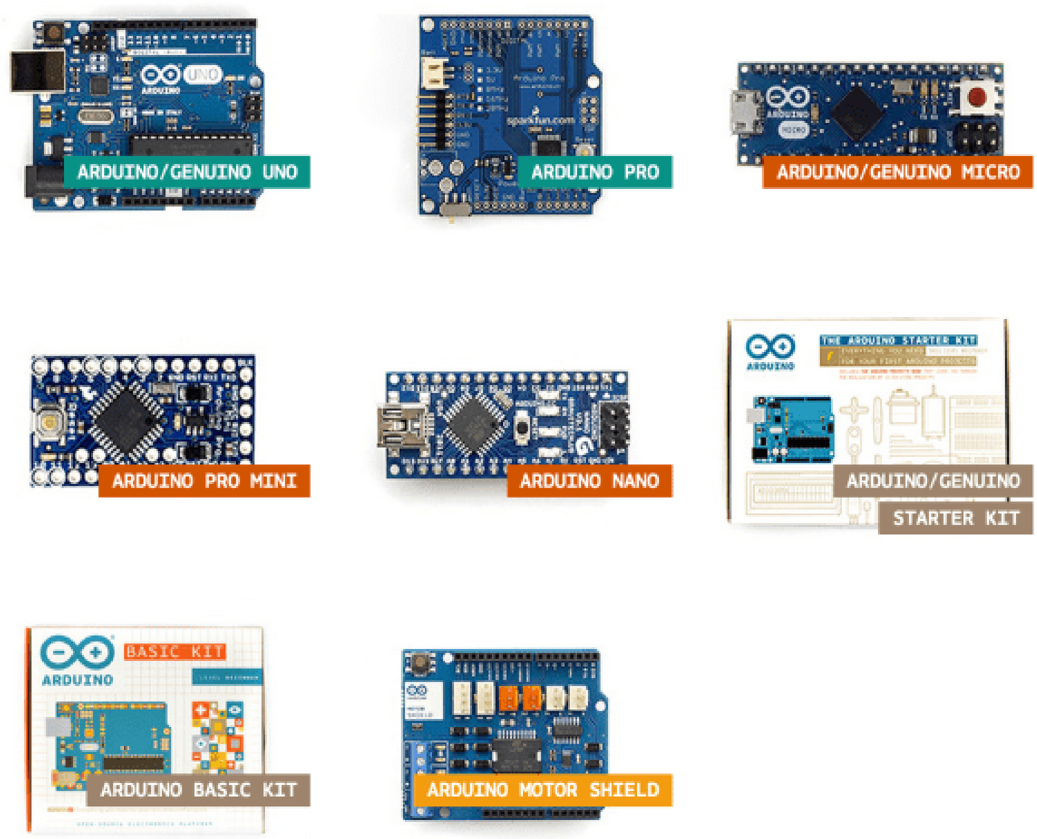


Fig.4.3 Arduino Family

4.1.5 Shields

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

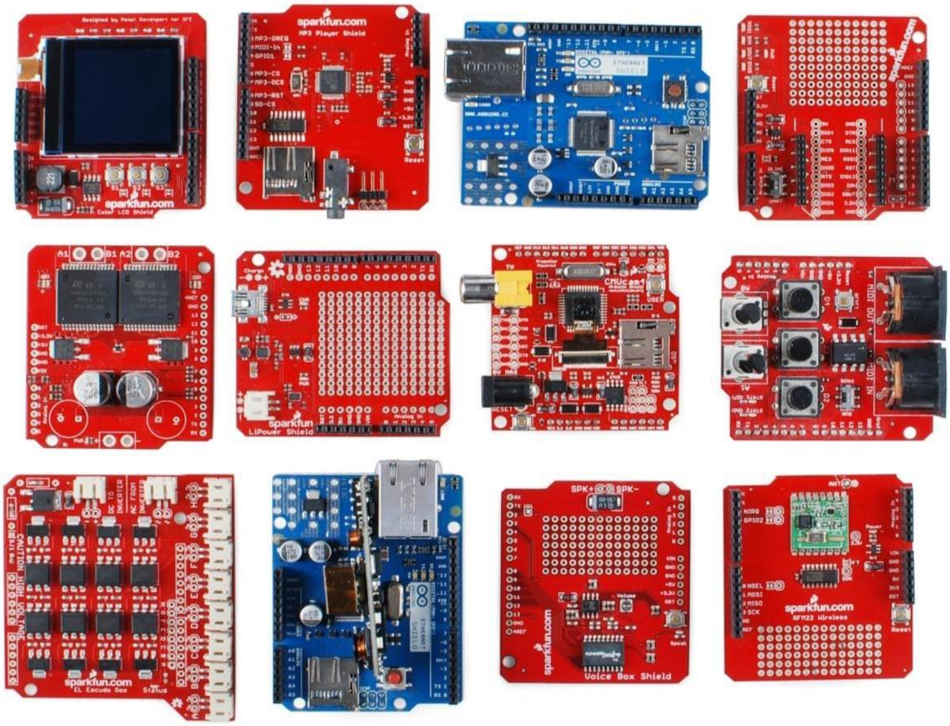


Fig.4.4 Arduino Shields

4.1.6 Pin Description Of Atmega328

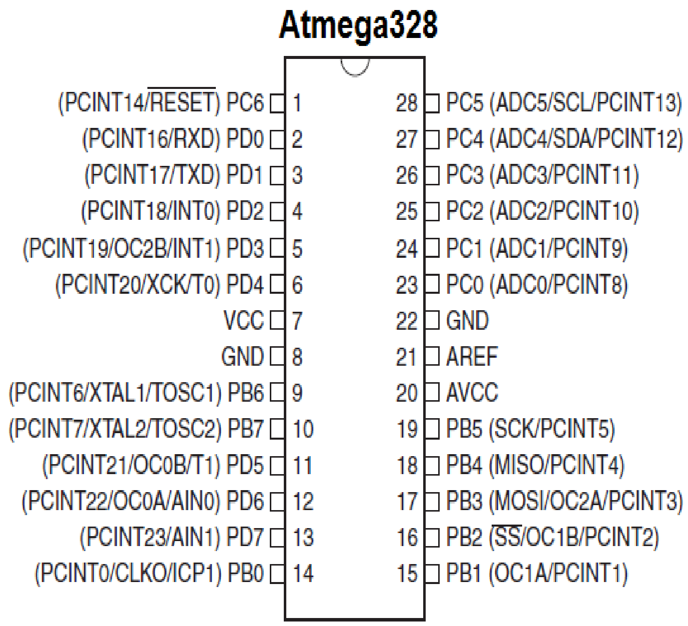


Fig.4.5 Pin description of ATMEGA328

4.1.7 Advantages Of Arduino

- It is cheap
- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
- For beginners, it is very simple to use.

4.1.8 Applications

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

4.2 POWER SUPPLY UNIT

4.2.1 Introduction:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices.

A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as “Regulated D.C Power Supply”.

- **Block Diagram Of Power Supply:**

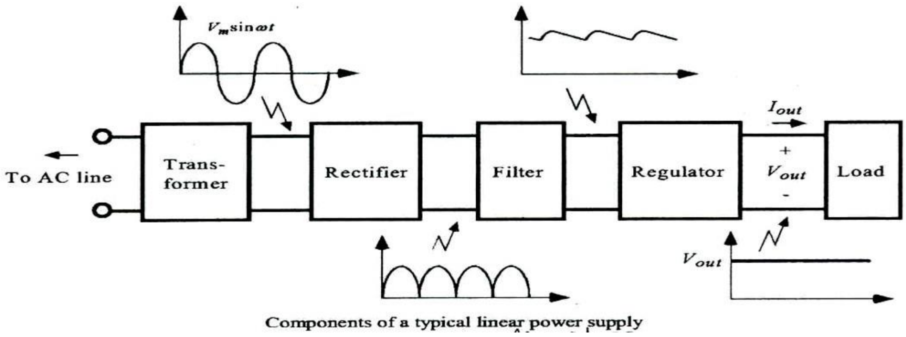


Fig.4.6 (a) Block Diagram of Power Supply

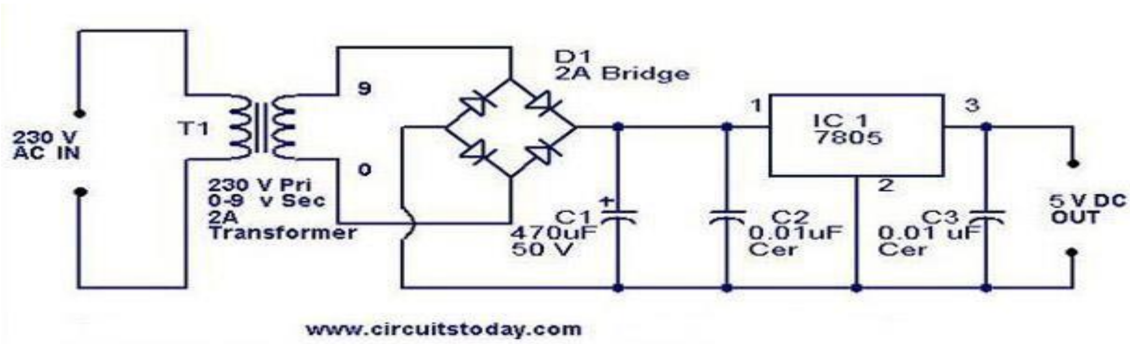


Fig.4.6 (b) Schematic Diagram of Power Supply

- **Description Of Power Supply:**

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a a different form of energy – such as solar, mechanical, or chemical - into electrical energy.

A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered.

For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

4.2.2 Transformer:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

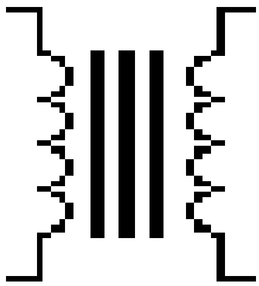


Fig.4.7 (a) Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

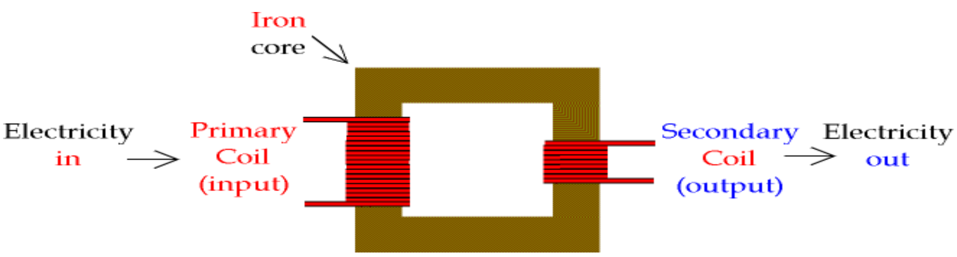


Fig.4.7 (b)Transformer

- **Basic Principle Of Transformer:**

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

From Faraday's Law

$$\frac{V_S}{V_P} = \frac{N_S}{N_P}$$

For ideal transformer

The voltage ratio is equal to the turns ratio, and power in equals power out.

From conservation of energy

$$P_P = V_P I_P = V_S I_S = P_S$$

- **Working Of Transformer:**

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below.

There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

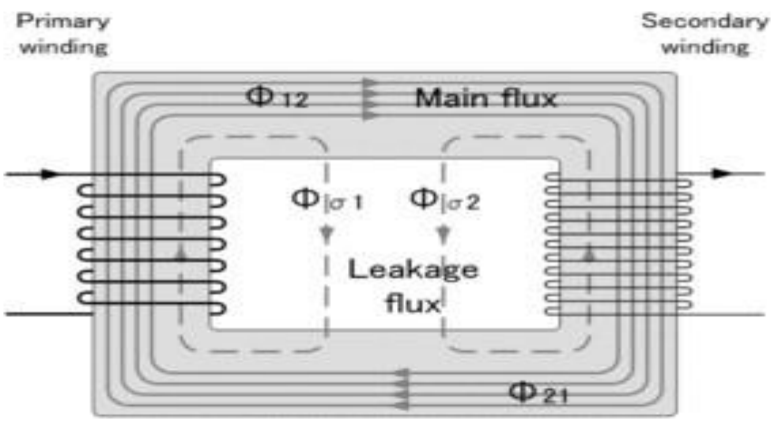


Fig.4.8 Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced EMF. The iron core is laminated to prevent 'eddy currents' flowing in the core.

These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

- ## Classification Of Transformer:

- Step-Up Transformer
- Step-Down Transformer

Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage

applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn’s ratio of the two sets of windings determines the amount of voltage transformation.

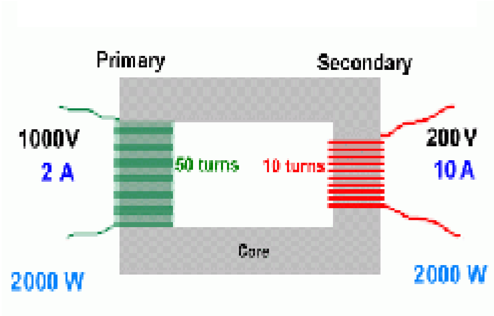


Fig.4.9 (a) Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.**Step down transformers can be considered nothing more than a voltage ratio device.**

With step down transformers the voltage ratio between primary and secondary will mirror the "turn’s ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn’s ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated.

Because of this it is possible to do some non-standard applications using standard transformers.Single phase steps down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase steps up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.

- **Step-Up Transformer:**

A step up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply. A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

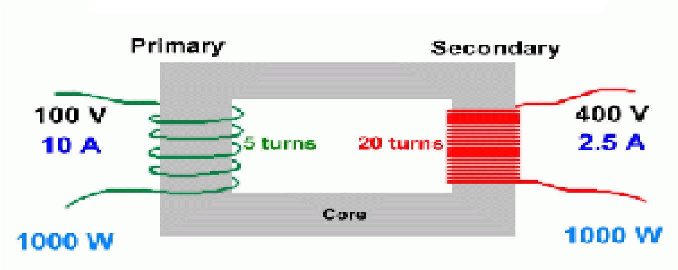


Fig.4.9 (b) Step-Up Transformer

- **Diodes:**

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

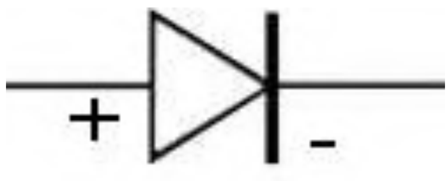


Fig.4.10 Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

- **Rectifier:**

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

  - **The Half-Wave Rectifier:**

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure.

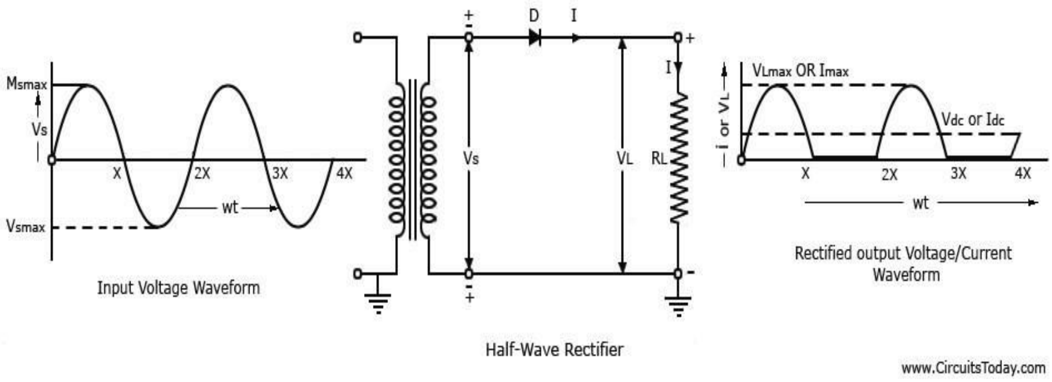


Fig.4.11 (a) Half Wave Rectifier

Figure 2 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

- ## The Full-Wave Rectifier:

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC.

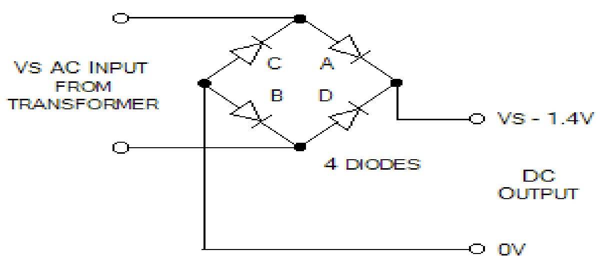


Fig.4.11 (b) Full-Wave Rectifier

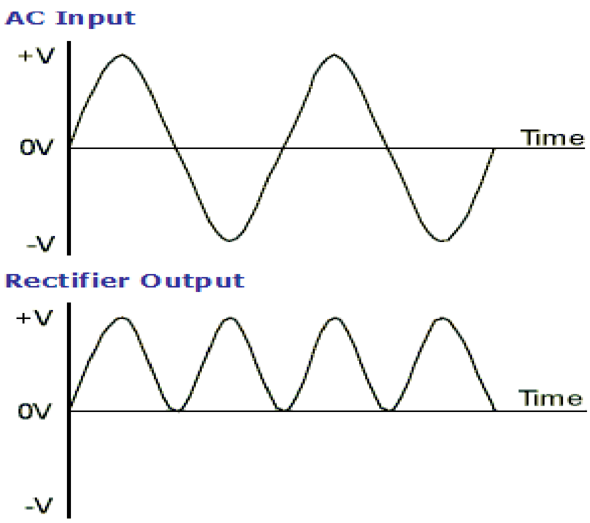


Fig.4.11 (c) Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased. While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

- **Capacitor Filter:**

The capacitor-input filter, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

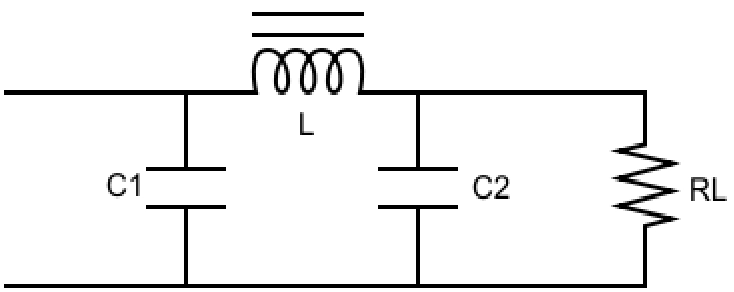


Fig.4.12 Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result, the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result, only the DC component appears across the load RL.

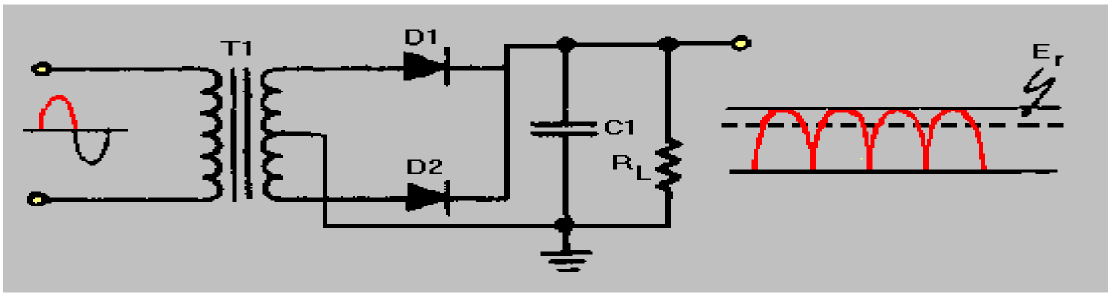


Fig.4.13 Centered Tapped Full-Wave Rectifier with a Capacitor Filter

- **Voltage Regulator:**

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages.

There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx' indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx' indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consist the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

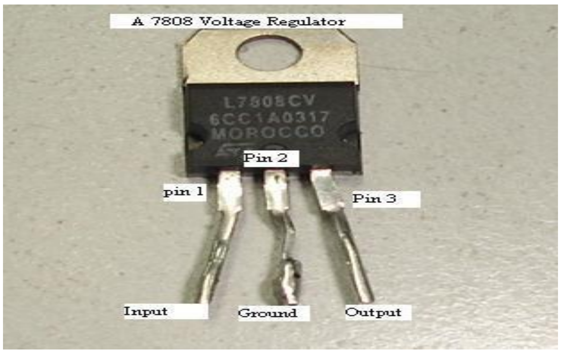


Fig.4.14 Regulator

4.3 TEMPERATURE SENSOR:

Temperature Sensor which converts temperature value into electrical signals. We used IC called LM 35 as a temperature sensor. LM35 series sensors are precision integrated-circuit temperature sensors whose output voltage is linearly proportional to the Celsius temperature. The LM35 requires no external calibration since it is internally calibrated. . The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°C over a full −55 to +150°C temperature range.

The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 μA from its supply, it has very low self-heating, less than 0.1°C in still air.

4.31 Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteed (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 μA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±1⁄4°C typical
- Low impedance output, 0.1 W for 1 mA load

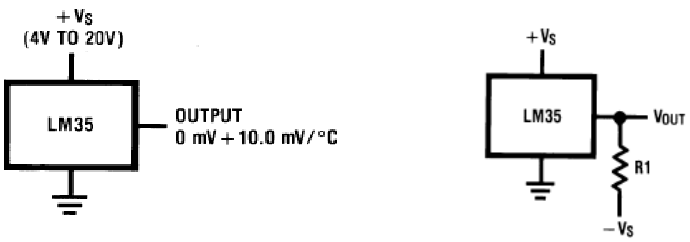


Fig.4.15. Temperature Sensor

The characteristic of this LM35 sensor is:

For each degree of centigrade temperature it outputs 10milli volts.ADC accepts the output from LM35 and converts that data into digital form which is sent to microcontroller for further processing.

Temperature sensing circuit:

The methods of temperature measurement may be divided into two main classes according as the exchange of heat between the testing body and the hot system takes place by contact, or by radiation across a space. In the contact methods, thermometers or thermocouples are used and they are immersed in solids or liquids. The thermodynamic equilibrium between the hot body and the testing body is established by material contact.

In the non-contact methods, the thermodynamic equilibrium is established by the radiation emitted as excited atom and molecules in the hot body return to the ground state.

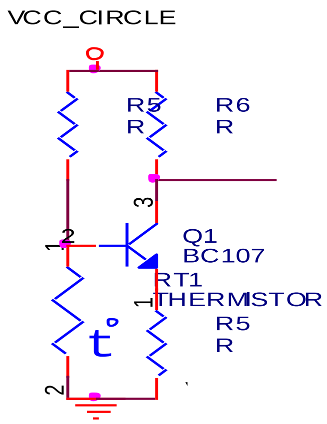


Fig.4.16. Temperature Sensing Circuit

4.4 16 * 2 ALPHANUMERIC LCD

4.4.1 Description

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

4.4.2 Block diagram

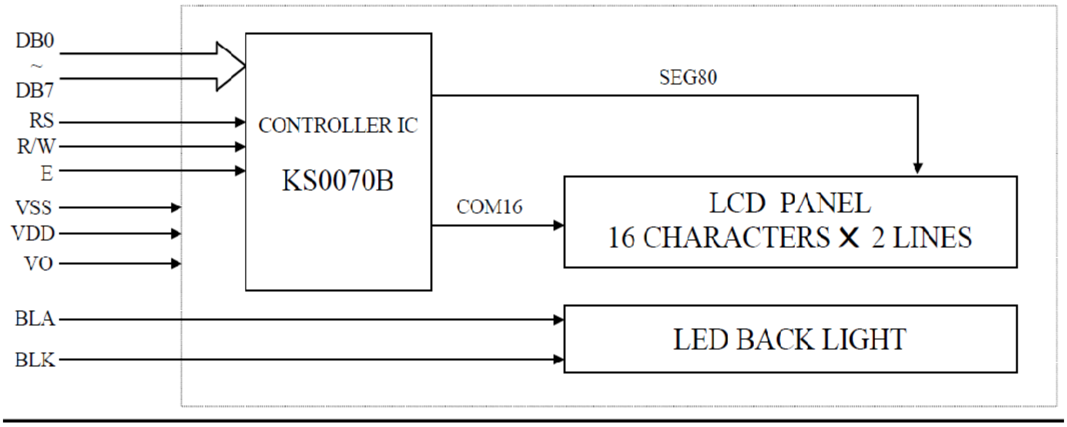


Fig.4.17 Block diagram of LCD display

- **Diplay Data Ram**

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD.

For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.

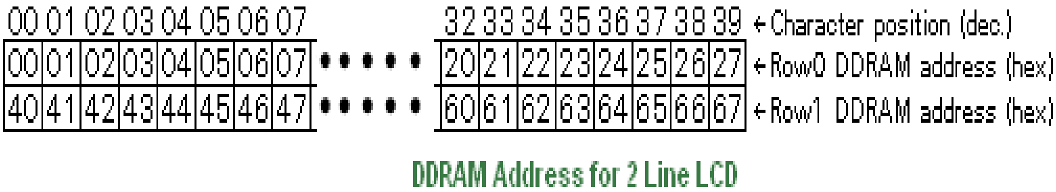


Fig.4.18 DDRAM Address of 2 Line LCD

- **Character Generator Rom**

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM.

- **Busy Flag**

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

- **Instruction Register (IR) And Data Register (DR)**

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD.

When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD

Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

4.4.3 16 x 2 Alphanumeric LCD Module

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

- **Schematic**

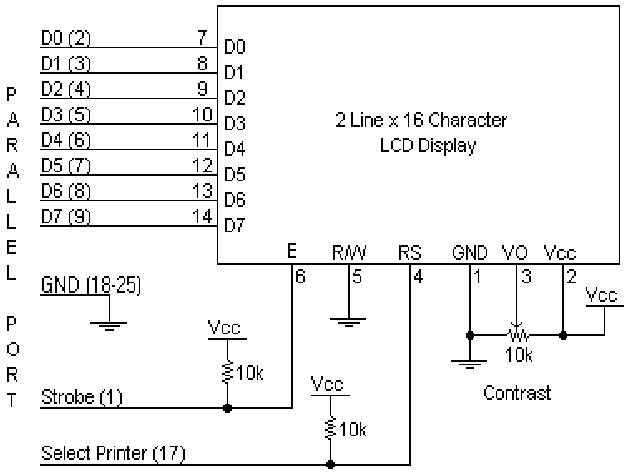


Fig.4.19 Schematic of 2 x16 Character LCD Display

- ## Specifications

Connector Pin Assignment:

Pin	*Symbol*	*Function*	*Pin*	*Symbol*	*Function*
14	DB7	Data Bus Line	6	E	Enable
13	DB6	Data Bus Line	5	R/W	Read/Write
12	DB5	Data Bus Line	4	RS	Register select input
11	DB4	Data Bus Line	3	Vcontrast	Contrast (0V for max contrast)
10	DB3	Data Bus Line	2	**GND (0V)**	Power supply
9	DB2	Data Bus Line	1	**Vcc +5V**	Power supply
8	DB1	Data Bus Line			
7	DB0	Data Bus Line			

Note1: Pin 1 is +5V and pin 2 is GND! This is different from most other HD44780 compatible models.

Note2: Pin 14 is marked with a "1" on the back of the PCB. Do not get confused by this.

Table.4.19 Connector pin Assignment

- ## Circuit Description

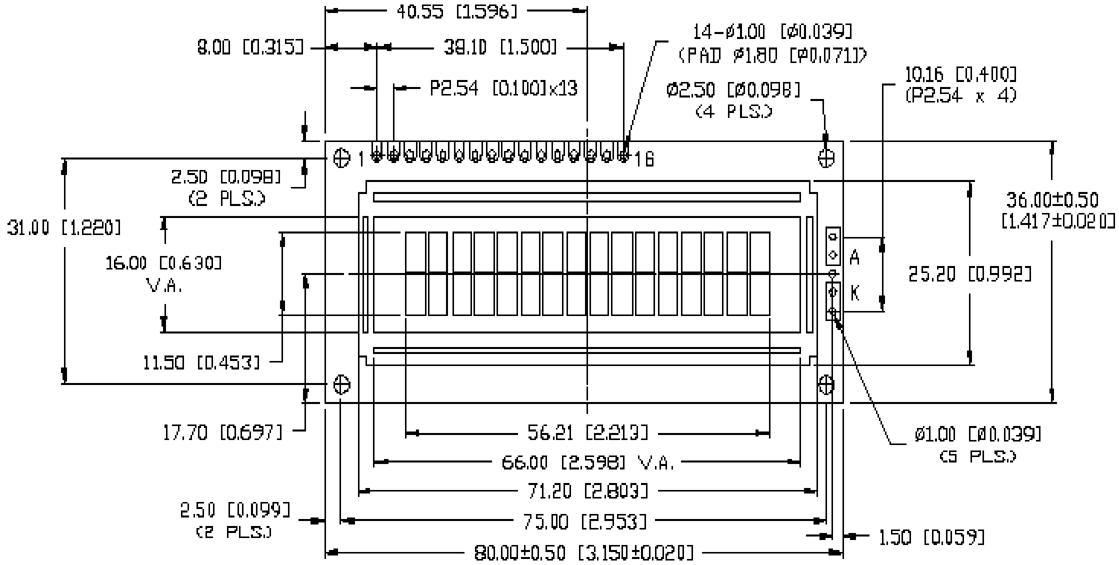


Fig.4.20 Internal Voltage and Current Distribution

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output.

While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly. The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

4.5 GSM

Definition of GSM:

GSM (Global System for Mobile communications) is an open, digital cellular technology used for transmitting mobile voice and data services.

GSM (Global System for Mobile communication) is a digital mobile telephone system that is widely used in Europe and other parts of the world. GSM uses a variation of Time Division Multiple Access (TDMA) and is the most widely used of the three digital wireless telephone technologies (TDMA, GSM, and CDMA). GSM digitizes and compresses data, then sends it down a channel with two other streams of user data, each in its own time slot. It operates at either the 900 MHz or 1,800 MHz frequency band. It supports voice calls and data transfer speeds of up to 9.6 kbit/s, together with the transmission of SMS (Short Message Service).

4.5.1 History

In 1982, the European Conference of Postal and Telecommunications Administrations (CEPT) created the Group Special Mobile (GSM) to develop a standard for a mobile telephone system that could be used across Europe. In 1987, a memorandum of understanding was signed by 13 countries to develop a common cellular telephone system across Europe. Finally the system created by SINTEF lead by Torleiv Maseng was selected.

In 1989, GSM responsibility was transferred to the European Telecommunications Standards Institute (ETSI) and phase I of the GSM specifications were published in 1990. The first GSM network was launched in 1991 by Radiolinja in Finland with joint technical infrastructure maintenance from Ericsson.

By the end of 1993, over a million subscribers were using GSM phone networks being operated by 70 carriers across 48 countries. As of the end of 1997, GSM service was available in more than 100 countries and has become the *de facto* standard in Europe and Asia.

GSM Frequencies

GSM networks operate in a number of different frequency ranges (separated into GSM frequency ranges for 2G and UMTS frequency bands for 3G). Most 2G GSM networks operate in the 900 MHz or 1800 MHz bands. Some countries in the Americas (including Canada and the United States) use the 850 MHz and 1900 MHz bands because the 900 and 1800 MHz frequency bands were already allocated. Most 3G GSM networks in Europe operate in the 2100 MHz frequency band. The rarer 400 and 450 MHz frequency bands are assigned in some countries where these frequencies were previously used for first-generation systems.

GSM-900 uses 890–915 MHz to send information from the mobile station to the base station (uplink) and 935–960 MHz for the other direction (downlink), providing 124 RF channels (channel numbers 1 to 124) spaced at 200 kHz. Duplex spacing of 45 MHz is used. In some countries the GSM-900 band has been extended to cover a larger frequency range. This 'extended GSM', E-GSM, uses 880–915 MHz (uplink) and 925–960 MHz (downlink), adding 50 channels (channel numbers 975 to 1023 and 0) to the original GSM-900 band.

Time division multiplexing is used to allow eight full-rate or sixteen half-rate speech channels per radio frequency channel. There are eight radio timeslots (giving eight burst periods) grouped into what is called a TDMA frame. Half rate channels use alternate frames in the same timeslot. The channel data rate for all 8 channels is 270.833 Kbit/s, and the frame duration is 4.615 ms.

The transmission power in the handset is limited to a maximum of 2 watts in GSM850/900 and 1 watt in GSM1800/1900. GSM operates in the 900MHz and 1.8GHz bands in Europe and the 1.9GHz and 850MHz bands in the US. The 850MHz band is also used for GSM and 3G in Australia, Canada and many South American countries. By having harmonized spectrum across most of the globe, GSM's international roaming capability allows users to access the same services when travelling abroad as at home. This gives consumers seamless and same number connectivity in more than 218 countries.

Terrestrial GSM networks now cover more than 80% of the world's population. GSM satellite roaming has also extended service access to areas where terrestrial coverage is not available.

Mobile Telephony Standards

Standard	Generation	Frequency band	Throughput	
GSM	2G	Allows transfer of voice or low-volume digital data.	9.6 kbps	9.6 kbps
GPRS	2.5G	Allows transfer of voice or moderate-volume digital data.	21.4-171.2 kbps	48 kbps
EDGE	2.75G	Allows simultaneous transfer of voice and digital data.	43.2-345.6 kbps	171 kbps
UMTS	3G	Allows simultaneous transfer of voice and high-speed digital data.	0.144-2 Mbps	384 kbps

1G

The first generation of mobile telephony (written **1G**) operated using analogue communications and portable devices that were relatively large. It used primarily the following standards:

- **AMPS** (Advanced Mobile Phone System), which appeared in 1976 in the United States, was the first cellular network standard. It was used primarily in the Americas, Russia and Asia. This first-generation analogue network had weak security mechanisms which allowed hacking of telephones lines.
- **TACS** (Total Access Communication System) is the European version of the AMPS model. Using the 900 MHz frequency band, this system was largely used in England and then in Asia (Hong-Kong and Japan).
- **ETACS** (Extended Total Access Communication System) is an improved version of the TACS standard developed in the United Kingdom that uses a larger number of communication channels.

The first-generation cellular networks were made obsolete by the appearance of an entirely digital second generation.

Second Generation of Mobile Networks (2G)

The second generation of mobile networks marked a break with the first generation of cellular telephones by switching from analogue to digital. The main 2G mobile telephony standards are:

- GSM (*Global System for Mobile communications*) is the most commonly used standard in Europe at the end of the 20th century and supported in the United States. This standard uses the 900 MHz and 1800 MHz frequency bands in Europe. In the United States, however, the frequency band used is the 1900 MHz band. Portable telephones that are able to operate in Europe and the United States are therefore called **tri-band**.

- **CDMA** (*Code Division Multiple Access*) uses a spread spectrum technique that allows a radio signal to be broadcast over a large frequency range.
- **TDMA** (*Time Division Multiple Access*) uses a technique of time division of communication channels to increase the volume of data transmitted simultaneously. TDMA technology is primarily used on the American continent, in New Zealand and in the Asia-Pacific region.

With the 2G networks, it is possible to transmit voice and low volume digital data, for example text messages (**SMS**, for *Short Message Service*) or multimedia messages (**MMS**, for *Multimedia Message Service*). The GSM standard allows a maximum data rate of 9.6 kbps.

Extensions have been made to the GSM standard to improve throughput. One of these is the **GPRS** (*General Packet Radio System*) service which allows theoretical data rates on the order of 114 Kbit/s but with throughput closer to 40 Kbit/s in practice. As this technology does not fit within the "3G" category, it is often referred to as **2.5G**

The **EDGE** (*Enhanced Data Rates for Global Evolution*) standard, billed as **2.75G**, quadruples the throughput improvements of GPRS with its theoretical data rate of 384 Kbps, thereby allowing the access for multimedia applications. In reality, the EDGE standard allows maximum theoretical data rates of 473 Kbit/s, but it has been limited in order to comply with the IMT-2000 (*International Mobile Telecommunications-2000*) specifications from the ITU (*International Telecommunications Union*).

3G

The IMT-2000 (*International Mobile Telecommunications for the year 2000*) specifications from the International Telecommunications Union (ITU) defined the characteristics of **3G** (third generation of mobile telephony). The most important of these characteristics are:

1. High transmission data rate.
2. 144 Kbps with total coverage for mobile use.
3. 384 Kbps with medium coverage for pedestrian use.
4. 2 Mbps with reduced coverage area for stationary use.
5. World compatibility.
6. Compatibility of 3rd generation mobile services with second generation networks.

3G offers data rates of more than 144 Kbit/s, thereby allowing the access to multimedia uses such as video transmission, video-conferencing or high-speed internet access. 3G networks use different frequency bands than the previous networks: 1885-2025 MHz and 2110-2200 MHz.

The main 3G standard used in Europe is called **UMTS** (*Universal Mobile Telecommunications System*) and uses **WCDMA** (*Wideband Code Division Multiple Access*) encoding. UMTS technology uses 5 MHz bands for transferring voice and data, with data rates that can range from 384 Kbps to 2 Mbps. **HSDPA** (*High Speed Downlink Packet Access*) is a third generation mobile telephony protocol, (considered as "3.5G"), which is able to reach data rates on the order of 8 to 10 Mbps. HSDPA technology uses the 5 GHz frequency band and uses WCDMA encoding.

Introduction to the GSM Standard

The **GSM** (*Global System for Mobile communications*) network is at the start of the 21st century, the most commonly used mobile telephony standard in Europe. It is called as Second Generation (2G) standard because communications occur in an entirely digital mode, unlike the first generation of portable telephones. When it was first standardized in 1982, it was called as **Group Special Mobile** and later, it became an international standard called **"Global System for Mobile communications"** in 1991.

In Europe, the GSM standard uses the 900 MHz and 1800 MHz frequency bands. In the United States, however, the frequency band used is the 1900 MHz band. For this reason, portable telephones that are able to operate in both Europe and the United States are called **tri-band** while those that operate only in Europe are called **bi-band**.

The GSM standard allows a maximum throughput of 9.6 kbps which allows transmission of voice and low-volume digital data like text messages (**SMS**, for *Short Message Service*) or multimedia messages (**MMS**, for *Multimedia Message Service*).

GSM Standards:

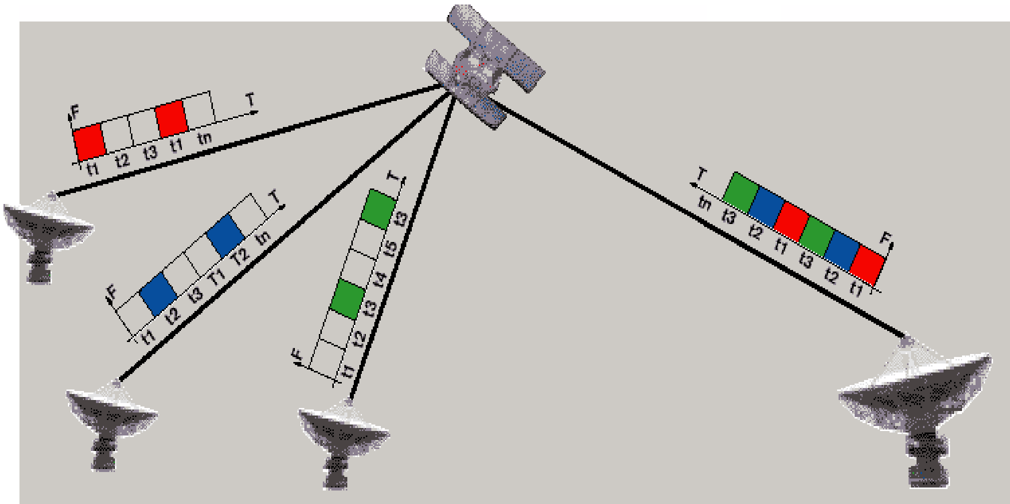
GSM uses narrowband TDMA, which allows eight simultaneous calls on the same radio frequency. There are three basic principles in multiple access, FDMA (Frequency Division Multiple Access), TDMA (Time Division Multiple Access), and CDMA (Code Division Multiple Access). All three principles allow multiple users to share the same physical channel. But the two competing technologies differ in the way user sharing the common resource.

TDMA allows the users to share the same frequency channel by dividing the signal into different time slots. Each user takes turn in a round robin fashion for transmitting and receiving over the channel. Here, users can only transmit in their respective time slot.

CDMA uses a spread spectrum technology that is it spreads the information contained in a particular signal of interest over a much greater bandwidth than the original signal. Unlike TDMA, in CDMA several users can transmit over the channel at the same time.

TDMA in brief:

In late1980’s, as a search to convert the existing analog network to digital as a means to improve capacity, the cellular telecommunications industry association chose TDMA over FDMA. Time Division Multiplex Access is a type of multiplexing where two or more channels of information are transmitted over the same link by allocating a different time interval for the transmission of each channel. The most complex implementation using TDMA principle is of GSM’s (Global System for Mobile communication). To reduce the effect of co-channel interference, fading and multipath, the GSM technology can use frequency hopping, where a call jumps from one channel to another channel in a short interval.



Time Division Multiple Access

TDMA systems still rely on switch to determine when to perform a handoff. Handoff occurs when a call is switched from one cell site to another while travelling. The TDMA handset constantly monitors the signals coming from other sites and reports it to the switch without caller’s awareness. The switch then uses this information for making better choices for handoff at appropriate times. TDMA handset performs hard handoff, i.e., whenever the user moves from one site to another, it breaks the connection and then provides a new connection with the new site.

Advantages of TDMA:

There are lots of advantages of TDMA in cellular technologies.

1. It can easily adapt to transmission of data as well as voice communication.
2. It has an ability to carry 64 kbps to 120 Mbps of data rates. This allows the operator to do services like fax, voice band data and SMS as well as bandwidth intensive application such as multimedia and video conferencing.
3. Since TDMA technology separates users according to time, it ensures that there will be no interference from simultaneous transmissions.
4. It provides users with an extended battery life, since it transmits only portion of the time during conversations. Since the cell size grows smaller, it proves to save base station equipment, space and maintenance.

TDMA is the most cost effective technology to convert an analog system to digital.

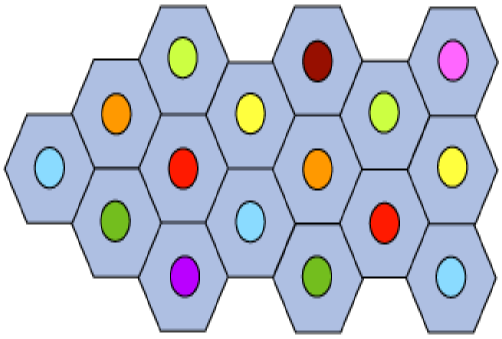
Disadvantages of TDMA:

One major disadvantage using TDMA technology is that the users has a predefined time slot. When moving from one cell site to other, if all the time slots in this cell are full the user might be disconnected. Likewise, if all the time slots in the cell in which the user is currently in are already occupied, the user will not receive a dial tone.

The second problem in TDMA is that it is subjected to multipath distortion. To overcome this distortion, a time limit can be used on the system. Once the time limit is expired, the signal is ignored.

The concept of cellular network

Mobile telephone networks are based on the concept of **cells**, circular zones that overlap to cover a geographical area.



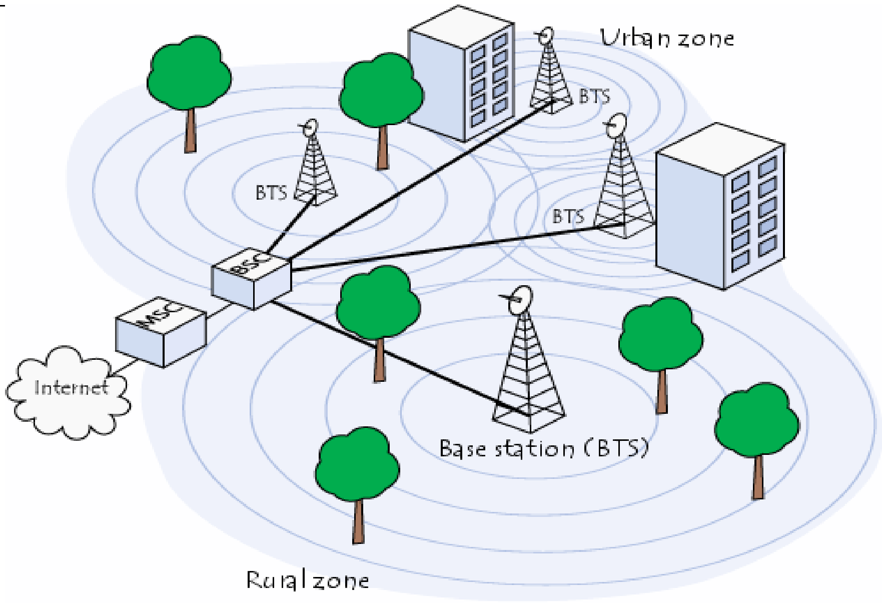
Cellular networks are based on the use of a central transmitter-receiver in each cell, called a "**base station**" (or *Base Transceiver Station*, written **BTS**). The smaller the radius of a cell, the higher is the available bandwidth. So, in highly populated urban areas, there are cells with a radius of a few hundred meters, while huge cells of up to 30 kilometers provide coverage in rural areas.

In a cellular network, each cell is surrounded by 6 neighbouring cells (thus a cell is generally drawn as a hexagon). To avoid interference, adjacent cells cannot use the same frequency. In practice, two cells using the same frequency range must be separated by a distance of two to three times the diameter of the cell.

Architecture of the GSM Network

In a GSM network, the user terminal is called a **mobile station**. A mobile station is made up of a **SIM** (*Subscriber Identity Module*) card allowing the user to be uniquely identified and a mobile terminal. The terminals (devices) are identified by a unique 15-digit identification number called **IMEI** (*International Mobile Equipment Identity*). Each SIM card also has a unique (and secret) identification number called **IMSI** (*International Mobile Subscriber Identity*). This code can be protected using a 4-digit key called a *PIN code*.

The SIM card therefore allows each user to be identified independently of the terminal used during communication with a base station. Communications occur through a radio link (air interface) between a mobile station and a base station.



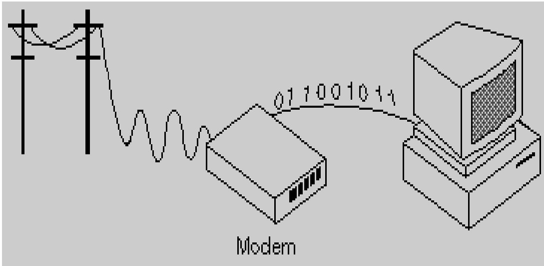
All the base stations of a cellular network are connected to a **base station controller** (**BSC**) which is responsible for managing distribution of the resources. The system consisting of the base station controller and its connected base stations is called the **Base Station Subsystem** (**BSS**).

Finally, the base station controllers are themselves physically connected to the **Mobile Switching Centre** (**MSC**), managed by the telephone network operator, which connects them to the public telephone network and the Internet. The MSC belongs to a **Network Station Subsystem** (**NSS**), which is responsible for managing user identities, their location and establishment of communications with other subscribers. The MSC is generally connected to databases that provide additional functions:

1. The **Home Location Register** (**HLR**) is a database containing information (geographic position, administrative information etc.) of the subscribers registered in the area of the switch (MSC).
2. The **Visitor Location Register** (**VLR**) is a database containing information of users other than the local subscribers. The VLR retrieves the data of a new user from the HLR of the user's subscriber zone. The data is maintained as long as the user is in the zone and is deleted when the user leaves or after a long period of inactivity (terminal off).
3. The **Equipment Identify Register** (**EIR**) is a database listing the mobile terminals.
4. The **Authentication Centre** (**AUC**) is responsible for verifying user identities.
5. The cellular network formed in this way is designed to support mobility via management of *handovers* (movements from one cell to another).

Finally, GSM networks support the concept of **roaming** i.e., movement from one operator network to another.

Introduction to Modem:



Modem stands for *modulator-demodulator.*

A modem is a device or program that enables a computer to transmit data over telephone or cable lines. Computer information is stored digitally, whereas information transmitted over telephone lines is transmitted in the form of analog waves. A modem converts between these two forms.

Fortunately, there is one standard interface for connecting external modems to computers called *RS-232*. Consequently, any external modem can be attached to any computer that has an RS-232 port, which almost all personal computers have. There are also modems that come as an expansion board that can be inserted into a vacant expansion slot. These are sometimes called *onboard* or *internal modems*.

While the modem interfaces are standardized, a number of different protocols for formatting data to be transmitted over telephone lines exist. Some, like CCITT V.34 are official standards, while others have been developed by private companies. Most modems have built-in support for the more common protocols at slow data transmission speeds at least, most modems can communicate with each other. At high transmission speeds, however, the protocols are less standardized.

Apart from the transmission protocols that they support, the following characteristics distinguish one modem from another:

- **Bps:** How fast the modem can transmit and receive data. At slow rates, modems are measured in terms of baud rates. The slowest rate is 300 baud (about 25 cps). At higher speeds, modems are measured in terms of bits per second (bps). The fastest modems run at 57,600 bps, although they can achieve even higher data transfer rates by compressing the data. Obviously, the faster the transmission rate, the faster the data can be sent and received. It should be noted that the data cannot be received at a faster rate than it is being sent.
- **Voice/data:** Many modems support a switch to change between voice and data modes. In data mode, the modem acts like a regular modem. In voice mode, the modem acts like a regular telephone. Modems that support a voice/data switch have a built-in loudspeaker and microphone for voice communication.
- **Auto-answer:** An auto-answer modem enables the computer to receive calls in the absence of the operator.

- **Data compression:** Some modems perform data compression, which enables them to send data at faster rates. However, the modem at the receiving end must be able to decompress the data using the same compression technique.
- **Flash memory:** Some modems come with *flash memory* rather than conventional ROM which means that the communications protocols can be easily updated if necessary.
- **Fax capability:** Most modern modems are fax modems, which mean that they can send and receive faxes.

GSM Modem:

A GSM modem is a wireless modem that works with a GSM wireless network. A wireless modem behaves like a dial-up modem. The main difference between them is that a dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves.

A GSM modem can be an external device or a PC Card / PCMCIA Card. Typically, an external GSM modem is connected to a computer through a serial cable or a USB cable. A GSM modem in the form of a PC Card / PCMCIA Card is designed for use with a laptop computer. It should be inserted into one of the PC Card / PCMCIA Card slots of a laptop computer. Like

4.6 L293D

L293D is basically a high current dual motor driver/controller Integrated Circuit (IC). It is able to drive load having current up to 1A at the voltage ranging from 4.5V to 36V. Motor driver usually act as current amplifier because they receive a low current signal as an input and provides high current signal at the output.

Motors usually operates on this higher current. L-293D has to Built in H-Bridge driver circuits and is able to control two DC motors at a time in both clockwise and counter clockwise direction. It has two enable pins and they should be kept high in order to control the motor. By changing the polarity of applied signal motor can be rotated in either clockwise or counter clockwise direction. If L 293D enable pin is high, its corresponding driver will provide the desired out. If the enable pin is low, there will be no output. L-293D has different features including internal ESD protection, large voltage supply range, large output current per channel, high noise immunity input etc. L 293D plays a vital role in electronics era and has several different applications e.g relay drivers, DC motor drivers, stepping motor drivers etc. The further detail about L 293D motor driver/controller will be given later in this tutorial.

4.6.1 Introduction to L293D

L293D is basically a motor driver or controller. It has two built in H-bridge circuits which are able to control two DC motors simultaneously in both clockwise and counter clockwise direction. It acts as an current high amplifier because it take low current signal at its input and provides higher current signal at the output in order to drives different load e.g stepper motor & DC motors. Its features include large input voltage supply range, large output current high noise immunity input signals etc.

Its common real life applications include stepping motor drivers, relay drivers, DC motor drivers etc. L-293D motor driver/controller is shown in the figure given below.

Fig.4.20 L293D

1. L293D Pins

- L-293D has sixteen (16) pins, having different individual functions.
- All of these sixteen pins along with their serial no, are given in the table shown below

2. L293D Pin Functions

- L 293D each pin has different tasks to perform when they are in working condition.
- The tasks assigned to each pin are given in the table shown below.

L293D Pins	
Pin. No	Pin Name
1	Enable 1,2
2	Input 1
3	Output 1
4	GND
5	GND
6	Output 2
7	Input 2
8	V_{CC2}
9	Enable 3,4
10	Input 3
11	Output 3
12	GND
13	GND
14	Output put 4
15	Input 4
16	V_{CC1}

Table.4.21 L293D Pin

3. L293D Pinout

- Pinout diagram of any device presents the pins configuration through a completely labelled diagram.
- L293D pinout diagram is shown in the figure given below.

Fig.4.21 L293D Pin Configuration

4.7MOTORS

4.7.1Definition

Motor is a device that creates motion, not an engine; it usually refers to either an electrical motor or an internal combustion engine.

It may also refer to:

- Electric motor, a machine that converts electricity into a mechanical motion
  - AC motor, an electric motor that is driven by alternating current
    - Synchronous motor, an alternating current motor distinguished by a rotor spinning with coils passing magnets at the same rate as the alternating current and resulting magnetic field which drives it
    - Induction motor, also called a squirrel-cage motor, a type of asynchronous alternating current motor where power is supplied to the rotating device by means of electromagnetic induction
  - DC motor, an electric motor that runs on direct current electricity
    - Brushed DC electric motor, an internally commutated electric motor designed to be run from a direct current power source
    - Brushless DC motor, a synchronous electric motor which is powered by direct current electricity and has an electronically controlled commutation system, instead of a mechanical commutation system based on brushes
  - Electrostatic motor, a type of electric motor based on the attraction and repulsion of electric charge
  - Servo motor, an electric motor that operates a servo, commonly used in robotics

- Internal fan-cooled electric motor, an electric motor that is self-cooled by a fan, typically used for motors with a high energy density

4.7.2 Types of Motors

Industrial motors come in a variety of basic types. These variations are suitable for many different applications. Naturally, some types of motors are more suited for certain applications than other motor types are. This document will hopefully give some guidance in selecting these motors.

1. **AC Motors**

The most common and simple industrial motor is the three phase AC induction motor, sometimes known as the "squirrel cage" motor. Substantial information can be found about any motor by checking its (nameplate).

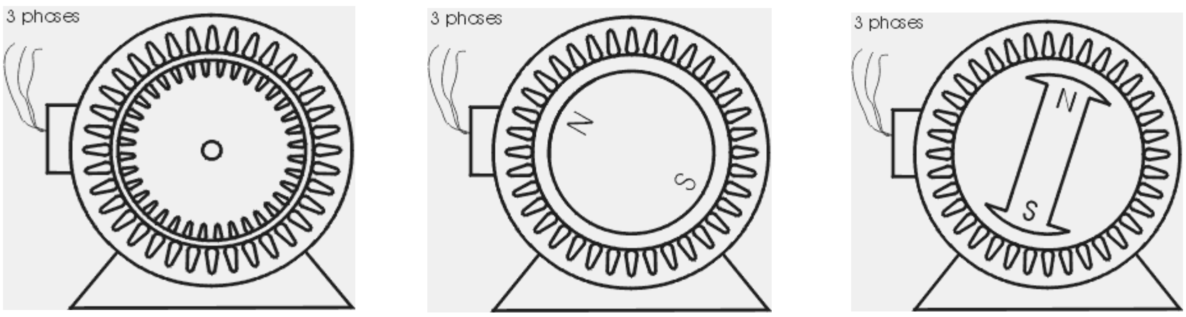


Fig.4.22. 3 Phase AC Induction Motor

Advantages

- Simple Design
- Low Cost
- Reliable Operation
- Easily Found Replacements
- Variety of Mounting Styles
- Many Different Environmental Enclosures

Simple Design

The simple design of the AC motor -- simply a series of three windings in the exterior (stator) section with a simple rotating section (rotor). The changing field caused by the 50 or 60 Hertz AC line voltage causes the rotor to rotate around the axis of the motor.

The speed of the AC motor depends only on three variables:

1. The fixed number of winding sets (known as poles) built into the motor, which determines the motor's base speed.
2. The frequency of the AC line voltage. Variable speed drives change this frequency to change the speed of the motor.
3. The amount of torque loading on the motor, which causes slip.

Low Cost

The AC motor has the advantage of being the lowest cost motor for applications requiring more than about 1/2 hp (325 watts) of power. This is due to the simple design of the motor. For this reason, AC motors are overwhelmingly preferred for fixed speed applications in industrial applications and for commercial and domestic applications where AC line power can be easily attached. Over 90% of all motors are AC induction motors. They are found in air conditioners, washers, dryers, industrial machinery, fans, blowers, vacuum cleaners, and many, many other applications.

- **Reliable Operation**

The simple design of the AC motor results in extremely reliable, low maintenance operation. Unlike the DC motor, there are no brushes to replace. If run in the appropriate environment for its enclosure, the AC motor can expect to need new bearings after several years of operation. If the application is well designed, an AC motor may not need new bearings for more than a decade.

- **Easily Found Replacements**

The wide use of the AC motor has resulted in easily found replacements. Many manufacturers adhere to either European (metric) or American (NEMA) standards. (For Replacement Motors) Variety of Mounting Styles

AC Motors are available in many different mounting styles such as:

- Foot Mount
- C-Face
- Large Flange
- Vertical
- Specialty

2. DC Motors

The brushed DC motor is one of the earliest motor designs. Today, it is the motor of choice in the majority of variable speed and torque control applications.

- **Advantages**

- Easy to understand design
- Easy to control speed
- Easy to control torque
- Simple, cheap drive design

- **Easy to understand design**

The design of the brushed DC motor is quite simple. A permanent magnetic field is created in the stator by either of two means:

- Permanent magnets
- Electro-magnetic windings

If the field is created by permanent magnets, the motor is said to be a "permanent magnet DC motor" (PMDC). If created by electromagnetic windings, the motor is often said to be a "shunt wound DC motor" (SWDC). Today, because of cost-effectiveness and reliability, the PMDC motor is the motor of choice for applications involving fractional horsepower DC motors, as well as most applications up to about three horsepower.

At five horsepower and greater, various forms of the shunt wound DC motor are most commonly used. This is because the electromagnetic windings are more cost effective than permanent magnets in this power range.

Caution: If a DC motor suffers a loss of field (if for example, the field power connections are broken), the DC motor will immediately begin to accelerate to the top speed which the loading will allow. This can result in the motor flying apart if the motor is lightly loaded. The possible loss of field must be accounted for, particularly with shunt wound DC motors.

Opposing the stator field is the armature field, which is generated by a changing electromagnetic flux coming from windings located on the rotor. The magnetic poles of the armature field will attempt to line up with the opposite magnetic poles generated by the stator field. If we stopped the design at this point, the motor would spin until the poles were opposite one another, settle into place, and then stop -- which would make a pretty useless motor!

However, we are smarter than that. The section of the rotor where the electricity enters the rotor windings is called the commutator. The electricity is carried between the rotor and the stator by conductive graphite-copper brushes (mounted on the rotor) which contact rings on stator. Imagine power is supplied:

The motor rotates toward the pole alignment point. Just as the motor would get to this point, the brushes jump across a gap in the stator rings.

Momentum carries the motor forward over this gap. When the brushes get to the other side of the gap, they contact the stator rings again and -- the polarity of the voltage is reversed in this set of rings! The motor begins accelerating again, this time trying to get to the opposite set of poles. (The momentum has carried the motor past the original pole alignment point.) This continues as the motor rotates.

In most DC motors, several sets of windings or permanent magnets are present to smooth out the motion.

- **Easy to control speed**

Controlling the speed of a brushed DC motor is simple. The higher the armature voltage, the faster the rotation. This relationship is linear to the motor's maximum speed.

The maximum armature voltage which corresponds to a motor's rated speed (these motors are usually given a rated speed and a maximum speed, such as 1750/2000 rpm) are available in certain standard voltages, which roughly increase in conjunction with horsepower. Thus, the smallest industrial motors are rated 90 VDC and 180 VDC. Larger units are rated at 250 VDC and sometimes higher.

Specialty motors for use in mobile applications are rated 12, 24, or 48 VDC. Other tiny motors may be rated 5 VDC.

Most industrial DC motors will operate reliably over a speed range of about 20:1 -- down to about 5-7% of base speed. This is much better performance than the comparable AC motor. This is partly due to the simplicity of control, but is also partly due to the fact that most industrial DC motors are designed with variable speed operation in mind, and have added heat dissipation features which allow lower operating speeds.

- **Easy to control torque**

In a brushed DC motor, torque control is also simple, since output torque is proportional to current. If you limit the current, you have just limited the torque which the motor can achieve. This makes this motor ideal for delicate applications such as textile manufacturing.

- **Simple, cheap drive design**

The result of this design is that variable speed or variable torque electronics are easy to design and manufacture. Varying the speed of a brushed DC motor requires little more than a large enough potentiometer. In practice, these have been replaced for all but sub-fractional horsepower applications by the SCR and PWM drives, which offer relatively precisely control voltage and current. Common DC drives are available at the low end (up to 2 horsepower) for under US$100 -- and sometimes under US$50 if precision is not important.

Large DC drives are available up to hundreds of horsepower. However, over about 10 horsepower careful consideration should be given to the price/performance trade offs with AC inverter systems, since the AC systems show a price advantage in the larger systems. (But they may not be capable of the application's performance requirments).

- **Disadvantages**
  - Expensive to produce
  - Can't reliably control at lowest speeds
  - Physically larger
  - High maintenance
  - Dust

4.7.3 WORKING OF DC MOTOR

In any electric motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. As you are well aware of from playing with magnets as a kid, opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion.

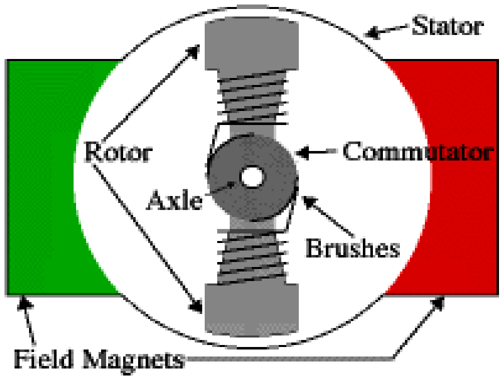


Fig.4.23 Internal Components of Motor

Principle

When a rectangular coil carrying current is placed in a magnetic field, a torque acts on the coil which rotates it continuously. When the coil rotates, the shaft attached to it also rotates and thus it is able to do mechanical work.

Every DC motor has six basic parts -- axle, rotor (a.k.a., armature), stator, commutator, field magnet(s), and brushes. In most common DC motors (and all that BEAMers will see), the external magnetic field is produced by high-strength permanent magnets[1]. The stator is the stationary part of the motor -- this includes the motor casing, as well as two or more permanent magnet pole pieces. The rotor (together with the axle and attached commutator) rotate with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a common motor layout-- with the rotor inside the stator (field) magnets.

The geometry of the brushes, commentator contacts, and rotor windings are such that when power is applied, the polarities of the energized winding and the stator magnet(s) are misaligned, and the rotor will rotate until it is almost aligned with the stator's field magnets. As the rotor reaches alignment, the brushes move to the next commentator contacts, and energize the next winding. Given our example two-pole motor, the rotation reverses the direction of current through the rotor winding, leading to a "flip" of the rotor's magnetic field, driving it to continue rotating.

In real life, though, DC motors will always have more than two poles (three is a very common number). In particular, this avoids "dead spots" in the commutator.

You can imagine how with our example two-pole motor, if the rotor is exactly at the middle of its rotation (perfectly aligned with the field magnets), it will get "stuck" there. Meanwhile, with a two-pole motor, there is a moment where the commutator shorts out the power supply (i.e., both brushes touch both commutator contacts simultaneously). This would be bad for the power supply, waste energy, and damage motor components as well. Yet another disadvantage of such a simple motor is that it would exhibit a high amount of torque "ripple" (the amount of torque it could produce is cyclic with the position of the rotor).

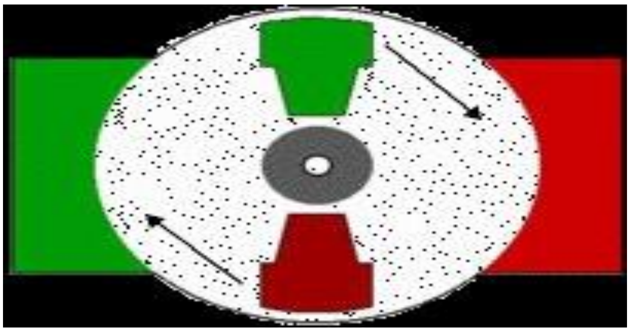


Fig.4.24 Rotation of two polar Motor

- **Construction and Working**

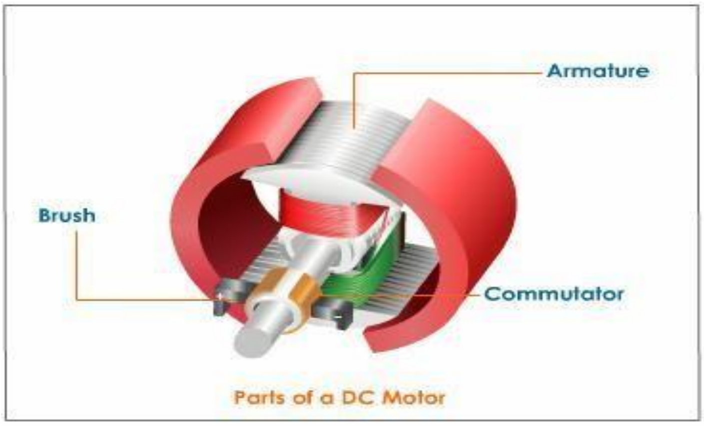


Fig.4.25 Parts of DC Motor

4.7.4 Parts of a DC Motor

- **Armature**

A D.C. motor consists of a rectangular coil made of insulated copper wire wound on a soft iron core. This coil wound on the soft iron core forms the armature. The coil is mounted on an axle and is placed between the cylindrical concave poles of a magnet.

- **Commutator**

A commutator is used to reverse the direction of flow of current. Commutator is a copper ring split into two parts C_1 and C_2. The split rings are insulated form each other and mounted on the axle of the motor. The two ends of the coil are soldered to these rings. They rotate along with the coil. Commutator rings are connected to a battery. The wires from the battery are not connected to the rings but to the brushes which are in contact with the rings.

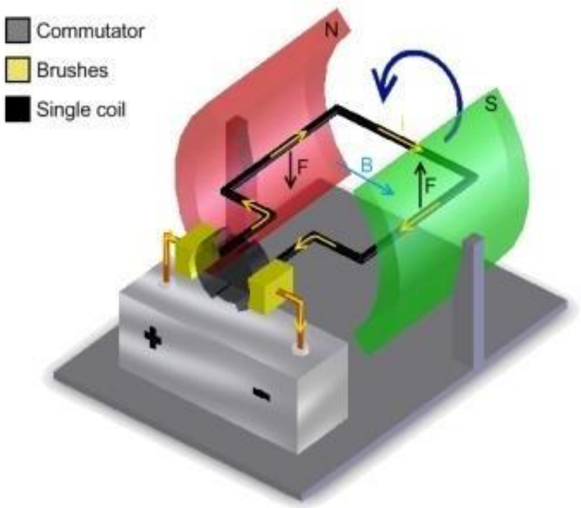


Fig.4.26 Commutator

- **Brushes**

Two small strips of carbon, known as brushes press slightly against the two split rings, and the split rings rotate between the brushes.

The carbon brushes are connected to a D.C. source.

- **Working of a DC Motor**

When the coil is powered, a magnetic field is generated around the armature. The left side of the armature is pushed away from the left magnet and drawn towards the right, causing rotation.

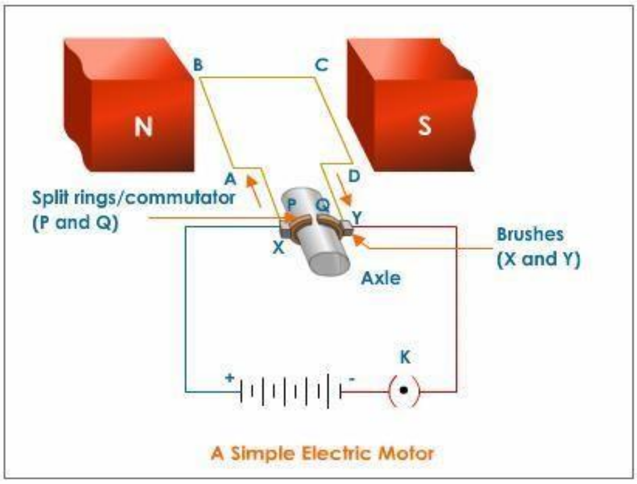


Fig.4.27 Simple Electric Motor

When the coil turns through 90^0, the brushes lose contact with the commutator and the current stops flowing through the coil.

However the coil keeps turning because of its own momentum. Now when the coil turns through 180^0, the sides get interchanged. As a result the commutator ring C_1 is now in contact with brush B_2 and commutator ring C_2 is in contact with brush B_1. Therefore, the current continues to flow in the same direction.

4.6.4 Parameters of the Dc Motors

- Direction of rotation
- Motor Speed
- Motor Torque
- Motor Start and Stop

▪ Direction of Rotation

A DC Motor has two wires. We can call them the positive terminal and the negative terminal, although these are pretty much arbitrary names (unlike a battery where these polarities are vital and not to be mixed!). On a motor, we say that when the + wire is connected to + terminal on a power source, and the - wire is connected to the - terminal source on the same power source, the motor rotates clockwise (if you are looking towards the motor shaft). If you reverse the wire polarities so that each wire is connected to the opposing power supply terminal, then the motor rotates counter clockwise. Notice this is just an arbitrary selection and that some motor manufacturers could easily choose the opposing convention. As long as you know what rotation you get with one polarity, you can always connect in such a fashion that you get the direction that you want on a per polarity basis.

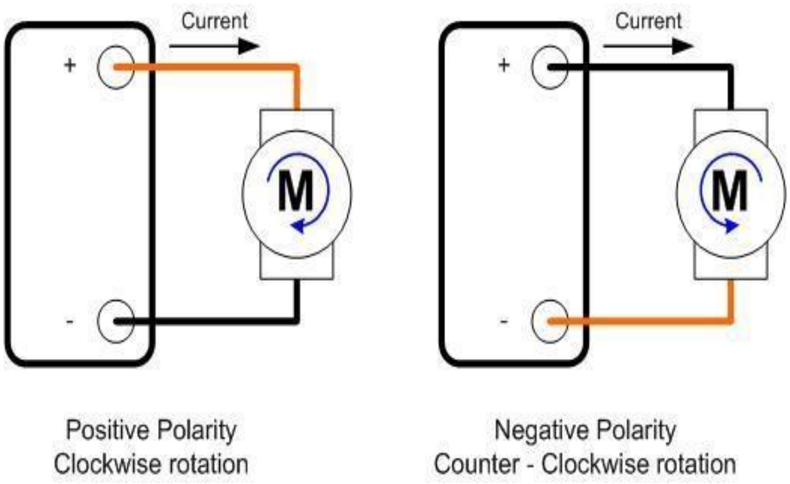


Fig.4.28Motor rotation with Polarity

▪ DC Motor Rotation vs Polarity

Facts:

- DC Motor rotation has nothing to do with the voltage magnitude or the current magnitude flowing through the motor.

- DC Motor rotation does have to do with the voltage polarity and the direction of the current flow.

DC Motor Speed

Whereas the voltage polarity controls DC motor rotation, voltage magnitude controls motor speed. Think of the voltage applied as a facilitator for the strengthening of the magnetic field. In other words, the higher the voltage, the quicker will the magnetic field become strong. Remember that a DC motor has an electromagnet and a series of permanent magnets. The applied voltage generates a magnetic field on the electromagnet portion. This electromagnet field is made to oppose the permanent magnet field. If the electromagnet field is very strong, then both magnetic entities will try to repel each other from one side, as well as attract each other from the other side. The stronger the induced magnetic field, the quicker will this separation/attraction will try to take place. As a result, motor speed is directly proportional to applied voltage.

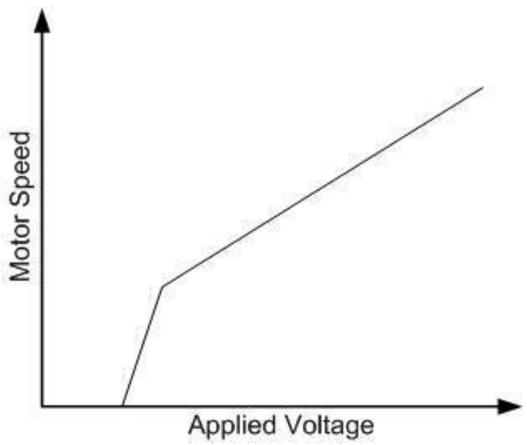


Fig.4.29 Motor Speed Vs Applied Voltage

Motor Speed Curve

One aspect to have in mind is that the motor speed is not entirely lineal. Each motor will have their own voltage/speed curve. One thing I can guarantee from each motor is that at very low voltages, the motor will supply not move. This is because the magnetic field strength is not enough to overcome friction.

Once friction is overcome, motor speed will start to increase as voltage increase. The following video shows the concept of speed control and offers some ideas on how this can be achieved.

- ## Motor Torque

In the previous segment I kind of described speed as having to do with the strength of the magnetic field, but this is in reality misleading. Speed has to do with how fast the magnetic field is built and the attraction/repel forces are installed into the two magnetic structures. Motor strength, on the other hand, has to do with magnetic field strength. The stronger the electromagnet attracts the permanent magnet, the more force is exerted on the motor load.

Per example, imagine a motor trying to lift 10 pounds of weight. This is a force that when multiplied by a distance (how much from the ground we are lifting the load) results in WORK. This WORK when exerted through a predetermined amount of time (for how long we are lifting the weight) gives us power. But whatever power came in, must come out as energy cannot be created or destroyed. So that you know, the power that we are supplying to the motor is computed by

$$\mathbf{P = IV}$$

Where P is power, I is motor current and V is motor voltage. Hence, if the voltage (motor speed) is maintained constant, how much load we are moving must come from the current.As you increase load (or torque requirements) current must also increase.

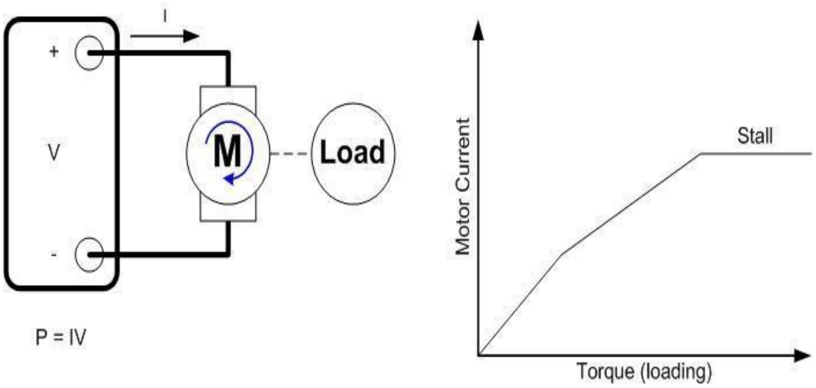


Fig.4.30 Torque Vs Motor Current

- **Motor Loading**

One aspect about DC motors which we must not forget is that loading or increase of torque cannot be infinite as there is a point in which the motor simply cannot move. When this happens, we call this loading “Stalling Torque”. At the same time this is the maximum amount of current the motor will see, and it is refer to Stalling Current. Stalling deserves a full chapter as this is a very important scenario that will define a great deal of the controller to be used. I promise I will later write a post on stalling and its intricacies.

- **Motor Start and Stop**

You are already well versed on how to control the motor speed, the motor torque and the motor direction of rotation. But this is all fine and dandy as long as the motor is actually moving. How about starting it and stopping it? Are these trivial matters? Can we just ignore them or should we be careful about these aspects as well? You bet we should!

Starting a motor is a very hazardous moment for the system. Since you have an inductance whose energy storage capacity is basically empty, the motor will first act as an inductor. In a sense, it should not worry us too much because current cannot change abruptly in an inductor, but the truth of the matter is that this is one of the instances in which you will see the highest currents flowing into the motor. The start is not necessarily bad for the motor itself as in fact the motor can easily take this Inrush Current. The power stage, on the other hand and if not properly designed for, may take a beating.

Once the motor has started, the motor current will go down from inrush levels to whatever load the motor is at. Per example, if the motor is moving a few gears, current will be proportional to that load and according to torque/current curves.

Stopping the motor is not as harsh as starting. In fact, stopping is pretty much a breeze. What we do need to concern ourselves is with how we want the motor to stop. Do we want it to coast down as energy is spent in the loop, or do we want the rotor to stop as fast as possible? If the later is the option, then we need braking. Braking is easily accomplished by shorting the motor outputs.

The reason why the motor stops so fast is because as a short is applied to the motor terminals, the Back EMF is shorted. Because Back EMF is directly proportional to speed, making Back EMF = 0, also means making speed = 0.

4.6.5 MOTOR DRIVER CIRCUIT

The name "H-Bridge" is derived from the actual shape of the switching circuit which control the motion of the motor. It is also known as "Full Bridge". Basically there are four switching elements in the H-Bridge as shown in the figure below.

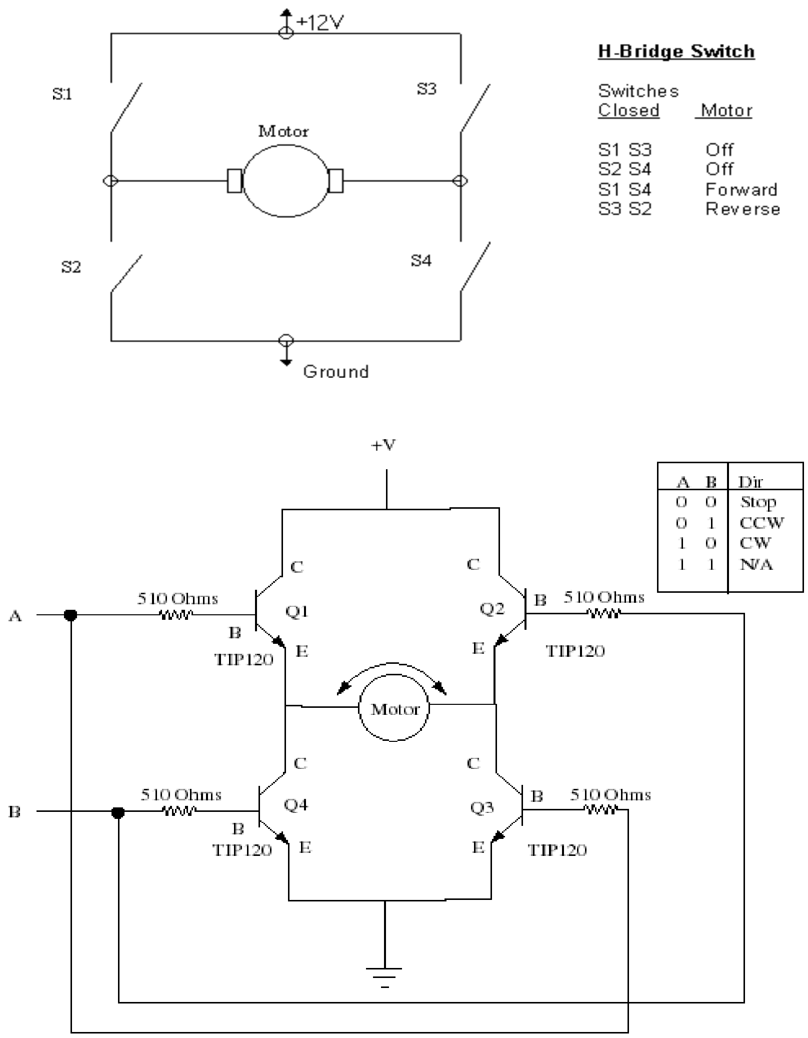


Fig.4.30 Motor driver Circuit

As you can see in the figure above there are four switching elements named as "High side left", "High side right", "Low side right", "Low side left". When these switches are turned on in pairs motor changes its direction accordingly. Like, if we switch on High side left and Low side right then motor rotate in forward direction, as current flows from Power supply

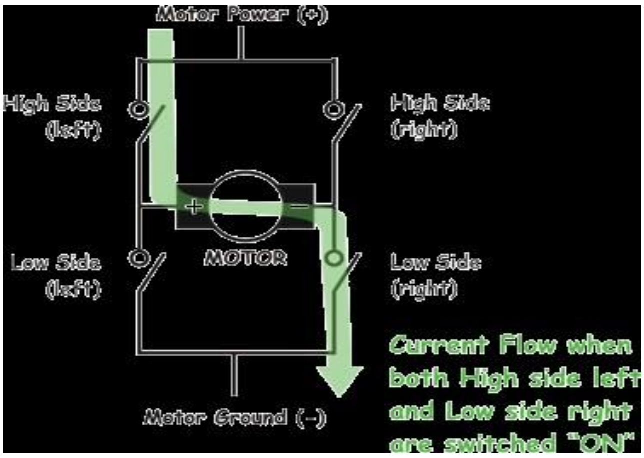


Fig.4.31 Current in Motor Circuit

Similarly, when you switch on low side left and high side right, the current flows in opposite direction and motor rotates in backward direction. This is the basic working of H-Bridge. We can also make a small truth table according to the switching of H-Bridge explained above.

Truth Table

High Left	High Right	Low Left	Low Right	Description
On	Off	Off	On	Motor runs clockwise
Off	On	On	Off	Motor runs anti-clockwise
On	On	Off	Off	Motor stops or decelerates
Off	Off	On	On	Motor stops or decelerates

Table.4.9 Truth Tables

As already said, H-bridge can be made with the help of transistors as well as MOSFETs, the only thing is the power handling capacity of the circuit. If motors are needed to run with high current then lot of dissipation is there. So head sinks are needed to cool the circuit.

Now you might be thinking why I did not discuss the cases like High side left on and Low side left on or high side right on and low side right on. Clearly seen in the diagram, you don't want to burn your power supply by shorting them. So that is why those combinations are not discussed in the truth table.

4.8 GAS SENSOR

Fig.4.32 Gas Sensor

4.8.1 Description

A gas detector is a device that detects the presence of gases in an area , often as part of a safety system .A gas detector can should an alarm to operators in the area where the leak is occurring, giving them the opportunity to leave.This type of device is important because there are many gases that can be harmful to organic life, such as humans or animals.A temperature sensor is an electronic device that measures c the temperature of its environment and converts the input data into electric data to record ,monitor temperature changes.

4.9 SOFTWARE EXPLINATION

Software Requirements

- Proteus simulation
- Arduino software
- Programming language

4.9.1 Arduino software

Arduino is an open-s4ource prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IOT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

4.9.2 Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux.

Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments.

Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handy board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand thectechnical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers c an make their own version

- Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money
- Getting Started with Arduino and Genuino products:-

Install the Arduino Software (IDE) on Windows PCs-

…This document explains how to install the Arduino Software (IDE) on Windows machines.

- Download the Arduino Software(IDE)
- Proceed with board specific instructions.

4.9.3 How to Download the Arduino Software (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

Installation:

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1

First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.4.33 (a) Standard USB Cable

Fig.4.34 (b) USB cable Mini-B

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

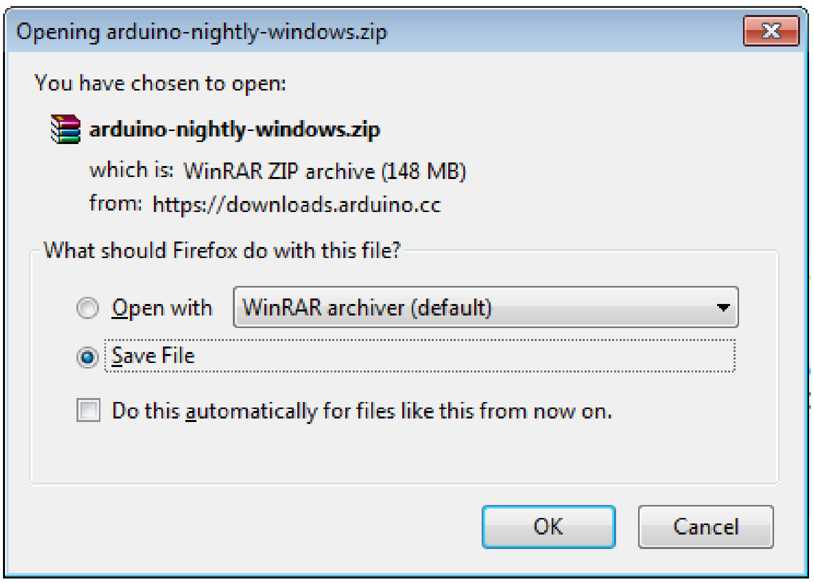


Fig.4.35 Process of Downloading the Arduino

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply.

If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe).

Double-click the icon to start the IDE.

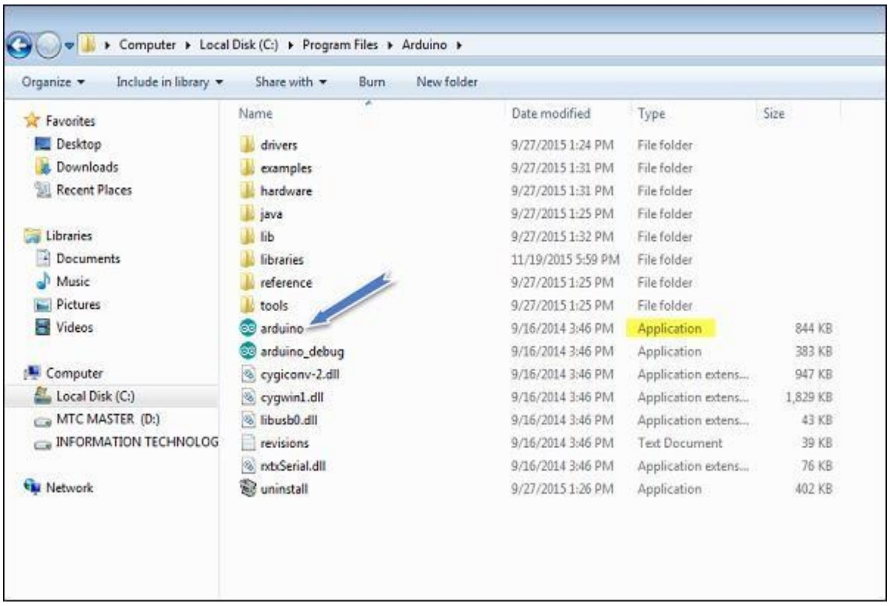


Fig.4.36 Launching Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.

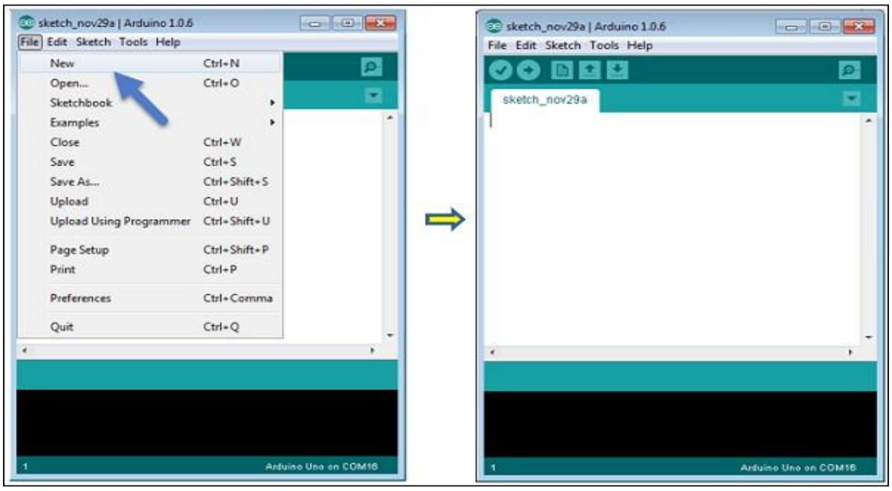


Fig.4.37 Opening Editor Window in IDE

To open an existing project example, select File → Example → Basics → Blink.

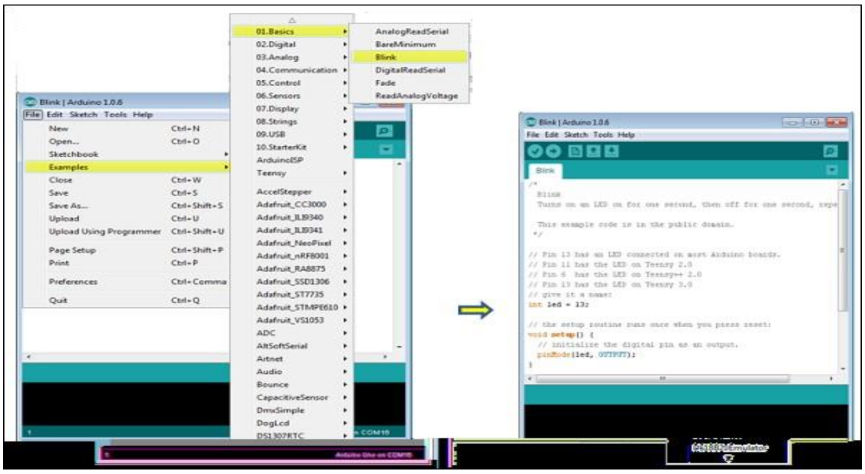


Fig.4.38 Opening Existing File in IDE

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

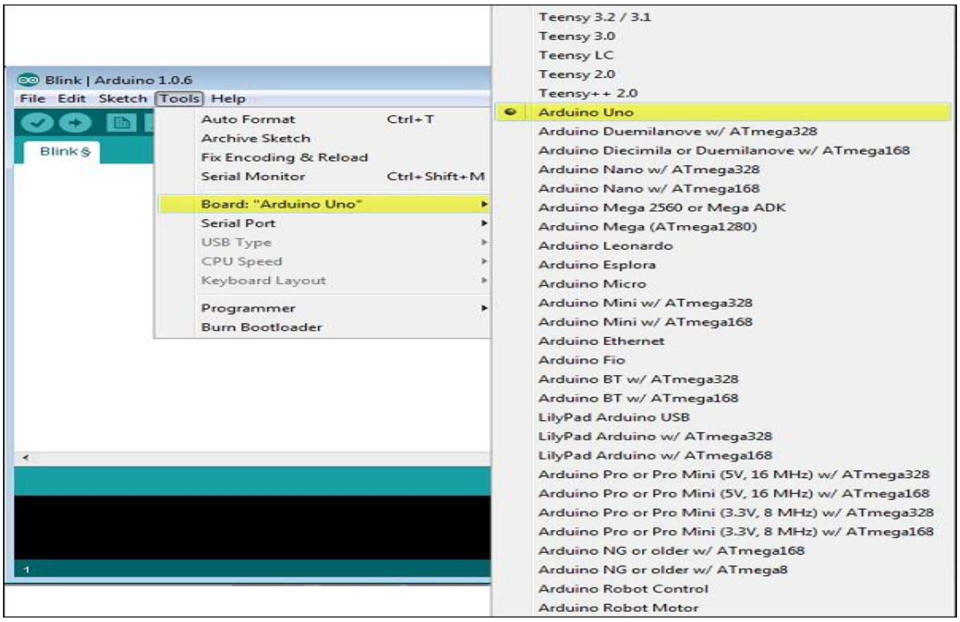


Fig.4.39 Selecting Arduino Board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports).

To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

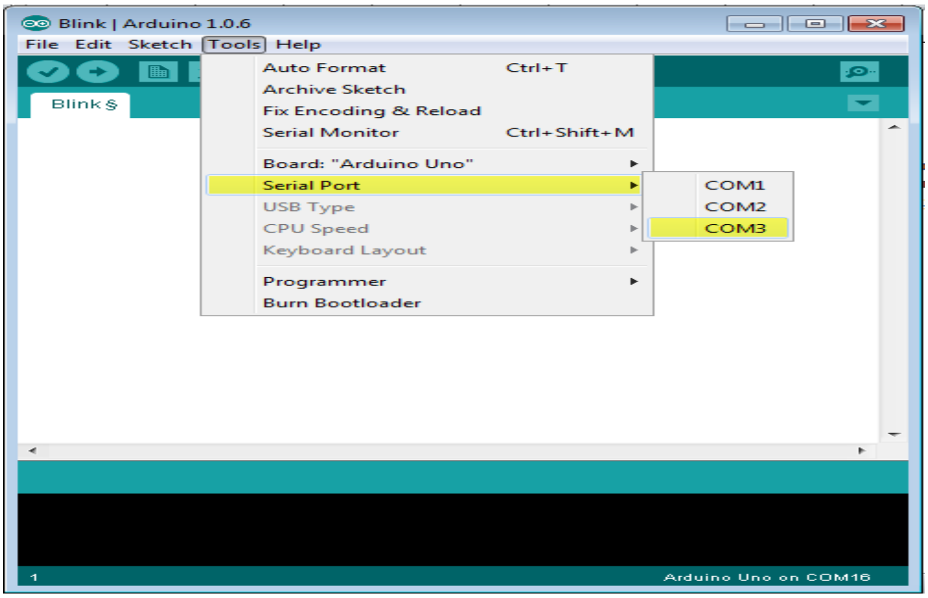


Fig.4.40 Selecting Serial Port

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

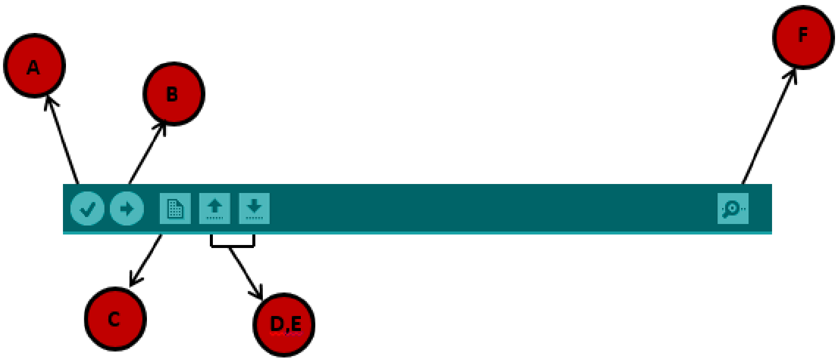


Fig.4.41 IDE Tool Bar

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

4.9.4 Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24 V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

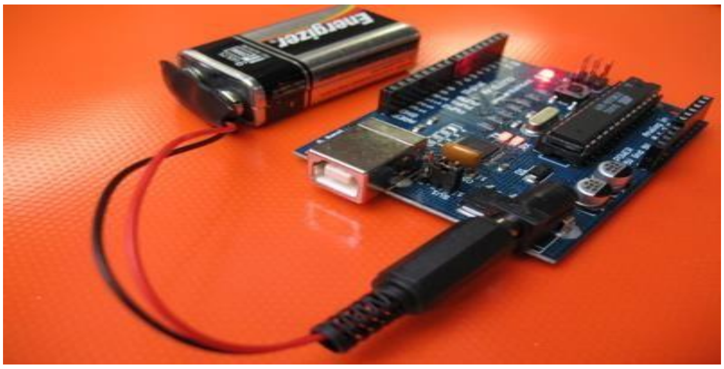


Fig.4.42 Arduino with battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called "sketches", but here we will just call them programs.

In the editing window that comes up, enter the following program show in Figure:4.66, paying attention to where semi-colons appear at the end of command lines.

Fig.4.43 Writing the Program in IDE

4.10 MC Programming Language: Embedded C

This is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements. Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems.

There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

4.11 Proteus:

Proteus is a simulation and design software tool developed by Lab Centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

4.11.1 About Proteus

It is a software suite containing schematic, simulation as well as PCB designing.
ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

4.11.2 Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components.

It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

4.11.3 Starting New Design

Step 1: Open ISIS software and select new design in File menu

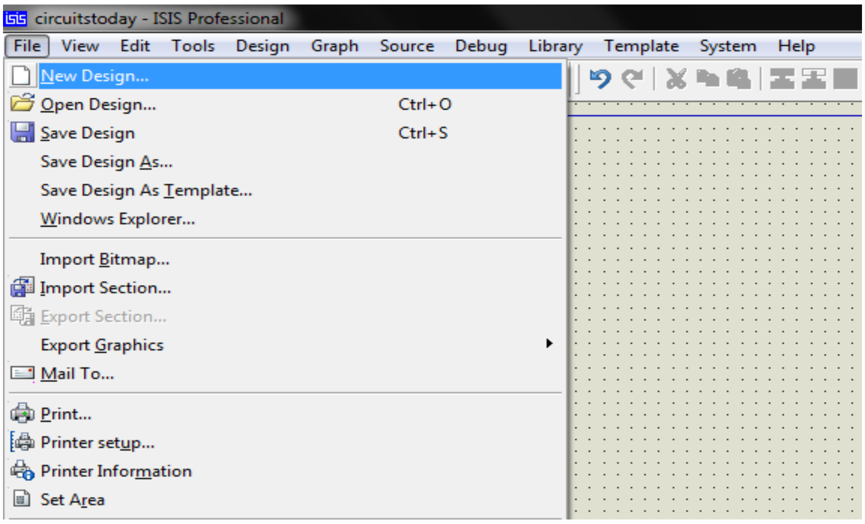


Fig .4.44 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

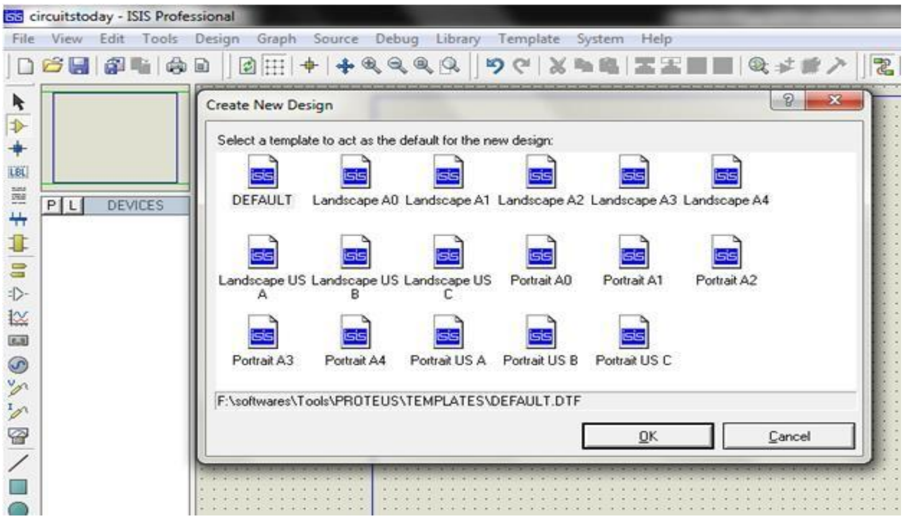


Fig.4.45 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

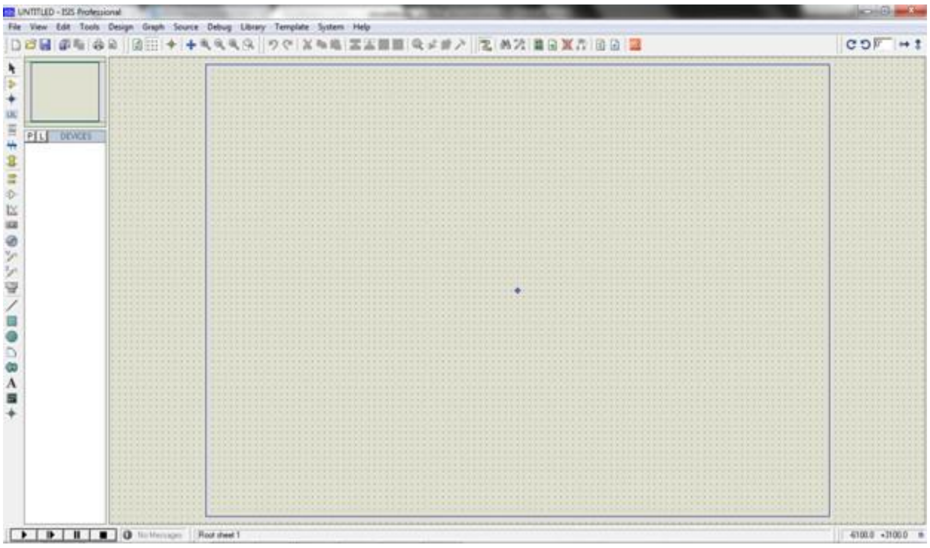


Fig 4.46 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

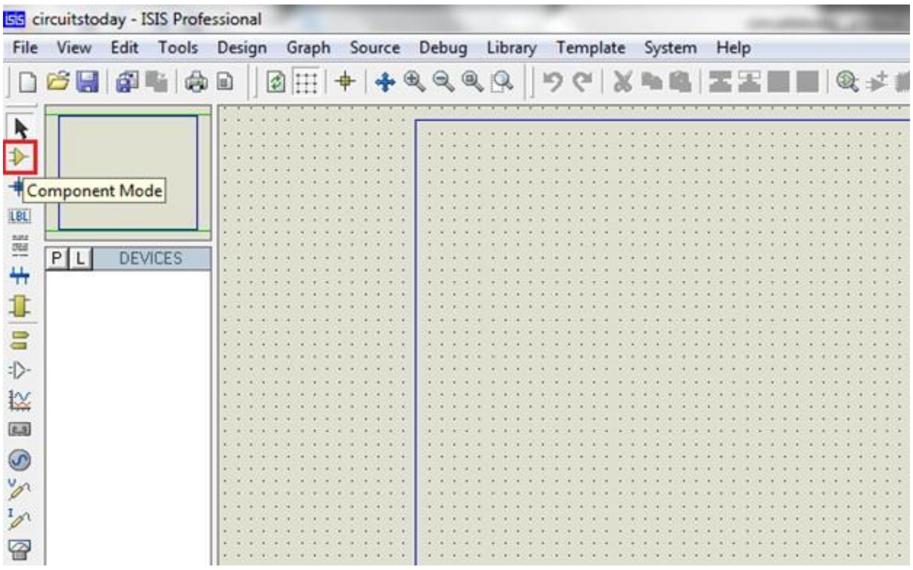


Fig.4.47 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

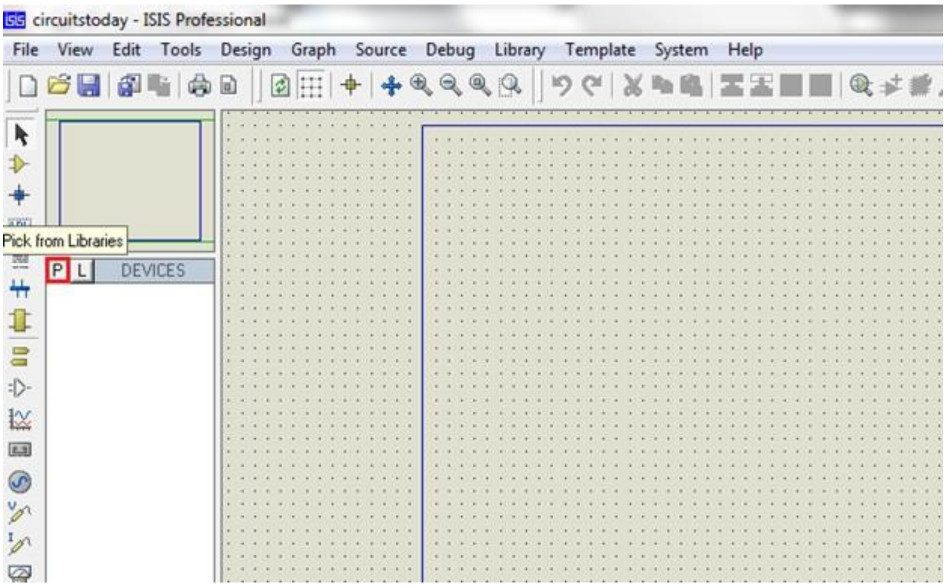


Fig4.47 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

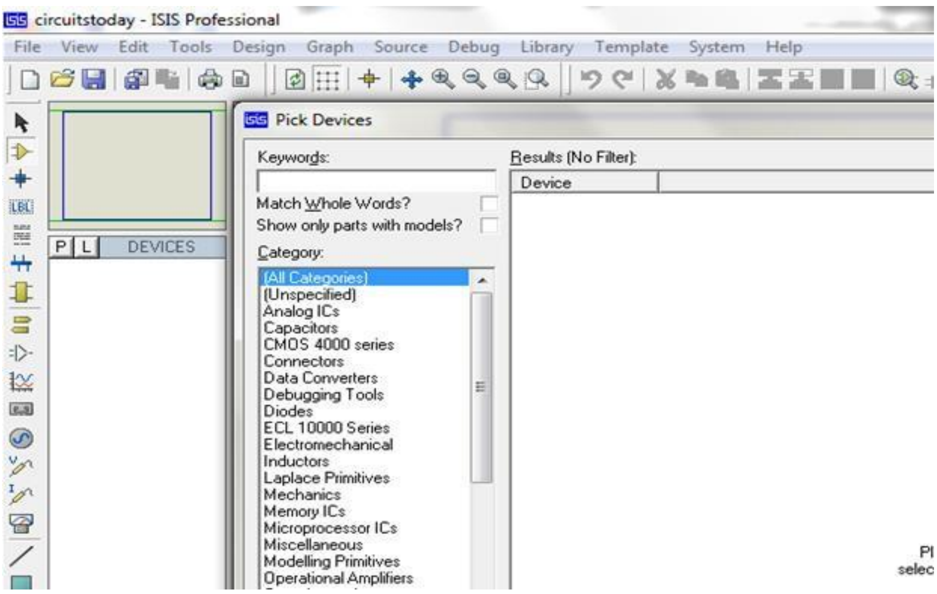


Fig 4.48 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

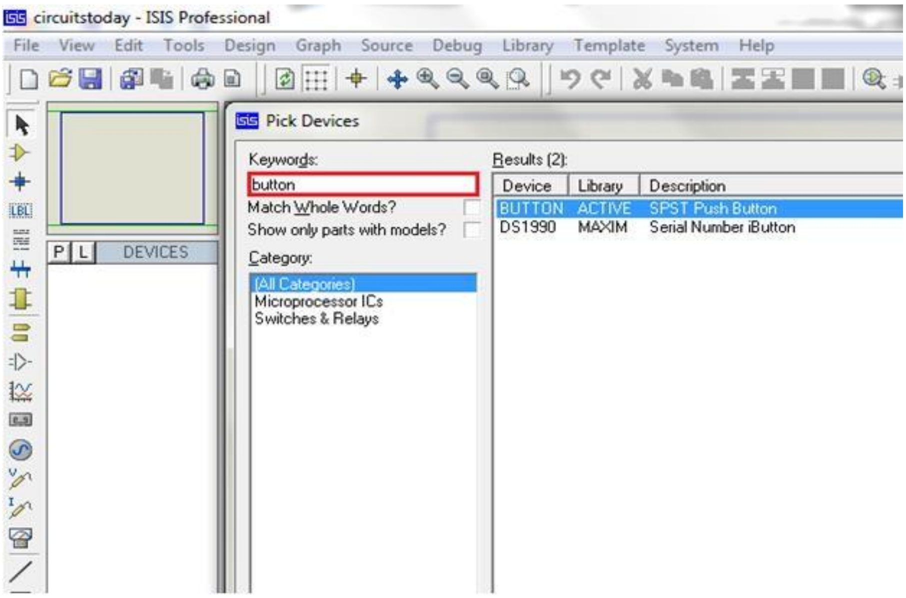


Fig4.49 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

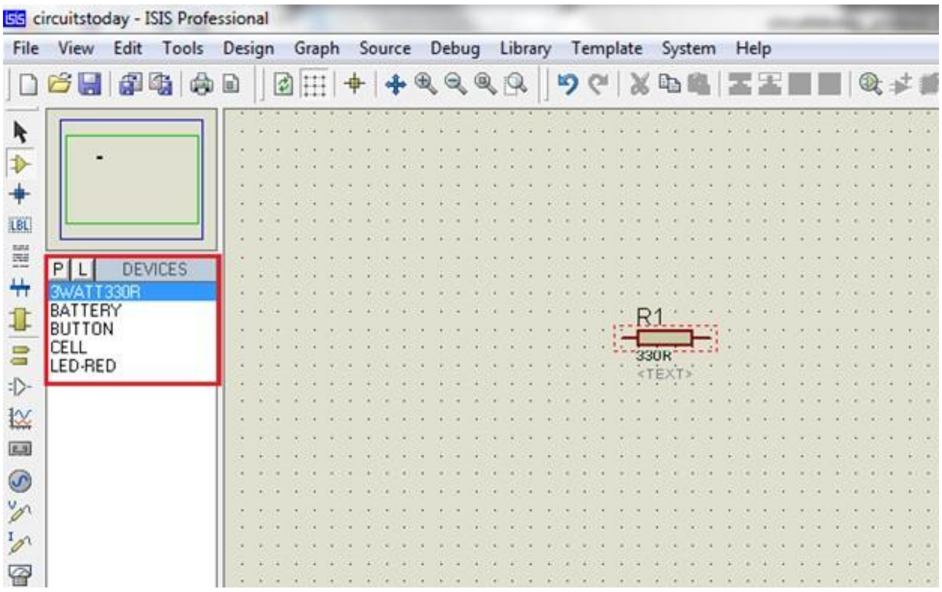


Fig4.50 Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

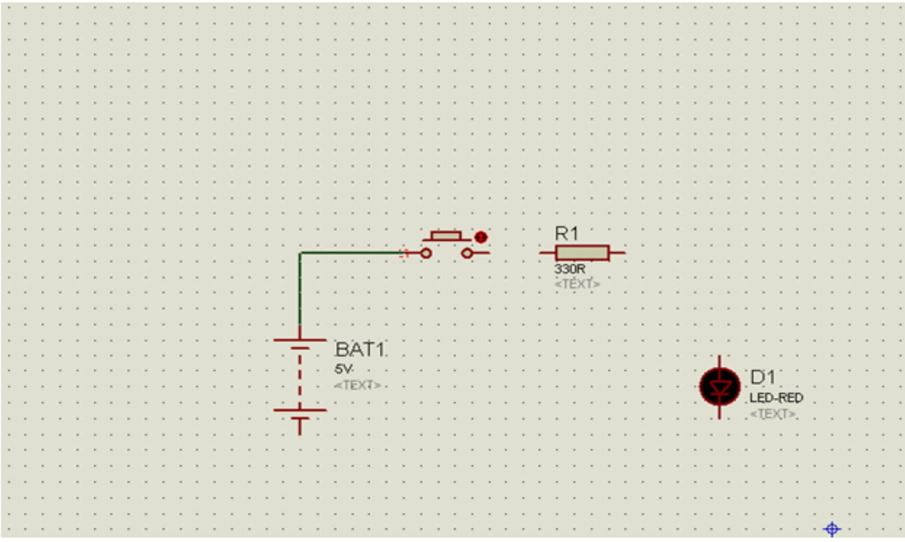


Fig4.51 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

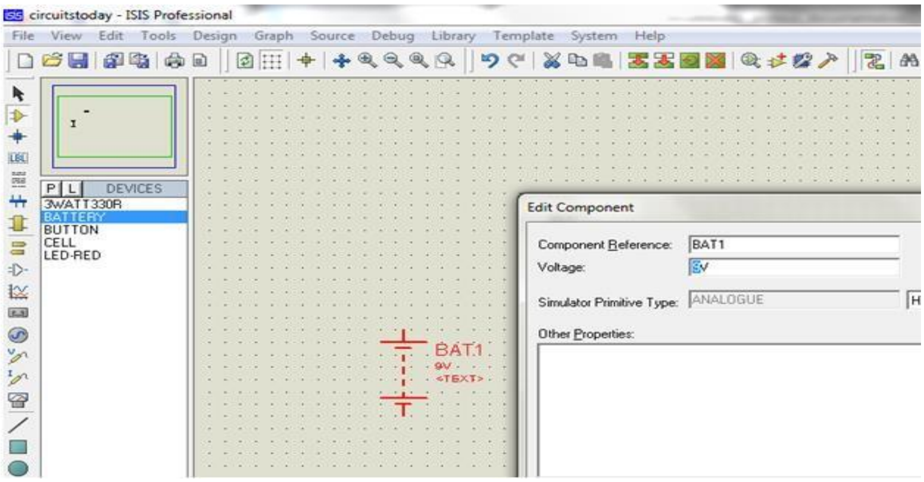


Fig4.52 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

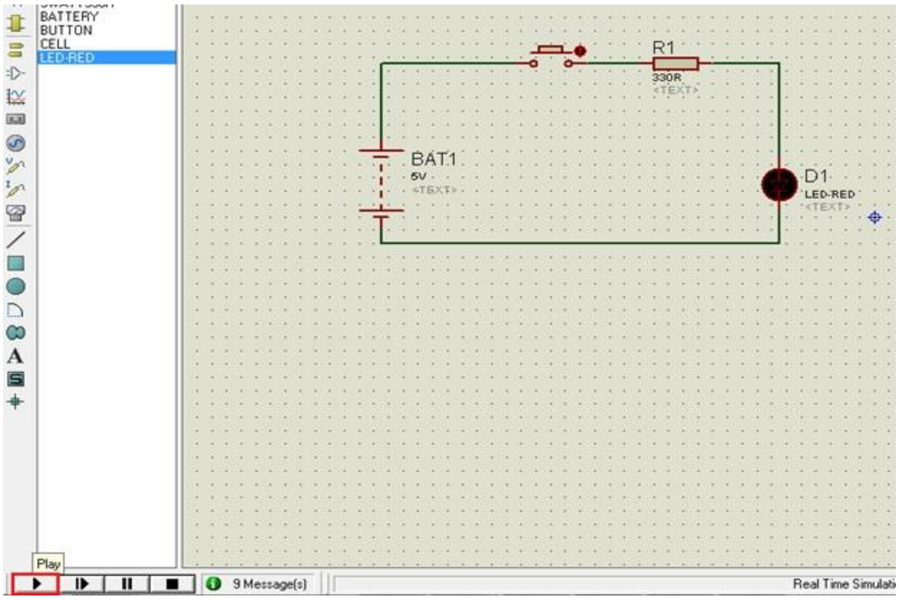


Fig 4.53 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

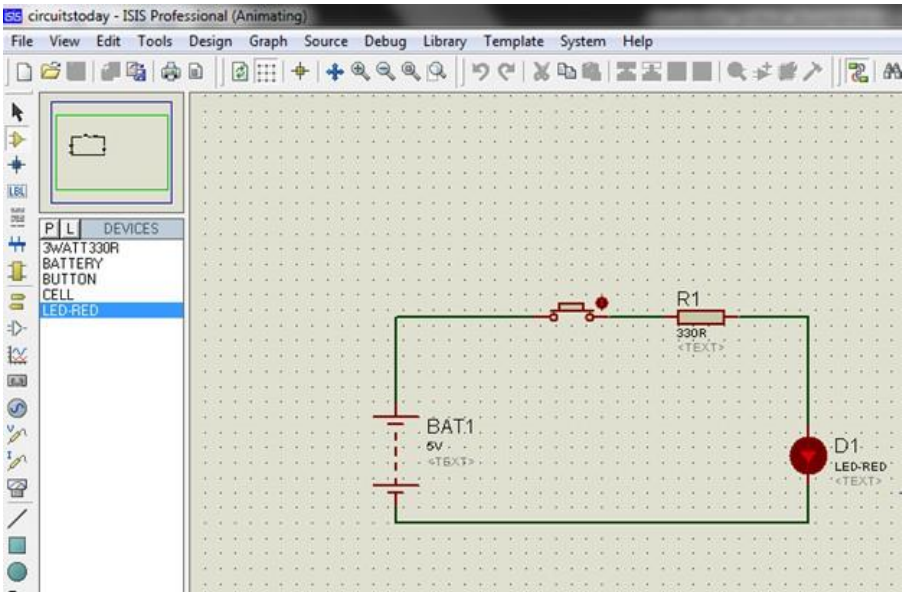


Fig4.54 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

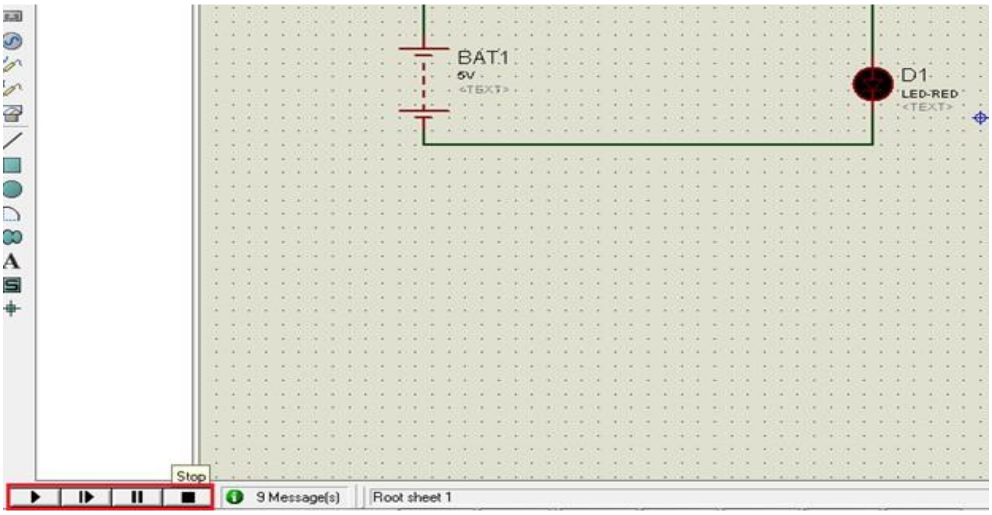


Fig4.55 Simulation Step-Pause-Stop Buttons

4.11.4 Project Code:

```
#include <LiquidCrystal.h>
 LiquidCrystal
lcd(13,12,11,10,9,8);//rs//en///d7//d6//d5//d4//
char res[130];
void serialFlush(){
  while(Serial.available() > 0) {
    char t = Serial.read();
  }
}
void sendmsg(char *num,char * msg);
int smoke=A4,temp=A5;
int m1 =4,m2=5;
```

```
char number1[15] = "6301186749\0";
void sendmsg(char *num,char * msg)
{
  Serial.print("AT+CMGS=\"");
  Serial.print(num);
  Serial.println("\"");delay(800);
  Serial.println(msg);delay(800);
  Serial.write(0x1a);
}
void setup() {
  char ret;
  pinMode(m1,OUTPUT);
  pinMode(m2,OUTPUT);
  //pinMode(m3,OUTPUT);
  //pinMode(m4,OUTPUT);

  digitalWrite(m1,LOW);
  digitalWrite(m2,LOW);
  //digitalWrite(m3,LOW);
  //digitalWrite(m4,LOW);
Serial.begin(9600);
  lcd.begin(16,2);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("WELCOME");
  delay(2000);
```

```
delay(2000);


lcd.clear();lcd.setCursor(0, 0);lcd.print("GSM INIT");

Serial.println("AT");delay(1000);

Serial.println("AT+CMGF=1");delay(1000);

Serial.println("AT+CNMI=1,2,0,0");delay(1000);

Serial.println("AT+CSMP=17,167,0,16");delay(1000);

lcd.clear();lcd.setCursor(0, 0);lcd.print("READY");

}

void loop() {


  int sd = digitalRead(smoke);

  int td = digitalRead(temp)/10;

    td   = td = td + 27;



//lcd.clear();lcd.setCursor(0, 0); lcd.print("S:");lcd.print(sd);
              //lcd.setCursor(8, 0); lcd.print("T:");lcd.print(td);


 if(sd == LOW)
 {
      digitalWrite(m1,LOW);
      digitalWrite(m2,LOW);
      //digitalWrite(m3,LOW);
      //digitalWrite(m4,LOW);
    lcd.clear();lcd.setCursor(0, 1); lcd.print("SMOKE FOUND");

    sendmsg(number1,"SMOKE FOUND");delay(1000);

  }

else if(td > 40)
```

CHAPTER-5

RESULTS & OUTPUTS

5.1 RESULT:

The outcome of the project "Automatic railway train safety system " after aligning all components is shown in figure.5.1 and figure.5.2, all the components are placed over the comfortable movements in any directions.

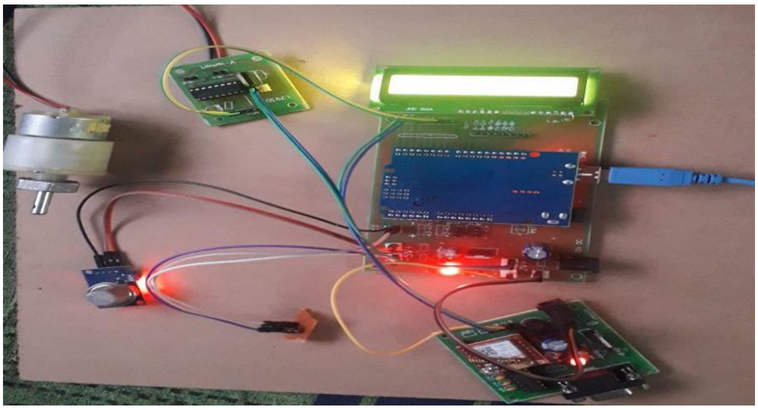


Fig.5.1 Automatic railway train safety system

The design of automation the system is used to stop the train whenever smoke and fire is accur. In these the input as LM35 we can use to sense the temperature and gas detection.in this situation temperature is high our motor will be turned on and it helps to pull the chain to stop the train. Along with breaking, the system sounds a buzzer that would alert nearby people. we get the output on LCD In that LCD we can found that smoke.in this situation GSM will be used to send the message to railway authority. These authority makes quick decisions to take control of fire.

Fig.5.2 :Smoke is found

Whenever smoke is found the GSM is send the message to railway authorities.and motor will be used to pull the chain and train will be stoped.the passangers are safe together.

CHAPTER-6

APPLICATIONS, ADVANTAGES AND LIMITATIONS

6.1 APPLICATIONS OF THE PROJECT:

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic services
- Communications
- Office systems and mobile equipment
- Banking finance and commercial

6.2 ADVANTAGES OF THE PROJECT:

- Automatic operation prevents errors due to manual operation.
- Lastly, no human resource is required . This makes its running cost very low compared to manned gates.
- We can avoid accident which sometimes happen.
- Power supply for the motor operation and signal lights is required.

6.3 LIMITATIONS:

- To establish the entire network is quite costly task.
- Energy requirements are high.
- Involves complex network of connections and require skilled workforce to build it in a good manner.

CHAPTER-7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION:

- It is a challenge to save the human life and vehicles from miserable train accidents in the area of modern science and technology.The obstacle detection system is very helpful to avoid accidents while the train is moving through the forest it will save the life of wild animals and also reduce the accidents in track.

.

7.2 FUTURE SCOPE:

- By using this project ,we can save man power.
- Hare,there is no need of man. The circuit is self sense the alarm. Once we switch on the circuit,it automacially performs all those actions without man handling.
- And we can implement a few more sensors and connect it to the satellite as a global feature of this system.
- It is the most advantages of the project.For this reason,in future, this project may be used in railways and also in aparments,military,hospitals , etc.

REFERENCES

[1] “pradeep Raj, “Increasing accidents in the unmanned level crossing of the railways”, 2012.

[2] Xishi Wang, Ning Bin, and Cheng Yinhang, “A new microprocessor based approach to an automatic control system.”, International Symposium on IndustrialElectronics, pp. 842-843, 1992

[3] Jeong Y., Choon-Sung Nam, Hee-JinJeong, and Dong Shin, “Train Auto Control System based on OSGi”, International Conference on AdvancedCommunication Technology, pp.276-279, 2008.

[4] Atul Kumar Dewangan, Meenu Gupta, and Pratibha Patel, “Automation of Railway Gate Control Using Micro-controller, International Journal of EngineeringResearch & Technology, pp.1-8, 2012.

[5] Gunyoung Kim, Kyungwoo Kang, “Railway Gate Con- trol System at Railroad-Highway Grade Crossing in Korea”. Railways to eliminate over 6,000 unmanned level crossings, article in the Indian Express, 30th July 2016 by PTI.

[6] Indian Railway develops warning system for unmanned level crossings, article in Times of India, 25th October 2015 by PTI

[7] J. Banuchandar, V. Kaliraj, P. Balasubramanian, S. Deepa, N. Thamilarasi,“ Automated Unmanned Railway Level Crossing System”, International Journal of Modern Engineering Research (IJMER)Volume.2, Issue.1, Jan-Feb 2012, pp-458-463.

[8] Ahmed Salih Mahdi. Al-Zuhairi,“Automatic Railway Gate and Crossing Control based Sensors & Microcontroller ”, International Journal of Computer Trends and Technology (IJCTT) – Volume 4 Issue 7–July 2013, pp.2135- 2140.

[9] Krishna, Shashi Yadav and Nidhi, “Automatic Railway Gate Control Using Microcontroller”, Oriental Journal Of Computer Science & Technology, Vol.6, No.4, December 2013, pp 435-440.

ST. MARTIN'S ENGINEERING COLLEGE

(Affiliated to JNTU, Hyderabad)

Dhulapally(V), Qutubullapur(M), Secunderabad

2020-2021

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

A

Major Project Report

On

UNDERWATER IMAGE RESTORATION USING SCENE DEPTH ESTIMATION TECHNIQUE

Submitted for partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY IN

ELECTRONICS AND COMMUNICATION ENGINEERING

BY

M. Bhargav Prasad	17K81A04F5
K. Vijay Kumar	17K81A04F7
P. Anand	17K81A04G3

UNDER THE GUIDANCE OF

Mr. G. Ramesh Reddy

Associate Professor,

Department of ECE

ST. MARTIN'S ENGINEERING COLLEGE
An UGC Autonomous Institute
Dhulapally, Secunderabad-500 100
NIRF & NAAC A+ Accredited
www.smec.ac.in

Certificate

This is to certify that the project entitled Sensitive Label Privacy Preservation with Anatomization for Data Publishing is being submitted by Batch-C7, ECE-4C in fulfilment of the requirement for the award of degree of **BACHELOR IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Guide Signature	Head of the Department
G.RAMESH REDDY	**DR.B. HARIKRISHNA**

Internal Examiner	**External Examiner**

ACKNOWLEDGEMENT

We sincerely thank the management of our college **ST. MARTIN'S ENGINEERING COLLEGE** for providing required facilities during our project work.

We derive our great pleasure in expressing our sincere gratitude to our principal **Dr.P. Santosh Kumar Patra** for his timely suggestions, which helped us to complete the project work successfully.

It is the very auspicious moment to express our gratitude to **Dr. B. Hari Krishna**, Head of the Department, ECE for his consistent encouragement during the progress of this project. We take it as a privilege to thank our project coordinator **Dr. A. Anand**, Professor, department of ECE for the ideas that led to complete the project work and we also thank them for their continuous guidance, support and unfailing patience, throughout the course of this work.

We sincerely thank our project Internal guide **Mr. G. Ramesh Reddy** M.Tech(Ph.D)**,** Department of ECE for guidance and encouragement in carrying out this project work.

M. Bhargav Prasad	17K81A04F5
K. Vijay Kumar	17K81A04F7
P. Anand	17K81A04G3

DECLARATION

We, the students of '**Bachelor of Technology in Department of Electronics and Communication Engineering**', session: 2017 – 2021, **St. Martin's Engineering College**, hereby declare that the project work entitled '**UNDERWATER IMAGE RESTORATION USING SCENE DEPTH ESTIMATION TECHNIQUE**' is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

M. Bhargav Prasad	**K. Vijay Kumar**	**P. Anand**
17K81A04F5	**17K81A04F7**	**17K81A04G3**

ABSTRACT

Underwater image restoration is a difficult task due to the shortage of reference image and changing underwater environment. The images are distorted by absorption and scattering eff ect of light. In deep water, light undergoes wavelength-dependent attenuation. Thus, red color light that has a large wavelength attenuates more than blue light and any other color. As depth increases, light attenuates most of the red content and image appears in bluish-green color. These images also have low contrast, color cast and hazy appearance. Existing methods may need specialized hardware or it may be based on multiple images of the same scene. Thus they cannot be used in real-time or video acquisition task. So, it's better to form an effective method for color enhancement and image restoration of images. Here depth map estimation along with image blurriness is proposed. As light travels deeper into the water, the image gets blurred and this is used to obtain depth map. The backlight is also obtained. These factors are substituted in the IFM (Image Formation Model) to restore the image. Keywords—IFM, Image restoration, image enhancement

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

In oceanic engineering, underwater imaging has a significant role. Underwater images have large scope in various research work. Nowadays several technologies are available to record underwater images and videos like waterproof cameras. But underwater images suff er from various demerits in case of visibility, contrast, illumination. This is because light rays on its path from the object to the camera get reflected by the organic particles dispersed in water. The major problem in image restoration is lack of reference image and underwater environment. In underwater, light propagation suffers from absorption and scattering. Scattering is because of collision with suspended particles and causes light rays to reflect in different directions, which can add a hazy layer to the image. Absorption causes energy loss of light rays and it depends upon the water medium properties such as density and turbidity. As turbidity increases absorption rate increases. Colors degrade gradually depending on wavelength as it moves deeper. Red light with large wavelength value attenuates more than shorter wavelength blue color light. Thus blue and green light reaches depth compared to red and orange. Thus, underwater images have appeared in bluish-green.

The methods used to increase the underwater image quality falls under two categories restoration and enhancement. Enhancement involves unsharp masking, histogram equalization, etc. for visually better results. The enhanced image is not necessarily to be exactly similar to the original image. But restoration can improve the image as same as the original image by modeling degradation. It restores an image by estimating underwater image formation model parameters. It can be modeled as a linear combination of the direct component and backscattered component. Direct component is the amount of light reflected from the object surface, only a portion of it reaches the camera and remaining gets scattered or absorbed. Scattering is of two types: forward and backward scattering. Backward scattering is responsible for low contrast thus it is included in the image formation model and the forward scattering component is neglected. Image formation model is given by

$$I(x) = S(x)\, t(x) + B\,(1-t(x)) \qquad (1)$$

I(x) is observed intensity at pixel x. J(x) is scene radiance, a fraction of light reflected from the scene into the camera. t(x) is a transmission map, a portion of the light that is not scattered or absorbed. B is a backlight, portion of light scattered into the camera by the suspended particles, it may add a hazy layer to the image. The main steps in this paper include image blurriness estimation, backlight estimation, depth map estimation, transmission map estimation, and scene radiance recovery. Finally, non-reference image quality assessment methods such as BRISQUE, entropy are used to measure the restored image quality.

Autonomous Underwater Vehicle (AUV) is an underwater unmanned vehicle for ocean exploration. It is suitable for detecting unknown environment and has the functions of autonomous decision-making, planning and obstacle avoidance in complex environment. Autonomous localization is of great significance to realize safe and efficient work of AUV. There are two main localization methods of AUV: dead reckoning and acoustic beacon positioning, but their complexity and the expensive instrumentation restrict the promotion of AUV. Compared with inertial navigation and acoustic beacon positioning, SLAM is a method with small-sized and affordable instrument. Simultaneous location and mapping (SLAM) refers to estimating a robot's pose (position and orientation) in unknown environment through repeated observation of environmental characteristics. Based on this foundation, an incremental map of the surrounding environment is constructed. Through the efforts of researchers, the SLAM technology has been widely used in fields such as unmanned aerial vehicle, sweeping robot, unmanned driving and intelligent wearing equipment. Compared with the SLAM of land and aerial unmanned system, the application of the SLAM in AUV positioning and navigation is just beginning. The research on AUV SLAM was mainly based on extended Kalman Filter and Particle Filter. Sonar is used as environment sensing sensor [1-2].

The earth is associate aquatic planet and the maximum amount as eightieth of its surface is roofed by water.. Moreover, there is a strong interest in knowing what lies in underwater. Present days, an image of deep waters has a scope to large investigation to explore the underwater for sea floor expedition and navigation. Enthusiasm of underwater imaging includes the inspection of plants, seabed exploration, the search for wrecks up and to the exploration of natural resources. There were several issues faced by the human in the underwater, if he dives deep into the ocean and stay there for a long time to perform experimentation. [1]. Due to the above reasons, unmanned remote vehicles are used to sea floor exploration.

A. Historical Development

Underwater image quality improvement approaches present a path to magnify the object recognition in underwater surrounding. A heap of research started for the upgradation of image visual quality, but a little amount of work has been carried out in this area. In the deep waters, image quality is degrades due to poor illumination conditions and the light properties differ in water compared to air.[2]. There were several parameters which decreases the quality of an image in underground waters. So in order to remove all these effects

there are several techniques has been implemented and practiced.

B. Need for Pre process

Initially processing is necessary for deep water images due to their poor quality during acquisition. Necessity for pre-processing of deep water images[1] are discussed below:

(i) Quality of images taken from deep water is deteriorated due to light ray attributes like scattering and absorption of light.

(ii) Specificity of surroundings such as lighting inequalities, water torridness, and blue complexion is more or less influential when vehicles move.

(iii) Video or image captured from deep waters like unknown rigid scene, and the depth of the scene and low light sensitivity due to Marine snow etc.

1.1 TRADITIONAL TECHNIQUES FOR IMAGE ENHANCEMENT

There are several techniques which are used very frequently for processing the image to improve the visual quality. Some of them are as follows:

(i) Contrast Stretching

(ii) Adaptive Histogram Equalization

A. Contrast stretching

The contrast stretching is a method to transform high intense region of image into brighter and less intense region into more darker by using a predefined transformation function T(r) [2]. Generally, the underwater images will have less grey values. There are 256 grey values. ‘0’ indicates black and ‘255’ indicates white. In this method the current grey value of the image is stretched towards 255 i.e., from black to white, pixel by pixel. That means the contrast of the image is stretched, so that the quality of the image is improved for better vision.

For example:

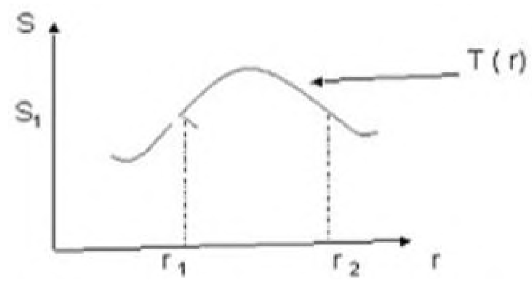


Fig.1. Two different gray levels look same

Here two different thresholds are considered for the entire image and the values between them are stretched to the maximum extent, so that the contrast increases. And more over by this method the entire global image contrast is enhanced.

Fig.2. (a) Raw image (b) Enhanced image

But the disadvantage here is that the transformation function is not unique. Depending on the application the suitable transformation function is chosen

B. Adaptive histogram equalization

Adaptive histogram equalization is a PC based image processing technique which is used to improve the quality of image properties like contrast. It is similar to contrast stretching method but with a slight difference. It computes several intensities of specific gray value, each corresponding to a distinct portion of an image, and with the help of them intensities are rearranged by applying a suitable transformation function. For example, a simple transformation function such as each pixel transformed based on the histogram of a square surrounding the pixel [3]. Existing values will be mapped to new values keeping actual number of intensities in the resulting image equal or less than the original number of intensities. The transformation function applied on the histogram is proportional to the cumulative distributive function (CDF) of pixel values in the neighborhood. Therefore it suits for enhancing the local details and enhancing the edge information of each region of an image.

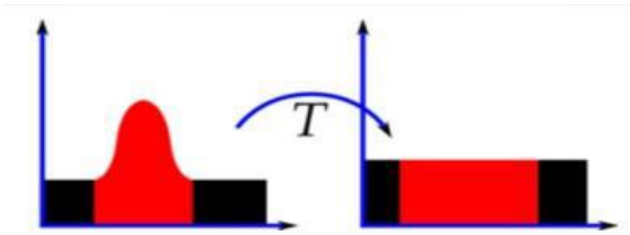


Fig.3. Histograms of an image before and after histogram equalization

Histogram equalization is a technique for changing the overall pixel intensities based on transformation function and contrast of an image. Histogram equalization is an effective technique which will benefit for the images with extreme contrast values. The limitation of this technique highlights the unwanted noise present in the background of an image and lead to loss in the information signal. It results in undesired effects in the resultant images [4].

Fig.4. (a) Raw image **(b) Enhanced image**

Here the noise in relatively homogeneous regions of the image are amplified which results in poor SNR. And also only the local objects of the image are enhanced and the background is left unenhanced.

CHAPTER 2

INTRODUCTION TO IMAGE PROCESSING

INTRODUCTION

2.1 IMAGE:

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Fig 1 General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

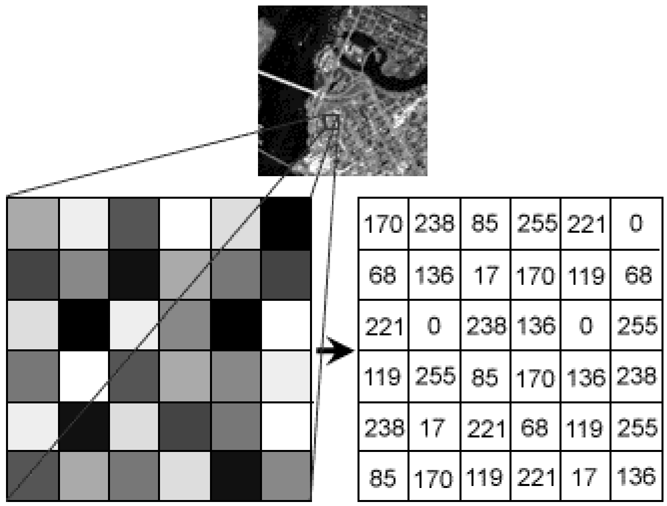


Fig 2.1 Image pixel

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

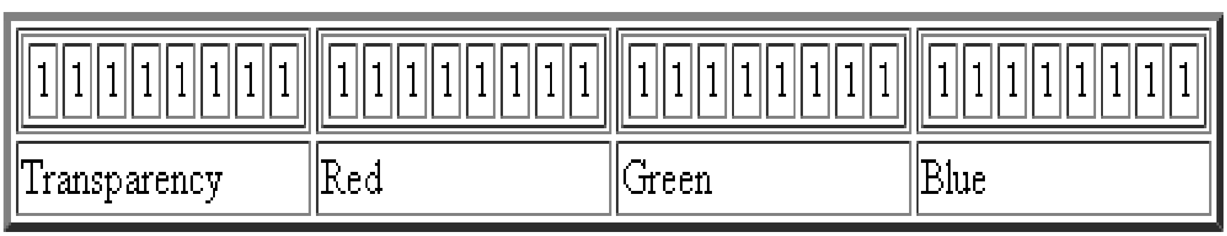


Fig2.2 Transparency image

2.2 IMAGE FILE SIZES:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

2.3 IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

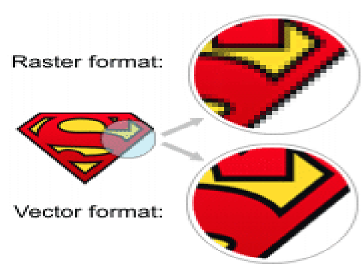


Fig2.3 Resolution image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

2.3.1 RASTER FORMATS:

These formats store images as bitmaps (also known as pixmaps)

- **JPEG/JFIF:**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

- **EXIF:**

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

- **TIFF:**

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

- **PNG:**

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

- **GIF:**

GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

- **BMP:**

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

2.3.2 VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM:**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

- **SVG:**

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

2.4 IMAGE PROCESSING:

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

2.5 FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:

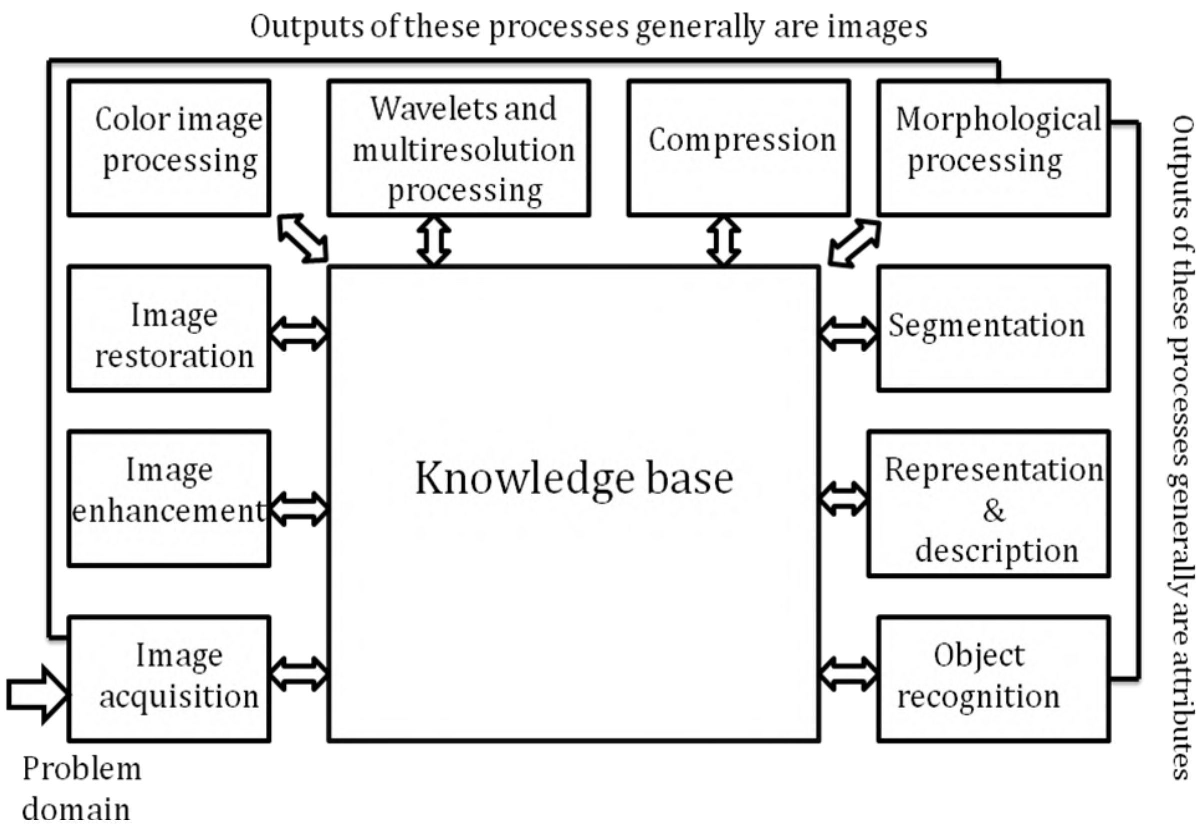


Fig 2.5 Image fundamental

2.5.1 Image Acquisition:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Fig 2.5.1 Digital camera image

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Fig 2.5.2 digital camera cell

2.5.2 Image Enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because “it looks better.” It is important to keep in mind that enhancement is a very subjective area of image processing.

Fig 2.5.3 Image enhancement

2.5.3 Image restoration:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Fig 2.5.4 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, where as removal of image blur by applying a deblurring function is considered a restoration technique.

2.5.4 Color image processing:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Fig 2.5.5 Color & Gray scale image

2.5.5 Wavelets and multiresolution processing:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform based image processing since the late1950's, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

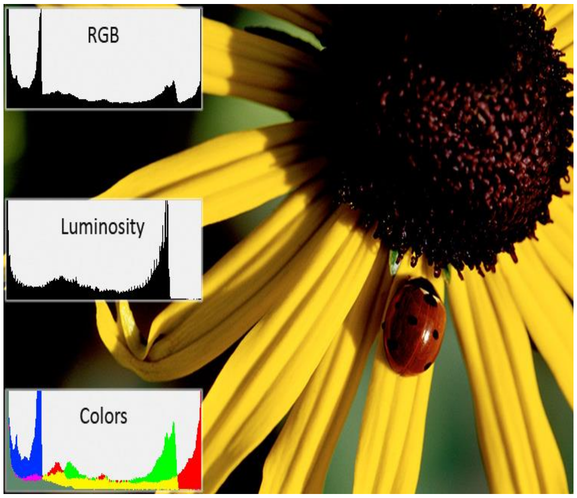


Fig 2.5.6 rgb histogram image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called **Multiresolution** theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

2.5.6 Compression:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

2.5.7 Morphological processing:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Fig 2.5.7 blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

2.5.8 Segmentation:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

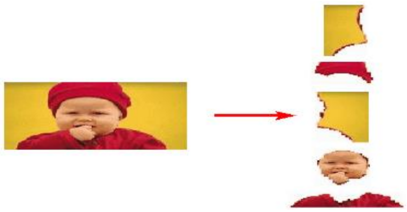


Fig 2.5.8 Image segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

2.5.9 Representation and description:

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

2.5.10 Object recognition:

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

2.5.11 Knowledgebase:

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an inter related to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

2.6 COMPONENTS OF AN IMAGE PROCESSING SYSTEM:

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

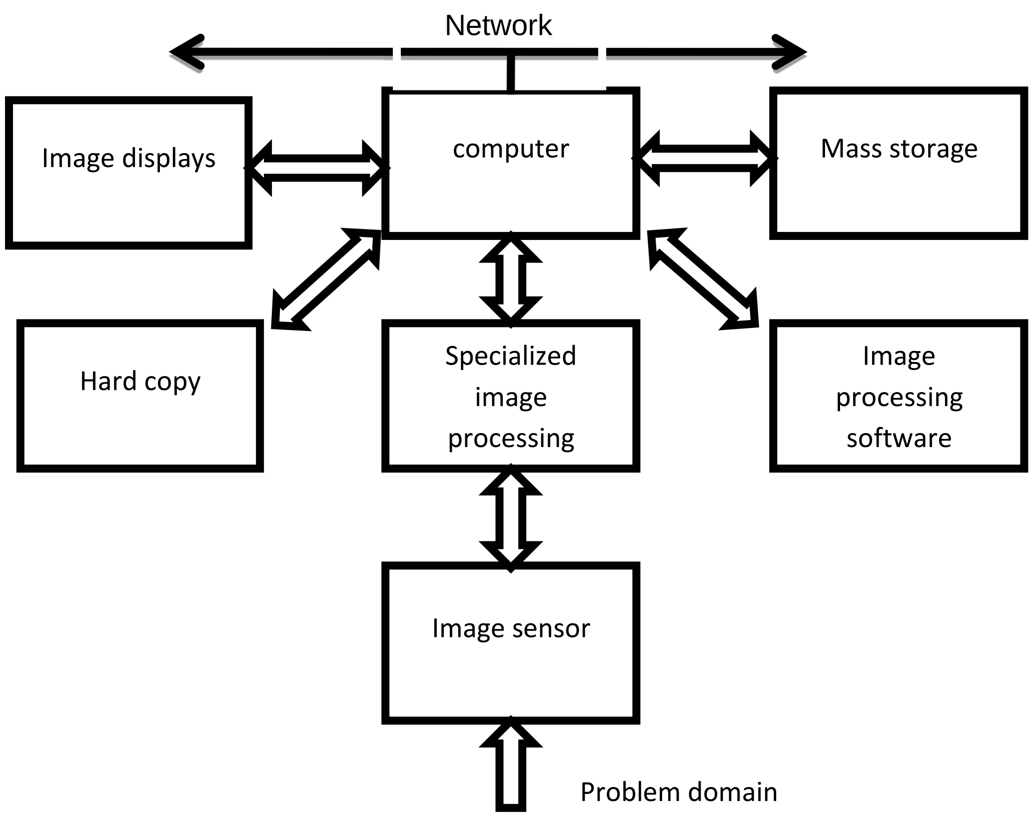


Fig 2.6 Component of image processing

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 1.24 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

- **Image sensors:**

With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

- **Specialized image processing hardware:**

Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

- **Computer:**

The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

- **Image processing software:**

Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

- **Mass storage:**

Mass storage capability is a must in image processing applications. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage for image processing applications fall into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), Gbytes (meaning giga, or one billion, bytes), and Tbytes (meaning tera, or one trillion, bytes)

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig. 1.24. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in "jukeboxes" are the usual media for archival applications.

- **Image displays:**

Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are there requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

- **Hardcopy:**

Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

- **Network:**

Networking is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of optical fiber and other broadband technologies.

Color and texture are two low-level features widely used for image classification, indexing and retrieval. Color is usually represented as a histogram, which is a first order statistical measure that captures global distribution of color in an image One of the main drawbacks of the histogram-based approaches is that the spatial distribution and local variations in color are ignored. Local spatial variation of pixel intensity is commonly used to capture texture information in an image. Grayscale Co-occurrence Matrix (GCM) is a well-known method for texture extraction in the spatial domain. A GCM stores the number of pixel neighborhoods in an image that have a particular grayscale combination. Let I be an image and let p and Np respectively denote any arbitrary pixel and its neighbor in a given direction. If GL denotes the total number of quantized gray levels and gl denotes the individual gray levels, where, gl $\in$ {0, . . .,GL _ 1}, then each component of GCM can be written as follows:

$$\mathrm{gcm}(i,j) = \Pr((\mathrm{gl}_{\mathrm{p}}, \mathrm{gl}_{N_{\mathrm{p}}}) = (i,j))$$

gcm(i, j) is the number of times the gray level of a pixel p denoted by glp equals i, and the gray level of its neighbor Np denoted by glNp equals j, as a fraction of the total number of pixels in the image. Thus, it estimates the probability that the gray level of an arbitrary pixel in an image is i, and that of its neighbor is j. One GCM matrix is generated for each possible neighborhood direction, namely, 0, 45, 90 and 135.Average and range of 14 features like Angular Second Moment, Contrast, Correlation, etc., are generated by combining all the four matrices to get a total of 28 features. In the GCM approach for texture extraction, color information is completely lost since only pixel gray levels are considered.

To incorporate spatial information along with the color of image pixels, a feature called color correlogram has recently been proposed. It is a three dimensional matrix that represents the probability of finding pixels of any two given colors at a distance 'd' apart Auto correlogram is a variation of correlogram, which represents the probability of finding two pixels with the same color at a distance 'd' apart. This approach can effectively represent color distribution in an image. However, correlogram features do not capture intensity variation many image databases often contain both color as well as gray scale images. The color correlogram method does not constitute a good descriptor in such databases.

Another method called Color Co-occurrence Matrix (CCM) has been proposed to capture color variation in an image. CCM is represented as a three-dimensional matrix, where color pair of the pixels p and Np are captured in the first two dimensions of the matrix and the spatial distance 'd' between these two pixels is captured in the third dimension. This approach is a generalization of the color correlogram and reduces to the pure color correlogram for d = 1. CCM is generated using only the Hue plane of the HSV (Hue, Saturation and Intensity Value) color space. The Hue axis is quantized into HL number of levels. If individual hue values are denoted by hl, where $\mathrm{hl} \in \{0, \ldots, \mathrm{HL} - 1\}$, then each component of CCM can be written as follows:

$$\mathrm{ccm}(i, j) = \Pr((\mathrm{hl}_{\mathrm{p}}, \mathrm{hl}_{N_{\mathrm{p}}}) = (i, j))$$

Four matrices representing neighbors at angles 0, 90, 180 and 270 are considered. This approach was further extended by separating the diagonal and the non-diagonal components of CCM to generate a Modified Color Co-occurrence Matrix (MCCM). MCCM, thus, may be written as follows: MCCM = (CCMD;CCMND)

Here, CCMD and CCMND correspond to the diagonal and off-diagonal components of CCM. The main drawback of this approach is that, like correlogram, it also captures only color information and intensity information is completely ignored.

An alternative approach is to capture intensity variation as a texture feature from an image and combine it with color features like histograms using suitable weights . One of the challenges of this approach is to determine suitable weights since these are highly application-dependent. In certain applications like Content-based Image Retrieval (CBIR), weights are often estimated from relevance feedback given by users.

While relevance feedback is sometimes effective, it makes the process of image retrieval user-dependent and iterative. There is also no guarantee on the convergence of the weight-learning algorithms. In order to overcome these problems, researchers have tried to combine color and texture features together during extraction.

proposed two approaches for capturing color and intensity variations from an image using the LUV color space. In the Single-channel Co-occurrence Matrix (SCM), variations for each color channel, namely, L, U and V are considered independently. In the Multi channel Co-occurrence Matrix (MCM), variations are captured taking two channels at a time – UV, LU and LV. Since the LUV color space separates out chrominance (L and U) from luminance (V), SCM in effect, generates one GCM and two CCMs from each image independently. As a result, correlation between the color channels is lost

However, in MCM, the count of pair wise occurrences of the values of different channels of the color space is captured. Thus, each component of MCM can be written as follows:

$$mcmUV(i; j) = Pr((up; vNp) = (i; j))$$

$$mcmLU(i; j) = Pr((lp; uNp) = (i; j))$$

$$mcmLV(i; j) = Pr((lp; vNp) = (i; j))$$

Here, mcmUV(i, j) is the number of times the U chromaticity value of a pixel p denoted by up equals i, and the V chromaticity value of its neighbor Np denoted by vNp equals j, as a fraction of the total number of pixels in the image. Similarly, mcmLU(i, j) and mcmLV(i, j) are defined. One MCM matrix is generated for each of the four neighborhood directions, namely, 0, 45, 90 and 135.

Deng and Manjunath (2001) proposed a two-stage method called JSEG, which combines color and texture after image segmentation. In the first stage, colors are quantized to the required levels for differentiating between various regions of an image. Pixel values of the regions are then replaced by their quantized color levels to form a color map. Spatial variation of color levels between different regions in the map is viewed as a type of texture composition of the image.

Yu et al. (2002) suggested the use of color texture moments to represent both color and texture of an image. This approach is based on the calculation of

Local Fourier Transformation (LFT) coefficients. Eight templates equivalent to LFT are operated over an image to generate a characteristic map of the image. Each template is a $3 \cdot 3$ filter that considers eight neighbors of the current pixel for LFT calculation. First and second order moments of the characteristic map are then used to generate a set of features.

In this paper, we propose an integrated approach for capturing spatial variation of both color and intensity levels in the neighborhood of each pixel using the HSV color space. In contrast to the other methods, for each pixel and its neighbor, the amount of color and intensity variation between them is estimated using a weight function. Suitable constraints are satisfied while choosing the weight function for effectively relating visual perception of color and the HSV color space properties. The color and intensity variations are represented in a single composite feature known as Integrated Color and Intensity Co-occurrence Matrix (ICICM). While the existing schemes generally treat color and intensity separately, the proposed method provides a composite view to both color and intensity variations in the same feature. The main advantage of using ICICM is that it avoids the use of weights to combine individual color and texture features. We use ICICM feature in an image retrieval application from large image databases.

Early result on this work was reported in (Vadivel et al., 2004a). In the next section, we describe the proposed feature extraction technique after introducing some of the

properties of the HSV color space. Choice of quantization levels for color and intensity axes, selection of parameter values and a brief overview of the image retrieval application

Integrated color and intensity co-occurrence matrix:

We propose to capture color and intensity variation around each pixel in a two-dimensional matrix called Integrated Color and Intensity Co-occurrence Matrix (ICICM). This is a generalization of the Grayscale Co-occurrence Matrix and the Color Co-occurrence Matrix techniques. For each pair of neighboring pixels, we consider

their contribution to both color perception as well as gray level perception to the human eye. Some of the useful properties of the HSV color space and their relationship to human color perception are utilized for extracting this feature. In the next sub-section, we briefly explain relevant properties of the HSV color space. In the subsequent subsection, we describe how the properties can be effectively used for generating ICICM.

HSV color space:

HSV Color space: Basically there are three properties or three dimensions of color that being hue, saturation and value HSV means Hue, Saturation and Value. It is important to look at because it describes the color based on three properties. It can create the full spectrum of colors by editing the HSV values. The first dimension is the Hue. Hue is the other name for the color or the complicated variation in the color. The quality of color as determined by its dominant wavelength. This Hue is broadly classified into three categories. They are primary Hue, Secondary Hue and Teritiary Hue. The first and the foremost is the primary Hue it consists of three colors they are red, yellow and blue. The secondary Hue is formed by the combination of the equal amount of colors of the primary Hue and the colors of the secondary Hue which was formed by the primary Hue are Orange, Green and violet. The remaining one is the teritiary Hue is formed by the combination of the primary Hue and the secondary Hue. The limitless number of colors are produced by mixing the colors of the primary Hue in different amounts. Saturation is the degree or the purity of color. Then the second dimension is the saturation. Saturation just gives the intensity to the colors. The saturation and intensity drops just by mixing the colors or by adding black to the color. By adding the white to the color in spite of more intense the color becomes lighter. Then finally the third dimension is the Value. The value is the brightness of the color. When the value is zero the color space is totally black with the increase in the color there is also increase in the brightness and shows the various colors. The value describes the contrast of the color. That means it describes the lightness and darkness of the color. As similar to the saturation this value consists of the tints and shades. Tints are the colors with the added white and shades are the colors with the added black.

Properties of the HSV color space:

Sensing of light from an image in the layers of human retina is a complex process with rod cells contributing to scotopic or dim-light vision and cone cells to photopic or bright-light vision (Gonzalez and Woods, 2002). At low levels of illumination, only the rod cells are excited so that only gray shades are perceived. As the illumination level increases, more and more cone cells are excited, resulting in increased color perception. Various color spaces have been introduced to represent and specify colors in a way suitable for storage, processing or transmission of color information in images. Out of these, HSV is one of the models that separate out the luminance component (Intensity) of a pixel color from its chrominance components (Hue and Saturation). Hue represents pure color, which is perceived when incident light is of sufficient illumination and contains a single wavelength. Saturation gives a measure of the degree by which a pure color is diluted by white light. For light with low illumination, corresponding intensity value in the HSV color space is also low.

The HSV color space can be represented as a Hexacone, with the central vertical axis denoting the luminance component, I (often denoted by V for Intensity Value). Hue, is a chrominance component defined as an angle in the range [0,2p] relative to the red axis with red at angle 0, green at 2p/3, blue at 4p/3 and red again at 2p. Saturation, S, is the other chrominance component, measured as a radial distance from the central axis of the hexacone with value between 0 at the center to 1 at the outer surface. For zero saturation, as the intensity is increased, we move from black to white through various shades of gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the color represented by its hue. When saturation is near 0, all the pixels in an image look alike even though their hue values are different.

As we increase saturation towards 1, the colors get separated out and are visually perceived as the true colors represented by their hues. Low saturation implies presence of a large number of spectral components in the incident light, causing loss of color information even though the illumination level is sufficiently high. Thus, for low values of saturation or intensity, we can approximate a pixel color by a gray level while for higher saturation and intensity, the pixel color can be approximated by its hue. For low intensities, even for a high saturation, a pixel color is close to its gray value. Similarly, for low saturation even for a high value of intensity, a pixel is perceived as gray. We use these properties to estimate the degree by which a pixel contributes to color perception and gray level perception.

One possible way of capturing color perception of a pixel is to choose suitable thresholds on the intensity and saturation. If the saturation and the intensity are above their respective thresholds, we may consider the pixel to have color dominance; else, it has gray level dominance. However, such a hard thresholding does not properly capture color perception near the threshold values. This is due to the fact that there is no fixed level of illumination above which the cone cells get excited. Instead, there is a gradual transition from scotopic to photopic vision. Similarly, there is no fixed threshold for the saturation of cone cells that leads to loss of chromatic information at higher levels of illumination caused by color dilution. We, therefore, use suitable weights that vary smoothly with saturation and intensity to represent both color and gray scale perception for each pixel.

NON INTERVAL QUANTIZATION:

Due to the large range for each component by directly calculating the characteristics for the retrieval then the computation will be very difficult to ensure rapid retrieval. It is essential to quantify HSV space component to reduce computation and improve efficiency. At the same time, because the human eye to distinguish colors is limited, do not need to calculate all segments. Unequal interval quantization according the human color perception has been applied on H , S ,V components.

Based on the color model of substantial analysis, we divide color into eight parts. Saturation and intensity is divided into three parts separately in accordance with the human eyes to distinguish. In accordance with the different colors and subjective color perception quantification, quantified hue(H), saturation(S) and value(V).

In accordance with the quantization level above, the H, S, V three-dimensional feature vector for different values of with different weights to form one dimensional feature vector and is given by the following equation:

$$G = Qs*Qv*H+Qv*s+V$$

Where Qs is the quantized series of S and Qv is the quantized series of V. And now by setting Qs = Qv = 3, Then G = 9H+3S+V

$$H = \begin{cases} 0 \ \ if \ \ h \in [316,20] \\ 1 \ \ if \ \ h \in [21,40] \\ 2 \ \ if \ \ h \in [41,75] \\ 3 \ \ if \ \ h \in [76,155] \\ 4 \ \ if \ \ h \in [156,190] \\ 5 \ \ if \ \ h \in [191,270] \\ 6 \ \ if \ \ h \in [271,295] \\ 7 \ \ if \ \ h \in [296,315] \end{cases} \qquad S = \begin{cases} 0 \ \ if \ \ s \in [0,0.2) \\ 1 \ \ if \ \ s \in [0.2,0.7) \\ 2 \ \ if \ \ s \in [0.7,1) \end{cases} \qquad V = \begin{cases} 0 \ \ if \ \ v \in [0,0.2) \\ 1 \ \ if \ \ v \in [0.2,0.7) \\ 2 \ \ if \ \ v \in [0.7,1) \end{cases}$$

In this way three component vector of the HSV from one dimensional vector, which quantize the whole color space for the 72 kinds of the main colors. So we can handle 72 bins of one dimensional histogram. This qualification is effective in reducing the images by the effect of the light intensity, but also reducing the computational time and complexity.

IMAGE RETRIEVAL:

Image retrieval is nothing but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords or the descriptions to the images so that the retrieval can be performed. Manual image annotation is time consuming, expensive and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc..,

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others. one can today generate volumes of images with content as diverse as family get-togethers and national park visits. Low-cost storage and easy Web hosting has fueled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchable image data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real-world image search system designers.

As far as technological advances are concerned, growth in Content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real world implications, applications, and constraints of the technology. Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future.

An image retrieval system designed to serve a personal collection should focus on features such as personalization, flexibility of browsing, and display methodology. For example, Google's Picasa system [Picasa 2004] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods. The notable few applications include an FPGA implementation of a color-histogram-based image retrieval system [Kotoulas and Andreadis 2003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 2003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 2002].

Discussion. Regardless of the nature of the collection, as the expected user-base grows, factors such as concurrent query support, efficient caching, and parallel and distributed processing of requests become critical. For future real-world image retrieval systems, both software and hardware approaches to address these issues are essential. More realistically, dedicated specialized servers, optimized memory and storage support, and highly parallelizable image search algorithms to exploit cluster computing powers are where the future of large-scale image search hardware support lies.

OVERVIEW OF TEXTURE:

We all know about the term Texture but for defining it is a hard time. One can differentiate the two different Textures by recognizing the similarities and differences. Commonly there are three ways for the usage of the Textures:

Based on the Textures the images can be segmented to differentiate between already segmented regions or to classify them. we can reproduce Textures by producing the descriptions. The texture can be analyzed in three different ways. They are Spectral, Structural and Statistical:

I

1	1	5	6	8
2	3	5	7	1
4	5	7	1	2
8	5	1	2	5

GLCM

	1	2	3	4	5	6	7	8
1	1	2	0	0	1	0	0	0
2	0	0	1	0	1	0	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0
5	1	0	0	0	0	1	2	0
6	0	0	0	0	0	0	0	1
7	2	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0

CHAPTER-3

DIGITAL IMAGE PROCESSING

Digital image processing

Background:

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An important characteristic underlying the design of image processing systems is the significant level of testing & experimentation that normally is required before arriving at an acceptable solution. This characteristic implies that the ability to formulate approaches &quickly prototype candidate solutions generally plays a major role in reducing the cost & time required to arrive at a viable system implementation.

What is DIP

An image may be defined as a two-dimensional function f(x, y), where x & y are spatial coordinates, & the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital image by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensor, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis& computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting

& somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low-level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low- level process is characterized by the fact that both its inputs & outputs are images.

Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images. Finally higher- level processing involves “Making sense” of an ensemble of recognized objects, as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

What is an image?

An image is represented as a two dimensional function f(x, y) where x and y are spatial co-ordinates and the amplitude of ‘f’ at any pair of coordinates (x, y) is called the intensity of the image at that point.

Gray scale image:

A grayscale image is a function I (xylem) of the two spatial coordinates of the image plane.

I(x, y) is the intensity of the image at the point (x, y) on the image plane.

I (xylem) takes non-negative values assume the image is bounded by a rectangle [0, a] ×[0, b]I: [0, a] × [0, b] → [0, info)

Color image:

It can be represented by three functions, R (xylem) for red, G (xylem) for green *and* B (xylem) for blue.

An image may be continuous with respect to the x and y coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate's values is called sampling. Digitizing the amplitude values is called quantization.

Coordinate convention:

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image f(x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size M X N. The values of the coordinates (xylem) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates.

In many image processing books, the image origin is defined to be at (xylem)=(0,0).The next coordinate values along the first row of the image are (xylem)=(0,1).It is important to keep in mind that the notation (0,1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Following figure shows the coordinate convention. Note that x ranges from 0 to M-1 and y from 0 to N-1 in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (xylem) the toolbox uses the notation (race) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate topples, (alb), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentation refers to the coordinates. Less frequently the toolbox also employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refers to rows. This is the opposite of our use of variables x and y.

Image as Matrices:

The preceding discussion leads to the following representation for a digitized image function:

$$f\ (\text{xylem}) = \begin{matrix} f(0,0) & f(0,1) & \dots\dots\dots.. & f(0,N-1) \\ f(1,0) & f(1,1) & \dots\dots\dots\dots & f(1,N-1) \\ . & . & & . \\ . & . & & . \\ f(M-1,0) & f(M-1,1) & \dots\dots\dots\dots & f(M-1,N-1) \end{matrix}$$

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements.

A digital image can be represented naturally as a MATLAB matrix:

$$f = \begin{matrix} f(1,1) & f(1,2) & \dots\dots. & f(1,N) \\ f(2,1) & f(2,2) & \dots\dots.. & f(2,N) \\ f(M,1) & f(M,2) & \dots\dots. & f(M,N) \end{matrix}$$

Where f (1,1) = f(0,0) (note the use of a monoscope font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation f(p ,q) denotes the element located in row p and the column q. For example f(6,2) is the element in the sixth row and second column of the matrix f. Typically we use the letters M and N respectively to denote the number of rows and columns in a matrix. A 1xN matrix is called a row vector whereas an Mx1 matrix is called a column vector. A 1x1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using mono-scope characters. We use conventional Roman, italic notation such as f(x ,y), for mathematical expressions

Reading Images:

Images are read into the MATLAB environment using function imread whose syntax is

Imread (‘filename’)

Format name	**Description**	**recognized extension**
TIFF	Tagged Image File Format	.tif, .tiff
JPEG	Joint Photograph Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	.xwd

Here filename is a spring containing the complete of the image file(including any applicable extension).For example the command line

```
>> f   = imread ('8. jpg');
```

Reads the JPEG (above table) image chestxray into image array f. Note the use of single quotes (') to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output If a semicolon is not included. MATLAB displays the results of the operation(s) specified in that line. The prompt symbol (>>) designates the beginning of a command line, as it appears in the MATLAB command window.

Data Classes:

Although we work with integers coordinates the values of pixels themselves are not restricted to be integers in MATLAB. Table above list various data classes supported by MATLAB and IPT are representing pixels values. The first eight entries in the table are refers to as numeric data classes. The ninth entry is the char class and, as shown, the last entry is referred to as logical data class.

All numeric computations in MATLAB are done in double quantities, so this is also a frequent data class encounter in image processing applications. Class unit 8 also is encountered frequently, especially when reading data from storages devices, as 8 bit images are most common representations found in practice. These two data classes, classes logical, and, to a lesser degree, class unit 16 constitute the primary data classes on which we focus. Many ipt functions however support all the data classes listed in table. Data class double requires 8 bytes to represent a number uint8 and int 8 require one byte each, uint16 and int16 requires 2bytes and unit 32.

Name	**Description**
Double	Double _ precision, floating_ point numbers the Approximate.
Uint8	unsigned 8_bit integers in the range [0,255] (1byte per Element).
Uint16	unsigned 16_bit integers in the range [0, 65535] (2byte per element).

Uint 32 unsigned 32_bit integers in the range [0, 4294967295](4 bytes per element). Int8 signed 8_bit integers in the range [-128,127] 1 byte per element)

Int 16 signed 16_byte integers in the range [32768, 32767] (2 bytes per element).

Int 32 Signed 32_byte integers in the range [-2147483648, 21474833647] (4 byte per element).

Single single _precision floating _point numbers with values

In the approximate range (4 bytes per elements)

Char characters (2 bytes per elements).

Logical values are 0 to 1 (1byte per element).

Int 32 and single required 4 bytes each. The char data class holds characters in Unicode representation. A character string is merely a 1*n array of characters logical array contains only the values 0 to 1,with each element being stored in memory using function logical or by using relational operators.

Image Types:

The toolbox supports four types of images:

1 .Intensity images;

2. Binary images;

3. Indexed images;

4. R G B images.

Most monochrome image processing operations are carried out using binary or intensity images, so our initial focus is on these two image types. Indexed and RGB colour images.

Intensity Images:

An intensity image is a data matrix whose values have been scaled to represent intentions.

When the elements of an intensity image are of class unit8, or class unit 16, they have integer values in the range [0,255] and [0, 65535], respectively. If the image is of class double, the values are floating point numbers. Values of scaled, double intensity images are in the range [0, 1] by convention.

Binary Images:

Binary images have a very specific meaning in MATLAB.A binary image is a logical array 0s and1s.Thus, an array of 0s and 1s whose values are of data class, say unit8, is not considered as a binary image in MATLAB .A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, we create an array B using the statement.

B=logical (A)

If A contains elements other than 0s and 1s.Use of the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.

Using relational and logical operators also creates logical arrays.

To test if an array is logical we use the I logical function: islogical(c).

If c is a logical array, this function returns a 1.Otherwise returns a 0. Logical array can be converted to numeric arrays using the data class conversion functions.

Indexed Images:

An indexed image has two components:

A data matrix integer, x

A color map matrix, map

Matrix map is an m*3 arrays of class double containing floating point values in the range [0, 1].The length m of the map are equal to the number of colors it defines. Each row of map specifies the red, green and blue components of a single color. An indexed images uses "direct mapping" of pixel intensity values color map values. The color of each pixel is determined by using the corresponding value the integer matrix x as a pointer in to map. If x is of class double ,then all of its components with values less than or equal to 1 point to the first row in map, all components with value 2 point to the second row and so on. If x is of class units or unit 16, then all components value 0 point to the first row in map, all components with value 1 point to the second and so on.

RGB Image:

An RGB color image is an M*N*3 array of color pixels where each color pixel is triplet corresponding to the red, green and blue components of an RGB image, at a specific spatial location. An RGB image may be viewed as "stack" of three gray scale images that when fed in to the red, green and blue inputs of a color monitor

Produce a color image on the screen. Convention the three images forming an RGB color image are referred to as the red, green and blue components images. The data class of the components images determines their range of values. If an RGB image is of class double the range of values is [0, 1].

Similarly the range of values is [0,255] or [0, 65535].For RGB images of class units or unit 16 respectively. The number of bits use to represents the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8bit image, the corresponding RGB image is said to be 24 bits deep.

Generally, the number of bits in all component images is the same. In this case the number of possible color in an RGB image is (2^b) ^3, where b is a number of bits in each component image. For the 8bit case the number is 16,777,216 colors

CHAPTER 4

SOFTWARE INTRODUTION:

4.1. Introduction to MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

4.2 The MATLAB system:

The MATLAB system consists of five main parts

- **Development Environment**:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library**:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language**:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

- **Graphics:**

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

4.3 GRAPHICAL USER INTERFACE (GUI):

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

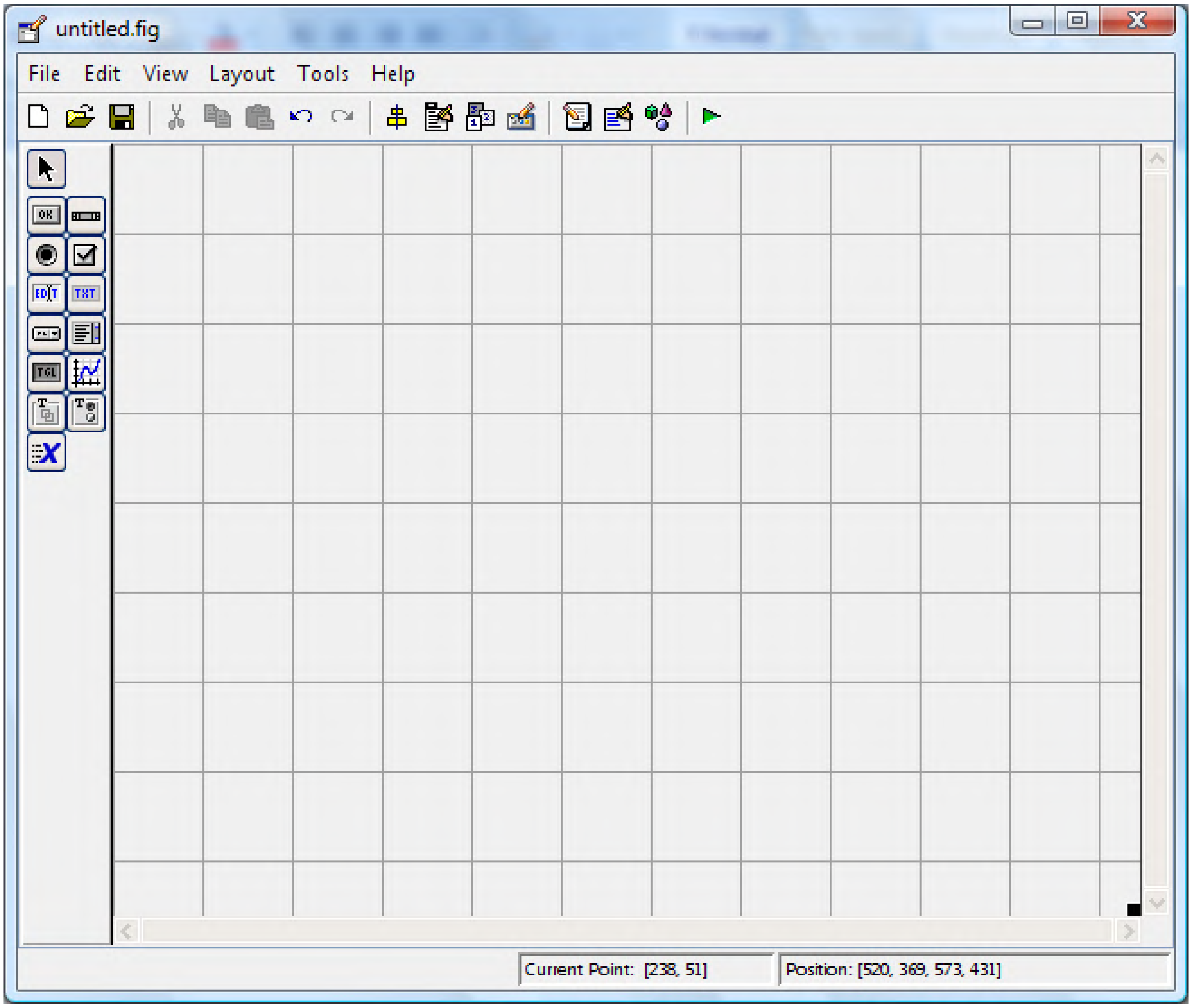
- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.

- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type

guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted, GUIDE opens a new (i.e., blank) window.



A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

4.4 Getting Started

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

4.4.1 Introduction - describes the components of the MATLAB system.

4.4.2 Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

4.4.3 Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

4.4.4 Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

4.4.5 Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

4.5 DEVELOPMENT ENVIRONMENT

4.5.1Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

4.5.2 Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

4.5.3 Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

4.5.4 MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

4.5.5 Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type helpbrowser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to runMATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

4.6 MANIPULATING MATRICES

4.6.1 **Entering Matrices**

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Dürer's matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

MATLAB displays the matrix you just entered.

```
A =

   16     3     2    13

    5    10    11     8

    9     6     7    12

    4    15    14     1
```

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

4.6.2 Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

4.6.3 Operators

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

4.6.4 Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.14159265...
I	Imaginary unit, √-1
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

4.7 GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

4.7.1 Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects.

You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

4.7.2 GUI Development Environment

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of its

children (uicontrols and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the

callbacks, which are defined as subfunctions. This M-file is referred to as the

application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

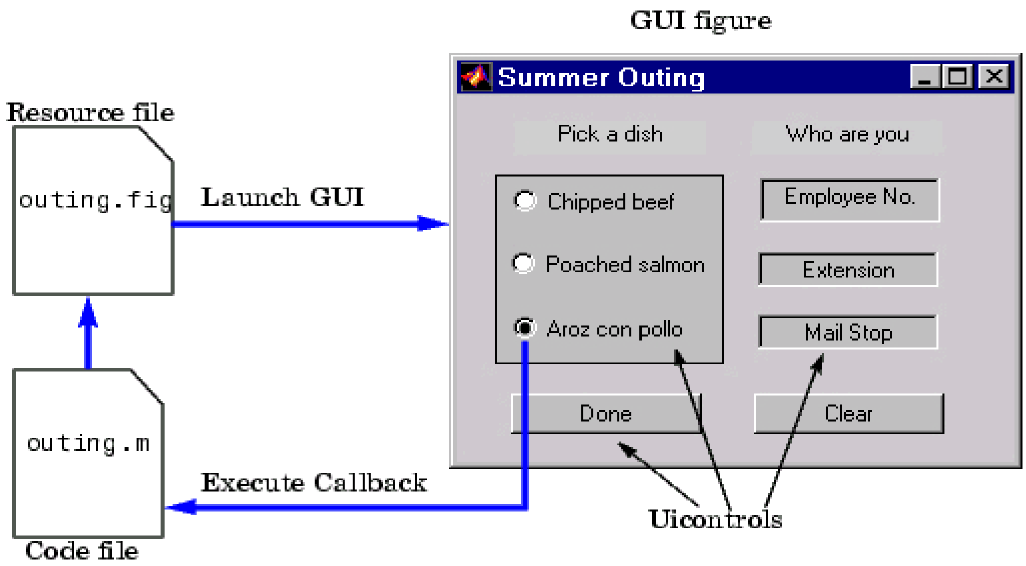


FIG 3.7.2 graphical user blocks

4.7.3 Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

4.7.4 Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques

used in the application M-file.

Application Examples - a collection of examples that illustrate techniques

which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

4.7.5 User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders

- Toggle Buttons
- Frames
- Radio Buttons
- Listboxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)

button_state = get(h,'Value');

if button_state == get(h,'Max')

   % toggle button is pressed

elseif button_state == get(h,'Min')

   % toggle button is not pressed

end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);

a(:,:,2) = rand(16,128);

a(:,:,3) = rand(16,128);

set(h,'CData',a)
```

Radio Buttons

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)

set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)

off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];
```

```
mutual_exclude(off)

% Continue with callback

    .

    .
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)

if (get(h,'Value') == get(h,'Max'))

   % then checkbox is checked-take approriate action

else

   % checkbox is not checked-take approriate action

end
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)

user_string = get(h,'string');

% proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)

user_entry = str2double(get(h,'string'));

if isnan(user_entry)

   errordlg('You must enter a numeric value','Bad Input','modal')

end

% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menubar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menubar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Non integer values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

Listboxes differentiate between single and double clicks on an item and set the figure SelectionType property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure SelectionType property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

switch val

case 1

% The user selected the first item

case 2

% The user selected the second item

% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

string_list = get(h,'String');

selected_string = string_list{val}; % convert from cell array to string

% etc.

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is

grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes ButtonDownFcn property to define the callback.

4.7.6 Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

```
axes(handles.axes1)
```

makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axes. See GUI with Multiple Axes for and example that uses two axes.

Figure

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control.

Chapter 5

PROPOSED METHOD

underwater image restoration using depth map estimation. Image blurriness is used as a before acquiring a depth map. As depth increases, image blurriness also gets increased. Here backlight is retrieved from the largest blurriness region and lowest variance region. A backlight is estimated from the top brightest pixel. If artificial lighting is used for illumination, foreground objects appear to be brighter than background objects. So, selecting the brightest pixels may lead to erroneous results. Then substituting backlight and depth map in IFM scene radiance is restored. Fig.1 shows the block diagram of the proposed method.

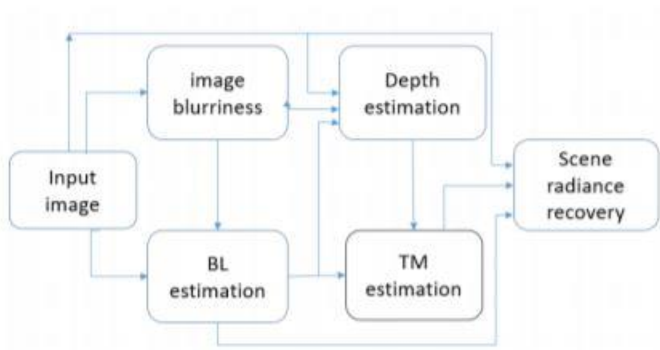


Fig.1. Block Diagram

A. Image Blurriness Estimation:

Image blurriness is due to scattering by suspended particles, minerals, rocks in the water. There is a linear relationship between blurriness and depth. Thus estimating blurriness in the image gives depth information. There are 5 types of image blurriness measuring techniques. In the first category, image blurriness is found by measuring the energy of the image. Blurring can smooth edges of the image and thus decreases high-frequency coefficients energy, thus a comparison of the energy of original and degraded image gives blurriness metric. The second category considers the edge of the image since blurriness may convert the sharp edges in the image into a smooth one. The edges and its width are obtained from vertical/horizontal gradients. The third category involves pixel intensity distribution. This method considers, a sharper image has a greater value of entropy and variance. The fourth category blurriness evaluation includes local gradient measures. Singular value decomposition is a local gradient measure used for blurriness estimation.

The fifth category uses a combination of the other 4 categories. But these methods are susceptible to noise since noise may increase high-frequency content in an image. In underwater image restoration, a reference image is not available thus above mentioned methods for blurriness calculation cannot be used. First, the input image is converted into YCbCr color space Y represents luminance and Cb, Cr represents chrominance values. Blurriness is calculated using the luminance component only, rather than considering chrominance values. Since it is perceptually more representative, it easy to tell blurry and sharp areas based on Y. Sharp edges in the image indicate blurriness is less. Then calculates the difference between the luminance component and Gaussian filtered version of that component. Gaussian filtering is a lowpass filter it removes high-frequency content such as edges or noise and retains low frequency information. This Difference contains sharp objects and fine details. Then a maximum filter is applied to this difference. This filter calculates the maximum value within the window and this window is slid over. A window size of 7 to 31 is suitable for any image size. Here window size equal to 7 is used. The output image is reconstructed by hole filling. The output image contains only sharp details and the blurred regions are removed. Thus in the blurriness map, small value means the corresponding pixels in the image are blurry, large value means they are not blurry. So sharp edges would have large values and a smooth region would have small values. However, a smooth region encompassed by sharp edges would be considered not blurry since they all belong to a single object. Fig.2 shows the image blurriness

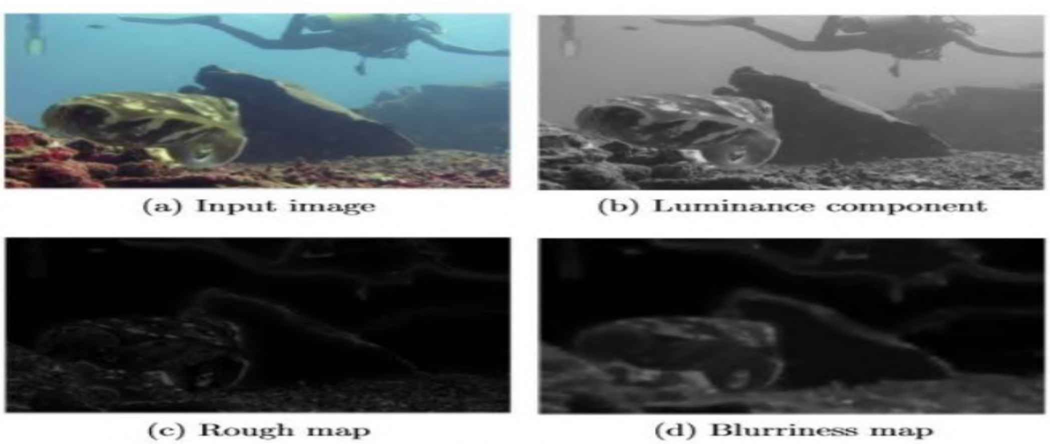
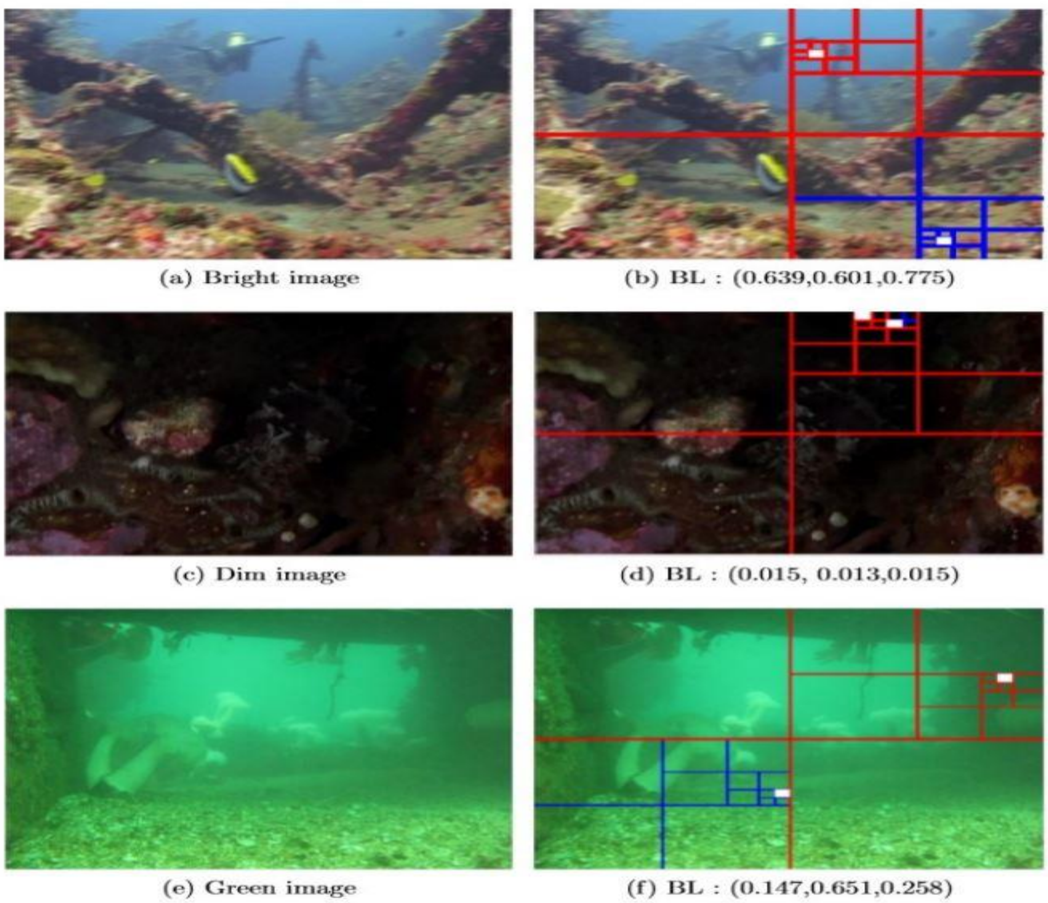


Fig.2. Image Blurriness Estimation

B. Backlight Estimation:

The backlight is due to the scattering of light by the particles in water. It adds a hazy layer and reduces image contrast. To overcome the drawbacks of backlight estimation methods (in the case of artificial lighting or brighter foreground object) this method considers variance and blurriness in addition to 1 percentage of top brightest pixels. Here largest blurriness and lowest variance regions are considered, these regions are obtained using quadtree subdivision. In low variance region pixel values do not change dramatically. Thus it is more likely to find backlight. Blur regions always have low variance but low variance does not always infer a region is blurry. Thus low variance and largest blurriness criteria are not interchangeable. The backlight has a dependence on depth because the scattering of ambient light increases as goes deeper into water. To take this dependence, backlight estimation considers the largest blurriness region.



(a) Bright image

(b) BL : (0.639,0.601,0.775)

(c) Dim image

(d) BL : (0.015, 0.013,0.015)

(e) Green image

(f) BL : (0.147,0.651,0.258)

Fig.3. Backlight Estimation

The image is first divided into four quadrants equally according to blurriness or variance. Then the quadrant with the lowest variance (in the case of blurriness region with the largest value is chosen.

Variance within a quadrant is obtained by the averaging variance of individual pixels. The quadrant with the lowest variance is further divided into four. This process is repeated until the quadrant size got lesser than a predefined threshold. The threshold value must lie between 0 and 1. From the largest blurriness region, lowest variance region, and top brightest pixels, the maximum, and minimum of the backlight are obtained. It is not able to find backlight anywhere in the image so maximum and minimum values are separately obtained for each color channel. Then calculates the ratio of pixels with intensity greater than a threshold (here chosen 0.5) to the total number of pixel values. If the image is taken under sufficient lighting this ratio would have larger value otherwise it would be a smaller value. When the percentage of bright pixels is high then the estimated backlight is bright is more suitable thus the maximum value of backlight is chosen as estimated backlight. If the image is taken without sufficient lighting (smaller ratio) then the minimum value of backlight is chosen as estimated backlight. In between these extreme conditions, the backlight is calculated as the combination of these maximum and minimum values. The figure shows backlight estimation where the blue line indicates variance and the red line indicates blurriness. The white portion indicates the lowest variance and largest blurriness region. Fig3.shows backlight estimation of bright, dim, and green input image. A bright image has larger backlight values in all color channels (RGB), the obtained BL for fig.3. a,b is [0.639,0.601,0.775]. For the dim input image, all three channels have lower backlight values and obtained backlight for fig.3. c,d is [0.015, 0.013,0.015]. Green input image has larger backlight value in the green channel only other channels have smaller values. Obtained backlight for fig.3. e,f is [0.147, 0.651,0.258].

C. Depth Estimation:

Attenuation of light in underwater depends upon light wavelength and distance it travels. Here depth is estimated by combining 3 methods. The first two methods are based on different wavelength have different attenuation rates and the third method is based on image blurriness. 1) Using red channel content: Red light has the longest wavelength in the visible spectrum hence it attenuates at a higher rate than other colors. As the light reaches greater depth most of the red channel content may be lost. Thus a measure of red channel content is used to obtain an estimate of depth. If the image has a considerable amount of red channel value it indicates that an image is captured at a lower depth. When the image I c (y) is captured at a larger depth the red channel information is almost lost.

Red channel content is,

$$R(x) = \max y \varepsilon \omega(x) I\ c\ (y) \qquad (2)$$

where ω(x) is a window of size 7,centered about x and c ε{r,g,b}. Then depth map is,

$$dR = 1 - K(R) \qquad (3)$$

Since three diff erent depth maps are combined, for them to be in the same range [0, 1] stretching function K is used. It is given by

$$K(L) = L - Lmin \,/\, Lmax - Lmin \qquad (4)$$

2)Using 3color channel: Difference between attenuation of different color channels (red, green, blue) can be used as a prior to calculate depth map since light undergoes attenuation which depends on color. Red light has a larger attenuation constant than green and blue light. For this depth estimation method, the difference between the maximum value of green and blue channels and that of the red channel is calculated. When the difference is larger it indicates that red channel intensity is low, the image is distorted badly and depth value is high. If this blue-green channel has a smaller value the red channel is less attenuated and depth is low. The prior value is,

$$BGR = \max y\varepsilon\omega(x),c\varepsilon(g,b)\ I\ c\ (y) - \max y\varepsilon\omega(x),c\varepsilon r\ I\ c\ (y) \ (5)$$

When the prior value increases depth also increases. Then the depth map is:

$$dBGR = K(BGR) \qquad (6)$$

3) Using image blurriness: As the image is more blurred, indicates that the image is from more depth and vice-versa. If image blurriness value is small, it's blurry and large value indicates the image is sharp. Thus, the depth map is; dB = 1 – K(Blurriness map) (7) When there is insufficient lighting, method of using red channel content is used. If there is an intense backlight, red content in the image is retained and this method is failed. In this case, the second method using 3 color channels is proposed. If there is a brighter backlight, all the 3 channels have larger values. Whether the backlight is dim or bright, when image lacks red content, the third method alone is used. As a combination of these three depth maps, the final depth map is processed. Fig.4 and Fig.5 show the depth map estimation.

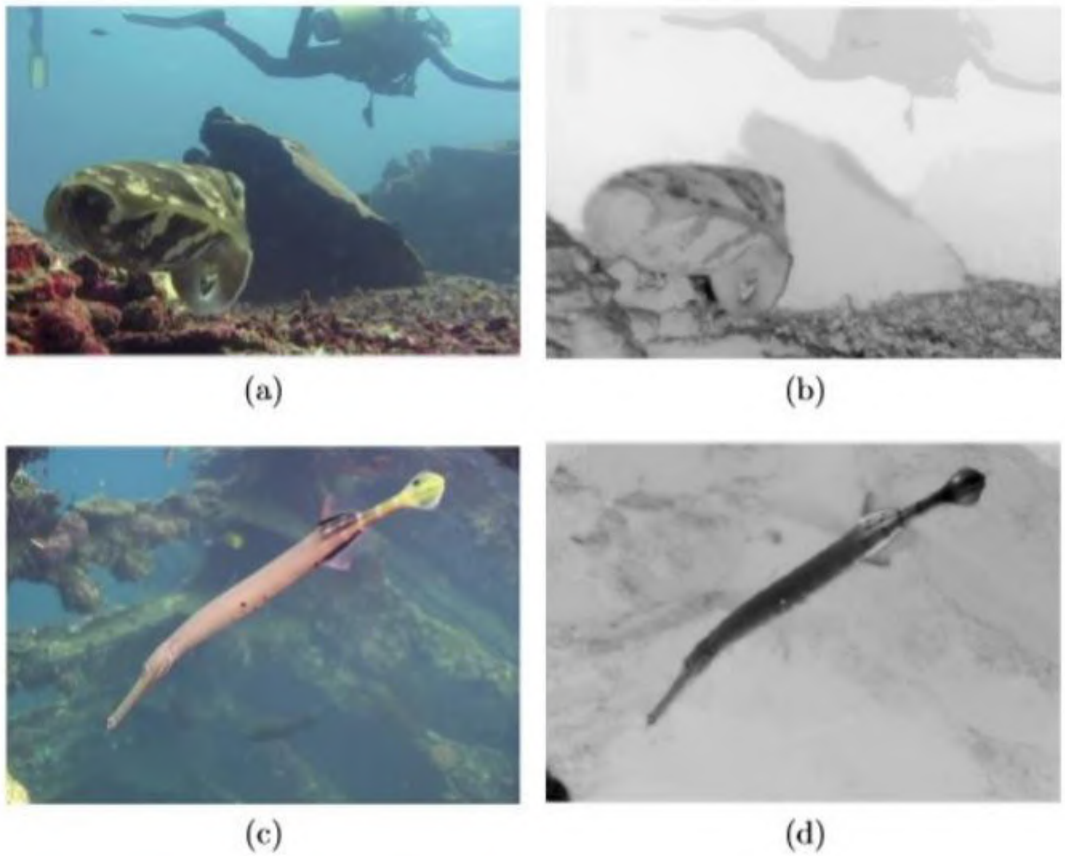


Fig.4. Depth Map Estimation

D. Transmission Map Estimation:

Transmission map is the part of the light that is not scattered or absorbed and finally reaches the camera. According to Beer-Lambert law light is exponentially attenuated with distance when it travels through a medium. $t_c(x) = \exp(-\beta_c\, d(x))$ (8) where $d(x)$ is scene depth and β is attenuation coefficient c represent RGB color channels. Transmission map is separately obtained for each color channel. Underwater image restoration using a single image makes use of the assumption that attenuation is wavelength independent (or color independent). But in underwater red light decays faster than green light and blue light. In this method wavelength-dependent attenuation is considered. The standard wavelength of red, green and blue are 620 nm, 540 nm, 450 nm respectively. Attenuation constant is proportional to wavelength and inversely proportional to the backlight. Using this relation the attenuation coefficient is:

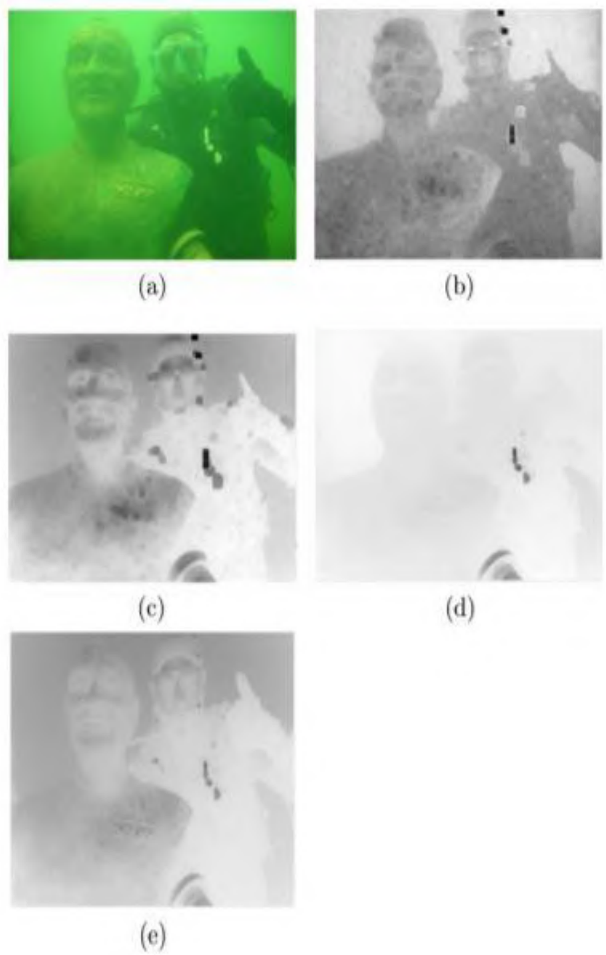
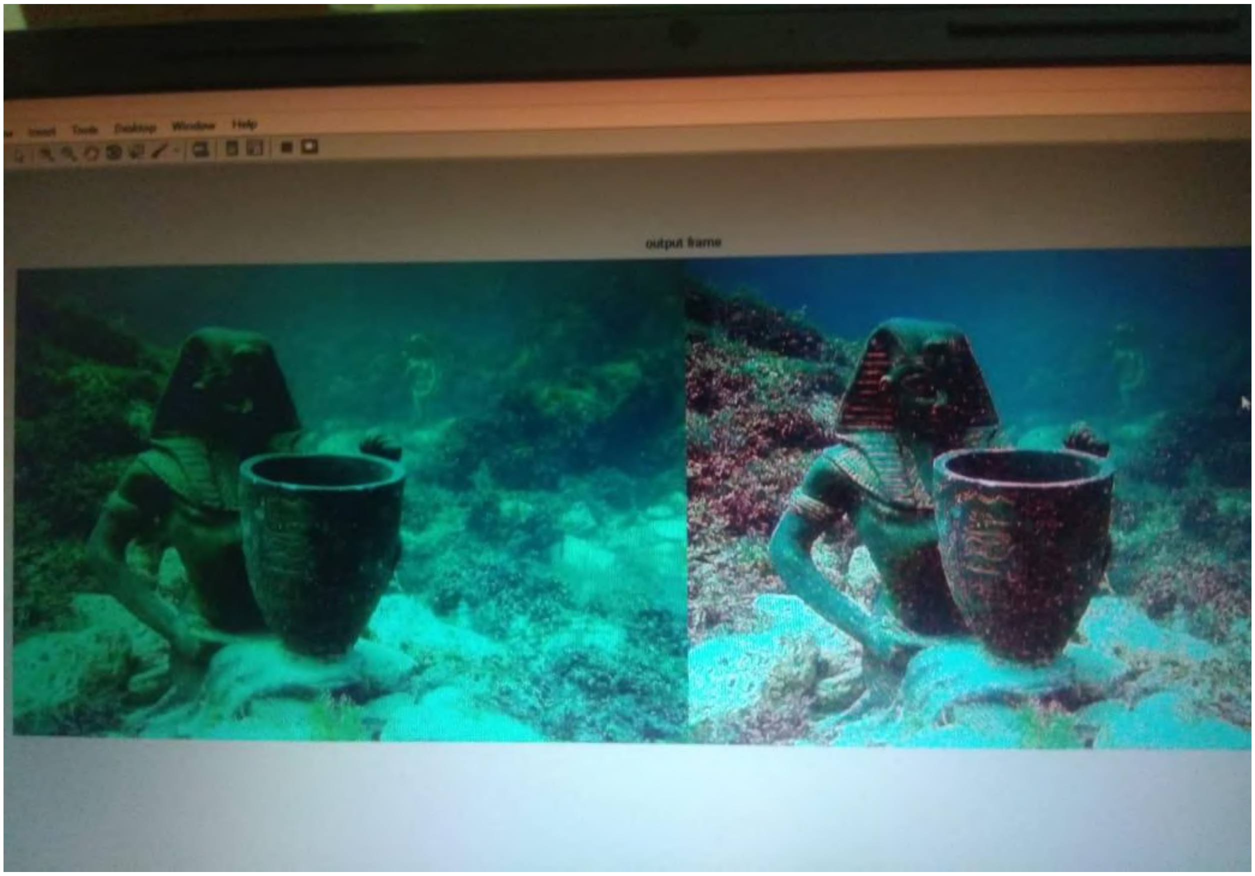


Fig.5: Depth map estimation (a) Input image (b) Depth map using red content (c) Depth map using 3 color channels (d) Depth map using image blurriness (e) Final depth map

Chapter 6

Experiment results



Chapter 7

CONCLUSIONS

Underwater image restoration is a difficult task as there is a shortage of reference images and a changing underwater environment. Existing methods need specialized hardware or multiple images of the same scene, and is inadequate for real time and video applications. Algorithms using a single image also deals with several problems due to the wavelength dependency of light and its attenuation. Here images are restored using depth estimation based on image blurriness and light absorption. Image blurriness is linearly dependent on depth. Thus image blurriness is obtained to provide a measure of a depth map. This technique utilizes the largest blurriness and lowest variance region to estimate backlight and thus avoids faults of bright foreground objects. Finally, the image is restored by providing estimated parameters in the image formation model. It requires only a single image of a scene. Quality of restored images is evaluated using non-reference image quality measures and works well with many underwater images without numerous images dataset of one scene itself. Therefore, this technique provides better restoration results compared to other existing methods.

REFERENCES

[1] Yoav Y. Schechner and Nir Karpel "Recovery of underwater visibility and structure by polarization analysis" IEEE Journal of Oceanic Engineering, 2006.

[2] Duo Min He, Gerald G L Seet "Divergent beam lidar imaging in turbid water" Elsevier Optics and Lasers in Engineering, 2010.

[3] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior" Computer Vision and Pattern Recognition, IEEE Computer Society Conference, 2009.

[4] Adrian Galdran, David Pardo, Artzai Picon, Aitor Alvarez Gila "Automatic red channel underwater image restoration" Journal on visual Communication Research gate, 2015.

[5] N. Carlevaris-Bianco, A. Mohan, and R. M. Eustice, "Initial results in underwater single image dehazing" in Proc. IEEE Oceans, 2010.

[6] Shuai Fang, Rong Deng "Effective Single Underwater Image Enhancement by Fusion" Journal of Computers 2012.

[7] S Emberton, L Chittka, and A Cavallaro, "Hierarchical rank based veiling light estimation for underwater dehazing," IEEE Transaction Image Processing, 2017.

[8] K Iqbal, M Odetayo, R A Salam, A James, and A Z H Talib, "Enhancing the Low Quality Images using Unsupervised Colour Correction Method", IEEE International Conference on Systems Man and Cybernetics, 2013.

[9] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain" IEEE Transaction Image Processing, 2012.

[10] WenhaoZhang,GeLi,ZhenqiangYing" Anew under water image enhancing method via color correction and illumination adjustment" IEEE Transaction on image processing, 2017.

[11] Taiebeh Askari Javaran, Hamid Hassanpour, Vahid Abolghasemi, "A noise immune no reference metric for estimating blurriness value of an image", Journal on Image Processing and Data Mining Elseiver, 2016.

[12] John Y. Chiang, Ying-Ching Chen, and Yung-Fu Chen, "Underwater image enhancement using wavelength compensation and image dehazing", IEEE conference on image processing, 2011.

[13] A. Levin, D. Lischinski, and Y. Weiss, "A closed form solution to natural image matting" IEEE Transaction on Pattern Anal. Mach. Intell., 2008.

[14] Yong Shi, Arialdis Japa, “Multi-dimensional Analysis on Data Sets for Information Retreival”. International Conference on Intelligent Sustainable Systems (ICISS 2017). December 7-8,2017, Tamil Nadu, India.

[15] Manoharan.S. (2019,”Image Detection, Classification and recognition for leak detection in automobiles”, Journal of Innovative Image Processing(JIIP),1(02),61-70.)

[16] K. Panetta, C. Gao, and S. Agaian, "Human visual system inspired underwater image quality measures", IEEE Journal Oceanic Eng., 2016.

ROAD MAP TO 6G: AI EMPOWERED WIRELESS NETWORK

A PROJECT REPORT

Submitted by

1. A. Nuthan Reddy (17K81A04C2)

2. B. Sai Krishna (17K81A04D2)

3. A. Sreeja Reddy (17K81A04G5)

in partial fulfillment for the award of the

degree of

BACHELOR OF TECHNOLOGY

IN

Department of Electronics and Communication Engineering

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

MAY 2021

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100 NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

BONAFIDE CERTIFICATE

Certified that this project report titled "Image Forensic for Digital Image Copy Move Forgery Detection" is a *bona fide* work of by A. Nuthan Reddy (17K81A04C2), B. Sai Krishna (17K81A04D0), A. Sreeja Reddy (17K81A04G5), who carried out the work under my supervision, for the partial fulfilment of the requirements for the award of the degree of *Bachelor of Technology* in *electronics and communication engineering.* Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion.

INTERNAL GUIDE

Mr. S. P. Manikanta

Associate Professor

HEAD OF THE DEPARTMENT ECE

Dr. B. Hari Krishna,

EXTERNAL EXAMINER

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **B.SAI KRISHNA** WITH ROLL NO.**17K81A04D0, NUTHAN REDDY** WITH ROLL NO.**17K81A04C2**, **SREEJA REDDY** WITH ROLL NO.**17K81A04G5**, OF B. TECH – IV YEAR**, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED **"THE ROAD MAP TO 6G: AI EMPOWERED WIRELESS NETWORK"** AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS LASYA
IT SOLUTIONS PVT LTD.

DECLARATION

We declare that this project report titled “**Roadmap to 6G: AI powered wireless network”** submitted in partial fulfillment of the degree of B. Tech in electronics and communication engineering, record of original work carried out by us under the guidance and supervision of Mr. S. P. Manikanta, and has not formed the name basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

1. A. Nuthan Reddy (17K81A04C2)	2. B. Sai Krishna (17K81A04D0)	3. P. Sreeja Reddy (17K81A04G5)

Place:

Date:

ACKNOWLEDGEMENTS

While bringing out this project report to its final form, we came across a number of people whose contributions in various ways helped my field of research and they deserve special thanks. It is a pleasure to convey our gratitude to all of them.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr. Santosh Kumar Patra**, **our beloved Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to Dr. B. Hari Krishna **Head &Professor**, the Dept. of electronics and communication engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept. project coordinator** Dr. A. Anand for his involvement with originality has nourished my intellectual maturity that will help me for a long time to come. We are highly grateful and obliged to **our class Project Coordinator** Mr. K. Nishakar, **Professor in ECE** and our **project guide** Mr. S. P. Manikanta, **Associate Professor in ECE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

Finally, I am deeply indebted to my mother, my father, my siblings and to my family members for their moral support and continuous encouragement while carrying out this study. I dedicate this thesis to almighty.

.

1. A. Nuthan Reddy (17K81A04C2)
2. B. Sai Krishna (17K81A04D0)
3. P. Sreeja Reddy (17K81A04G5)

ABSTRACT

The recent upsurge of diversified mobile applications, especially those supported by AI, is spurring heated discussions on the future evolution of wireless communications. While 5G is being deployed around the world, efforts from industry and academia have started to look beyond 5G and conceptualize 6G. We envision 6G to undergo an unprecedented transformation that will make it substantially different from the previous generations of wireless cellular systems. In particular, 6G will go beyond mobile Internet and will be required to support ubiquitous AI services from the core to the end devices of the network. Meanwhile, AI will play a critical role in designing and optimizing 6G architectures, protocols, and operations. In this article, we discuss potential technologies for 6G to enable mobile AI applications, as well as AI-enabled methodologies for 6G network design and optimization. Key trends in the evolution to 6G will also be discussed.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO THE PROJECT:

The wireless communications industry is one of the few industry sectors that have kept a fast growing trend with creative features for a number of decades. The current 4G LTE networks have led to the thriving of mobile Internet, enabling various innovative applications, such as mobile shopping and payment, smart home/city, mobile gaming, and so on. The great success of mobile Internet has in turn been a driving force behind the evolution of wireless technologies. The upcoming 5G network will support a wide range of services, including eMBB (enhanced mobile broadband), uRLLC (ultra-reliable and low-latency communications), and mMTC (massive machinetype communications).

While 5G is still at an initial stage, to maintain the sustainability and competitiveness of wireless communication systems, it is time for both the industry and academia to think about what 6G will be. There are already initiatives describing the roadmap toward 6G along with the emerging trends and requirements, as well as various enabling techniques and architectures. In contrast to previous generations, 6G will be transformative and will revolutionize the wireless evolution from "connected things" to "connected intelligence" with more stringent requirements specified as follows:

- Very high data rates, up to 1 Tb/s;
- Very high energy efficiency, with the ability to support battery-free IoT devices;
- Massive low-latency control (less than 1 msec end-to-end latency);
- Very broad frequency bands (e.g., 73GHz140GHz and 1THz-3THz [4]);
- Ubiquitous always-on broadband global network coverage by integrating terrestrial wireless with satellite systems;

• Connected intelligence with machine learning capability. 6G will also require the support of three new service types beyond the eMBB, uRLLC, and mMTC services, as described below. Computation Oriented Communications (COC): New smart devices call for distributed computation to enable the key functionalities, such as federated learning.

Instead of targeting classical quality of service (QoS) provisioning, CoC will flexibly choose an operating point in the rate-latency-reliability space depending on the availability of various communications resources to achieve a certain computational accuracy. Contextually Agile eMBB Communications (CAeC): The provision of 6G eMBB services is expected to be more agile and adaptive to the network context, including the communication network context such as link congestion and network topology; the physical environment context such as surrounding location and mobility; and the social network context such as social neighborhood and sentiments. Event Defined uRLLC (EDuRLLC): In contrast to the 5G uRLLC application scenario where redundant resources are in place to offset many uncertainties, 6G will need to support uRLLC in extreme or emergency events with spatially and temporally changing device densities, traffic patterns, and spectrum and infrastructure availability. The above service types represent emerging driving applications of 6G. They can hardly be offered by 5G, not only because of their stringent requirements for higher data rates, lower latency, denser connection, and so on, but also due to their extreme demand for new performance metrics that have never been considered in 5G, for example, delay jitter, context awareness, UAV/ satellite compatibility, and so on. Inspired by these trends, in this article, we attempt to conceptualize 6G as an intelligent information system that is both driven by and a driver of the modern AI technologies. A roadmap for 6G is depicted in Fig. 1, which is plotted based on the strategic plans of various standards bodies and is also projected based on the 5G status. Key performance indicators (KPIs) and service types are also illustrated. Meanwhile, a potential network architecture for 6G is shown in Fig. 2. We envision that AI will greatly enhance the situational awareness of the network operators, and enable closed-loop optimization to support the new service types as mentioned above. As such, 6G will unleash the full potential of mobile communications, computing, and control in a host of exciting applications, including smart cities, autonomous driving, UAVs , seamless virtual and augmented reality, Internet of Vehicles, space-air-ground integrated networks and much more.

Wireless Sensor Networks (WSN) consists of a large number of miniature size, low-power, smart sensor nodes interfacing with one another and deployed over a certain inaccessible geographical locale having inadequate power and storage space. Energy efficiency is a foremost concern in WSN.To maximize the network lifetime, one possible solution is to harvest energy from the ambient boundless accessible energy sources in the environment nearby to the sensor nodes. The use of energy harvesting (EH) nodes as cooperative relays is a promising and budding solution in WSN. Energy harvest from one source is not sufficient to congregate the power requirement of sensor nodes. In this work, in order to improve the network lifetime electromagnetic radiation energy harvesting (EMR EH) method has been proposed. Communication is carried out by signal cooperation through relays in order to assist the sources to exchange their data for longer time periods. The current slot's transmission is based on transmitter knowledge about the present channel signal to noise ratio (SNR), past harvested energy and present energy stored in the battery. In order to solve the low power problem while maximizing the lifetime of wireless sensor nodes power management schemes like water filling energy harvesting strategies is included. The network performance metrics such as the probability of successful data exchange and the network lifetime gain has been derived and simulated using MATLAB.Simulations results illustrate that proposed technique afford enhanced result compared to existing energy harvesting schemes.A WSN is capable of gathering, data processing and communicating through nodes which have fascinated much interest during the previous decade in forming the notion of smart spaces. It consists of spatially sprinkled autonomous sensors to examine physical or environmental conditions, and to cooperatively overtake their data through the network to a main location. The swift development of wireless networks devices comprise their own batteries to deliver energy. For each message sent or else received, every communication and computation performed in the operation drains the battery, unique concern is mandatory in the utilization of power.

Accomplishing sensor lifetime of numerous years signifies one of the utmost challenges for designers. Hence the project present research guidelines for alleviating the energy problems in growth of WSN, including wireless sensor architecture power management techniques, and environmental energy harvesting approaches. The rest of the manuscript is organised as follows. Section II gives the system model adopted. Section III introduces a brief overview of techniques used to improve the network lifetime.

The latter technique exploits the low power modes of wireless transceivers, whose components can be switched off for energy saving. When the node is in a low power (or “sleep”) mode its consumption is significantly lower than when the transceiver is on.

However, when asleep the node cannot transmit or receive packets. The duty cycle expresses the ratio between the time when the node is on and the sum of the times when the node is on and asleep. Adopting protocols that operate at very low duty cycles is the leading type of solution for enabling long lasting WSNs.

However, this approach suffers from two main drawbacks.

1) There is an inherent tradeoff between energy efficiency (i.e., low duty cycling) and data latency.

2) Battery operated WSNs fail to provide the needed answer to the requirements of many emerging applications that demand network lifetimes of decades or more.

Battery leakage depletes batteries within a few years even if they are seldom used For these reasons recent research on long-lasting WSNs is taking a different approach, proposing energy harvesters combined with the use of rechargeable batteries and super capacitors (for energy storage) as the key enabler to “perpetual” WSN operations. Energy Harvesting-based WSNs (EHWSNs) are the result of endowing WSN nodes with the capability of extracting energy from the surrounding environment. Energy harvesting can exploit different sources of energy, such as solar power, wind, mechanical vibrations, temperature variations, magnetic fields, etc.

Continuously providing energy, and storing it for future use, energy harvesting subsystems enable WSN nodes to last potentially forever. This chapter explores the opportunities and challenges of EHWSNs, explaining why the design of protocol stacks for traditional WSNs has to be radically revisited. We start by describing the architecture of a EHWSN node, and especially that of its energy subsystem . We then present the various forms of energy that are available and ways for harvesting them , Models for predicting availability of wind and solar energy are described in Section 1.4. We then survey task allocation, MAC and routing protocols proposed so far for EHWSNs.

A Wireless Sensor Network (WSN) is composed of a large number of small dispersed devices known as sensor nodes that are closely deployed in the environment sense changes. The location of nodes in a sensor network may not be predetermined and usually have a unique sensor node called Base Station (BS). All member nodes will forward data to BS either directly or through multi hop transmission. A base station may be either static or dynamic sensor node and provides wireless connectivity to its users. It is usually more capable than other sensor nodes in the WSN . When sensor nodes in a WSN run out of energy, they stop working which causes the whole sensor network to fail. Therefore, the main issue of WSN is energy conservation. Hence, such protocols should be designed which use minimum energy and should be utilized during sensing, processing and transmission.

In Wireless Sensor Network, energy consumption is a vital challenge for sensor node for sensing and transmitting the data to the closest SN or the base station. The key requirement of a WSN is to minimize energy consumption and prolong network lifetime. Communication among sensor nodes stops when these sensor nodes lose their battery power.

A Challenges to Wireless Sensor Network Several challenges still need to be faced by WSNs. The main challenges and essential design limitation which affect the performance of Wireless Sensor Network are discussed here :

1) Resource limitations: Sensing nodes are bounded by energy, data processing abilities, memory and the data transmission rate to be gained.

2) Security: The Wireless Sensor Network must be secure to control illegal entities spreading false data to the sensor node or giving wrong information to the other sensor nodes and possibly causing significant damage to the sensor nodes.

3) Self-Management: After deployment of the sensor nodes, the user has less interaction with these sensor nodes with no infrastructure support or the capability to keep and repair itself. Therefore, the sensor nodes must be self-organised in a way that must be configured, work together with other sensor nodes.

4) Heterogeneity: It is a group of sensor nodes which are not identical and do not have similar capabilities, i.e. some sensor nodes are more powerful than others. Heterogeneity arises when two different sensor networks need to communicate with each other. There will be some mechanism which is required to enable efficient information exchange among these networks.

These sensor nodes are able to sense the information from one end transmitter side and process it and send it to the selected destinations. Wireless sensors have characteristics behaviors such as macro dimension, very low cost, highly reliable, flexibility to install anywhere and low energy consumption. Because of these energy saving behaviors, they could be deployed in various fields such as oil fields, military, defense, agricultural, industrial, and biomedical and many more real and non real time application domains. The sensor they could easily be used in different environments such as unmanned and critical regions.

So the cost of network is less, complicated configuration and installation for these sensors in the network connects each other, we could use them with lesser cost in comparison with traditional or commercial networks. Our research efforts focused on establishing efficient routing paths for transmitting packets from a sensor node to a sensor destination. Routes to find the best and well selected profitable way for data transmission from source node to the destination node in the network by considering different parameters (e.g. flexibility, reliability overhead, throughput, stability, consumed and wasted power, data speed, etc). Our proposed sensor parameters to increase the wireless sensor networks Lifetime. Life time is one of the most important factors of deployed wireless sensor networks. Finding the best shortest route length to reduce the transmission overhead and reduced delay time. To increase the ratio of packets in packet delivery fraction, these wireless sensor networks must be designed and used in an efficient way to optimize the sensor to reduced power consumption and increased life time of the sensor network player.

Our idea to proposed work, using probability selection Game Theory approach for WSN, optimal discovery of route in WSN is discovered. Our approach towards this proposed project, we developed network to select route based on routing and sensor nodes the same things are assumed to be the game and players of the networks respectively. Each and every node in the network we considered wants to increase their benefit in the present network. So we use a different mixed strategy model as well as network profit and loss calculation for each player that alive in the networks. In this proposed method, the destination node player node pays a recognition to the source player node for each data packet successfully reception from the source player to destination player.

Moreover, the source player pays attention of this credit to each intermediate player node or relay player node that participates in sending data packet transaction. Yet to decide, each node sustains a transmission cost for each data packet transaction to other intermediate or destination receiving node. The cost is called networks Transmission Cost or path loss and related to different type of parameters.

Each player transmits the received data packet to the next or intermediate hop with the selected player game probability, calculated by the reliability of the multiple player that playing in the network. Our enhanced parameter checked on player characteristics or behavior, e.g. link error checking, probability, duty cycle, etc. ehwsns are composed of individual nodes that in addition to sensing and wireless communications are capable of extracting energy from multiple sources and converting it into usable electrical power. In this section we describe in details the architecture of a wireless sensor node with energy harvesting capabilities, including models for the harvesting hardware and for batteries.

The system architecture of a wireless sensor node includes the following components

1) The energy harvester(s), in charge of converting external ambient or human-generated energy to electricity;

2) a power management module, that collects electrical energy from the harvester and either stores it or delivers it to the other system components for immediate usage;

3) energy storage, for conserving the harvested energy for future usage;

4) a microcontroller;

5) a radio transceiver, for transmitting and receiving information;

6) sensory equipment;

7) an A/D converter to digitize the analog signal generated by the sensors and makes it available to the microcontroller for further processing, and

8) memory to store sensed information, application related data, and code. In the next section we focus on the energy harvesting components (the energy subsystem) of a EHWSN node, describing abstractions that have been proposed for modeling them.

The general architecture of the energy subsystem of a wireless sensor node with energy harvesting capabilities is shown. The energy subsystem includes one or multiple harvesters that convert energy available from the environment to electrical energy. The energy obtained by the harvester may be used to directly supply energy to the node or it may be stored for later use. Although in some application it is possible to directly power the sensor node using the harvested energy, with no energy storage (harvest-use architecture)in general this is not a viable solution.

A more reasonable architecture enables the node to directly use the harvested energy, but also includes a storage component that acts as an energy buffer for the system, with the main purpose of accumulating and preserving the harvested energy. When the harvesting rate is greater than the current usage, the buffer component can store excess energy for later use (e.g., when harvesting opportunities do not exist), thus supporting variations in the power level emitted by the environmental source. The two alternatives commonly used for energy storage are secondary rechargeable batteries and super capacitors (also known as ultra capacitors). Super capacitors are similar to regular capacitors, but they offer very high capacitance in a small size.

They offer several advantages with respect to rechargeable batteries . First of all, super capacitors can be recharged and discharged virtually an unlimited number of times, while typical lifetimes of an electrochemical battery is less than 1000 cycles. Second, they can be charged quickly using simple charging circuits, thus reducing system complexity, and do not need full-charge or deep-discharge protection circuits. They also have higher charging and discharging efficiency than electrochemical batteries. Another additional benefit is the reduction of environmental issues related to battery disposal. Thanks to these characteristics, many platforms with harvesting capabilities use super capacitors as energy storage, either by themselves or in combination with batteries.

Other systems, instead, focus on platforms using only rechargeable batteries. Both types of storage devices deviate from ideal energy buffers in a number of ways: They have a finite size BMax and can hold a finite amount of energy; they have a charging efficiency $\eta c < 1$ and a discharging efficiency $\eta d < 1$, i.e., some energy is lost while charging and discharging the buffer, and they suffer from leakage and self-discharge, i.e., some stored energy is lost even if the buffer is not in use. Leakage and self-discharge are phenomena that affect both batteries and super capacitors. All batteries suffer from self-discharge: A cell that simply sits on the shelf, without any connection between the electrodes, experiences a reduction in its stored charge due to internal chemical reactions, at a rate depending on the cell chemistry and the temperature.

A similar phenomenon affects electrochemical super-capacitors in charged state. They suffer gradual loss of energy and reduction of the inter-plate voltage. In order to reduce the energy lost trough buffer inefficiencies, many platforms allow the node to directly use the energy harvested. In particular, if the current energy consumption is greater or equal than the energy currently harvested, then the node can use the harvested energy for its operations.

This is the most efficient way of using the environmental energy, because it is used directly and there is no energy loss. Otherwise, if the amount of energy harvested is greater than the current energy consumption, some energy is directly used to sustain the node operations, while excess energy is stored in the buffer for later use.

A super capacitor leakage is strongly variable and depends on several factors, including the capacitance value of the super capacitor, the amount of energy stored, the operating temperature, the charge duration, etc. For this reason, the leakage pattern of a particular super capacitor must often be determined experimentally .Additionally, the leakage current varies with time: It is considerably higher immediately after the super capacitor has been charged, then it decreases to a plateau. Several model for the leakage from a charged super capacitor have been proposed in the literature, modeling the leakage as a constant current, or as an exponential function of the current super capacitor voltage , or by using a polynomial approximation of its empirical leakage pattern , or, finally, by using a piecewise linear approximation of its empirical leakage pattern.

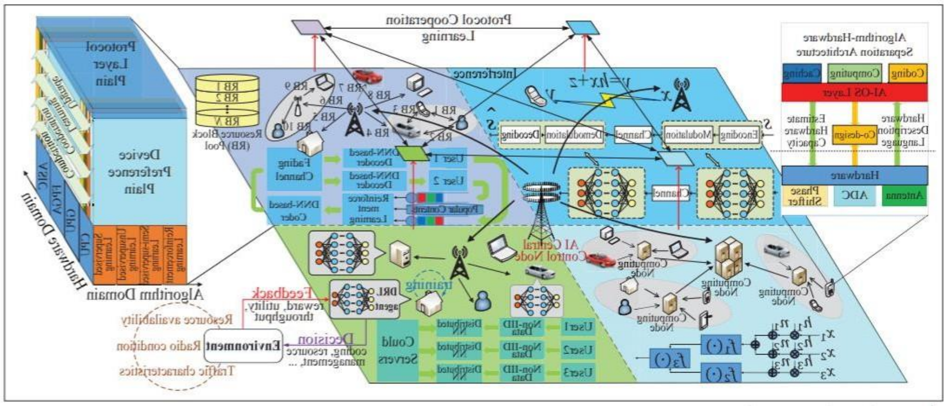


Fig: Architecture of 6G network

These models have been proposed after experimental observations of actual super capacitor leakage, such as those shown in Figure showing the self-discharge experienced by a charged 25F super capacitor over a two-weeks period. Another aspect to consider in the super capacitors vs. battery comparison is that in many application scenarios it is not possible to use the full energy stored in the super capacitor. The voltage of a super capacitor drops from full voltage to zero linearly, without the flat curve that is typical of most electrochemical batteries. The fraction of the charge available to the sensor node depends on the voltage requirements of the platform. For example, a Telos B mote requires a minimal voltage ranging from 1.8 V to 2.1 V. When the super capacitor voltage drops below this threshold, its residual energy can no longer be used to power the node.

This aspect may be partially mitigated by using a DC-DC converter to increase the voltage range, at the cost of introducing inefficiencies and an additional source of power consumption.

Wireless sensor network (WSN) refers to a group of spatially dispersed and dedicated sensors for monitoring and recording the physical conditions of the environment and organizing the collected data at a central location. WSNs measure environmental conditions like temperature, sound, pollution levels, humidity, wind, and so on.

These are similar to wireless ad hoc networks in the sense that they rely on wireless connectivity and spontaneous formation of networks so that sensor data can be transported wirelessly. WSNs are spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. And to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created.

The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding.

Energy harvesting (also known as power harvesting or energy scavenging or ambient power) is the process by which energy is derived from external sources (e.g., solar power, thermal energy, wind energy, salinity gradients, and kinetic energy, also known as ambient energy), captured, and stored for small, wireless autonomous devices, like those used in wearable electronics and wireless sensor networks.

Energy harvesters provide a very small amount of power for low-energy electronics. While the input fuel to some large-scale generation costs resources (oil, coal, etc.), the energy source for energy harvesters is present as ambient background. For example, temperature gradients exist from the operation of a combustion engine and in urban areas, there is a large amount of electromagnetic energy in the environment because of radio and television broadcasting.

One of the earliest applications of ambient power collected from ambient electromagnetic radiation (EMR) is the crystal radio.

The principles of energy harvesting from ambient EMR can be demonstrated with basic components

Energy can also be harvested to power small autonomous sensors such as those developed using MEMS technology. These systems are often very small and require little power, but their applications are limited by the reliance on battery power. Scavenging energy from ambient vibrations, wind, heat or light could enable smart sensors to be functional indefinitely. Several academic and commercial groups have been involved in the analysis and development of vibration-powered energy harvesting technology, including the Control and Power Group and Optical and Semiconductor Devices Group at Imperial College London, IMEC and the partnering Holst Centre, Adaptive Energy, LLC, ARVENI, MIT Boston, Victoria University of Wellington, Georgia Tech, UC Berkeley, Southampton University, University of Bristol, Micro Energy System Lab at The University of Tokyo, Nanyang Technological University, PMG Perpetuum, ReVibe Energy, Vest fold University College, National University of Singapore, NiPS Laboratory at the University of Perugia, Columbia University, Universidad Autónoma de Barcelona and USN & Renewable Energy Lab at the University of Ulsan (Ulsan, South Korea). The National Science Foundation also supports an Industry/University Cooperative Research Center led by Virginia Tech and The University of Texas at Dallas called the Center for Energy Harvesting Materials and Systems.

Typical power densities available from energy harvesting devices are highly dependent upon the specific application (affecting the generator's size) and the design itself of the harvesting generator. In general, for motion powered devices, typical values are a few $\mu W/cm^3$ for human body powered applications and hundreds of $\mu W/cm^3$ for generators powered from machinery. Most energy scavenging devices for wearable electronics generate very little power.

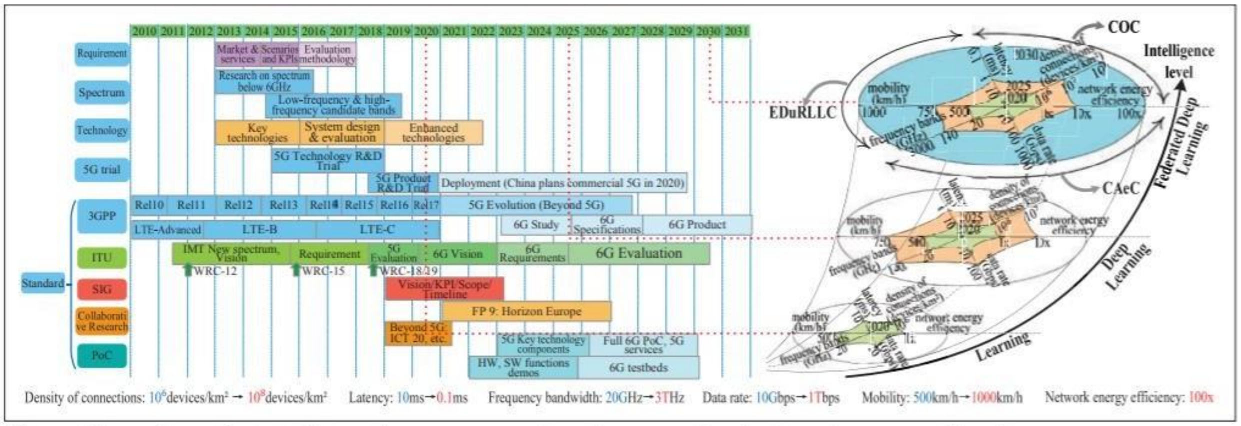


Fig: Spectrum Explanation of learning, Deep learning, federated deep learning

1.2 The Architecture of 6G Networks :

In this section, we introduce a potential architecture for 6G as shown in Fig.2, in which network intelligentization, subnetwork evolution, and intelligent radio are embraced. From Network Softwarization to Network Intelligentization We envision that 6G will take network softwarization to a new level, namely toward network intelligentization. In 5G, the “non-radio” aspect has become more and more important, and has been the key driver behind the recent efforts on “softwarization”. More specifically, two key 5G technologies are Software-Defined Networking (SDN) and Network Functions Virtualization (NFV), which have moved modern communications networks toward software-based virtual networks. They also enable network slicing, which can provide a powerful virtualization capability to allow multiple virtual networks to be created atop a shared physical infrastructure. Nevertheless, as the network is becoming more complex and more heterogeneous, softwarization is not going to be sufficient for 6G. In particular, to support AI-based applications, the network entities have to support diverse capabilities, including communications, content caching, computing, and even wireless power transfer. Furthermore, 6G will embrace new radio access interfaces such as THz communications and intel ligent surfaces. It will also need to support more advanced IoT functionalities including sensing, data collection, analytics, and storage. All of the aforementioned challenges call for an architecture that is flexible, adaptive, and more importantly, intelligent. Existing technologies, such as SDN, NFV, and network slicing will need to be further improved to meet these challenges. By enabling fast learning and adaptation, AI-based methods will render network slicing much more versatile in 6G.

The design of the 6G architecture shall follow an “AI native” approach where intelligentization will allow the network to be smart, agile, and able to learn and adapt itself according to the changing network dynamics. It will evolve into a “network of subnetworks,” allowing more efficient and flexible upgrades, and a new framework based on intelligent radio and algorithm-hardware separation to cope with the heterogeneous and upgradable hardware capabilities. Both of these two features will exploit AI techniques, as further illustrated in the following subsections. A Network of Subnetworks: Local vs Global Evolution Given its expected ultra-high heterogeneity, one key feature of 6G will be its capability to exploit a flexible subnetwork-wide evolution to effectively adapt to the local environments and user demands, thereby resulting in a “network of subnetworks.” In particular, local subnetworks in 6G may evolve individually to upgrade themselves. The local evolution may happen in a few neighboring cells in order to flexibly apply cutting-edge developments on new waveforms, coding, and multi-access protocols in subnetworks without extensive time-consuming tests. In contrast to the global evolution from 1G to 5G, in which both hardware and software of all the cells are upgraded simultaneously, there is no need to find a onesize-fit-all solution for all the cells and rebuild the whole system when local evolution is exploited. To achieve this goal, we need to address the following three challenges:

• Each subnetwork should collect and analyze its local data, which may include the wireless environments, user requests, mobility patterns, and so on, and then exploit AI methods to upgrade itself locally and dynamically.

• When the local PHY or MAC protocols are changed, the inter-subnetwork interaction is expected to maintain new inter-subnetwork coordination. One possible solution is to adopt game and learning approaches, which can assure the convergence of the sub networks upgrades.

• The local evolution of 6G requires a relatively stable control plane to support the evolution in the “network of subnetworks” level. One possible solution relies on the “learning from scratch” method developed in Alpha Zero.

CHAPTER 2

INTRODUCTION TO MATLAB

2.1 Introduction to MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

2.2 THE MATLAB SYSTEM:

The MATLAB system consists of five main parts

- Development Environment:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- The MATLAB Mathematical Function Library:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- The MATLAB Language:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both ―programming in the small‖ to rapidly create quick and dirty throw-away programs, and ―programming in the large‖ to create large and complex application programs.

- Graphics:

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for twodimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- The MATLAB Application Program Interface (API):

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using IMAGE PROCESSING toolbox.

2.3. GRAPHICAL USER INTERFACE (GUI):

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based Mfunction is executed.
- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted, GUIDE opens a new (i.e., blank) window.

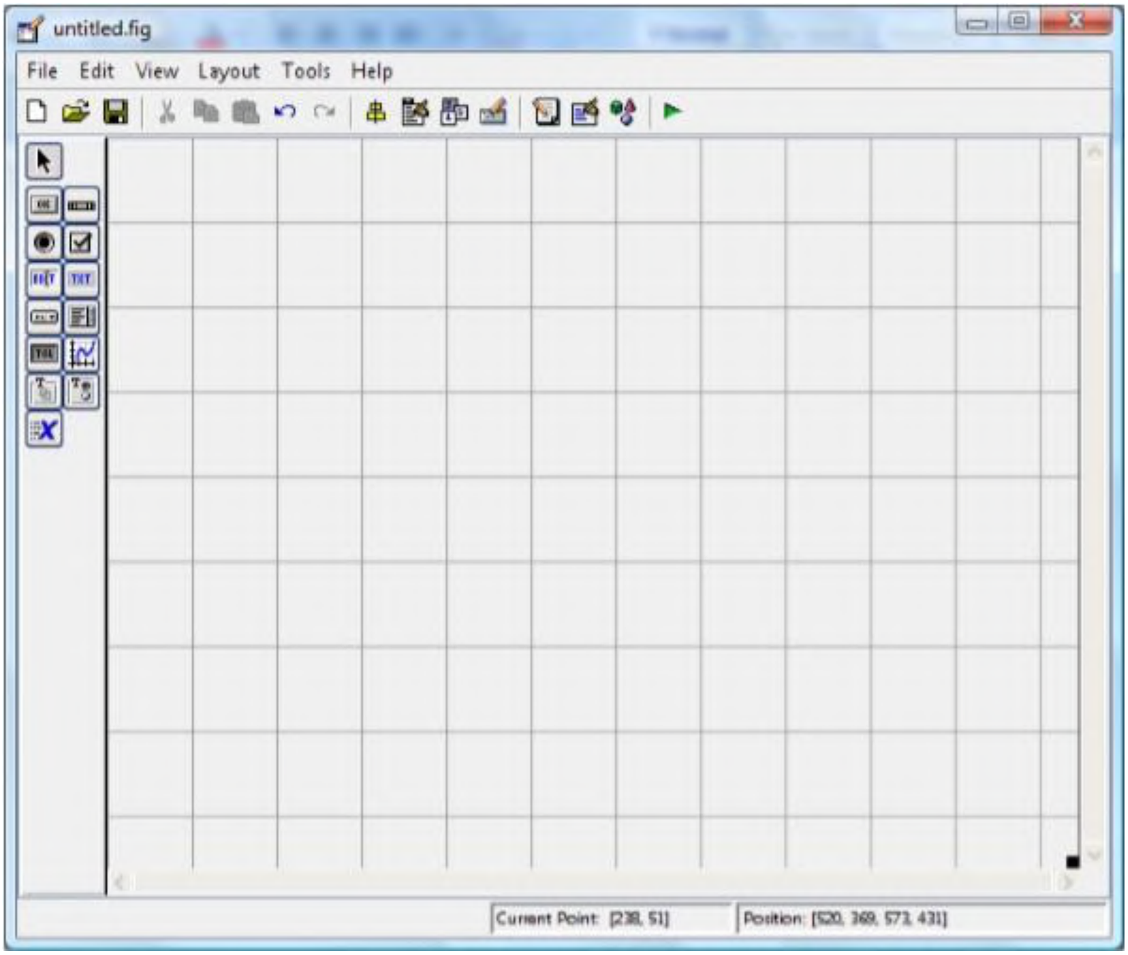


Figure : Matlab procedure

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

2.4 GETTING STARTED

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

2.4.1 INTRODUCTION - describes the components of the MATLAB system.

2.4.2 DEVELOPMENT ENVIRONMENT - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

2.4.3 MANIPULATING MATRICES - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

2.4.4 GRAPHICS - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images. 2.4.5 Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

2.4.5 DEVELOPMENT ENVIRONMENT – A Development environment is a collection of procedures and tools for developing ,testing and debugging an application or program

2.5.1 INTRODUCTION

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

2.5.2 STARTING MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.On a UNIX platform, to start MATLAB, type matlab

At the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

2.5.3 QUITTING MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

2.5.4 MATLAB DESKTOP

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB. The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

2.5.5 DESKTOP TOOLS

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

- Running External Programs - You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacsmagik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.
- Launch Pad - MATLAB's Launch Pad provides easy access to tools, demos, and documentation.
- Help Browser - Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents. To open the Help browser, click the help button in the toolbar, or type help browser in theCommand Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.
- Help Navigator - Use to Help Navigator to find information. It includes:
- Product filter - Set the filter to show documentation only for the products you specify.
- Contents tab - View the titles and tables of contents of documentation for your products.
- Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.
- Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.
- Favorites tab - View a list of documents you previously designated as favorites.
- Display Pane - After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:
- Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.
- Bookmark pages - Click the Add to Favorites button in the toolbar.
- Print pages - Click the print button in the toolbar.
- Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

- Current Directory Browser - MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.
- Search Path - To determine how to execute functions you call, MATLAB uses a search path to find Mfiles and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.
- Workspace Browser - The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

- Array Editor - Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

- Editor and debugger- MATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

CHAPTER 3

3.1 ENERGY HARVESTING

The generation of electrical energy involves an energy-to-energy conversion process such as mechanical-to-electrical (ME), chemical-to-electrical (CE), solar-to-electrical (SE), radio frequency-to-electrical (RFE), and thermal-to-electrical (TE). ME conversion is used in hydroelectric and wind turbines for large-scale generation to meet the demands of cities. CE conversion is used in batteries to provide portable electrical energy. TE conversion is a technology under development.

SE conversion uses solar energy to generate electrical energy. Solar farms provide electrical generation on a large industrial scale as well as for individual home owners. RFE converts electromagnetic energy in the millimeter (mm)-to-micron-wavelength range of the electromagnetic spectrum to electrical energy. Currently, energy harvesting refers to the nonchemical generation of small amounts of electrical energy on a local scale using one of the above energy conversion principles. Among the above energy conversion processes for electrical energy harvesting, even under very low light levels, SE conversion has generally been found to be the best choice and is widely used in consumer goods and many other products.

For example, SE conversion is used to power wrist watches, calculators, road signs, and in practically any other application where solar illumination is available and its space and power requirements can be met. The energy output of SE-based energy harvesters is limited by the size of the solar cell. However, in many energy-harvesting applications, a few microwatts of power may suffice and can be obtained. For 24/7 operation, storage devices such as rechargeable batteries and super-capacitors have been used. It is, however, important to note that in many applications, the use of SE conversion is not practical, such as in enclosed environments where enough light is never available.

The use of SE conversion may also be impractical due to the required size of the solar collector and its collection and storage components. Examples of such cases include self-powered and networked sensors used on machinery to monitor vibration and health in industrial plants; sensors for use inside an enclosed space such as in vehicle tires or inside a machine body; or self-powering of encapsulated microelectromechanical systems (MEMS)-type sensors of various types.

In many applications, the energy that has been made available for harvesting is in the form of mechanical kinetic and/or potential energy, or even impulsive loading of some form. As a result, in many applications, energy harvesters based on converting kinetic and/or potential energy to electrical energy become the only viable option. Certain ME-based energy-harvesting devices, such as those employing piezoelectric transducers, also have the unique capability of being used as sensors,

for example, for detecting and measuring acceleration or force generated due to a specific emergency event. Today, energy harvesters may be designed to provide perpetual power for autonomous wireless sensor nodes, or pulse power for short-lived one-time use, as in initiation or emergency systems. Design of energy harvesters is determined by both the local energy source (host system) and by the interfacing mechanism, particularly for motion-based energy-harvesting systems. Available sources of energy for harvesters can be grouped into either ambient or man-made.

The former category includes natural sources such as solar radiation, thermal gradients, wind, and ocean waves. Man-made sources may arise as by-products of various activities and processes, for example, background energy from radio-frequency signals generated primarily for use in telecommunication systems, or vibrational energy resulting from vehicles or large industrial systems, such as production machinery.

The choice of a particular energy source is determined by the operational environment, available energy density, and the required energy level to power the intended device. Figure 1.1 shows some of the popular energy sources and the corresponding energy density. It should be noted that, while the solar

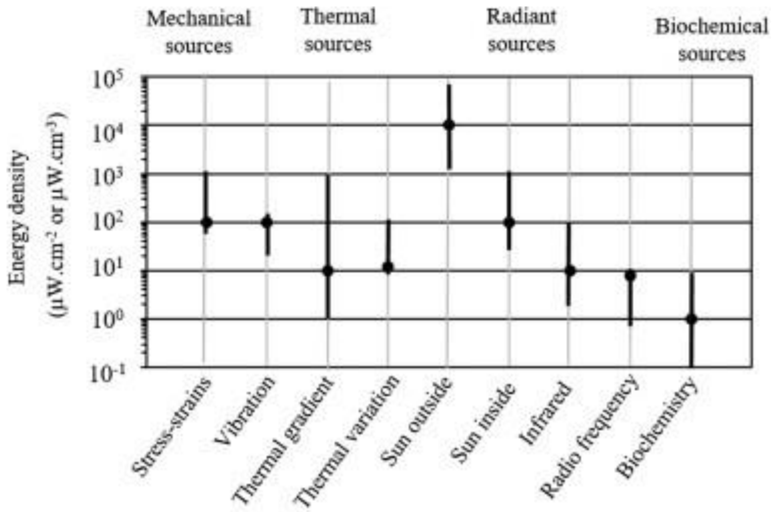


Figure 1.1 Energy densities of typical ambient energy sources.[1]

luminance has the highest available power density, it cannot be changed, and its availability is not always guaranteed. The primary focus of this book is on harnessing mechanical potential and/or kintic energy. Other sources of energy are also available, such as thermal gradient and radiated electromagnetic energy. The latter sources of energy are briefly described below. A brief description of the application of energy harvesting for self-powering implantable devices and sensors in the human body is also provided. The reader is referred to available literature for an in-depth treatment of the subject and for other sources of available energy for harvesting.

3.2 Thermal-to-Electrical-based Energy Harvesting:

The thermoelectric effect2 is used to convert temperature difference into voltage. An implementation of the effect in a loop constructed of two dissimilar conductors generates an electromotive force Vemf when a constant temperature gradient exists between the two common points. The generated voltage is given where a1 and a2 are the Seebeck coefficients for the two dissimilar conductors. The value of a may range from –100 mV/K to 1000 mV/K for common conductors.3 Very large temperature differences are needed to produce useful operating voltages compatible with electronic integrated circuits.

In order to make a useful generator, conductors are typically replaced by n- and p-type semiconductors, as illustrated in Fig. 1.2. This configuration allows heat transfer in the same direction, while the voltage difference across the n- and p-type materials is additive. The power output of an n-p-based thermoelectric generator is proportional to the square of the temperature difference between the hot and cold surfaces and is proportional to the physical cross-sectional area of the n-p legs. For example, the generated power density at a temperature difference of 200 °C is on the order of 100 mW/cm2.

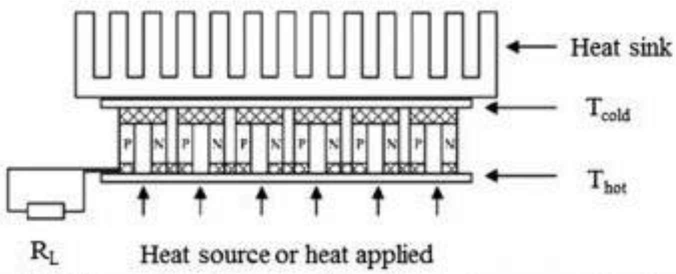


Figure 1.2 Thermoelectric-generating cell n- and p-type semiconductors.[4]

3.3 Solar-to-Electrical-based Energy Harvesting:

The most dominant source of radiated electromagnetic energy is solar energy, which illuminates the surface of the earth at a nominal value of 1 kW · m2 . Solar energy may be harvested indoors and outdoors using photovoltaic devices. This makes solar energy the first choice for harvesting, as long as it satisfies the constraints of the devices or systems to be powered. For example, the device to be powered may not have direct solar radiation or may require uninterrupted power during daytime as well as during the night without the use of a storage device due to size or weight limitations. Photovoltaics technology primarily targets the visible-to-near infrared part of the electromagnetic spectrum.

The transducers, commonly referred to as photovoltaics or photocells, are quantum devices that directly generate electron–hole pairs from the absorption of incident photons within the depletion region of a p-n junction device. These devices are generally modeled as a current source shunted by an ideal p-n junction diode. For indoor applications, electrical energy generated will also depend on the spatial characteristics of the light source as well as the distance of the transducer from the source. For example, a 100-W incandescent bulb is expected to generate a few microwatts at the output of a 2-mm diameter photovoltaic cell placed at a distance of 2 m.

3.4 Radio-Frequency-to-Electrical-based Energy Harvesting:

RFE technology harvests electromagnetic energy in the radio-frequency band from megahertz to microwave. RFE devices typically have a tuned receiving antenna for converting the received RF energy into electrical energy. The power generated by RFE harvesters is extremely low unless the receiver is in close proximity to the source and/or the receiver is very large. Some of these energy-harvesting devices use the ambient electromagnetic energy emitted by nearby sources and are finding use in autonomous sensor nodes. However, these types of harvesters cannot be placed inside conductive enclosures. Directed radiofrequency emissions have also been used for collection by matched receiving antennas (the so-called rectenna). For example, active and passive radio-frequency identification (RFID) systems use such technologies.

3.5 Sources of Energy from Human Activity:

Due to the recent proliferation of wearable health monitoring devices and portable electronics, considerable effort is being devoted to harnessing power from voluntary and involuntary human activities, for example, from the pulsating motion of the heart. Table compares nominal values of power available from human activity and the corresponding power needs of some typical applications. In addition to harnessing energy from human locomotion, devices to convert body heat to electrical energy through thermoelectric conversion and flexible piezoelectric materials embedded into fabrics may soon be coming to the market. In addition to harnessing energy from human activity, there is considerable interest in attaching self-powered health-monitoring sensors directly to organs, such as the heart. The sensors may also assist the organ’s function by providing electrical stimulus, as is the case with heart pacemakers. Figure shows an example of an autonomous device mounted on a bovine heart. This heart-assist device uses a flexible piezoelectric energy harvester. On another note, the hidden cost of attaching an energy-harvesting device to a system such as the heart, which is optimized for a particular function, should not be underestimated or overlooked. For instance, the heart, which is a pulsating oscillator with a life cycle of over 5 billion beats has taken nature over 65 million years to perfect. It pumps blood through a circulatory system

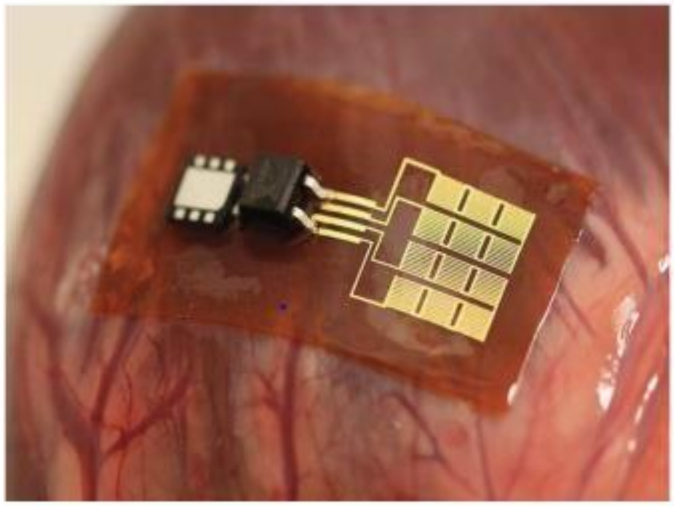


Figure 1.3 A piezoelectric energy harvester, with rectifier and microbattery, mounted on a bovine heart. (Reprinted with permission from Ref. 12.)

of vessels to deliver oxygen and nutrients to individual cells and removes metabolic waste.13 Attaching an external device, no matter how small, may produce a reactive chain of events from the cardiovascular system. The longterm effects of loading the cardiovascular system are difficult to predict and will require the accumulation of clinical data over many years.

3.6 Mechanical-to-Electrical-based Energy Harvesting:

The design of energy harvesters intended for converting mechanical kinetic and/or potential energy from a host system to electrical energy for direct consumption or storage for later use. The process of converting mechanical energy to electrical energy may be described in three distinct phases, as shown in Fig. 1.4. In the first phase, an interfacing mechanism properly transfers the mechanical energy to the transducer. In the second phase, the transducer generates electrical energy. In the third phase, the generated electrical energy is collected and conditioned to be either stored in an energy storage device such as a rechargeable battery or a capacitor or to be delivered directly to the intended electrical energy consuming device (load). The remainder of this book is divided into four chapters. Chapter 2 describes the three primary types of transducers typically used for converting mechanical energy to electrical energy, that is, piezoelectric, electromagnetic, and electrostatic transducers. Magnetostrictive-based transducers are also briefly introduced. Chapter 3 presents an in-depth analysis of the interfacing mechanisms used for coupling mechanical kinetic and/or potential energy to the transducer for effective energy transfer. Chapter

4 addresses coupling and conditioning circuits needed to extract the generated electrical energy for delivery to the load. The theme of these chapters shows the connection between the three components of an energy-harvesting system, namely, the interfacing mechanism, the transducer, and the collection and conditioning circuit. Chapter presents case studies of some available energy harvesting system solutions.

Energy harvesting (also known as power harvesting or energy scavenging or ambient power) is the process by which energy is derived from external sources (e.g., solar power, thermal energy, wind energy, salinity gradients, and kinetic energy, also known as ambient energy), captured, and stored for small, wireless autonomous devices, like those used in wearable electronics and wireless sensor networks.

Energy harvesters provide a very small amount of power for low-energy electronics. While the input fuel to some large-scale generation costs resources (oil, coal, etc.), the energy source for energy harvesters is present as ambient background. For example, temperature gradients exist from the operation of a combustion engine and in urban areas, there is a large amount of electromagnetic energy in the environment because of radio and television broadcasting.

One of the earliest applications of ambient power collected from ambient electromagnetic radiation (EMR) is the crystal radio.

The principles of energy harvesting from ambient EMR can be demonstrated with basic components

3.6.1 OPERATION:

Energy harvesting devices converting ambient energy into electrical energy have attracted much interest in both the military and commercial sectors. Some systems convert motion, such as that of ocean waves, into electricity to be used by oceanographic monitoring sensors for autonomous operation. Future applications may include high power output devices (or arrays of such devices) deployed at remote locations to serve as reliable power stations for large systems. Another application is in wearable electronics, where energy harvesting devices can power or recharge cellphones, mobile computers, radio communication equipment, etc. All of these devices must be sufficiently robust to endure long-term exposure to hostile environments and have a broad range of dynamic sensitivity to exploit the entire spectrum of wave motions.

3.7 Accumulating energy:

Energy can also be harvested to power small autonomous sensors such as those developed using MEMS technology. These systems are often very small and require little power, but their applications are limited by the reliance on battery power. Scavenging energy from ambient vibrations, wind, heat or light could enable smart sensors to be functional indefinitely. Several academic and commercial groups have been involved in the analysis and development of vibration-powered energy harvesting technology, including the Control and Power Group and Optical and Semiconductor Devices Group at Imperial College London, IMEC and the partnering Holst Centre, Adaptiv Energy, LLC, ARVENI, MIT Boston, Victoria University of Wellington, Georgia Tech, UC Berkeley, Southampton University, University of Bristol, Micro Energy System Lab at The University of Tokyo, Nanyang Technological University, PMG Perpetuum, ReVibe Energy, Vestfold University College, National University of Singapore, NiPS Laboratory at the University of Perugia,] Columbia University, Universidad Autonoma de Barcelona and USN & Renewable Energy Lab at the University of Ulsan (Ulsan, South Korea). The National Science Foundation also supports an Industry/University Cooperative Research Center led by Virginia Tech and The University of Texas at Dallas called the Center for Energy Harvesting Materials and Systems.

Typical power densities available from energy harvesting devices are highly dependent upon the specific application (affecting the generator's size) and the design itself of the harvesting generator. In general, for motion powered devices, typical values are a few $\mu W/cm^3$ for human body powered applications and hundreds of $\mu W/cm^3$ for generators powered from machinery. Most energy scavenging devices for wearable electronics generate very little power.

3.8 Storage of power:

In general, energy can be stored in a capacitor, super capacitor, or battery. Capacitors are used when the application needs to provide huge energy spikes. Batteries leak less energy and are therefore used when the device needs to provide a steady flow of energy. Compared to batteries, super capacitors have virtually unlimited charge-discharge cycles and can therefore operate forever enabling a maintenance-free operation in IoT and wireless sensor devices.

3.9 Use of the power:

Current interest in low power energy harvesting is for independent sensor networks. In these applications an energy harvesting scheme puts power stored into a capacitor then boosted/regulated to a second storage capacitor or battery for the use in the microprocessor or in the data transmission. The power is usually used in a sensor application and the data stored or is transmitted possibly through a wireless method.

The history of energy harvesting dates back to the windmill and the waterwheel. People have searched for ways to store the energy from heat and vibrations for many decades. One driving force behind the search for new energy harvesting devices is the desire to power sensor networks and mobile devices without batteries. Energy harvesting is also motivated by a desire to address the issue of climate change and global warming.

There are many small-scale energy sources that generally cannot be scaled up to industrial size:

- Some wristwatches are powered by kinetic energy (called automatic watches), in this case movement of the arm is used. The arm movement causes winding of its mainspring. A newer design introduced by Seiko ("Kinetic") uses movement of a magnet in the electromagnetic generator instead to power the quartz movement. The motion provides a rate of change of flux, which results in some induced emf on the coils. The concept is related to Faraday's Law.
- Photovoltaics is a method of generating electrical power by converting solar radiation (both indoors and outdoors) into direct current electricity using semiconductors that exhibit the photovoltaic effect. Photovoltaic power generation employs solar panels composed of a number of cells containing a photovoltaic material. Note that photovoltaics have been scaled up to industrial size and that large solar farms exist.
- Thermoelectric generators (TEGs) consist of the junction of two dissimilar materials and the presence of a thermal gradient. Large voltage outputs are possible by connecting many junctions electrically in series and thermally in parallel. Typical performance is 100-300 μV/K per junction. These can be utilized to capture mW.s of energy from industrial equipment,

structures, and even the human body. They are typically coupled with heat sinks to improve temperature gradient.

- Micro wind turbine are used to harvest wind energy readily available in the environment in the form of kinetic energy to power the low power electronic devices such as wireless sensor nodes. When air flows across the blades of the turbine, a net pressure difference is developed between the wind speeds above and below the blades. This will result in a lift force generated which in turn rotate the blades. Similar to photovoltaics, wind farms have been constructed on an industrial scale and are being used to generate substantial amounts of electrical energy.
- Piezoelectric crystals or fibers generate a small voltage whenever they are mechanically deformed. Vibration from engines can stimulate piezoelectric materials, as can the heel of a shoe, or the pushing of a button.
- Special antennas can collect energy from stray radio waves. this can also be done with a Rectenna and theoretically at even higher frequency EM radiation with a Nantenna.
- Power from keys pressed during use of a portable electronic device or remote controller, using magnet and coil or piezoelectric energy converters, may be used to help power the device.

3.9.1 Ambient-radiation sources:

A possible source of energy comes from ubiquitous radio transmitters. Historically, either a large collection area or close proximity to the radiating wireless energy source is needed to get useful power levels from this source. The n antenna is one proposed development which would overcome this limitation by making use of the abundant natural radiation (such as solar radiation).

One idea is to deliberately broadcast RF energy to power and collect information from remote devices:[14] This is now commonplace in passive radio-frequency identification (RFID) systems, but the Safety and US Federal Communications Commission (and equivalent bodies worldwide) limit the maximum power that can be transmitted this way to civilian use. This method has been used to power individual nodes in a wireless sensor network

3.9.2 Fluid flow

Airflow can be harvested by various turbine and non-turbine generator technologies. For example, Zephyr Energy Corporation's patented Windbeam micro generator captures energy from airflow to recharge batteries and power electronic devices. The Windbeam's novel design allows it to operate silently in wind speeds as low as 2 mph. The generator consists of a lightweight beam suspended by durable long-lasting springs within an outer frame. The beam oscillates rapidly when exposed to airflow due to the effects of multiple fluid flow phenomena. A linear alternator assembly converts the oscillating beam motion into usable electrical energy. A lack of bearings and gears eliminates frictional inefficiencies and noise. The generator can operate in low-light environments unsuitable for solar panels (e.g. HVAC ducts) and is inexpensive due to low cost components and simple construction. The scalable technology can be optimized to satisfy the energy requirements and design constraints of a given application.

The flow of blood can also be used to power devices. For instance, the pacemaker developed at the University of Bern, uses blood flow to wind up a spring which in turn drives an electrical micro-generator

3.9.3 Photovoltaic

Photovoltaic (PV) energy harvesting wireless technology offers significant advantages over wired or solely battery-powered sensor solutions: virtually inexhaustible sources of power with little or no adverse environmental effects. Indoor PV harvesting solutions have to date been powered by specially tuned amorphous silicon (aSi)a technology most used in Solar Calculators. In recent years new PV technologies have come to the forefront in Energy Harvesting such as Dye Sensitized Solar Cells (DSSC). The dyes absorbs light much like chlorophyll does in plants. Electrons released on impact escape to the layer of TiO_2 and from there diffuse, through the electrolyte, as the dye can be tuned to the visible spectrum much higher power can be produced. At 200 lux a DSSC can provide over 10 µW per cm².

3.9.4 Piezoelectric

The piezoelectric effect converts mechanical strain into electric current or voltage. This strain can come from many different sources. Human motion, low-frequency seismic vibrations, and acoustic noise are everyday examples. Except in rare instances the piezoelectric effect operates in AC requiring time-varying inputs at mechanical resonance to be efficient.

Most piezoelectric electricity sources produce power on the order of milliwatts, too small for system application, but enough for hand-held devices such as some commercially available self-winding wristwatches. One proposal is that they are used for micro-scale devices, such as in a device harvesting micro-hydraulic energy. In this device, the flow of pressurized hydraulic fluid drives a reciprocating piston supported by three piezoelectric elements which convert the pressure fluctuations into an alternating current.

As piezo energy harvesting has been investigated only since the late 1990s, it remains an emerging technology. Nevertheless, some interesting improvements were made with the self-powered electronic switch at INSA school of engineering, implemented by the spin-off Arveni. In 2006, the proof of concept of a battery-less wireless doorbell push button was created, and recently, a product showed that classical wireless wallswitch can be powered by a piezo harvester. Other industrial applications appeared between 2000 and 2005 to harvest energy from vibration and supply sensors for example, or to harvest energy from shock.

Piezoelectric systems can convert motion from the human body into electrical power. DARPA has funded efforts to harness energy from leg and arm motion, shoe impacts, and blood pressure for low level power to implantable or wearable sensors. The nanobrushes are another example of a piezoelectric energy harvester. They can be integrated into clothing. Multiple other nanostructures have been exploited to build an energy-harvesting device, for example, a single crystal PMN-PT nanobelt was fabricated and assembled into a piezoelectric energy harvester in 2016. Careful design is needed to minimise user discomfort. These energy harvesting sources by association affect the body.

The Vibration Energy Scavenging Project is another project that is set up to try to scavenge electrical energy from environmental vibrations and movements. Microbelt can be used to gather electricity from respiration. Besides, as the vibration of motion from human comes in three directions, a single piezoelectric cantilever based omni-directional energy harvester is created by using 1:2 internal resonance. Finally, a millimeter-scale piezoelectric energy harvester has also already been created.

The use of piezoelectric materials to harvest power has already become popular. Piezoelectric materials have the ability to transform mechanical strain energy into electrical charge. Piezo elements are being embedded in walkways to recover the "people energy" of footsteps. They can also be embedded in shoes to recover "walking energy". Researchers at MIT developed the first micro-scale piezoelectric energy harvester using thin film PZT in 2005. Arman Hajati and Sang-Gook Kim invented the Ultra Wide-Bandwidth micro-scale piezoelectric energy harvesting device by exploiting the nonlinear stiffness of a doubly clamped microelectromechanical systems (MEMSs) resonator. The stretching strain in a doubly clamped beam shows a nonlinear stiffness, which provides a passive feedback and results in amplitude-stiffened Duffing mode resonance.

Typically, piezoelectric cantilevers are adopted for the above-mentioned energy harvesting system. One drawback is that the piezoelectric cantilever has gradient strain distribution, i.e., the piezoelectric transducer is not fully utilized. To address this issue, triangle shaped and L-shaped cantilever are proposed for uniform strain distribution.

In 2018, Soochow University researchers reported hybridizing a triboelectric nanogenerator and a silicon solar cell by sharing a mutual electrode. This device can collect solar energy *or* convert the mechanical energy of falling raindrops into electricity

Energy from smart roads and piezoelectricity

Brothers Pierre Curie and Jacques Curie gave the concept of piezoelectric effect in 1880. Piezoelectric effect converts mechanical strain into voltage or electric current and generates electric energy from motion, weight, vibration and temperature changes as shown in the figure.

Considering piezoelectric effect in thin film lead zirconate titanate PZT, microelectromechanical systems (MEMS) power generating device has been developed. During recent improvement in piezoelectric technology, Aqsa Abbasi (also known as Aqsa Aitbar, General secretory at IMS, IEEE MUET Chapter and Director Media at HYD MUN) diffentiated two modes called and in vibration converters and re-designed to resonate at specific frequencies from an external vibration energy source, thereby creating electrical energy via the piezoelectric effect using electromechanical damped mass.

However, Aqsa further developed beam-structured electrostatic devices that are more difficult to fabricate than PZT MEMS devices versus a similar because general silicon processing involves many more mask steps that do not require PZT film. Piezoelectric type sensors and actuators have a cantilever beam structure that consists of a membrane bottom electrode, film, piezoelectric film, and top electrode. More than (3-5 masks) mask steps are required for patterning of each layer while have very low induced voltage. Pyroelectric crystals that have a unique polar axis and have spontaneous polarization, along which the spontaneous polarization exists. These are the crystals of classes 6mm, 4mm, mm2, 6, 4, 3m, 3,2, m. The special polar axis crystallo physical axis X3 coincides with the axes L6,L4, L3, and L2 of the crystals or lies in the unique straight plane P (class “m”). Consequently, the electric centers of positive and negative charges are displaced of an elementary cell from equilibrium positions, i.e., the spontaneous polarization of the crystal changes. Therefore, all considered crystals have spontaneous polarization .

Since piezoelectric effect in pyroelectric crystals arises as a result of changes in their spontaneous polarization under external effects (electric fields, mechanical stresses). As a result of displacement, Aqsa Abbasi introduced change in the components along all three axes . Suppose that is proportional to the mechanical stresses causing in a first approximation, which results where Tkl represents the mechanical stress and dikl represents the piezoelectric modules.

PZT thin films have attracted attention for applications such as force sensors, accelerometers, gyroscopes actuators, tunable optics, micro pumps, ferroelectric RAM, display systems and smart roads, when energy sources are limited, energy harvesting plays an important role in the environment. Smart roads have the potential to play an important role in power generation. Embedding piezoelectric material in the road can convert pressure exerted by moving vehicles into voltage and current.

3.9.5 Smart transportation intelligent system

Piezoelectric sensors are most useful in Smart-road technologies that can be used to create systems that are intelligent and improve productivity in the long run. Imagine highways that alert motorists of a traffic jam before it forms. Or bridges that report when they are at risk of collapse, or an electric grid that fixes itself when blackouts hit. For many decades, scientists and experts have argued that the best way to fight congestion is intelligent transportation systems, such as roadside sensors to measure traffic and synchronized traffic lights to control the flow of vehicles. But the spread of these technologies has been limited by cost. There are also some other smart-technology shovel ready projects which could be deployed fairly quickly, but most of the technologies are still at the development stage and might not be practically available for five years or more.

Energy and global warming are two subjects that recently drew lots of attention in different sections of the society. Among all energy dependent units, automobiles are in fact one of the biggest consumers. Some considerable amount of the energy that an automobile produces for motion is wasted through heat and vibrations. Although different parts of

automobiles are subjected to vibrations (engine, passenger seats, and the body of the vehicle) and constant deformations (tires), no significant research has been done on recycling those wastes. Some early attempts show feasibility of energy harvesting using piezoelectric energy harvester in cars. Kim et al. investigated the capability of piezoelectric transducer to scavenge electric power from automobile engine vibrations. Furthermore, hybrid cars are other examples of energy harvesting in automobiles but they are not in the scope of this research topic. Ambient deflection energy of pneumatic tire which is normally lost could be utilized to develop fully self-powered electronics.

In this study, we propose a solution to harvest tire deflection energy by using devices which are independent from tire structure. This research facilitates the application of smart tires where reliable and sustainable power sources are essentially vital for different wired and wireless sensor. Some of the low power components such as tire pressure sensors are already utilized in the new automobiles. Those sensors are not communicating with the Engine Control Unit (ECU) continually (every 10-30 sec) so the battery can live longer. Other important features regarding to the tire condition that are needed for better isolation and preventing roll over are not implemented due to serious power issues. The appropriate sensors for those applications require reliable and unlimited life power sources because of being in constant communication with ECU and the actuators.

So far, only a few patents are available in the literature about tire energy harvesting. Their fundamental problem is the dependency of the proposed energy harvesting devices on the tire structure. As a result, replacing the tire, they are throwing away their energy harvesting devices as well. In addition, there is no analytical study conducted over the feasibility and the amount of energy which can be harvested from tires. The lack of this information in addition to the design shortcomings makes all those solutions not appealing for the automotive and tire industries for commercialization.

Among different candidates that can potentially be utilized for tire energy harvesting applications, piezoelectric is an ideal material. In fact, this material converts mechanical strain (tire deflection) directly to electrical current and as a result can be used to generate power from tire flexure. Many research has been done on the applications of piezoelectric for power harvesting systems as well as the estimation of electric charge output for piezoelectric energy harvesting devices , and the techniques to increase the efficiency of the charging the batteries by improving the circuits. Similar studies performed on the efficiency of battery recharging. Such research is still in progress and needs more improvement. One early study performed by Sodano et al. investigates the power harvesting from vibration using piezoelectric materials.

The results indicate the feasibility of using the piezoelectric material to power the portable electronics and wireless sensors. Consequently, Sodano et al estimated the electric charge output for piezoelectric energy harvesting. The constitutive equations of a bimorph PZT cantilever beam with base excitation are derived by authors. Later on, Sodano et al. compared the performance of different piezoelectric materials for recharging batteries. This study was done on PZT, QP and MFC.

The conclusion was that PZT is the most effective device for power harvesting applications. The objectives of the research described here are calculations of the wasted tire deformation energy, analytical study on the feasibility of the tire energy harvesting, and the amount of energy using the piezoelectric materials. The paper initially presents the concept of tire deflection and the amount of energy wasted in a car because of tire deflections. The paper then introduces piezoelectric transducer and a model for PZT from literature which is tailored for tire application. Finally, the paper presents the amount of energy which can be captured from tires using PZT.

3.10 ESTIMATION OF WASTED DEFLECTION ENERGY FROM ROAD LOADS:

Investigation of the energy harvesting of the tires is the main goal of this research. However a basic background of the forces which cause the deflection as well as evaluation of the energy used for the deflection is required for better understanding of the problem. Although all the energy of the WLUH GHIOHFWLRQ FDQ W EH UHFRYHUHG it justifies the significance of further studies in this area. It also gives a measure to calculate the efficiency of the power harvesting process which later on can be used as a metric for comparing different methods of harvesting and the appropriate materials for this application. There are three forces and one torque dominantly acting on an automobile tire. These forces as shown in Fig. 1, are the result of different characteristics of a car. The vertical force is due to the sprung mass of the vehicle, aligning torque is caused by the steering torque on the tire, longitudinal force is originated by the traction/acceleration or braking/deceleration, and the lateral force is the reaction of the forces to turn the vehicle.

All these forces contribute to the deformation of the tire (sidewall 13%, shoulder region, 12%, and the beads 2%); however the share of the vertical component is dominant (tread region 73%).

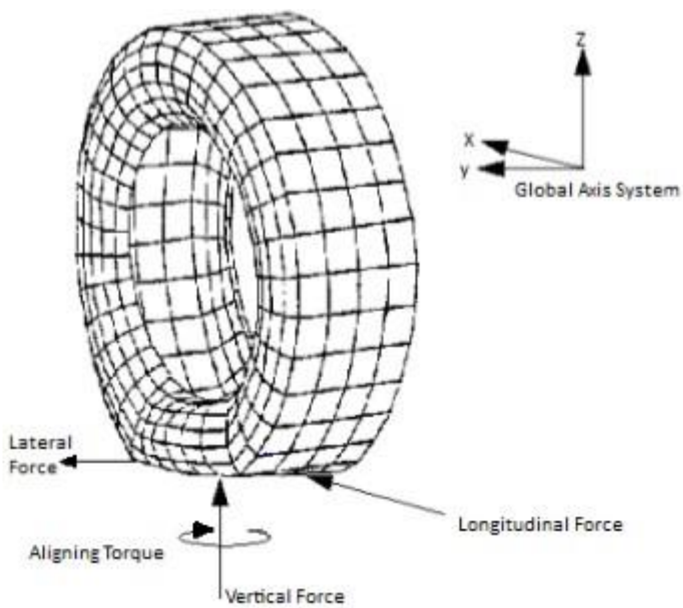


Fig. 1   Forces and Torque at the Center of the Tire-Road Surface Contact Patch

Calculation of the tire deflection in real world is difficult and there is no accurate model to relate different important factors to the energy loss in tire deflection. However, the relation between the tire deflection energy loss and the rolling resistance is well established in the vehicle textbooks. To understand the rolling resistance and its relevance to the tire deflection, It should be noted that when the tire is rolling, the tire carcass is deflected due to the vertical component of the forces on the tire. When tire deformed, the normal pressure distribution in the whole contact patch is not similar. In fact, the normal pressure in the leading half is higher than in the trailing half. Consequently, the center of normal pressure is transferred in the rolling direction. As a result of this transfer, a moment is produced about the tire rotation axis. This moment is the rolling resistance. Although other factors such as tire-road friction (2±10%), and circulation of air inside the tire and the tire fan effect (1.5±3.5%), contribute to the rolling resistance, it is evident that the tire deflection has a dominating effect (90± 95%). As a result, about 65-69% of the rolling resistance is from the tire deflection caused by vertical forces. Considering the whole vehicle wheel, the total rolling resistance is the sum of the resistances from all the wheels which can be presented as:

Where R is the rolling resistance force, fr is the coefficient of rolling resistance, and W is weight of the wheel. The coefficient of rolling resistance, fr, is a dimensionless factor that expresses the effects of the complicated and interdependent physical properties of tire and ground. Several equations for estimating rolling resistance have been developed over the years. To estimate fr, the studies on the rolling loss characteristics of solid rubber tires suggest an equation

Where C is a Constant reflecting loss and elastic characteristics of the tire material, D is the Outside diameter, ht is the tire section height, and w is the tire section width. Over broader speed ranges, the coefficient rises in a manner that is closer to a speed-squared relationship. Another study by the institute of technology in Stuttgart suggests the following equation for rolling on a concrete surface Where V is the speed in mph, fo is the basic coefficient, and fs is the speed effect coefficient. The two coefficients, fo and fs, depend on inflation pressure and are determined from the graph shown in Fig. 2

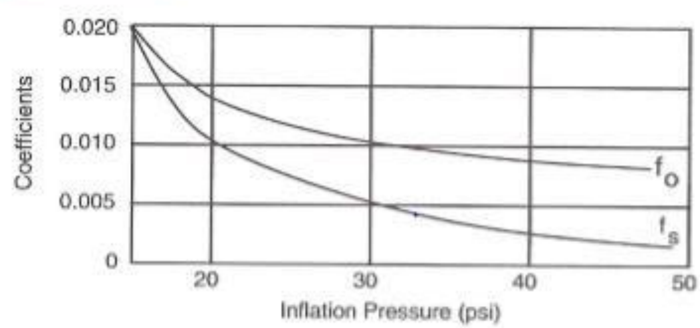


Fig. 2 tire deflection coefficients versus inflation pressure

The road load horsepower is computed by multiplying Eq. (1) by the vehicle velocity and applying the appropriate conversion factor to obtain horsepower.

Where V is in mph. These equations are used to estimate the energy of tire deflection. For instance, in a car weighing 1361kg rolls along a road at a speed of 50mph, the air temperature is 55°F and the inflation pressure is 30 psi, with radial-ply tires,the rolling resistance coefficient is:

3.10.1 PIEZOELECTRIC TRANSDUCER:

Piezoelectric transducers are classified as devices which can convert mechanical energy to electricity and vice versa. Given an electrical voltage, the piezoelectric transducers generate large forces in a small displacement. This characteristic makes the piezoelectric material a good candidate for actuation and micro position application. As a mechanical to electrical transducer, piezoelectric material can be used to convert small displacements to electricity. In fact, change of stress in the element of piezoelectric provides a transient potential difference across the electrodes. This electricity can then be stored or used to extend the life of the power supply of electronic devices. In fact, using piezoelectric material is ideal for the situations where continuous source of deflections exists; for instance mechanical vibrations. It can provide energy for the wireless devices which are hard to reach for battery replacement and their life span are mainly limited to the capacity of their power supply. In addition, piezoelectric devices are inexpensive and reliable to the environmental factors such as electromagnetic fields and humidity. Being robust to damage and environment factors are two desirable characteristics for energy harvesting applications. For using piezoelectric devices to capture energy from tire deflections, some concepts may exist. Here the idea is to have arrays of piezoelectric stacks connected together by small springs and makes a flexible ring. As soon as the tire is inflated the piezoelectric stacks are connected to the tire. The coil springs restore the tension and produce the required distance for the two adjacent piezoelectric stacks. The advantage of this arrangement is the simplicity of installation and replacement. If something goes wrong with the tire, flat tire for instance, the energy harvesting system can be pulled off and replaced in the new tire. Consider piezoelectric energy harvester model PEH (Mide´ Technology Corporation, Medford, MA, USA) which includes 1 stack of 2 PZT active elements. The size of this device is 9.20 x 4.38 x 0.99 cm. By the concept of having PZT arrays as the energy harvesting unit, inside a 195/60-15 tire structure which has the perimeter of 193 cm, about 14 stacks of piezo with a small spring with a half length of stack size between each two stacks can be mounted. The schematic view of this design and the connection among different sections of the power harvesting system is illustrated in Fig. 3. In this configuration, electrical energy is transferred by the wires to the battery charger. There are different strategies and circuits available for this purpose. The efficiency of the harvesting highly depends on the recharging technique. It seems the method purposed by Sodano achieves an acceptable efficiency with less complexity. It can be easily implemented in automotive application where the simplicity and low cost is always required. The energy can then be used to power the electronics and sensors which are hard to reach inside the tire.

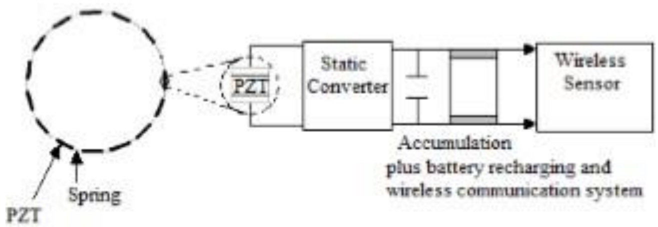


Fig. 3 System for harvesting energy from tire

3.10.2 AMOUNT OF HARVESTED ENERGY USING PZT:

One of the main objectives in this paper is giving an estimate of the amount of energy which can be harvested from tire deflections using piezoelectric materials. This study will help the researchers and manufacturers to have a better picture about the feasibility of the subject. To estimate this energy a realistic model is required. For this application, the model should combine constitutive equations of piezoelectric and the equation of motion of the cantilever beam including bimorph PZT components (Fig. 4). Sodano et al. got an accurate model of the power harvesting system for a bimorph PZT cantilever stack displacement of the beam as the summation of product of the modes and the temporal coordinates (RayleighRitz method) 2- Euiler-Bernolli beam model for The PZT stack vibrations 3- Constant electric energy across the PZT Eqs.(5) and (6) are derived for all vibration modes of a cantilever beam. For the tire energy harvesting applications, the PZT inside the tire only experiences the first mode. Other modes of vibration will not happen due to the physical constraints inside the tire. For the cantilever beam the first mode of vibration can be written.

CHAPTER 4

ELECTROMAGNETIC RADIATION

Electromagnetic radiation (EM radiation or EMR) refers to the waves (or their quanta, photons) of the electromagnetic field, propagating (radiating) through space, carrying electromagnetic radiant energy. It includes radio waves, microwaves, infrared, (visible) light, ultraviolet, X-rays, and gamma rays

Classically, electromagnetic radiation consists of electromagnetic waves, which are synchronized oscillations of electric and magnetic fields that propagate at the speed of light, which, in a vacuum, is commonly denoted c. In homogeneous, isotropic media, the oscillations of the two fields are perpendicular to each other and perpendicular to the direction of energy and wave propagation, forming a transverse wave. The wavefront of electromagnetic waves emitted from a point source (such as a light bulb) is a sphere. The position of an electromagnetic wave within the electromagnetic spectrum can be characterized by either its frequency of oscillation or its wavelength. Electromagnetic waves of different frequency are called by different names since they have different sources and effects on matter. In order of increasing frequency and decreasing wavelength these are: radio waves, microwaves, infrared radiation, visible light, ultraviolet radiation, X-rays and gamma rays.

Electromagnetic waves are emitted by electrically charged particles undergoing acceleration, and these waves can subsequently interact with other charged particles, exerting force on them. EM waves carry energy, momentum and angular momentum away from their source particle and can impart those quantities to matter with which they interact. Electromagnetic radiation is associated with those EM waves that are free to propagate themselves ("radiate") without the continuing influence of the moving charges that produced them, because they have achieved sufficient distance from those charges. Thus, EMR is sometimes referred to as the far field. In this language, the near field refers to EM fields near the charges and current that directly produced them specifically, electromagnetic induction and electrostatic induction phenomena.

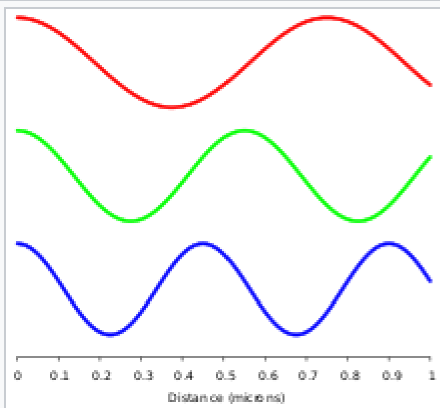
In quantum mechanics, an alternate way of viewing EMR is that it consists of photons, uncharged elementary particles with zero rest mass which are the quanta of the electromagnetic force, responsible for all electromagnetic interactions. Quantum electrodynamics is the theory of how EMR interacts with matter on an atomic level. Quantum effects provide additional sources of EMR, such as the transition of electrons to lower energy levels in an atom and black-body radiation. The energy of an individual photon is quantized and is greater for photons of higher frequency. This relationship is given by Planck's equation $E = h\nu$, where E is the energy per photon, ν is the frequency of the photon, and h is Planck's constant. A single gamma ray photon, for example, might carry 100,000 times the energy of a single photon of visible light.

The effects of EMR upon chemical compounds and biological organisms depend both upon the radiation's power and its frequency. EMR of visible or lower frequencies (i.e., visible light, infrared, microwaves, and radio waves) is called non-ionizing radiation, because its photons do not individually have enough energy to ionize atoms or molecules or break chemical bonds. The effects of these radiations on chemical systems and living tissue are caused primarily by heating effects from the combined energy transfer of many photons. In contrast, high frequency ultraviolet, X-rays and gamma rays are called ionizing radiation, since individual photons of such high frequency have enough energy to ionize molecules or break chemical bonds. These radiations have the ability to cause chemical reactions and damage living cells beyond that resulting from simple heating, and can be a health hazard.

4.1 Maxwell’s equations:

James Clerk Maxwell derived a wave form of the electric and magnetic equations, thus uncovering the wave-like nature of electric and magnetic fields and their symmetry. Because the speed of EM waves predicted by the wave equation coincided with the measured speed of light, Maxwell concluded that light itself is an EM wave. Maxwell's equations were confirmed by Heinrich Hertz through experiments with radio waves.

According to Maxwell's equations, a spatially varying electric field is always associated with a magnetic field that changes over time. Likewise, a spatially varying magnetic field is associated with specific changes over time in the electric field. In an electromagnetic wave, the changes in the electric field are always accompanied by a wave in the magnetic field in one direction, and vice versa. This relationship between the two occurs without either type of field causing the other; rather, they occur together in the same way that time and space changes occur together and are interlinked in special relativity. In fact, magnetic fields can be viewed as electric fields in another frame of reference, and electric fields can be viewed as magnetic fields in another frame of reference, but they have equal significance as physics is the same in all frames of reference, so the close relationship between space and time changes here is more than an analogy. Together, these fields form a propagating electromagnetic wave, which moves out into space and need never again interact with the source. The distant EM field formed in this way by the acceleration of a charge carries energy with it that "radiates" away through space, hence the term.



Shows the relative wavelengths of the electromagnetic waves of three different colours of light (blue, green, and red) with a distance scale in micrometers along the x-axis.

Fig: Relation Between Wavelengths

4.2 Near and far fields:

Maxwell's equations established that some charges and currents ("sources") produce a local type of electromagnetic field near them that does not have the behaviour of EMR. Currents directly produce a magnetic field, but it is of a magnetic dipole type that dies out with distance from the current. In a similar manner, moving charges pushed apart in a conductor by a changing electrical potential (such as in an antenna) produce an electric dipole type electrical field, but this also declines with distance. These fields make up the near-field near the EMR source. Neither of these behaviours are responsible for EM radiation. Instead, they cause electromagnetic field behaviour that only efficiently transfers power to a receiver very close to the source, such as the magnetic induction inside a transformer, or the feedback behaviour that happens close to the coil of a metal detector. Typically, near-fields have a powerful effect on their own sources, causing an increased “load” (decreased electrical reactance) in the source or transmitter, whenever energy is withdrawn from the EM field by a receiver. Otherwise, these fields do not “propagate” freely out into space, carrying their energy away without distance-limit, but rather oscillate, returning their energy to the transmitter if it is not received by a receiver.

By contrast, the EM far-field is composed of radiation that is free of the transmitter in the sense that (unlike the case in an electrical transformer) the transmitter requires the same power to send these changes in the fields out, whether the signal is immediately picked up or not. This distant part of the electromagnetic field is "electromagnetic radiation" (also called the far-field). The far-fields propagate (radiate) without allowing the transmitter to affect them. This causes them to be independent in the sense that their existence and their energy, after they have left the transmitter, is completely independent of both transmitter and receiver. Due to conservation of energy, the amount of power passing through any spherical surface drawn around the source is the same. Because such a surface has an area proportional to the square of its distance from the source, the power density of EM radiation always decreases with the inverse square of distance from the source; this is called the inverse-square law.

This is in contrast to dipole parts of the EM field close to the source (the near-field), which varies in power according to an inverse cube power law, and thus does not transport a conserved amount of energy over distances, but instead fades with distance, with its energy (as noted) rapidly returning to the transmitter or absorbed by a nearby receiver (such as a transformer secondary coil).

The far-field (EMR) depends on a different mechanism for its production than the near-field, and upon different terms in Maxwell's equations. Whereas the magnetic part of the near-field is due to currents in the source, the magnetic field in EMR is due only to the local change in the electric field. In a similar way, while the electric field in the near-field is due directly to the charges and charge-separation in the source, the electric field in EMR is due to a change in the local magnetic field. Both processes for producing electric and magnetic EMR fields have a different dependence on distance than do near-field dipole electric and magnetic fields. That is why the EMR type of EM field becomes dominant in power “far” from sources. The term “far from sources” refers to how far from the source (moving at the speed of light) any portion of the outward-moving EM field is located, by the time that source currents are changed by the varying source potential, and the source has therefore begun to generate an outwardly moving EM field of a different phase.

A more compact view of EMR is that the far-field that composes EMR is generally that part of the EM field that has traveled sufficient distance from the source, that it has become completely disconnected from any feedback to the charges and currents that were originally responsible for it. Now independent of the source charges, the EM field, as it moves farther away, is dependent only upon the accelerations of the charges that produced it. It no longer has a strong connection to the direct fields of the charges, or to the velocity of the charges (currents).

In the Lienard–Wiechert potential formulation of the electric and magnetic fields due to motion of a single particle (according to Maxwell's equations), the terms associated with acceleration of the particle are those that are responsible for the part of the field that is regarded as electromagnetic radiation. By contrast, the term associated with the changing static electric field of the particle and the magnetic term that results from the particle's uniform velocity, are both associated with the electromagnetic near-field, and do not comprise EM radiation.

4.3 Properties:

Electrodynamics is the physics of electromagnetic radiation, and electromagnetism is the physical phenomenon associated with the theory of electrodynamics. Electric and magnetic fields obey the properties of superposition. Thus, a field due to any particular particle or time-varying electric or magnetic field contributes to the fields present in the same space due to other causes. Further, as they are vector fields, all magnetic and electric field vectors add together according to vector addition. For example, in optics two or more coherent lightwaves may interact and by constructive or destructive interference yield a resultant irradiance deviating from the sum of the component irradiances of the individual lightwaves.

Since light is an oscillation it is not affected by traveling through static electric or magnetic fields in a linear medium such as a vacuum. However, in nonlinear media, such as some crystals, interactions can occur between light and static electric and magnetic fields — these interactions include the Faraday effect and the Kerr effect.

In refraction, a wave crossing from one medium to another of different density alters its speed and direction upon entering the new medium. The ratio of the refractive indices of the media determines the degree of refraction, and is summarized by Snell's law. Light of composite wavelengths (natural sunlight) disperses into a visible spectrum passing through a prism, because of the wavelength-dependent refractive index of the prism material (dispersion); that is, each component wave within the composite light is bent a different amount.

EM radiation exhibits both wave properties and particle properties at the same time (see wave-particle duality). Both wave and particle characteristics have been confirmed in many experiments. Wave characteristics are more apparent when EM radiation is measured over relatively large timescales and over large distances while particle characteristics are more evident when measuring small timescales and distances. For example, when electromagnetic radiation is absorbed by matter, particle-like properties will be more obvious when the average number of photons in the cube of the relevant wavelength is much smaller than 1. It is not too difficult to experimentally observe non-uniform deposition of energy when light is absorbed, however this alone is not evidence of "particulate" behavior. Rather, it reflects the quantum nature of matter. Demonstrating that the light itself is quantized, not merely its interaction with matter, is a more subtle affair.

Some experiments display both the wave and particle natures of electromagnetic waves, such as the self-interference of a single photon. When a single photon is sent through an interferometer, it passes through both paths, interfering with itself, as waves do, yet is detected by a photomultiplier or other sensitive detector only once.

A quantum theory of the interaction between electromagnetic radiation and matter such as electrons is described by the theory of quantum electrodynamics.

Electromagnetic waves can be polarized, reflected, refracted, diffracted or interfere with each other.

4.4 Wave model:

In homogeneous, isotropic media, electromagnetic radiation is a transverse wave, meaning that its oscillations are perpendicular to the direction of energy transfer and travel. The electric and magnetic parts of the field stand in a fixed ratio of strengths in order to satisfy the two Maxwell equations that specify how one is produced from the other. In dissipation less (lossless) media, these E and B fields are also in phase, with both reaching maxima and minima at the same points in space (see illustrations). A common misconception is that the E and Bfields in electromagnetic radiation are out of phase because a change in one produces the other, and this would produce a phase difference between them as sinusoidal functions (as indeed happens in electromagnetic induction, and in the near-field close to antennas). However, in the far-field EM radiation which is described by the two source-free Maxwell curl operator equations, a more correct description is that a time-change in one type of field is proportional to a space-change in the other. These derivatives require that the E and B fields in EMR are in-phase (see math section below).

An important aspect of light's nature is its frequency. The frequency of a wave is its rate of oscillation and is measured in hertz, the SI unit of frequency, where one hertz is equal to one oscillation per second. Light usually has multiple frequencies that sum to form the resultant wave. Different frequencies undergo different angles of refraction, a phenomenon known as dispersion.

A wave consists of successive troughs and crests, and the distance between two adjacent crests or troughs is called the wavelength. Waves of the electromagnetic spectrum vary in size, from very long radio waves the size of buildings to very short gamma rays smaller than atom nuclei. Frequency is inversely proportional to wavelength, according to the equation:

Electromagnetic waves in free space must be solutions of Maxwell's electromagnetic wave equation. Two main classes of solutions are known, namely plane waves and spherical waves. The plane waves may be viewed as the limiting case of spherical waves at a very large (ideally infinite) distance from the source. Both types of waves can have a waveform which is an arbitrary time function (so long as it is sufficiently differentiable to conform to the wave equation). As with any time function, this can be decomposed by means of Fourier analysis into its frequency spectrum, or individual sinusoidal components, each of which contains a single frequency, amplitude and phase. Such a component wave is said to be monochromatic. A monochromatic electromagnetic wave can be characterized by its frequency or wavelength, its peak amplitude, its phase relative to some reference phase, its direction of propagation and its polarization.

Interference is the superposition of two or more waves resulting in a new wave pattern. If the fields have components in the same direction, they constructively interfere, while opposite directions cause destructive interference. An example of interference caused by EMR is electromagnetic interference (EMI) or as it is more commonly known as, radio-frequency interference (RFI). Additionally, multiple polarization signals can be combined (i.e. interfered) to form new states of polarization, which is known as parallel polarization state generation.

The energy in electromagnetic waves is sometimes called radiant energy.

An anomaly arose in the late 19th century involving a contradiction between the wave theory of light and measurements of the electromagnetic spectra that were being emitted by thermal radiators known as black bodies. Physicists struggled with this problem, which later became known as the ultraviolet catastrophe, unsuccessfully for many years. In 1900, Max Planck developed a new theory of black-body radiation that explained the observed spectrum. Planck's theory was based on the idea that black bodies emit light (and other electromagnetic radiation) only as discrete bundles or packets of energy. These packets were called quanta. Later, Albert Einstein proposed that light quanta be regarded as real particles. Later the particle of light was given the name photon, to correspond with other particles being described around this time, such as the electron and proton. A photon has an energy, E, proportional to its frequency, f.

4.5 Wave and particle effects of electromagnetic radiation:

Together, wave and particle effects fully explain the emission and absorption spectra of EM radiation. The matter-composition of the medium through which the light travels determines the nature of the absorption and emission spectrum. These bands correspond to the allowed energy levels in the atoms. Dark bands in the absorption spectrum are due to the atoms in an intervening medium between source and observer. The atoms absorb certain frequencies of the light between emitter and detector/eye, then emit them in all directions. A dark band appears to the detector, due to the radiation scattered out of the beam. For instance, dark bands in the light emitted by a distant star are due to the atoms in the star's atmosphere. A similar phenomenon occurs for emission, which is seen when an emitting gas glows due to excitation of the atoms from any mechanism, including heat. As electrons descend to lower energy levels, a spectrum is emitted that represents the jumps between the energy levels of the electrons, but lines are seen because again emission happens only at particular energies after excitation. An example is the emission spectrum of nebulae. Rapidly moving electrons are most sharply accelerated when they encounter a region of force, so they are responsible for producing much of the highest frequency electromagnetic radiation observed in nature.

These phenomena can aid various chemical determinations for the composition of gases lit from behind (absorption spectra) and for glowing gases (emission spectra). Spectroscopy (for example) determines what chemical elements comprise a particular star. Spectroscopy is also used in the determination of the distance of a star, using the red shift.

CHAPTER 5

WIRELESS SENSOR NETWORKS

Wireless Sensor Networks (WSNs) can be defined as a self-configured and infrastructureless wireless networks to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location or sink where the data can be observed and analysed. A sink or base station acts like an interface between users and the network. One can retrieve required information from the network by injecting queries and gathering results from the sink. Typically a wireless sensor network contains hundreds of thousands of sensor nodes. The sensor nodes can communicate among themselves using radio signals. A wireless sensor node is equipped with sensing and computing devices, radio transceivers and power components. The individual nodes in a wireless sensor network (WSN) are inherently resource constrained: they have limited processing speed, storage capacity, and communication bandwidth. After the sensor nodes are deployed, they are responsible for self-organizing an appropriate network infrastructure often with multi-hop communication with them. Then the onboard sensors start collecting information of interest. Wireless sensor devices also respond to queries sent from a “control site” to perform specific instructions or provide sensing samples. The working mode of the sensor nodes may be either continuous or event driven. Global Positioning System (GPS) and local positioning algorithms can be used to obtain location and positioning information. Wireless sensor devices can be equipped with actuators to “act” upon certain conditions. These networks are sometimes more specifically referred as Wireless Sensor and Actuator Networks as described in (Akkaya et al., 2005). Wireless sensor networks (WSNs) enable new applications and require non-conventional paradigms for protocol design due to several constraints. Owing to the requirement for low device complexity together with low energy consumption (i.e. long network lifetime), a proper balance between communication and signal/data processing capabilities must be found. This motivates a huge effort in research activities, standardization process, and industrial investments on this field since the last decade (Chiara et. al. 2009). At present time, most of the research on WSNs has concentrated on the design of energy- and computationally efficient algorithms and protocols, and the application domain has been restricted to simple data-oriented monitoring and reporting applications (Labrador et. al. 2009). The authors in (Chen et al., 2011) propose a Cable Mode Transition (CMT) algorithm, which determines the minimal number of active sensors to maintain K-coverage of a terrain as well as K-connectivity of the network. Specifically, it allocates periods of inactivity for cable sensors without affecting the coverage and connectivity requirements of the network based only on local information. In (Cheng et al., 2011), a delay-aware data collection network structure for wireless sensor networks is proposed. The objective of the proposed network structure is to minimize delays in the data collection processes of wireless sensor networks which extends the lifetime of the network. In (Matin et al., 2011), the authors have considered relay nodes to mitigate the network geometric deficiencies and used Particle Swarm Optimization (PSO) based algorithms to locate the optimal sink location with respect to those relay nodes to overcome the lifetime challenge. Energy efficient communication has also been addressed in (Paul et al., 2011; Fabbri et al. 2009). In (Paul et al., 2011), the authors proposed a geometrical solution for locating the optimum sink placement for maximizing the network lifetime. Most of the time, the research on wireless sensor networks have considered homogeneous sensor nodes. But nowadays researchers have focused on heterogeneous sensor networks where the sensor nodes are unlike to each other in terms of their energy. In (Han et al., 2010), the authors addresses the problem of deploying relay nodes to provide fault tolerance with higher network connectivity in heterogeneous wireless sensor networks, where sensor nodes possess different transmission radii. New network architectures with heterogeneous devices and the recent advancement in this technology eliminate the current limitations and expand the spectrum of possible applications for WSNs considerably and all these are changing very rapidly.

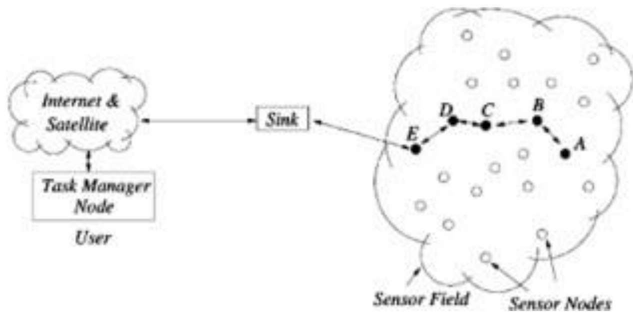


Figure: A typical Wireless Sensor Network

5.1 Applications of wireless sensor network:

Wireless sensor networks have gained considerable popularity due to their flexibility in solving problems in different application domains and have the potential to change our lives in many different ways. WSNs have been successfully applied in various application domains (Akyildiz et al. 2002; Bharathidasan et al., 2001), (Yick et al., 2008; Boukerche, 2009), (Sohraby et al., 2007), and (Chiara et al., 2009; Verdone et al., 2008), such as: Military applications: Wireless sensor networks be likely an integral part of military command, control, communications, computing, intelligence, battlefield surveillance, reconnaissance and targeting systems. Area monitoring: In area monitoring, the sensor nodes are deployed over a region where some phenomenon is to be monitored. When the sensors detect the event being monitored (heat, pressure etc), the event is reported to one of the base stations, which then takes appropriate action. Transportation: Real-time traffic information is being collected by WSNs to later feed transportation models and alert drivers of congestion and traffic problems. Health applications: Some of the health applications for sensor networks are supporting interfaces for the disabled, integrated patient monitoring, diagnostics, and drug administration in hospitals, tele-monitoring of human physiological data, and tracking & monitoring doctors or patients inside a hospital. Environmental sensing: The term Environmental Sensor Networks has developed to cover many applications of WSNs to earth science research. This includes sensing volcanoes, oceans, glaciers, forests etc. Some other major areas are listed below:

Air pollution monitoring Forest fires detection

- Greenhouse monitoring

- Landslide detection

- Structural monitoring: Wireless sensors can be utilized to monitor the movement within buildings and infrastructure such as bridges, flyovers, embankments, tunnels etc enabling Engineering practices to monitor assets remotely with out the need for costly sitevisits. Industrial monitoring: Wireless sensor networks have been developed for machinery condition-based maintenance (CBM) as they offer significant cost savings and enable new functionalities. In wired systems, the installation of enough sensors is often limited by the cost of wiring. Agricultural sector: using a wireless network frees the farmer from the maintenance of wiring in a difficult environment. Irrigation automation enables more efficient water use and reduces waste.

5.2 Design issues of a wireless sensor network:

There are a lot of challenges placed by the deployment of sensor networks which are a superset of those found in wireless ad hoc networks. Sensor nodes communicate over wireless, lossy lines with no infrastructure. An additional challenge is related to the limited, usually non-renewable energy supply of the sensor nodes. In order to maximize the lifetime of the network, the protocols need to be designed from the beginning with the objective of efficient management of the energy resources (Akyildiz et al., 2002). Wireless Sensor Network Design issues are mentioned in (Akkaya et al., 2005), (Akyildizet al., 2002), (SensorSim; Tossim, Younis et al., 2004), (Pan et al., 2003) and different possible platforms for simulation and testing of routing protocols for WSNs are discussed in (NS-2, Zeng et al., 1998, SensorSim, Tossiim). Let us now discuss the individual design issues in greater detail. Fault Tolerance: Sensor nodes are vulnerable and frequently deployed in dangerous environment. Nodes can fail due to hardware problems or physical damage or by exhausting their energy supply. We expect the node failures to be much higher than the one normally considered in wired or infrastructure-based wireless networks. The protocols deployed in a sensor network should be able to detect these failures as soon as possible and be robust enough to handle a relatively large number of failures while maintaining the overall functionality of the network. This is especially relevant to the routing protocol design, which has to ensure that alternate paths are available for rerouting of the packets. Different deployment environments pose different fault tolerance requirements. Scalability: Sensor networks vary in scale from several nodes to potentially several hundred thousand. In addition, the deployment density is also variable.

For collecting high resolution data, the node density might reach the level where a node has several thousand neighbours in their transmission range. The protocols deployed in sensor networks need to be scalable to these levels and be able to maintain adequate performance. Production Costs: Because many deployment models consider the sensor nodes to be disposable devices, sensor networks can compete with traditional information gathering approaches only if the individual sensor nodes can be produced very cheaply. The target price envisioned for a sensor node should ideally be less than $1. Hardware Constraints: At minimum, every sensor node needs to have a sensing unit, a processing unit, a transmission unit, and a power supply. Optionally, the nodes may have several built-in sensors or additional devices such as a localization system to enable location-aware routing. However, every additional functionality comes with additional cost and increases the power consumption and physical size of the node. Thus, additional functionality needs to be always balanced against cost and low-power requirements. Sensor Network Topology: Although WSNs have evolved in many aspects, they continue to be networks with constrained resources in terms of energy, computing power, memory, and communications capabilities. Of these constraints, energy consumption is of paramount importance, which is demonstrated by the large number of algorithms, techniques, and protocols that have been developed to save energy, and thereby extend the lifetime of the network. Topology Maintenance is one of the most important issues researched to reduce energy consumption in wireless sensor networks. Transmission Media: The communication between the nodes is normally implemented using radio communication over the popular ISM bands. However, some sensor networks use optical or infrared communication, with the latter having the advantage of being robust and virtually interference free. Power Consumption: As we have already seen, many of the challenges of sensor networks revolve around the limited power resources. The size of the nodes limits the size of the battery. The software and hardware design needs to carefully consider the issues of efficient energy use. For instance, data compression might reduce the amount of energy used for radio transmission, but uses additional energy for computation and/or filtering. The energy policy also depends on the application; in some applications, it might be acceptable to turn off a subset of nodes in order to conserve energy while other applications require all nodes operating simultaneously.

5.3 Structure of a wireless sensor network:

Structure of a Wireless Sensor Network includes different topologies for radio communications networks. A short discussion of the network topologies that apply to wireless sensor networks are outlined below:

Star network (single point-to-multipoint) (Wilson, 2005) A star network is a communications topology where a single base station can send and/or receive a message to a number of remote nodes. The remote nodes are not permitted to send messages to each other. The advantage of this type of network for wireless sensor networks includes simplicity, ability to keep the remote node's power consumption to a minimum. It also allows low latency communications between the remote node and the base station. The disadvantage of such a network is that the base station must be within radio transmission range of all the individual nodes and is not as robust as other networks due to its dependency on a single node to manage the network.

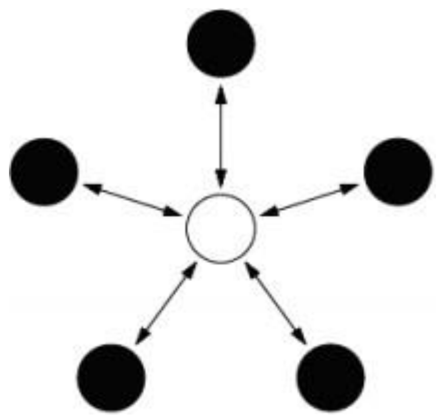


Figure : A Star network topology

5.4 Mesh network:

A mesh network allows transmitting data to one node to other node in the network that is within its radio transmission range. This allows for what is known as multi-hop communications, that is, if a node wants to send a message to another node that is out of radio communications range, it can use an intermediate node to forward the message to the desired node. This network topology has the advantage of redundancy and scalability. If an individual node fails, a remote node still can communicate to any other node in its range, which in turn, can forward the message to the desired location. In addition, the range of the network is not necessarily limited by the range in between single nodes; it can simply be extended by adding more nodes to the system. The disadvantage of this type of network is in power consumption for the nodes that implement the multi-hop communications are generally higher than for the nodes that don't have this capability, often limiting the battery life. Additionally, as the number of communication hops to a destination increases, the time to deliver the message also increases, especially if low power operation of the nodes is a requirement.

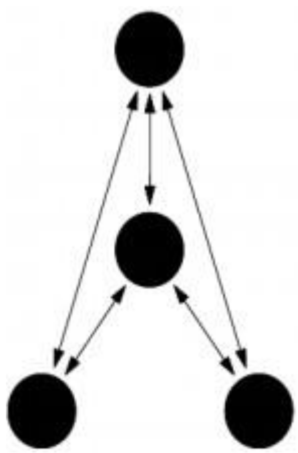


Figure: A Mesh network topology

5.5 Hybrid star – Mesh network:

A hybrid between the star and mesh network provides a robust and versatile communications network, while maintaining the ability to keep the wireless sensor nodes power consumption to a minimum. In this network topology, the sensor nodes with lowest power are not enabled with the ability to forward messages. This allows for minimal power consumption to be maintained. However, other nodes on the network are enabled with multi-hop capability, allowing them to forward messages from the low power nodes to other nodes on the network. Generally, the nodes with the multi-hop capability are higher power, and if possible, are often plugged into the electrical mains line. This is the topology implemented by the up and coming mesh networking standard known as ZigBee.

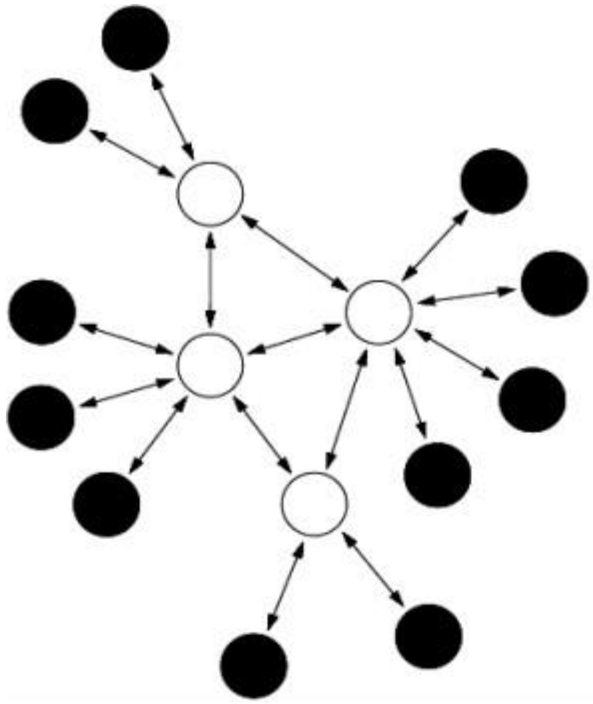


Figure: A Hybrid Star – Mesh network topology

Structure of a wireless sensor node: A sensor node is made up of four basic components such as sensing unit, processing unit, transceiver unit and a power unit which is shown in Fig. 5. It also has application dependent additional components such as a location finding system, a power generator and a mobilizer. Sensing units are usually composed of two subunits: sensors and analogue to digital converters (ADCs) (Akyildiz et al., 2002). The analogue signals produced by the sensors are converted to digital signals by the ADC, and then fed into the processing unit. The processing unit is generally associated with a small storage unit and it can manage the procedures that make the sensor node collaborate with the other nodes to carry out the assigned sensing tasks. A transceiver unit connects the node to the network. One of the most important components of a sensor node is the power unit. Power units can be supported by a power scavenging unit such as solar cells. The other subunits, of the node are application dependent. A functional block diagram of a versatile wireless sensing node is provided in Fig. 6. Modular design approach provides a flexible and versatile platform to address the needs of a wide variety of applications. For example, depending on the sensors to be deployed, the signal conditioning block can be re-programmed or replaced. This allows for a wide variety of different sensors to be used with the wireless sensing node. Similarly, the radio link may be swapped out as required for a given applications' wireless range requirement and the need for bidirectional communications.

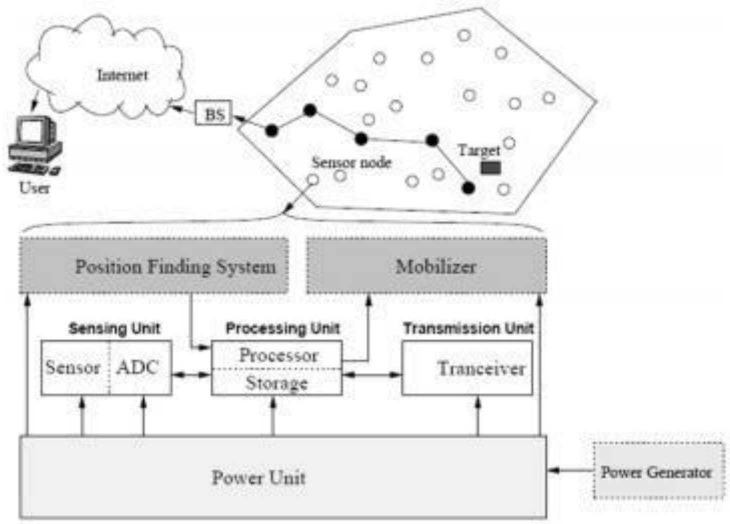


Figure: The components of a sensor node

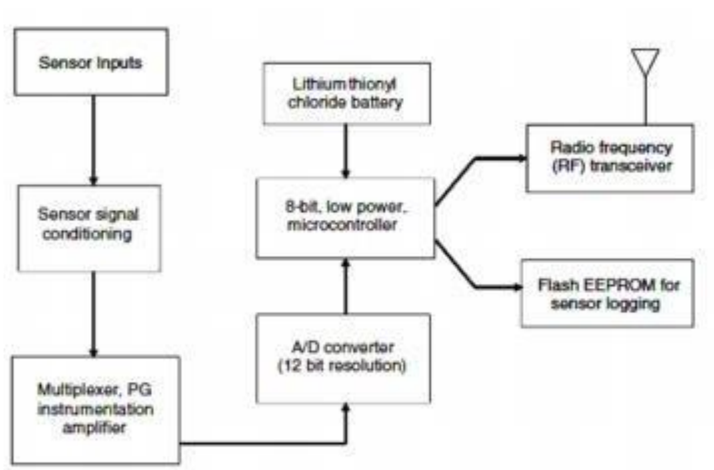


Figure: Functional block diagram of a sensor node

Using flash memory, the remote nodes acquire data on command from a base station, or by an event sensed by one or more inputs to the node. Moreover, the embedded firmware can be upgraded through the wireless network in the field. The microprocessor has a number of functions including:

- Managing data collection from the sensors

- performing power management functions

- interfacing the sensor data to the physical radio layer

- managing the radio network protocol

- A key aspect of any wireless sensing node is to minimize the power consumed by the system.

Usually, the radio subsystem requires the largest amount of power. Therefore, data is sent over the radio network only when it is required. An algorithm is to be loaded into the node to determine when to send data based on the sensed event. Furthermore, it is important to minimize the power consumed by the sensor itself. Therefore, the hardware should be designed to allow the microprocessor to judiciously control power to the radio, sensor, and sensor signal conditioner (Akyildiz et al., 2002). 6. Communication structure of a wireless sensor network The sensor nodes are usually scattered in a sensor field as shown in Fig. 1. Each of these scattered sensor nodes has the capabilities to collect data and route data back to the sink and the end users. Data are routed back to the end user by a multi-hop infrastructure-less architecture through the sink as shown in Fig. 1. The sink may communicate with the task manager node via Internet or Satellite.

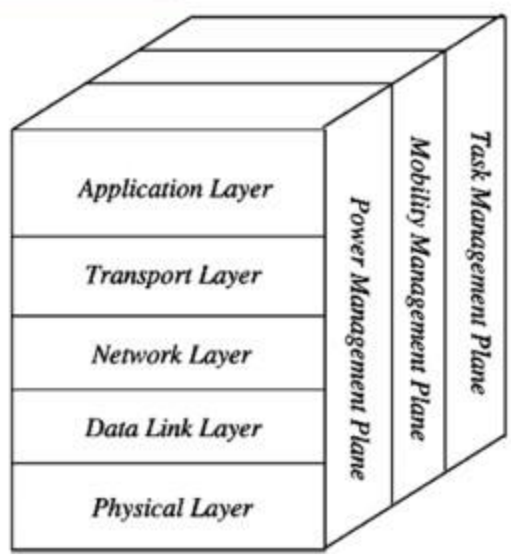


Figure: Wireless Sensor Network protocol stack

The protocol stack used by the sink and the sensor nodes is given in Fig. 7. This protocol stack combines power and routing awareness, integrates data with networking protocols, communicates power efficiently through the wireless medium and promotes cooperative efforts of sensor nodes. The protocol stack consists of the application layer, transport layer network layer, data link layer, physical layer, power management plane, mobility management plane, and task management plane (Akyildiz et al., 2002). Different types of application software can be built and used on the application layer depending on the sensing tasks. This layer makes hardware and software of the lowest layer transparent to the end-user. The transport layer helps to maintain the flow of data if the sensor networks application requires it. The network layer takes care of routing the data supplied by the transport layer, specific multi-hop wireless routing protocols between sensor nodes and sink. The data link layer is responsible for multiplexing of data streams, frame detection, Media Access Control (MAC) and error control. Since the environment is noisy and sensor nodes can be mobile, the MAC protocol must be power aware and able to minimize collision with neighbours' broadcast. The physical layer addresses the needs of a simple but robust modulation, frequency selection, data encryption, transmission and receiving techniques. In addition, the power, mobility, and task management planes monitor the power, movement, and task distribution among the sensor nodes. These planes help the sensor nodes coordinate the sensing task and lower the overall energy consumption.

Energy consumption issues in wireless sensor network Energy consumption is the most important factor to determine the life of a se ensor network because usually sensor nodes are driven by battery. Sometimes energy optimization is more complicated in sensor networks because it involved not only reduction of energy consumption but also prolonging the life of the network as much as possible. The optimization can be done by having energy awareness in every aspect of design and operation. This ensures that energy awareness is also incorporated into groups of communicating sensor nodes and the entire network and not only in the individual nodes (Bharathidasan et al. 2001). A sensor node usually consists of four sub-systems (Bharathidasan et al. 2001): a computing subsystem : It consists of a microprocessor(microcontroller unit, MCU)

- which is responsible for the control of the sensors and implementation of communication protocols. MCUs usually operate under various modes for power management purposes. As these operating modes involves consumption of power, the energy consumption levels of the various modes should be considered while looking at the battery lifetime of each node. a communication subsystem: It consists of a short range radio which communicate with

• neighboring nodes and the outside world. Radios can operate under the different modes. It is important to completely shut down the radio rather than putting it in the Idle mode when it is not transmitting or receiving for saving power. a sensing subsystem : It consists of a group of sensors and actuators and link the node

• to the outside world. Energy consumption can be reduced by using low power components and saving power at the cost of performance which is not required. a power supply subsystem : It consists of a battery which supplies power to the node.

It should be seen that the amount of power drawn from a battery is checked because if high current is drawn from a battery for a long time, the battery will die faster even though it could have gone on for a longer time. Usually the rated current capacity of a battery being used for a sensor node is less than the minimum energy consumption. The lifetime of a battery can be increased by reducing the current drastically or even turning it off often. To minimize the overall energy consumption of the sensor network, different types of protocols and algorithms have been studied so far all over the world. The lifetime of a sensor network can be increased significantly if the operating system, the application layer and the network protocols are designed to be energy aware.

These protocols and algorithms have to be aware of the hardware and able to use special features of the micro-processors and transceivers to minimize the sensor node's energy consumption. This may push toward a custom solution for different types of sensor node design. Different types of sensor nodes deployed also lead to different types of sensor networks. This may also lead to the different types of collaborative algorithms in wireless sensor networks arena. 8. Protocols & algorithms of wireless sensor network In WSN, the main task of a sensor node is to sense data and sends it to the base station in multi hop environment for which routing path is essential. For computing the routing path from the source node to the base station there is huge numbers of proposed routing protocols exist (Sharma et al., 2011).

The design of routing protocols for WSNs must consider the power and resource limitations of the network nodes, the time-varying quality of the wireless channel, and the possibility for packet loss and delay. To address these design requirements, several routing strategies for WSNs have been proposed in (Labrador et al., 2009), (Akkaya et al., 2005), (Akyildiz et al. 2002), (Boukerche, 2009, Al-karaki et al., 2004, Pan et al., 2003) and (Waharte et al., 2006). The first class of routing protocols adopts a flat network architecture in which all nodes are considered peers. Flat network architecture has several advantages, including minimal overhead to maintain the infrastructure and the potential for the discovery of multiple routes between communicating nodes for fault tolerance.

A second class of routing protocols imposes a structure on the network to achieve energy efficiency, stability, and scalability. In this class of protocols, network nodes are organized in clusters in which a node with higher residual energy, for example, assumes the role of a cluster head. The cluster head is responsible for coordinating activities within the cluster and forwarding information between clusters. Clustering has potential to reduce energy consumption and extend the lifetime of the network. A third class of routing protocols uses a data-centric approach to disseminate interest within the network. The approach uses attribute-based naming, whereby a source node queries an attribute for the phenomenon rather than an individual sensor node. The interest dissemination is achieved by assigning tasks to sensor nodes and expressing queries to relative to specific attributes. Different strategies can be used to communicate interests to the sensor nodes, including broadcasting, attribute-based multicasting, geo-casting, and any casting.

A fourth class of routing protocols uses location to address a sensor node. Location-based routing is useful in applications where the position of the node within the geographical coverage of the network is relevant to the query issued by the source node. Such a query may specify a specific area where a phenomenon of interest may occur or the vicinity to a specific point in the network environment. In the rest of this section we discuss some of the major routing protocols and algorithms to deal with the energy conservation issue in the literatures.

5.6 Flooding:

Flooding is a common technique frequently used for path discovery and information dissemination in wired and wireless ad hoc networks which has been discussed in (Akyildiz et al., 2002). The routing strategy of flooding is simple and does not rely on costly network topology maintenance and complex route discovery algorithms. Flooding uses a reactive approach whereby each node receiving a data or control packet sends the packet to all its neighbors. After transmission, a packet follows all possible paths. Unless the network is disconnected, the packet will eventually reach its destination. Furthermore, as the network topology changes, the packet transmitted follows the new routes. Fig. 8 illustrates the concept of flooding in data communications network. As shown in the figure, flooding in its simplest form may cause packets to be replicated indefinitely by network nodes.

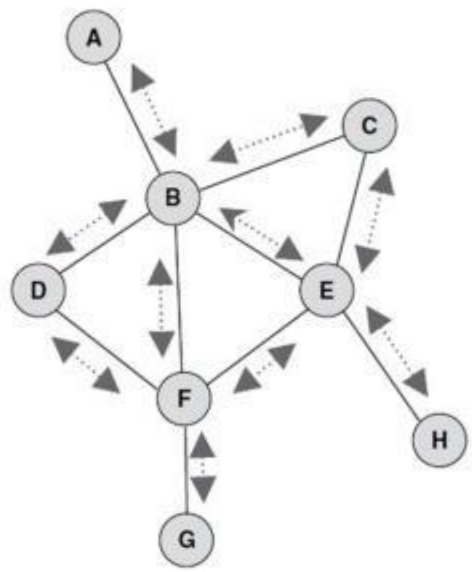


Figure : Flooding in data communication networks

5.7 Gossiping:

To address the shortcomings of flooding, a derivative approach, referred to as gossiping, has been proposed in (Braginsky et al., 2002). Similar to flooding, gossiping uses a simple forwarding rule and does not require costly topology maintenance or complex route discovery algorithms. Contrary to flooding, where a data packet is broadcast to all neighbors, gossiping requires that each node sends the incoming packet to a randomly selected neighbor. Upon receiving the packet, the neighbor selected randomly chooses one of its own neighbors and forwards the packet to the neighbor chosen. This process continues iteratively until the packet reaches its intended destination or the maximum hop count is exceeded. 2. Protocols for Information via Negotiation (SPIN): Sensor protocols for information via negotiation (SPIN), is a data-centric negotiation-based family of information dissemination protocols for WSNs (Kulik et al., 2002). The main objective of these protocols is to efficiently disseminate observations gathered by individual sensor nodes to all the sensor nodes in the network. Simple protocols such as flooding and gossiping are commonly proposed to achieve information dissemination in WSNs. Flooding requires that each node sends a copy of the data packet to all its neighbors until the information reaches all nodes in the network. Gossiping, on the other hand, uses randomization to reduce the number of duplicate packets and requires only that a node receiving a data packet forward it to a randomly selected neighbor.

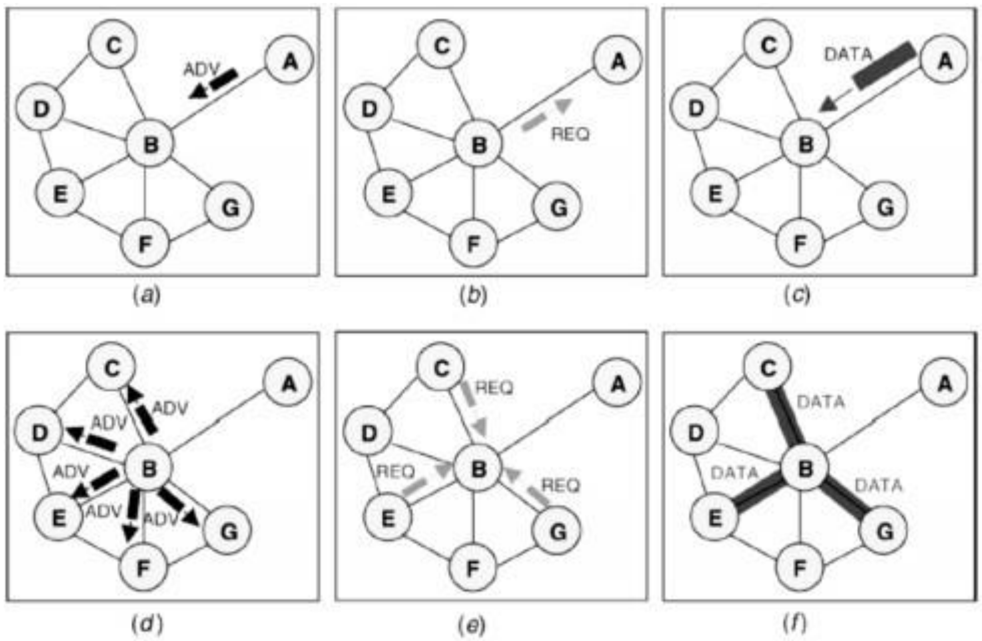


Figure. SPIN basic protocol operation

5.8 Low-Energy Adaptive Clustering Hierarchy (LEACH)

Low-energy adaptive clustering hierarchy (LEACH) is a routing algorithm designed to collect and deliver data to the data sink, typically a base station (Heinzelman et. al. 2000).

The main objectives of LEACH are:

- Extension of the network lifetime
- Reduced energy consumption by each network sensor node
- Use of data aggregation to reduce the number of communication messages

To achieve these objectives, LEACH adopts a hierarchical approach to organize the network into a set of clusters. Each cluster is managed by a selected cluster head. The cluster head assumes the responsibility to carry out multiple tasks. The first task consists of periodic collection of data from the members of the cluster. Upon gathering the data, the cluster head aggregates it in an effort to remove redundancy among correlated values. The second main task of a cluster head is to transmit the aggregated data directly to the base station over single hop. The third main task of the cluster head is to create a TDMA-based schedule whereby each node of the cluster is assigned a time slot that it can use for transmission. The cluster head announces the schedule to its cluster members through broadcasting.

To reduce the likelihood of collisions among sensors within and outside the cluster, LEACH nodes use a code-division multiple access–based scheme for communication. The basic operations of LEACH are organized in two distinct phases. The first phase, the setup phase, consists of two steps, cluster-head selection and cluster formation. The second phase, the steady-state phase, focuses on data collection, aggregation, and delivery to the base station. The duration of the setup is assumed to be relatively shorter than the steadystate phase to minimize the protocol overhead. At the beginning of the setup phase, a round of cluster-head selection starts. To decide whether a node to become cluster head or not a threshold T(s) is addressed in (Heinzelman et. al. 2000) which is as follows:

$$T(s) = \begin{cases} \dfrac{p_{opt}}{1 - p_{opt}.(r.\mathrm{mod}.\dfrac{1}{p_{opt}})}, ifs \in G' \\ 0, otherwise \end{cases}$$

Where r is the current round number and G is the set of nodes that have not become cluster head within the last 1/popt rounds. At the beginning of each round, each node which belongs to the set

G selects a random number 0 or 1. If the random number is less than the threshold T(s) then the node becomes a cluster head in the current round.

5.9 Threshold-sensitive Energy Efficient Protocols (TEEN and APTEEN):

Two hierarchical routing protocols called TEEN (Threshold-sensitive Energy Efficient sensor Network protocol), and APTEEN (Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network protocol) are proposed in (Manjeshwar et al., 2001) and (Manjeshwar et al., 2002) , respectively. These protocols were proposed for time-critical applications. In TEEN, sensor nodes sense the medium continuously, but the data transmission is done less frequently. A cluster head sensor sends its members a hard threshold, which is the threshold value of the sensed attribute and a soft threshold, which is a small change in the value of the sensed attribute that triggers the node to switch on its transmitter and transmit. Thus the hard threshold tries to reduce the number of transmissions by allowing the nodes to transmit only when the sensed attribute is in the range of interest. The soft threshold further reduces the number of transmissions that might have otherwise occurred when there is little or no change in the sensed attribute. A smaller value of the soft threshold gives a more accurate picture of the network, at the expense of increased energy consumption. Thus, the user can control the trade-off between energy efficiency and data accuracy. When cluster-heads are to change, new values for the above parameters are broadcast. The main drawback of this scheme is that, if the thresholds are not received, the nodes will never communicate, and the user will not get any data from the network at all.

5.10 Power-Efficient Gathering in Sensor Information Systems (PEGASIS):

Power-efficient gathering in sensor information systems (PEGASIS) (Lindsey et al., 2002) and its extension, hierarchical PEGASIS, are a family of routing and information-gathering protocols for WSNs. The main objectives of PEGASIS are twofold. First, the protocol aims at extending the lifetime of a network by achieving a high level of energy efficiency and uniform energy consumption across all network nodes. Second, the protocol strives to reduce the delay that data incur on their way to the sink. The network model considered by PEGASIS assumes a homogeneous set of nodes deployed across a geographical area. Nodes are assumed to have global knowledge about other sensors' positions. Furthermore, they have the ability to control their power to cover arbitrary ranges. The nodes may also be equipped with CDMA-capable radio transceivers. The nodes' responsibility is to gather and deliver data to a sink, typically a wireless base station. The goal is to develop a routing structure and an aggregation scheme to reduce energy consumption and deliver the aggregated data to the base station with minimal delay while balancing energy consumption among the sensor nodes. Contrary to other protocols, which rely on a tree structure or a cluster-based hierarchical organization of the network for data gathering and dissemination, PEGASIS uses a chain structure.

5.11 Directed Diffusion:

Directed diffusion (Intanagonwiwat et al., 2000) is a data-centric routing protocol for information gathering and dissemination in WSNs. The main objective of the protocol is to achieve substantial energy savings in order to extend the lifetime of the network. To achieve this objective, directed diffusion keeps interactions between nodes, in terms of message exchanges, localized within limited network vicinity. Using localized interaction, direct diffusion can still realize robust multi-path delivery and adapt to a minimal subset of network paths. This unique feature of the protocol, combined with the ability of the nodes to aggregate response to queries, results into significant energy savings.

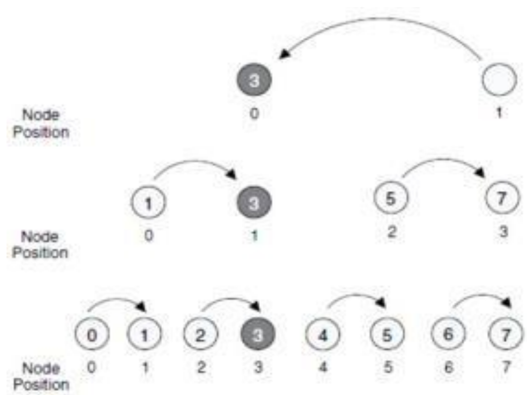


Figure 10. Chain-based data gathering and aggregation scheme

The main elements of direct diffusion include interests, data messages, gradients, and reinforcements. Directed diffusion uses a publish-and-subscribe information model in which an inquirer expresses an interest using attribute–value pairs. An interest can be viewed as a query or an interrogation that specifies what the inquirer wants.

5.12 Geographic Adaptive Fidelity (GAF):

GAF (Xu et al., 2001) is an energy-aware location-based routing algorithm designed mainly for mobile ad hoc networks, but may be applicable to sensor networks as well. The network area is first divided into fixed zones and forms a virtual grid. Inside each zone, nodes collaborate with each other to play different roles. For example, nodes will elect one sensor node to stay awake for a certain period of time and then they go to sleep. This node is responsible for monitoring and reporting data to the BS on behalf of the nodes in the zone. Hence, GAF conserves energy by turning off unnecessary nodes in the network without affecting the level of routing fidelity.

5.13 Security issues in wireless sensor network :

Security issues in sensor networks depend on the need to know what we are going to protect. In (Zia et al., 2006), the authors defined four security goals in sensor networks which are Confidentiality, Integrity, Authentication and Availability. Another security goal in sensor network is introduced in (Sharma et al., 2011).Confidentiality is the ability to conceal message from a passive attacker, where the message communicated on sensor networks remain confidential. Integrity refers to the ability to confirm the message has not been tampered, altered or changed while it was on the network. Authentication Need to know if the messages are from the node it claims to be from, determining the reliability of message's origin. Availability is to determine if a node has the ability to use the resources and the network is available for the messages to move on.

Freshness implies that receiver receives the recent and fresh data and ensures that no adversary can replay the old data. This requirement is especially important when the WSN nodes use shared-keys for message communication, where a potential adversary can launch a replay attack using the old key as the new key is being refreshed and propagated to all the nodes in the WSN (Sen, 2009). To achieve the freshness the mechanism like nonce or time stamp should add to each data packet. Having built a foundation of security goals in sensor network, the major possible security attacks in sensor networks are identified in (Undercoffer et al., 2002) . Routing loops attacks target the information exchanged between nodes. False error messages are generated when an attacker alters and replays the routing information. Routing loops attract or repel the network traffic and increases node to node latency.

Selective forwarding attack influences the network traffic by believing that all the participating nodes in network are reliable to forward the message. In selective forwarding attack malicious nodes simply drop certain messages instead of forwarding every message. Once a malicious node cherry picks on the messages, it reduces the latency and deceives the neighboring nodes that they are on a shorter route. Effectiveness of this attack depends on two factors. First the location of the malicious node, the closer it is to the base stations the more traffic it will attract. Second is the percentage of messages it drops. When selective forwarder drops more messages and forwards less, it retains its energy level thus remaining powerful to trick the neighboring nodes. In sinkhole attacks, adversary attracts the traffic to a compromised node. The simplest way of creating sinkhole is to place a malicious node where it can attract most of the traffic, possibly closer to the base station or malicious node itself deceiving as a base station. One reason for sinkhole attacks is to make selective forwarding possible to attract the traffic towards a compromised node.

The nature of sensor networks where all the traffic flows towards one base station makes this type of attacks more susceptible. Sybil attacks are a type of attacks where a node creates multiple illegitimate identities in sensor networks either by fabricating or stealing the identities of legitimate nodes. Sybil attacks can be used against routing algorithms and topology maintenance; it reduces the effectiveness of fault tolerant schemes such as distributed storage and disparity. Another malicious factor is geographic routing where a Sybil node can appear at more than one place simultaneously. In wormhole attacks an adversary positioned closer to the base station can completely disrupt the traffic by tunneling messages over a low latency link. Here an adversary convinces the nodes which are multi hop away that they are closer to the base station. This creates a sinkhole because adversary on the other side of the sinkhole provides a better route to the base station.

In Hello flood attacks a Broadcasted message with stronger transmission power is pretending that the HELLO message is coming from the base station. Message receiving nodes assume that the HELLO message sending node is the closest one and they try to send all their messages through this node. In this type of attacks all nodes will be responding to HELLO floods and wasting the energies. The real base station will also be broadcasting the similar messages but will have only few nodes responding to it. Denial of service (DoS) attacks occur at physical level causing radio jamming, interfering with the network protocol, battery exhaustion etc. An specific type of DoS attack, Denial-of-service attack has been explored in (Raymond et al., 2009), in which a sensor node's power supply is targeted. Attacks of this type can reduce the sensor lifetime from years to days and have a devastating impact on a sensor network.

5.14 Layering based security approach

5.14.1 Application layer:

Data is collected and managed at application layer therefore it is important to ensure the reliability of data. Wagner (Wanger, 2004) has presented a resilient aggregation scheme which is applicable to a cluster based network where a cluster leader acts as an aggregator in sensor networks. However this technique is applicable if the aggregating node is in the range with all the source nodes and there is no intervening aggregator between the aggregator and source nodes. To prove the validity of the aggregation, cluster leaders use the cryptographic techniques to ensure the data reliability.

5.14.2 Network layer:

Network layer is responsible for routing of messages from node to node, node to cluster leader, cluster leaders to cluster leaders, cluster leaders to the base station and vice versa.

5.15.2 Data link layer:

Data link layer does the error detection and correction, and encoding of data. Link layer is vulnerable to jamming and DoS attacks. TinySec (Karlof et al., 2004) has introduced link layer encryption which depends on a key management scheme. However, an attacker having better energy efficiency can still rage an attack. Protocols like LMAC (Hoesel et al., 2004) have better anti-jamming properties which are viable countermeasure at this layer.

5.15.3 Physical Layer:

The physical layer emphasizes on the transmission media between sending and receiving nodes, the data rate, signal strength, frequency types are also addressed in this layer. Ideally FHSS frequency hopping spread spectrum is used in sensor networks.

CHAPTER 6

RESULT AND CONCLUSION:

6.1 RESULTS:

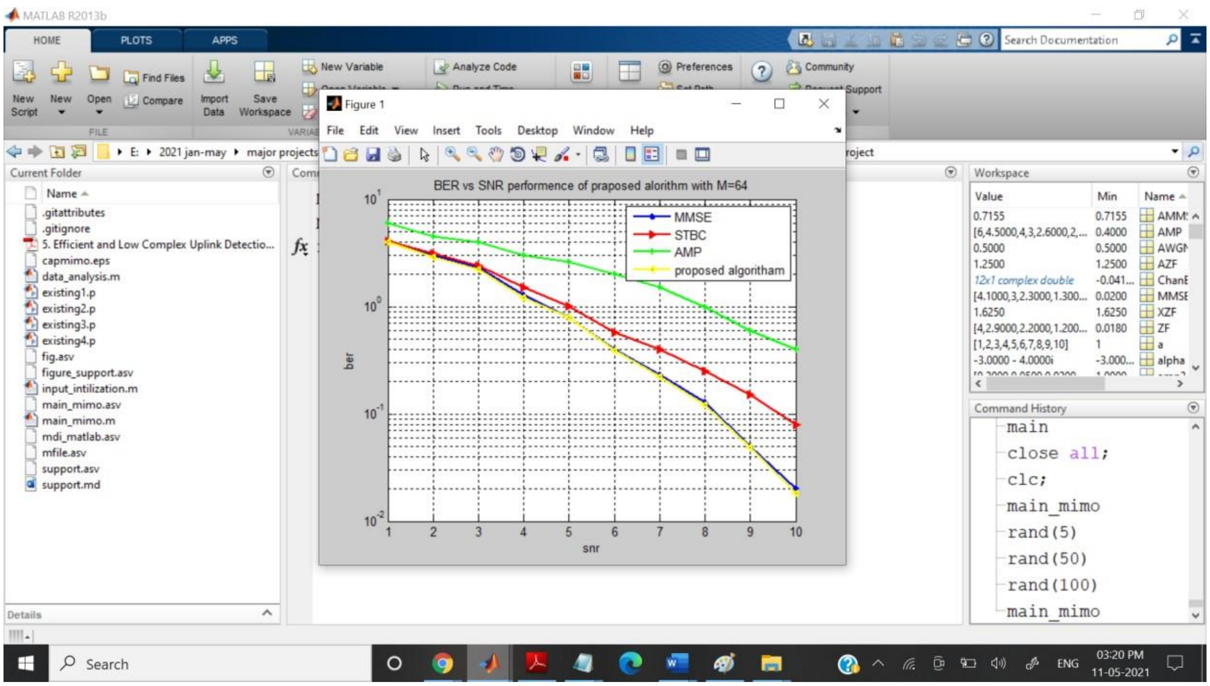


Fig:BER vs SNR performance of proposed algorithm with M=64

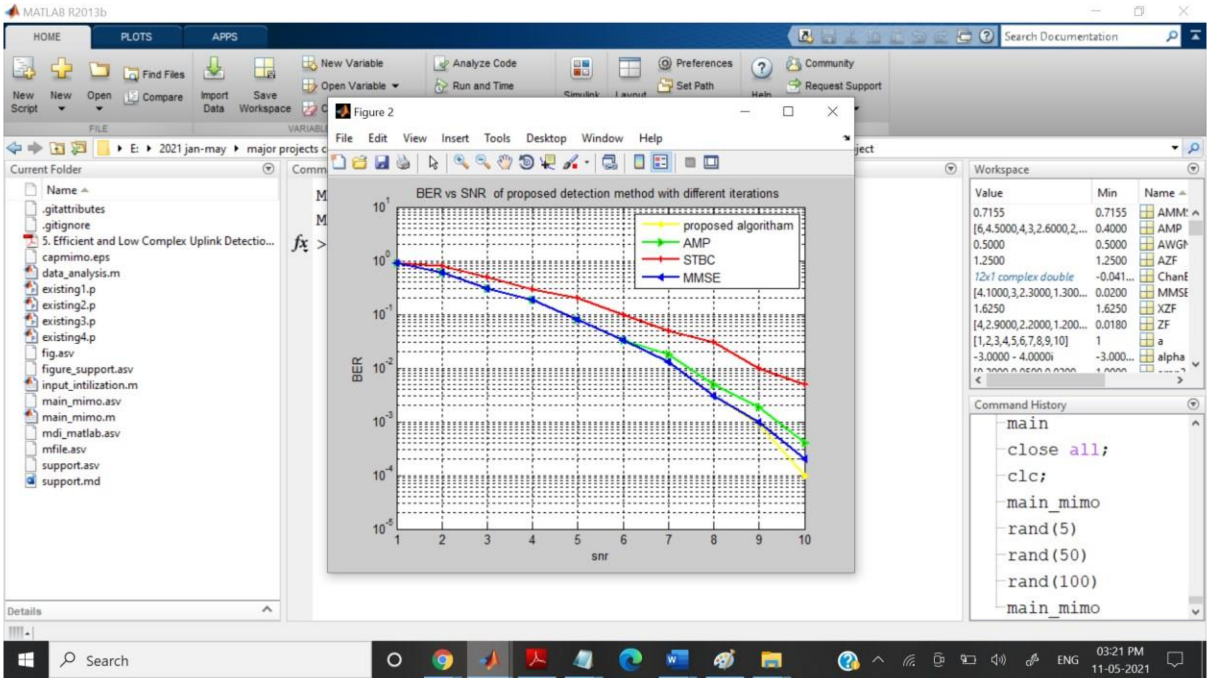


Fig: BER vs SNR of proposed detection method with different iterations

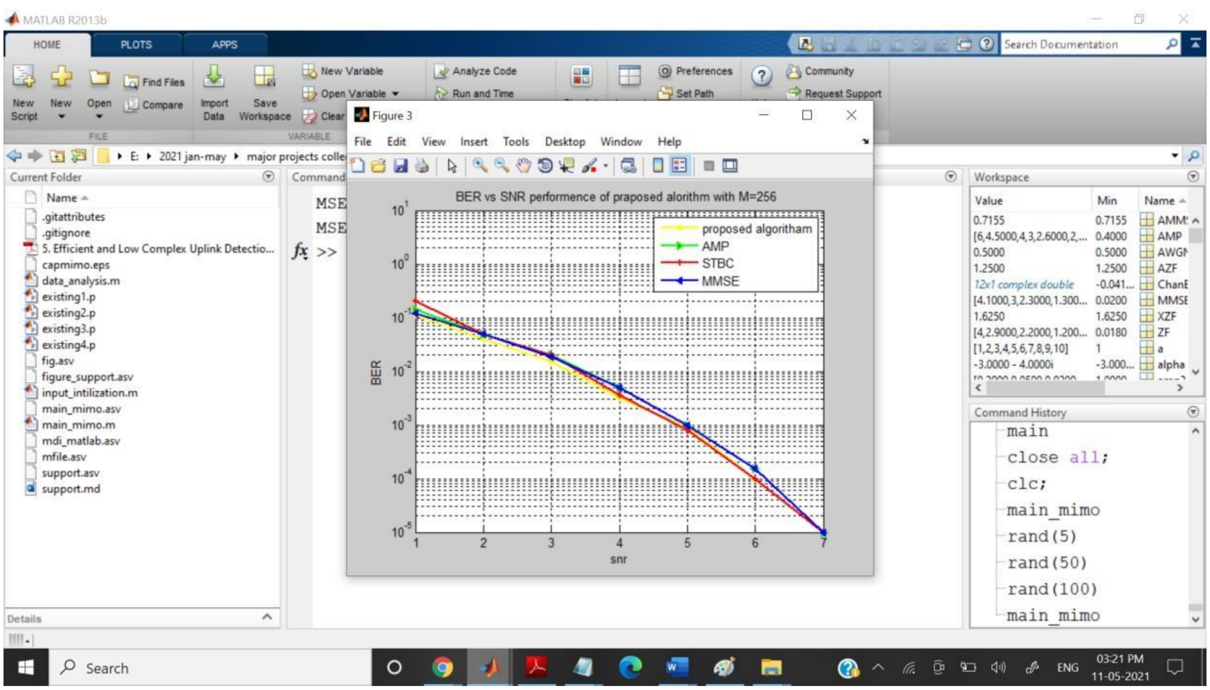


Fig: BER vs SNR performance of proposed algorithm with M=256

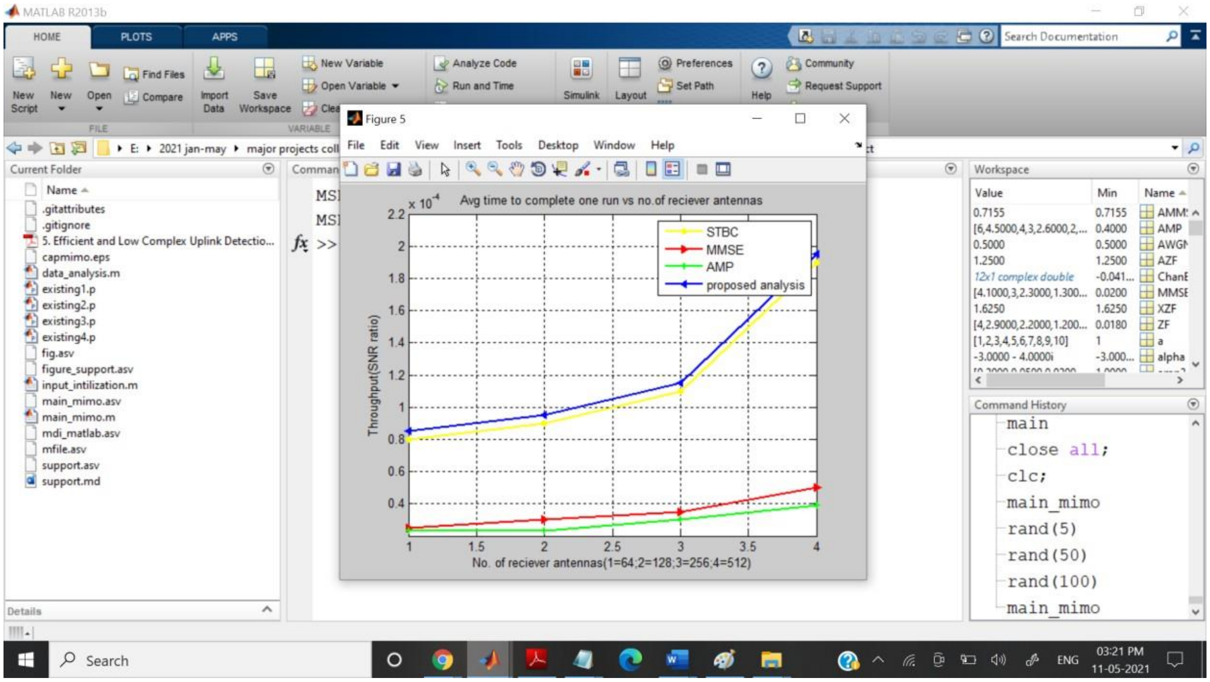


Fig: Aug time to complete one run vs no of receiver antennas

CONCLUSION:

This article is a humble attempt to provide a forward looking research roadmap for 6G. New features of the 6G evolution were identified, and enabling technologies were discussed. While a partial picture was presented, we hope our discussion will spur interests and further investigations on the future evolution of cellular networks.

REFERENCES:

[1] A.Nasir, X. Zhou, S. Durrani, and R. A. Kennedy, “Relaying protocols for wireless energy harvesting and information processing,” IEEE Trans. Wireless Commun., vol. 12, no. 7, pp. 3622–3636, 2013.

[2] B.Medepally and N. B. Mehta, “Voluntary energy harvesting relays and selection in cooperative wireless networks,” IEEE Trans. Wireless Commun., vol. 9, no. 11, pp. 3543–3553, Nov. 2010.

[3] C. K. Ho and R. Zhang, “Optimal energy allocation for wireless communications with energy harvesting constraints,” IEEE Trans. Signal Process., vol. 60, no. 9, pp. 4808–4818, Sept. 2012.

[4] J. Yang and S. Ulukus, “Optimal packet scheduling in an energy harvesting communication system,” IEEE Trans. Commun., vol. 60, no.1, pp. 220–230, Jan. 2012.

[5] K. Tutuncuoglu and A. Yener, “Optimum transmission policies for battery limited energy harvesting nodes,” IEEE Trans. Wireless Commun., vol. 11, no. 3, pp. 1180–1189, Mar. 2012.

[6] K. Chen et al., “Ultra-Low Latency Mobile Networking,” IEEE Network, vol. 33, Mar. 2018, pp. 181–87.

[7] J. Andrews et al., “What Will 5G Be?” IEEE JSAC, vol. 32, June 2014, pp. 1065–82. [3] K. David and H. Berndt, “6G Vision and Requirements: Is There Any Need for Beyond 5G?,” IEEE Veh. Technol. Mag., vol. 13, Sept. 2018, pp. 72–80.

[8] T. S. Rappaport et al., “Wireless Communications and Applications above 100 GHz: Opportunities and Challenges for 6G and Beyond,” IEEE Access, vol. 7, July 2019, pp. 78729–57.

[9] B. McMahan et al., “Communication-Efficient Learning of Deep Networks from Decentralized Data,” Proc. Int’l. Conf. Artificial Intell. Stat. (AISTATS), vol. 54, 2017, pp. 1273–82

A Project Report

On

Minimum Time Delay and More Efficient Image Filtering Brain Tumor Detection with the help of MATLAB

Submitted By

G. Vishwas (17K81A04E7)

A. Mary Sophia (17K81A04C3)

B. Srinish Kumar (17K81A04D5)

in partial fulfilment for the award of degree

of

BACHELOR OF TECHNOLOGY

IN

Electronics And Communication Engineering

Under The Guidance of
Mr.K.Nishakar
M. Tech, (Ph.D)
Associate Professor
Department of Electronics & Communication Engineering

St. MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad-500 100.

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE

(Permanently Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An Autonomous Institute

NBA & NAAC A+ Accredited

Dhulapally, Secunderabad-500 100.

Department of Electronics & Communication Engineering

BONAFIDE CERTIFICATE

This is to certify that the project entitled **"Minimum Time Delay and More Efficient Image Filtering Brain Tumor Detection With the Help Of MATLAB"**, is being submitted by **Mr.G.Vishwas (17K81A04E7), Ms.A.Mary Sophia (17K81A04C3), Mr.B.Srinish Kumar (17K81A04D5)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN Electronics And Communication Engineering** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory

INTERNAL GUIDE
Mr.K. Nishakar
Associate Professor

HEAD OF THE DEPARTMENT
Dr.B. Hari Krishna
Professor

EXTERNAL EXAMNIER

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **B SRINISH KUMAR** WITH ROLL NO.**17K81A04D5, G VISHWAS** WITH ROLL NO.**17K81A04E7**, **MARY SOPHIA** WITH ROLL NO.**17K81A04C3,** OF B.TECH–IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**MINIMUM TIME DELAY AND MORE EFFICIENT IMAGE FILTERING BRAIN TUMOR DETECTION WITH THE HELP OF MATLAB**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS LASYA IT
SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We declare that this project report titled **Minimum Time Delay and More Efficient Image Filtering Brain Tumor Detection With The Help Of MATLAB** submitted in partial fulfillment of the degree of B. Tech in Electronics And Communication Engineering record of original work carried out by us under the guidance and supervision of Mr. K. Nishakar, and has not formed the name basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

.

G. Vishwas	A.Mary Sophia	B.Srinish Kumar
(17KA81A0E7)	(17K81A04C3)	(17K81A04D5)

Place: Secunderabad

Date: 25-06-2021

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

We extended our deep sense of gratitude to Principal, **Dr. P. SANTOSH KUMAR PATRA**, St. Martin's Engineering College, Dhulapally, for permitting us to undertake this project.

We are also thankful to **Dr.B.Hari Krishna** Head of the Department, Electronics And Communications, St. Martin's Engineering College, Dhulapally, for his support and guidance throughout our project and as well as our project coordinator **Dr.A.Anand**, Professor in Electronics And Communication Engineering, for his valuable support.

We would like to express our sincere gratitude and indebtedness to our project supervisor **Mr.K.Nishakar**, Associate Professor, Electronics And Communications St. Martin's Engineering College, Dhulapally, for his support and guidance throughout our project.

Finally, we express thanks to all those who have helped us successfully to completing this project. Furthermore, we would like to thank our family and friends for their moral support and encouragement.

We express thanks to all those who have helped us in successfully completing the project.

.

1. G. Vishwas
2. A. Mary Sophia
3. B. Srinish Kumar

ABSTRACT

Brain tumour detection is one of the intriguing task in the medical field still now. Earlier the technologies pneumoencephalography and cerebral angiography were used but they had drawbacks. The next technologies such as CT and MRI scan techniques with the help of surgeons came into use to provide a higher quality result in image processing. However it is difficult in distinguishing between brain tumour tissue and normal tissue because it they are similar in colour. Hence Brain tumour must be analysed more precisely in order to cure it. In this paper Tumour Detection with help of MATLAB image processing catches three stages Pre-processing, Processing and Post-Postprocessing in morphological detection. After the getting MRI report first stage is pre-processing which is converting the original RGB image to gray-scale image and then using a Gaussian high pass filter for noise reduction. In the second stage processing for pixel enhancement we use Median filter and in third stage i.e. the post-processing where different filters such as Entropy Filter., Standard Deviation Filter (SDF), Weiner Filter, Gradient Magnitude, Regional Maxima are used for various different-different results. In this post processing which is followed by algorithm not only creates the report automatically, but has very less delay time and gives the output more efficiently.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1
INTRODUCTION

Brain tumor is a life- threatening disease. Brain tumors can be malignant or benign. When tumor cells grow it cause pressure inside the skull, this leads to brain damage. Two types of Brain tumor namely Primary and secondary. Primary brain tumors are benign and that are originates in brain. Secondary brain tumors occur when cancer cells spread over the brain from other organ such as lung or breast. It is also called as metastatic brain tumor. Brain tumor can be occurred in any ages. If brain tumor detected early stage it is treatable. Brain tumor cause more death in children and adults under the age of 40 than any other cancer. In India the tumors ranges from 5 to 10 per 100,000 populations with an increasing trend.

Diagnosing brain tumor is a tedious process. Computer Tomography scan (CT scan), Magnetic Resonance Imaging(MRI), tests like an Angiogram, Spinal tab and Biopsy are used to diagnosis brain tumors. CT scan uses x-ray to produce images. CT scan expose patients to ionizing radiation. High radiation are involved in CT scan . A new study in the Journal of the National Cancer Institute suggests that CT scans, commonly used in medical imaging, may increase the risk of brain tumors.

Magnetic Resonance Imaging (MRI) plays an important role in the medical era. It is noninvasive method to detect Brain tumor. MRI uses Magnetic field and Radio waves to produce Brain images. MRIs create more detailed pictures than CT scans. MRI is the preferred screening examination for brain tumor. The goal is to detect brain tumor before clinical signs are noticeable. The size of the tumor can be measured by MRI. A special dye called contrast medium is injected into the patient's vein or given as a pill or fluid to swallow before an MRI scan.

Any area with abnormal tissue can cause for cancer. The abnormalities will be examined by radiologist. A focused white area on MRI can be a lump or tumor. Tumors can be cancerous or benign. If a tumor is benign, it is not a risky and is unlikely to grow or change shape. The radiologist will check its shape and pattern,as they can sometimes be a sign of cancer.

1.1 Project Overview

Brain tumour detection is one of the intriguing task in the medical field still now. Earlier the technologies pneumoencephalography and cerebral angiography were used but they had drawbacks. The next technologies such as CT and MRI scan techniques with the help of surgeons came into use to provide a higher quality result in image processing. However it is difficult in distinguishing between brain tumour tissue and normal tissue because it they are similar in colour. Hence Brain tumour must be analyzed more precisely in order to cure it. In this project Tumor Detection with help of MATLAB image processing catches three stages Pre-processing, Processing and Post-Postprocessing in morphological detection. After the getting MRI report first stage is pre-processing which is converting the original RGB image to gray-scale image and then using a Weiner filter for noise reduction. In the second stage processing for pixel enhancement we use Median filter Entropy Filter., Standard Deviation Filter (SDF), Gradient Magnitude, Regional Maxima are used for various different-different results and in third stage we do image fusion and add some standard no. of pixels to the resultant image. We then compare the input image and resultant image and find some parameters which tells about the efficiency and noise present.

1.2 Project Objective

The Objective of our project is to

- Study the literature that is based on image processing.
- Identify the major areas of image processing in medicine and in particularly on brain tumor detection.
- Identify the current issues in brain tumor identification.
- Propose future directions and improvements that can be made in this area.
- Critically analyze and review identification of brain tumor using image processing.
- To detect the brain tumour with very less time delay and more efficiently.

1.3 Organisation Of Chapters

1.3.1 Introduction

1.3.2 Literature Survey

1.3.3 Software Introduction

1.3.4 Introduction to Image Processing

1.3.5 Digital Image Processing

1.3.6 Methodology

1.3.7 Results

1.3.8 Conclusion and Future Scope

CHAPTER-2
LITERATURE SURVEY

Tumors are abnormal cells which expands in our body but in brain tumor a mass of tissue in our brain cell are expanded rapidly. The symptoms of brain tumor depends upon tumor type, size and location of the tumor. The symptoms are headache vomiting nausea some other changes are speech vision and hearing problems sometime they can changes their activities like personality, ability to concentrate and problem with memory. A brain tumor is categorized primary and secondary depends upon location size and its origin.

A. Benign: Benign tumors which type of non-cancerous mass of cells that it was expands very slowly in the brain. It was does not spread and stay in one place. Normally benign brain tumors can be detected by CT and MRI scans.

B. Maligant: Malignant brain tumor was rapidly expand like cancer that spreads one area to other area of the brain and spine it [3].

C. Materials and Methods: The MRI report is obtained from Government Medical College, Jagdalpur under the guidance of Rajat Kumar Pandey(MBBS, Final Part II ,Student) that includes one normal image and one abnormal image of MRI and CT report. All the image processing operation are done in MATLAB.

2.1 Existing Method:-

- The existing method first employed the wavelet transform to extract features from images, followed by applying principle component analysis (PCA) to reduce the dimensions of features. The reduced features were submitted to a kernel support vector machine (KSVM).
- Another method employed a discrete wavelet transform to extract features from images, and then applied the technique of principle component analysis (PCA) to reduce the size of the features. The reduced features were sent to an FNN, of which the parameters were optimized via an improved artificial bee colony (ABC) algorithm based on both fitness scaling and chaotic theory.

2.2 Proposed Method:-

- Now in our project we proposed some new types of filtering method which may perform the tumor detection more efficiently than the previous methods and give us a detailed output of the tumor cells.
- These filtering techniques such as Weiner filter, Gaussian filter reduce the time in giving us the required output and the output formed is more clear with less amount of noise present in it. Thus, our system lays an emphasis in detecting tumor in an easier manner than the previous ones.

CHAPTER-3
SOFTWARE INTRODUCTION

3.1. Introduction to MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn

and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

3.2 The MATLAB System:

The MATLAB system consists of five main parts

- **Development Environment**:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library:**

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language:**

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

- **Graphics:**

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

3.3 Graphical User Interface (GUI)

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .Figure, called a FIGURE-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIGURE-file contains binary data that does not need to be parsed whenhe associated GUI-based M-function is executed.
- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type

guide filename

Where filename is the name of an existing FIGURE-file on the current path. If filename is omitted,

GUIDE opens a new (i.e., blank) window.

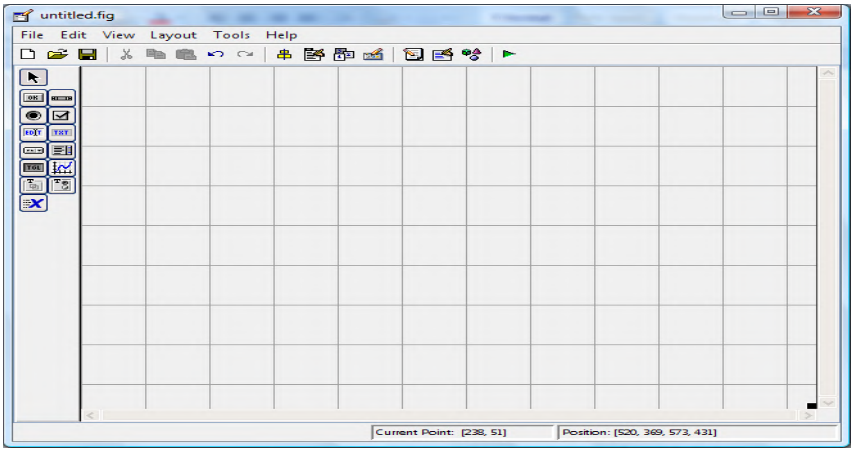


Figure 3. 1 Graphical User Interface

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

3.4 Getting Started

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

[1] Introduction - describes the components of the MATLAB system.

[2] Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

[3] Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

[4] Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

[5] Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cellarrays and multidimensional arrays.

3.5- Development Environment

3.5.1 Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

3.5.2 Starting and Quitting MATLAB

- **Starting MATLAB**

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop. On a UNIX platform, to start MATLAB, type MATLAB at the operating system prompt. After starting MATLAB,the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

- **Quitting MATLAB**

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

3.6 MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB. The first time MATLAB starts, the desktop appears as shown in thefollowing illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

3.6.1 Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- **Current Directory Browser**
- **Workspace Browser**
- **Array Editor**
- **Editor/Debugger**
- **Command Window**
- **Command History**
- **Launch Pad**
- **Help Browser**

  - **Command Window**

Use the Command Window to enter variables and run functions and M-files.

- **Command History**

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

- **Running External Programs**

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

- **Launch Pad**

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

- **Help Browser**

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type help browser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, whereyou view the information.

- **Help Navigator**

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

3.6.2 Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

3.6.3 Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

3.6.4 Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select SaveWorkspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There areoptions for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

3.6.5 Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

3.6.6 Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to runMATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

3.7 Manipulating Matrices

3.7.1 Entering Matrices

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Dürer's matrix as a list of its elements. You have only to follow afew basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ]. To enter Dürer's matrix, simply type in the Command Window

A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]

MATLAB displays the matrix you just entered.

```
A =

16    3    2   13

 5   10   11    8

 9    6    7   12

 4   15   14    1
```

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can referto it simply as A.

3.8 Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

3.8.1 Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

3.8.2 Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

3.8.3 Operators

Expressions use familiar arithmetic operators and precedence rules.

Table 3.8. 1 Operators

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose

3.8.4 Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Table 3.8. 2 Functions

I	Imaginary unit, √-1
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

3.9 GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

3.9.1 Creating GUIs with GUIDE

MATLAB implements GUIs as Figureure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

3.9.2 GUI Development Environment

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

3.9.3 The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIGURE-file - contains a complete description of the GUI Figureure and all of its children (uicontrols and axes), as well as the values of all object properties. **An M-file** - contains the functions that launch and control the GUI and the callbacks, which are defined as subfunctions. This M-file is referred to as theapplication M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIGURE-file.

The following diagram illustrates the parts of a GUI implementation.

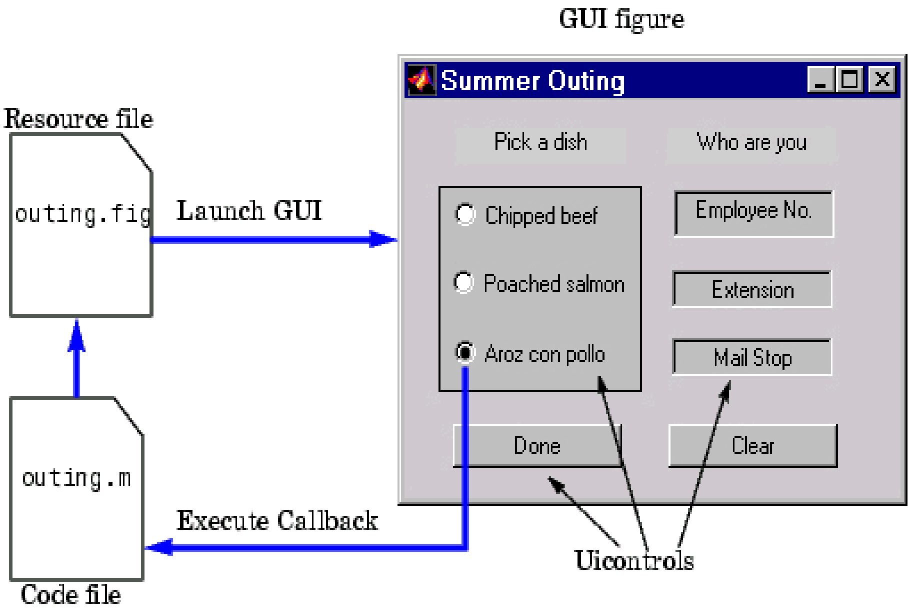


Figure 3. 2 Graphical user blocks

3.9.4 Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see ConFigureuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIGURE-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIGURE-file generated by the Layout Editor.

3.9.5 Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIGURE-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques used in the application M-file.

Application Examples - a collection of examples that illustrate techniques which are useful for implementing GUIs.

3.9.6 Command-Line Accessibility

When MATLAB creates a graph, the Figureure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current Figureure and axes.

GUIs are also created in Figureure windows. Generally, you do not want GUI Figureures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI Figureure, resulting in the graph appearingin the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

3.9.7User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides informationon these components.

- Push Buttons

- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- Listboxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figureures

- **Push Buttons**

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

- **Programming the Callback**

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

- **Toggle Buttons**

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

- **Programming the Callback**

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)

button_state = get(h,'Value');

if button_state == get(h,'Max')

   % toggle button is pressed

elseif button_state == get(h,'Min')

   % toggle button is not pressed

end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);
a(:,:,2) = rand(16,128);
a(:,:,3) = rand(16,128);
set(h,'CData',a)
```

- **Radio Buttons**

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)

set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)

off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];

mutual_exclude(off)

% Continue with callback
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

- **Checkboxes**

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function  checkbox1_Callback(h,eventdata,handles,varargin)

if (get(h,'Value') == get(h,'Max'))

   % then checkbox is checked-take approriate action
else

   % checkbox is not checked-take approriate action

end
```

- **Edit Text**

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)

user_string = get(h,'string');

% proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function  edittext1_Callback(h,eventdata,handles,varargin)

user_entry = str2double(get(h,'string'));

if isnan(user_entry)

   errordlg('You must enter a numeric value','Bad Input','modal')

end

% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menubar of the Figureure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menubar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

- **Static Text**

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it.

- **Frames**

Frames are boxes that enclose regions of a Figureure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

- **List Boxes**

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

Listboxes differentiate between single and double clicks on an item and set the Figureure SelectionType property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the Figureure SelectionType property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

- **Popup Menus**

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then youcan use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

switch val

case 1

% The user selected the first item

case 2

% The user selected the second item

% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

string_list = get(h,'String');

selected_string = string_list{val}; % convert from cell array to string

% etc.
```

- **Enabling or Disabling Controls**

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is grayed out

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the Figureure WindowButtonDownFcn callback executes. Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes ButtonDownFcn property to define the callback.

3.9.8 Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example, axes(handles.axes1) makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you wantto target a different axes. See GUI with Multiple Axes for and example that uses two axes.

Figure

Figure are the windows that contain the GUI you design with the Layout Editor. See the description of Figure properties for information on what Figure characteristics you can control.

CHAPTER-4
INTRODUCTION TO IMAGE PROCESSING

Introduction

4.1. Image

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and naturalobjects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional Figureure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Figure 4.1. 1 General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display.However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each

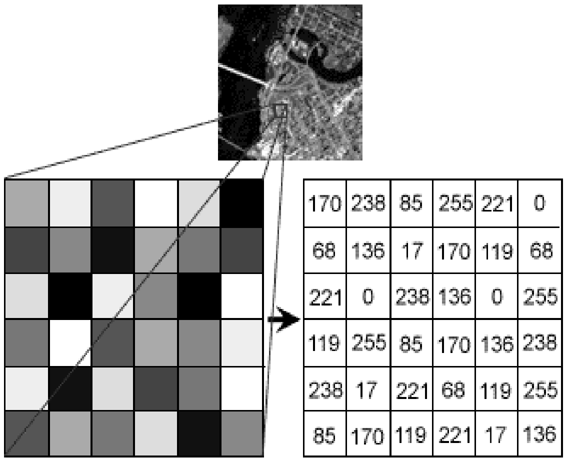


Figure 4.1. 2 Image pixel

pixelconsists of numbers representing magnitudes of brightness and color.

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

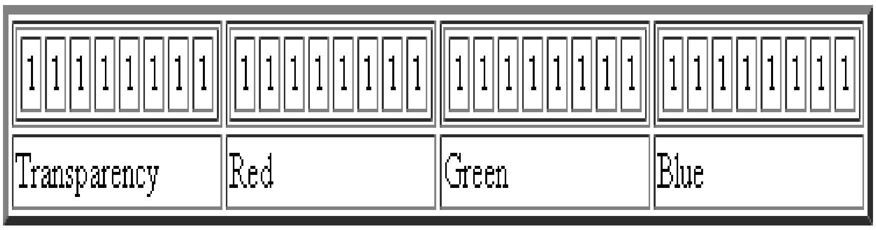


Figure 4.1. 3 Pixel Transparency

4.2 Image File Sizes

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color.

For example, an image recorded by a 12 MPcamera; since each pixel uses 3 bytes to record true color, the uncompressed imagewould occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

4.3 Image File Formats

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

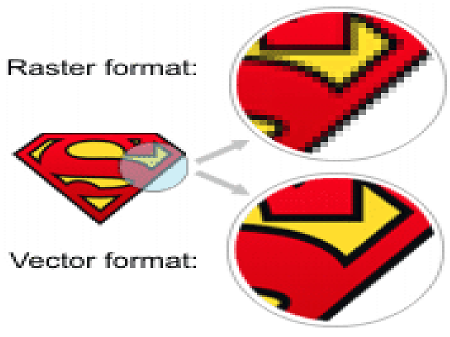


Figure 4.3. 1 Resolution Image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

RASTER FORMATS:

These formats store images as bitmaps (also known as pixmaps)

JPEG/JFIF:

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

EXIF:

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

TIFF:

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

PNG:

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

GIF:

GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

BMP:

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

CGM:

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

SVG:

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

4.4 Image Processing

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man’s ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

4.4.1 Fundamental Steps In Digital Image Processing

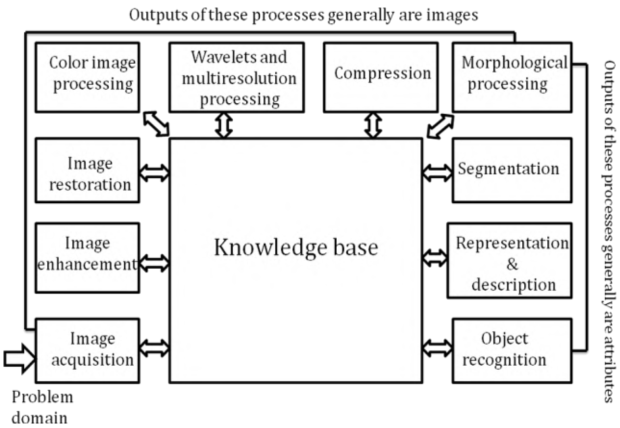


Figure 4.4. 1 Image Segmentation

4.4.2 Image Acquisition:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Figure 4.4. 2 Digital Camera Image

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

4.4.3 Image Enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because “it looks better.” It is important to keep in mind that enhancement is a very subjective area of image processing.

Figure 4.4. 3 Image Enhancement

4.4.4 Image restoration:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Figure 4.4. 4 Image Restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, where as removal of image blur by applying a deblurring function is considered a restoration technique.

4.4.5 Color image processing:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Figure 4.4. 5 Colour Image Processing

4.4.6 Wavelets and multiresolution processing:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform based image processing since the late1950's, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

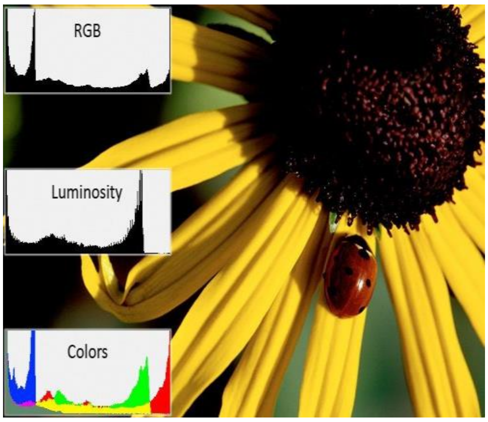


Figure 4.4. 6 RGB Histogram Image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called **Multiresolution** theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

4.4.7 Compression:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

4.4.8 Morphological processing:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Figure 4.4. 7 Blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

4.4.9 Segmentation:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

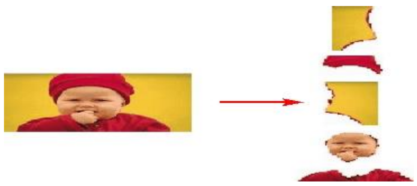


Figure 4.4. 8 Image Segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

4.4.10 Representation and description:

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region.

Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

4.4.11 Object recognition:

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

4.4.12 Knowledgebase:

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an inter related to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

4.5 Components Of An Image Processing System

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

- Image sensors:

With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

- Specialized image processing hardware:

Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

- Computer:

The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

- Image processing software:

Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

- Mass storage:

Mass storage capability is a must in image processing applications. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage forimage processing applications fall into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), Gbytes (meaning giga, or one billion, bytes), and Tbytes (meaning tera, or one trillion, bytes)

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Figure. 1.24. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in “jukeboxes” are the usual media for archival applications.

- Image displays:

Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are there requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

- Hardcopy:

Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

- Network:

Networking is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient.

CHAPTER-5
DIGITAL IMAGE PROCESSING

5.1 Digital image processing

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An important characteristic underlying the design of image processing systems is the significant level of testing & experimentation that normally is required before arriving at an acceptable solution. This characteristic implies that the ability to formulate approaches &quickly prototype candidate solutions generally plays a major role in reducing the cost & time required to arrive at a viable system implementation.

5.2 An overview of Digital Image Processing

An image may be defined as a two-dimensional function f(x, y), where x & y are spatial coordinates, & the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital image by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensor, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis& computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting

& somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low-level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low- level process is characterized by the fact that both its inputs & outputs are images.

Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images. Finally higher- level processing involves “Making sense” of an ensemble of recognized objects, as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

5.3 The definition of an Image

An image is represented as a two dimensional function f(x, y) where x and y are spatial co-ordinates and the amplitude of ‘f’ at any pair of coordinates (x, y) is called the intensity of the image at that point.

5.4 Gray scale image:

A grayscale image is a function I (xylem) of the two spatial coordinates of the image plane.

I(x, y) is the intensity of the image at the point (x, y) on the image plane.

I (xylem) takes non-negative values assume the image is bounded by a rectangle [0, a] ×[0, b]I: [0, a] × [0, b] → [0, info)

5.5 Color image:

It can be represented by three functions, R (xylem) for red, G (xylem) for green *and* B (xylem) for blue.

An image may be continuous with respect to the x and y coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate's values is called sampling. Digitizing the amplitude values is called quantization.

5.6 Coordinate convention:

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image f(x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size M X N. The values of the coordinates (xylem) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates.

In many image processing books, the image origin is defined to be at (xylem)=(0,0).The next coordinate values along the first row of the image are (xylem)=(0,1).It is important to keep in mind that the notation (0,1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Following Figureure shows the coordinate convention. Note that x ranges from 0 to M-1 and y from 0 to N-1 in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (xylem) the toolbox uses the notation (race) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate topples, (alb), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentation refers to the coordinates. Less frequently the toolbox also employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refers to rows. This is the opposite of our use of variables x and y.

5.7 Image as Matrices:

The preceding discussion leads to the following representation for a digitized image function:

f (0,0)   f(0,1)   ………..   f(0,N-1)

f (1,0)   f(1,1)   …………   f(1,N-1)

f (xylem)=   .   . .

.   .

f (M-1,0)   f(M-1,1) .... f(M-1,N-1)

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements. A digital image can be represented naturally as a MATLAB matrix:

f (1,1)   f(1,2)...........f(1,N)

f (2,1)   f(2,2)...........f (2,N)

.   .   .

f= f (M,1) f(M,2) …….f(M,N)

Where f (1,1) = f(0,0) (note the use of a monoscope font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation f(p ,q) denotes the element located in row p and the column q. For example f(6,2) is the element in the sixth row and second column of the matrix f. Typically we use the letters M and N respectively to denote the number of rows and columns in a matrix. A 1xN matrix is called a row vector whereas an Mx1 matrix is called a column vector. A 1x1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using mono-scope characters. We use conventional Roman, italic notation such as f(x ,y), for mathematical expressions

5.8 Reading Images:

Images are read into the MATLAB environment using function imread whose syntax is

Imread (‘filename’)

Table 5. 1 Classification of Different Formats

Format name	**Description Extension**	**Recognized**
TIFF	Tagged Image File Format	.tif
JPEG	Joint Photograph Experts Group	.jpg
GIF	Graphics Interchange Format	gif
BMP	Windows Bitmap	.bmp

Here filename is a spring containing the complete of the image file(including any applicable extension).For example the command line

>> f = imread (‘8. jpg’);

Reads the JPEG (above table) image chestxray into image array f. Note the use of single quotes (‘) to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output If a semicolon is not included. MATLAB displays the results of the operation(s) specified in that line. The prompt symbol (>>) designates the beginning of a command line, as it appears in the MATLAB command window.

5.9 Data Classes:

Although we work with integers coordinates the values of pixels themselves are not restricted to be integers in MATLAB. Table above list various data classes supported by MATLAB and IPT are representing pixels values. The first eight entries in the table are refers to as numeric data classes. The ninth entry is the char class and, as shown, the last entry is referred to as logical data class.

All numeric computations in MATLAB are done in double quantities, so this is also a frequent data class encounter in image processing applications. Class unit 8 also is encountered frequently, especially when reading data from storages devices, as 8 bit images are most common representations found in practice. These two data classes, classes logical, and, to a lesser degree, class unit 16 constitute the primary data classes on which we focus. Many ipt functions however support all the data classes listed in table. Data class double requires 8 bytes to represent a number uint8 and int 8 require one byte each, uint16 and int16 requires 2bytes and unit 32.

Table 5. 2 Data Classes

Name	Description
Double	Double _ precision, floating_ point numbers the
Unit 8	unsigned 8_bit integers in the range[0,255](1byte per Element).
Unit 16	unsigned 16_bit integers in the range[0,65535](2byteper element)

5.10 Image Types:

The toolbox supports four types of images:

1. Intensity images
2. Binary images;
3. Indexed images;
4. R G B images.

Most monochrome image processing operations are carried out using binary or intensity images, so our initial focus is on these two image types. Indexed and RGB colour images.

- Intensity Images:

An intensity image is a data matrix whose values have been scaled to represent intentions. When the elements of an intensity image are of class unit8, or class unit 16, they have integer values in the range [0,255] and [0, 65535], respectively. If the image is of class double, the values are floating point numbers. Values of scaled, double intensity images are in the range [0, 1] by convention.

- Binary Images:

Binary images have a very specific meaning in MATLAB.A binary image is a logical array 0s and1s.Thus, an array of 0s and 1s whose values are of data class, say unit8, is not considered as a binary image in MATLAB .A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, we create an array B using the statement.

B=logical (A)

If A contains elements other than 0s and 1s.Use of the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.

Using relational and logical operators also creates logical arrays.

To test if an array is logical we use the I logical function: islogical(c).

If c is a logical array, this function returns a 1.Otherwise returns a 0. Logical array can be converted to numeric arrays using the data class conversion functions.

- Indexed Images:

An indexed image has two components:

A data matrix integer, x

A color map matrix, map

Matrix map is an m*3 arrays of class double containing floating point values in the range [0, 1].The length m of the map are equal to the number of colors it defines. Each row of map specifies the red, green and blue components of a single color. An indexed images uses “direct mapping” of pixel intensity values color map values. The color of each pixel is determined by using the corresponding value the integer matrix x as a pointer in to map. If x is of class double ,then all of its components with values less than or equal to 1 point to the first row in map, all components with value 2 point to the second row and so on. If x is of class units or unit 16, then all components value 0 point to the first row in map, all components with value 1 point to the second and so on.

- RGB Image:

An RGB color image is an M*N*3 array of color pixels where each color pixel is triplet corresponding to the red, green and blue components of an RGB image, at a specific spatial location. An RGB image may be viewed as “stack” of three gray scale images that when fed in to the red, green and blue inputs of a color monitor

Produce a color image on the screen. Convention the three images forming an RGB color image are referred to as the red, green and blue components images. The data class of the components images determines their range of values. If an RGB image is of class double the range of values is [0, 1].

Similarly the range of values is [0,255] or [0, 65535].For RGB images of class units or unit 16 respectively. The number of bits use to represents the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8bit image, the corresponding RGB image is said to be 24 bits deep.

Generally, the number of bits in all component images is the same. In this case the number of possible color in an RGB image is (2^b) ^3, where b is a number of bits in each component image. For the 8bit case the number is 16,777,216 colors

CHAPTER-6
METHODOLOGY

6.1 Pre-Processing:

I. Pre-processing: The aim of pre-processing is to convert the RGB image into gray image using Gray-Scale conversion. The converted image is then processed to reduce noise and implement the contrast enhancement [4,5] to improve the quality of the image.

The working process of pre-processing is as follows:

1. Gray-scale conversion
2. Addition of Poisson Noise

Gray Scale Conversion*:* The Gray scale image includes only gray-scale values, but MRI report consist of primary colors (RGB) [6]. Gray scale is a type of monochrome image where each pixel of any image consists some information of the intensity of light hence it is similar to that of conventional black and white image. By converting the image into gray it becomes easier for further processing. Here we take the help of 'RGB2gray' or 'binarize' the commands which converts the three dimensional image into two dimensional or gray scale image.

Poisson noise addition and applying a high pass filter *:* Filtering is a technique in image processing where we try to reduce the effect of noise as minimum as possible. Let us assume that the image consists of Gaussian noise present in it. After the conversion of image into gray, we try to minimize the noise effect by passing it through a high pass filter known as Gaussian HPF. Median filter is the type of filter provides a median value of the pixels of an image and it is used because the mean values obtained using averaging filters it makes image blur [7]. In MRI, Gaussian noise are more commonly seen or observed. Salt and Pepper noise canbe reduced with the help of median filter, considering that Gaussian noise is reduced by a Gaussian high pass filter (HPF).

Weiner Filtering*:* On comparing and analyzing various filters and their capabilities we came to a conclusion that Weiner filter is the most suitable filter for further stages of processing. Hence the addition of Weiner filter completes the 1st stage i.e. pre-processing.

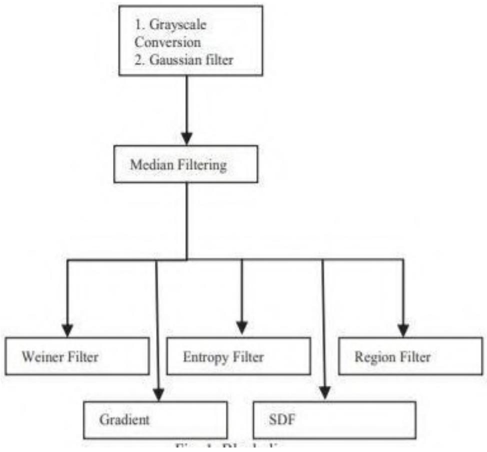


Figure 6. 1 Block Diagram

6.2 Processing:

Processing: The 2nd stage starts with processing of the Weiner filter image. In stage 2 we applied median and gradient filters to reduce the noise at the edges and to improve the intensity of the image at the later stages. Hence Processing stage consists of Median and Gradient filtering.

1. Median Filtering.
2. Gradient Filtering.

Median Filtering*:* The median filter is a type of filter which is used to reduce or minimize the salt and pepper noises present in the image, which are caused by motion artifacts (movement of patient during scan). In-order to reduce these kinds of noise we used this filter in our MATLAB code following the command '**medfilt2**'. We are sub-dividing the images into 4 variants (Low-Low Image Low-High Image, High-Low image High-High Image) using **2-Dimensional Discrete Wavelet Transform**. The command 'Dwt2' subdivides the image based on its intensity values. The image obtained is combination of all the intensities

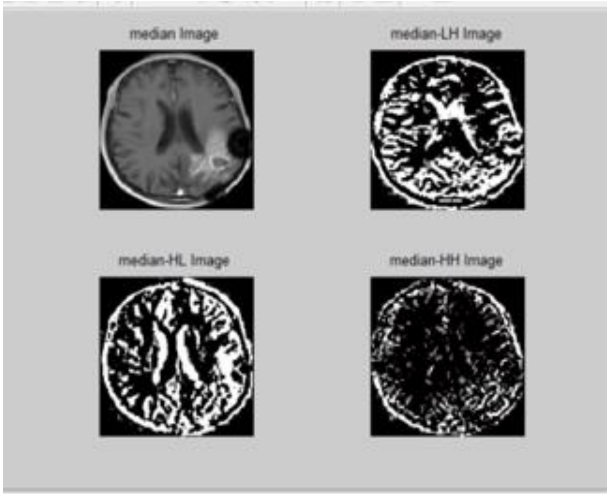


Figure 6. 2 Median Image

Gradient Filtering*:* The gradient of any vector of an image contains the information which can let us know about the change in the intensity of the image at any given point of time. By using the 'imgradient' command present in the MATLAB and converting the matrix values (which are the intensity values) into the same components as we stated in median filtering, we can know the intensities of the image at any given point of time. For our project we've taken the combination of Low-High and High-Low intensity values lying between (90-200). These values are used to produce the combined image and is further processed to stage 3. The processing stage ends here and the following stage is post-processing.

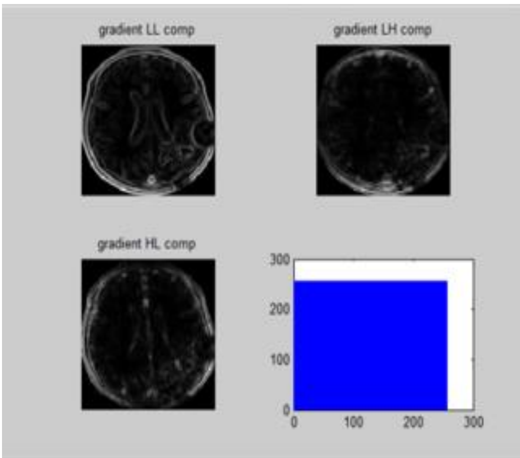


Figure 6. 3 Gradient filter output

6.3 Post-Processing:

Post-Processing: In post processing stage, the obtained gradient output images are combined together to obtain a standard output image. This is done with the help of for loop and having iterations in the program. To obtain the combination of all the pixels into single output we used four for loops and 72 iterations. These commands run and execute the code to give a single result image. For the image obtained we add some standard number of pixels to enhance the image. The addition of standard number of pixels to the image is done using "imadd()" command. The resultant output image is also resized using 'imresize(image_name,[256,256])' command.

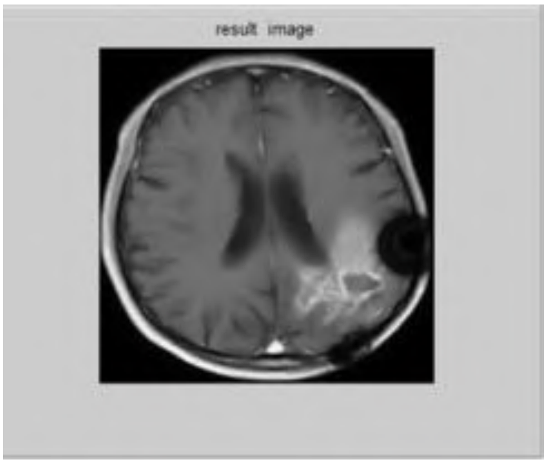


Figure 6. 4 Resultant image

This completes the 3rd stage which is post-processing. The obtained image is then compared with the noise image and gives us the Peak Signal to Noise Ratio, and Mean Square Error. Hence the margin or error and the value of noise reduced can be known.

6.4 Comparison

The following table consists of the comparison between the normal CT and MRI scanned images in tally with the abnormal CT and MRI scanned images. The parameters such as mean. standard deviation, area, entropy, median, moment fifth are been showed. This table gives us the range of values where one can know if the scanned image consists of the tumor or is the person normal.

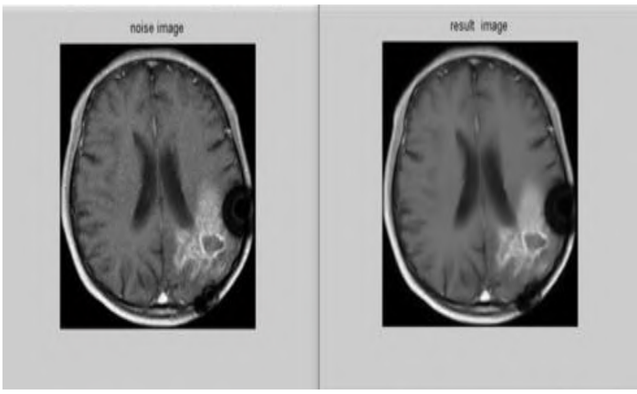


Figure 6. 5 Comparison of noise and result image

The comparison of the result and noise image in the above figure lets us in calculating the PSNR and MSE values so that the robustness of the image obtained can be known when some external disturbances occur.

Table 6. 1 Comparison of Statistical properties of MRI and CT-Scan

S. N O.	**Parameters**	**MRI**		**CT**	
		Normal	**Abnormal**	**Normal**	**Abnormal**
01	**Mean**	63.008	75.916	482.22	78.1154
02	**Standard deviation**	440.578	227.134	8.033	477.3392
03	**Median**	8	7	3	2
04	**Moment fifth**	21.068	35.882	21.536	21.765
05	**Area**	113	36	18	156
06	**Entropy**	5.987	6.9627	4.3295	6.9948

CHAPTER-7
RESULTS

- **Input Image**

The input image is converted from RGB to grey using grey scale conversion

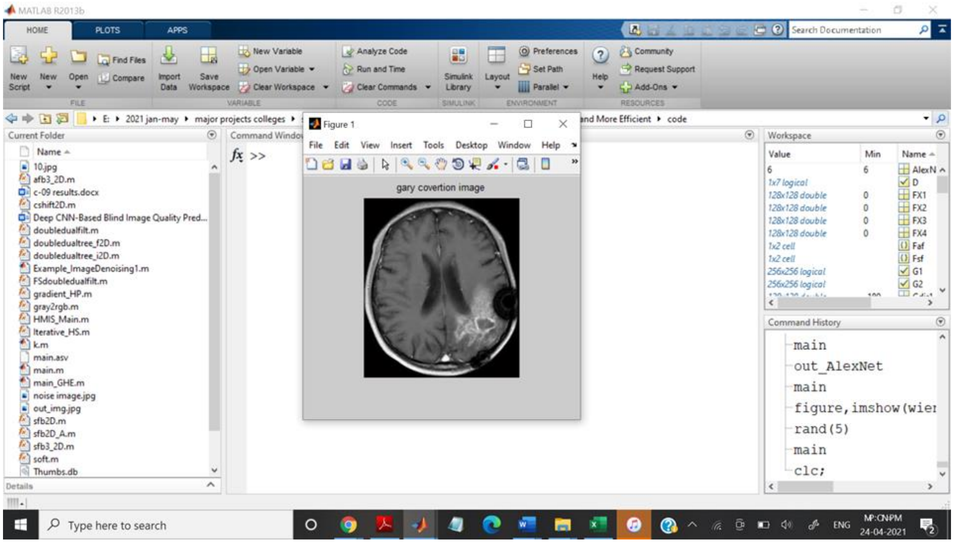


Figure 7.1 Grey Conversion Image

- **Noise Image**

After adding poison noise to the Grey image we get Noise image.

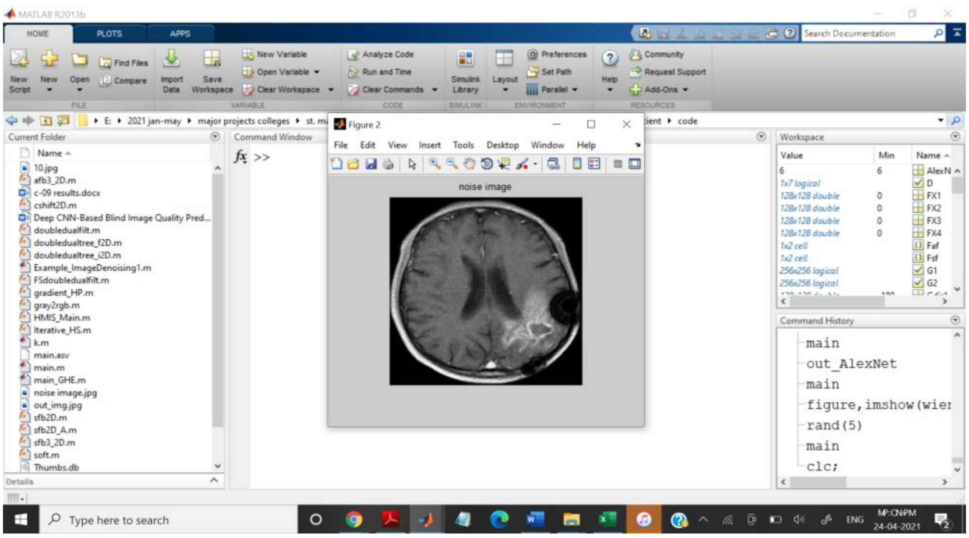


Figure 7.2 Noise Image

- **Weiner Filter Image:**

  This image shows the Weiner filter output where the errors are reduced as minimal as possible and only high frequency parts of the image are visible

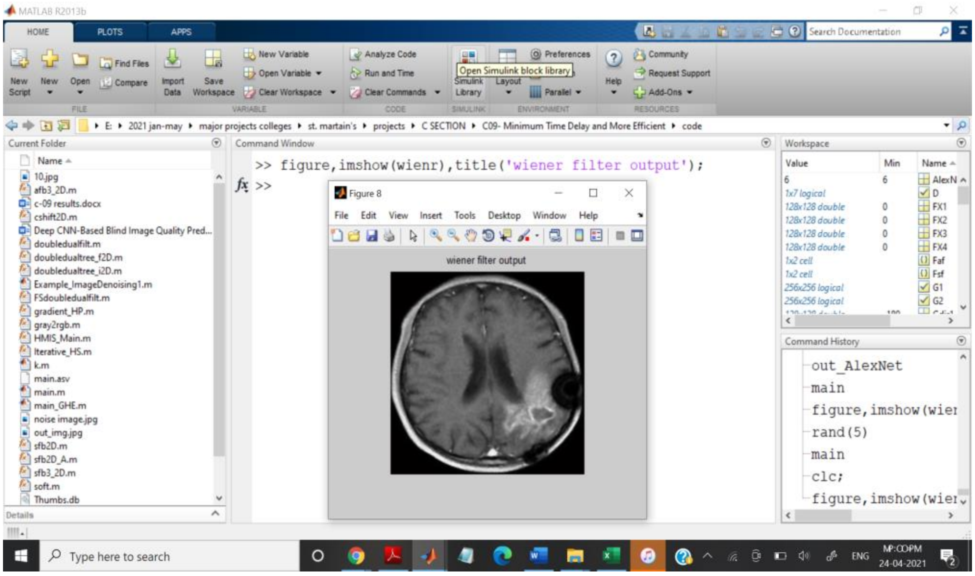


Figure 7.3 Wiener Filter Output Image

- **Region Based Filter:**

  This image shows the region based enhancement where any part of the imagecan be cropped out for having a better analysis of that part.

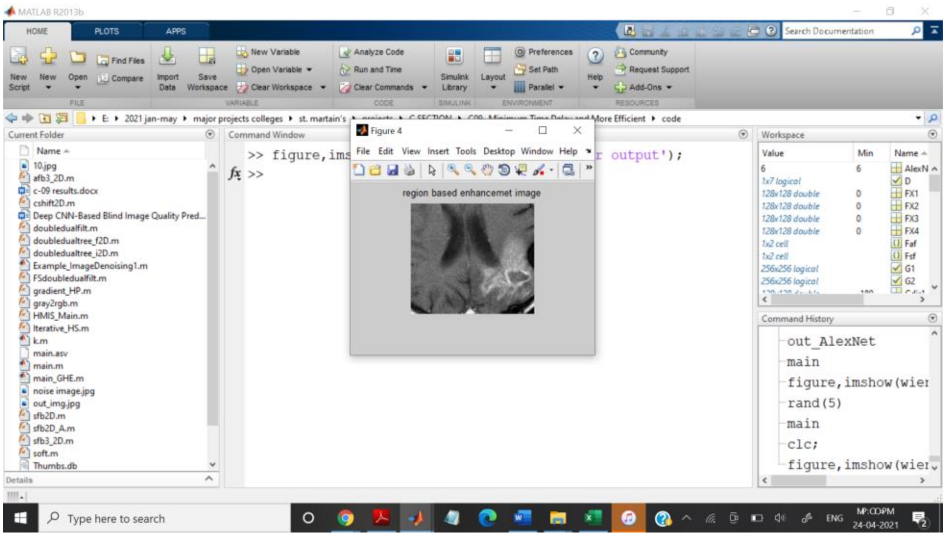


Figure 7.4 Region Based Enhancement Image

- **Median Filter Outputs:**

These images are the outputs of median filter where the image is free from salt and paper noise and divided into median Low-Low image, median High-Low image, median High-High image according to pixels intensity.

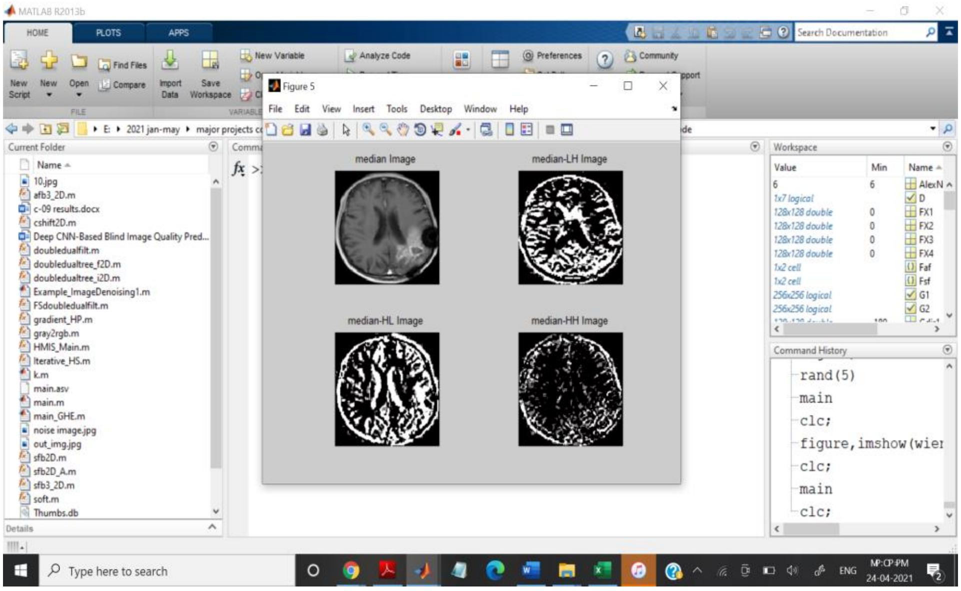


Figure 7.5 Median Images

- **Gradient Filter Outputs:**

These images are the outputs of gradient filter. From this image we can seethe tumor region more efficiently.

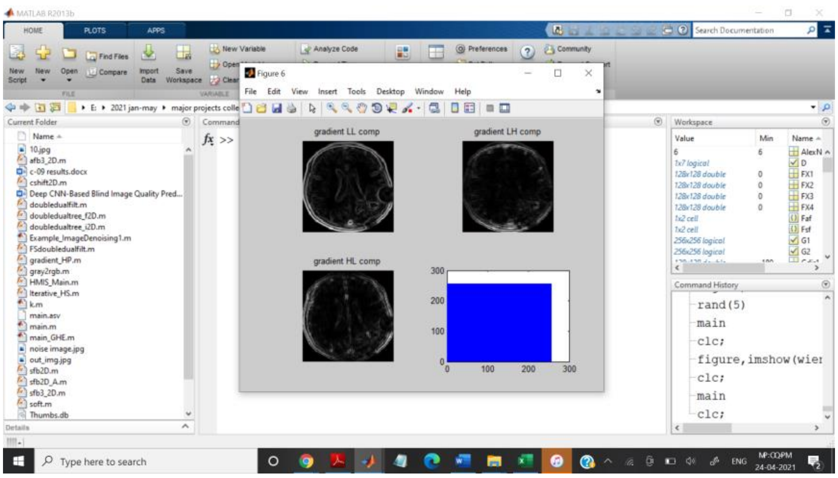


Figure 7.6 Gradient Scale Images

- **Resultant Image:**

  The image shows the obtained result image after applying all the filers.

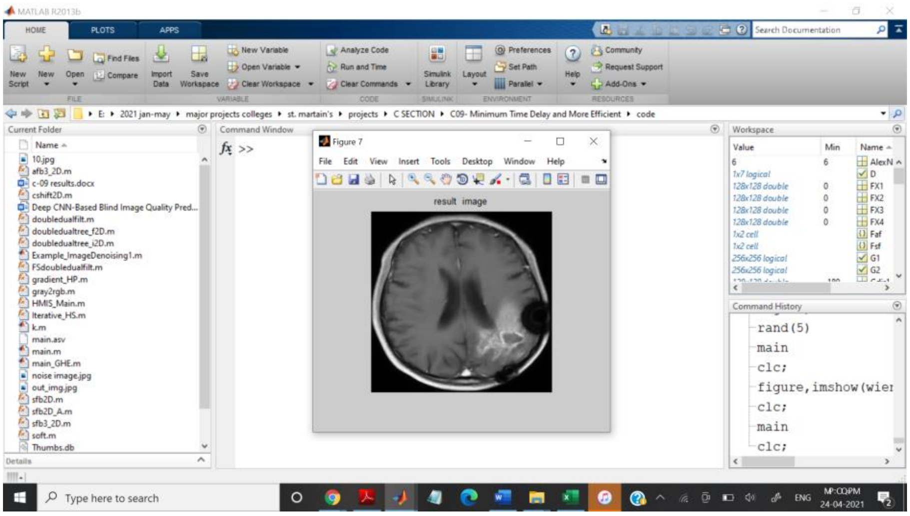


Figure 7.7 Result Image

- **Comparison between Noise and Resultant Image:**

  The image shows the comparison between the noise and the result image. From this image we can see how the noise image is Enhanced.

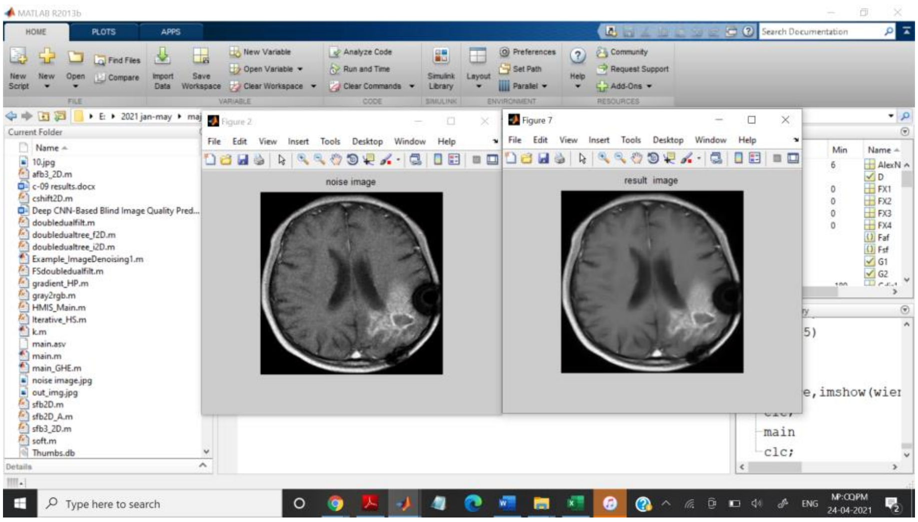


Figure 7.8 Noise vs Result Image

- **Parameters Measured:**

This image shows the calculated PSNR and MSE values which gives us theefficiency of the resultant image i.e., the performance of the filters.

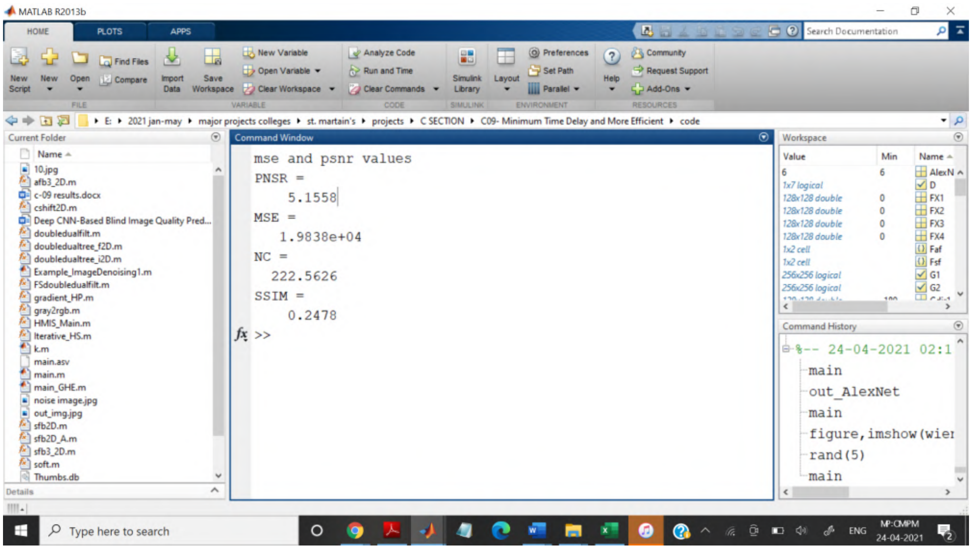


Figure 7.9 PSNR and MSE Values

CHAPTER-8
CONCLUSION AND FUTURE SCOPE

8.1 Conclusion:

All the discussion is aiming to detect the brain Tumor from or the abnormal behavior of the any one's brain through CT and MRI brain images. The brain tumor is detected more accurately using image processing and various type of the morphological operations, image filtering are performed for analysis or to observe and to obtain the various parameters like Mean, Standard deviation, Third moment, Area, Entropy of the image.

8.2 Future Scope:

- In future this method can be enhanced and can be used to detect brain tumor in colored MRI images.

- As we are currently using this method to detect the tumour in brain only, in future this method can be further processed to detect the tumour in various parts of the body with very less time delay and more efficiency.

REFERENCES

[1] Karuna Ankita Joshi, M., Automatic detection and severity analysis of brain tumors using GUI in MATLAB, IJRET: International Journal of Research in Engineering and Technology eISSN: 2319-1163, PISSN: 2321-7308.

[2] Ashraf Anwar and Arsalan Iqbal, 2013. Image Processing Technique for Brain Abnormality Detection. International Journal of Image Processing (IJIP).algorithm" IEEE Trans on Acoustics, Speech, and Signal Processing.

[3] Brown, Robert Grover; Hwang, Patrick Y.C. (1996). Introduction to Random Signals and Applied Kalman Filtering (3 ed.). New York: John Wiley & Sons. ISBN 978-0-471-12839-7.

[4] Gonzalez, R. C., et al. Digital Image Processing Using MATLAB. New Jersey, Prentice Hall, 2003, Chapter 11.

[5] Gopinath, N., 2012. Extraction of Cancer Cells from MRI Prostate Image Using MATLAB. International Journal of Engineering Science and Innovative Technology (IJESIT)

[6] Digital Image Processing, XXXX. by Rafael C. Gonzalez, Richard E. Woods, ISBN-10:013168728X.

[7] Rajesh C. Patil and Dr. A.S. Bhalchandra, XXXX.Brain Tumor Extraction from MRI Images Using MATLAB. International Journal of Electronics, Communication &Soft Computing Science and Engineering ISSN: 2277-9477, 2(1).

[8] Gopinath, N., 2012. Extraction of Cancer Cells from MRI Prostate Image Using MATLAB. InternationalJournal of Engineering Science and Innovative Technology (IJESIT) 1(1).

[9] Rajesh Patil and Dr. AS. Bhalchandra, 2014. BrainTumor Extraction From MRI images using MATLAB, International Journal of Electronics& Communication Of Engineering And Soft Computing, IJECSSE, 2(1).

BIBILOGRAPHY

CT – Computer Tomography

MRI – Magnetic Resonance Imaging

MATLAB – Matrix Laboratory

RGB – Red Green Blue

SDF – Standard Deviation Filter

PCA – Principle Component Analysis

KVSM – Kernel Support Vector Machine

FNN – Feedforward Neural Network

ABC – Artificial Bee Colony

BLAS – Basic Linear Algebra Subprograms0

LAPACK – Linear Algebra Package

LINPACK – Linear Equation Software Package

ESPACK – Eigen System Package

GUI – Graphical User Interface

GUIDE – Graphical User Interface Development Environment

API – Application User Interface

MP – Mega Pixel

PNG – Portable Network Graphics

JPG /JPEG – Joint photographic Experts Group

GIF – Graphical Interchange Format

JFIF – JPEG File Interchange Format

EXIF – Exchangeable Image File Format

TIFF – Tagged Image File Format

BMP – Bit Map

OS – Operating System

CRT – Cathode Ray Tube

CGM – Computer Graphics Metafile

SVG – Scalable Vector Graphics

XML – Extensible Markup Language

CCD – Charge Coupled Devices

ALU – Arithmetic Logic Unit

PC – Personal Computer

CD-ROM – Compact Disc Read Only Memory

GCM – Grayscale Co-occurrence Matrix

CCM – Color Co-occurrence Matrix

HSV – Hue, Saturation and Intensity Value

MCCM – Modified Color Co-occurrence Matrix

CBIR – Content-based Image Retrieval

SCM – Single-channel Co-occurrence Matrix

MCM – Multi channel Co-occurrence Matrix

LFT – Local Fourier Transformation

ICICM – Integrated Color and Intensity Co-occurrence Matrix

FPGA – Field Programmable Gate Arrays

DIP – Digital Image Processing

EM – Electro Magnetic

HPF – High Pass Filter

PSNR – Peak Signal to Noise Ratio

MSE – Mean Square Error

APPENDICES

- The command used to read the input file is

  Syntax: imread(File_name);
- The command used to convert rgb to grey image is

  Syntax: rgb2gray(image_name);
- The command used to show the image in figure window is

  Syntax: imshow(image_name);
- To change the image to a particular size we use

  Syntax: imresize(image_name,[256,256]);
- Adding noise can be done by using the command

  Syntax: imnoise(image_name,'poisson');
- The filter command used to get the weiner filter output

  Syntax:weiner2(image_name);
- The command used to crop the image is

  Syntax: imcrop(image_name);
- The command used to apply wavelet transfrom is

  Syntax: dwt2(image_name,'haar');
- The filter command used to get the median filter output

  Syntax: medfilt2(image_name);
- The filter command used to get the gradient filter output

  Syntax: imgradient(image_name,'prewitt');
- The command used to transform the domain specified vectors x and y into arrays or 2-D grid is

  Syntax: meshgrid(x,y);
- To get mean of the elements present w use

  Syntax: mean(A);
- The command used to convert grey image to rgb is

  Syntax: gray2rgb(image_name);
- The filter command used to apply Standard deviation filter

  Syntax: std(image_name);
- For image fusion we are using four for loops to combine 4 images**.**

Syntax: for index = values

statements

end

- To add standard number of pixels to the image we use Syntax: imadd(image-name,33);
- To measure PSNR,MSE values we use

  Syntax: measerr(input_image,output_image)

A MAJOR PROJECT REPORT

ON

SOLAR POWERED AUTOMATED MUTITASKING AGRICULTURE ROBOT USING IOT

Submitted by

1) Ms. A.LAHARIKA (17K81A04C4)
2) Mr. K.YASHWANTH REDDY (17K81A04E9)
3) Mr. P. AJAY GOUD (18K85A0429)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

P.KIRANMAYEE

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST.MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100 NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

BONAFIDE CERTIFICATE

This is to certify that the project entitled Design and Implementation of SOLAR POWERED AUTOMATED MULTITASKING AGRICULTURE ROBOT USING IOT, is being submitted by **Ms .A. LAHARIKA - 17K81A04C4 , Mr. K. YASHWANTH REDDY-17K81A04E9, Mr. P. AJAY GOUD - 18K85A0429,** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of the work carried out by them. The result embodied in this report have been verified and found satisfactory.

Internal Guide

Mrs. P.KIRANMAYEE

Assistant professor

Head of the Department

Dr. B.HARI KRISHNA

Professor

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **A.LAHARIKA** WITH ROLL NO.**17K81A04C4, K.YASHWANTH REDDY** WITH ROLL NO.**17K81A04E9**, **P.AJAY GOUD** WITH ROLL NO.**18K85A0429,** OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**SOLAR POWERED AUTOMATED MULTITASKING AGRICULTURAL ROBOT USING IOT**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of **'Bachelor of Technology in Department of Electronics and Communication Engineering'**, session: 2017 – 2021, **St. Martin's Engineering College**, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled **'SOLAR POWERED AUTOMATED MULTITASKING AGRICULTURE ROBOT USING IOT'** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

A.LAHARIKA	(17K81A04C4)
K.YASHWANTH REDDY	(17K81A04E9)
P.AJAY GOUD	(18K85A0429)

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr. B.Hari Krishna**, **Head &Professor**, the Dept. of Information Technology for his/her advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept. project coordinator Dr.Maalyadri, Professor** for his/her involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Mrs.P.Kiranmayee**, **Assistant Professor in ECE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. A.LAHARIKA
2. K.YASHWANTH REDDY
3. P.AJAY GOUD

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract:

Agriculture plays a vital role in socio-economic development of India. Agriculture is the cultivation of animals, plants, fungi, and other life forms for food, fiber, biofuel, medicinal and other products used to sustain and enhance human life. Agriculture is the most essential and foremost economic activity of all times. Until the Industrial Revolution, the vast majority of the human population laboured in agriculture. Pre-industrial agriculture was typically subsistence agriculture/self- sufficiency in which farmers raised most of their crops for their own consumption instead of cash crops for trade. But now, agriculture is undergoing a structural change leading to a crisis situation. The growth rate of agricultural output is gradually declining in the recent years due to labour scarcity and more expensive.Seed and chemical prices are rising too, creating the need for their efficient use. The recent survey shows that the world should double their agriculture productivity to feed the entire booming population by 2050. At that time with efficient management of natural resources will meet increasing demand by adopting modern technology in farming to improve agricultural productivity. Hence, as the farmers are struggle with higher seed and chemical prices, the prospect of labour shortages and growing world demand for food will drive to think an innovative and more efficient farming method. These problems in agriculture led us to propose an automated design “Agribot” which can perform most of the farming activities.

CHAPTER-1

INTRODUCTION

1.1. Introduction to the project:

The primary occupation in a developing country like India is agriculture. But nowadays the number of people involved in agricultural sector is declining due tovarious reasons. It is essential to improve the efficiency and productivity of agriculture. By using this project, we can perform various tasks for agricultural purposes. Despite large-scale mechanization in agricultural field in some parts of the country, most of the agricultural operations in large number of parts are carried on by humans by using simple tools and implements like wooden plough, sickle, etc. Little or no use of machines is made in ploughing, sowing, irrigating, weeding the crops. This is specially the case with small and marginal farmers. It results in huge wastage of men power and in lower yields per capita labour force. So, we must mechanize the agricultural operations so that wastage of labour force is avoided, and farming is made convenient and efficient. We believe some progress will be madeby applying this vehicle in agricultural field.

Current agricultural practices in India are neither economically nor sustainable and India's yield for many agricultural commodities are low. The main factors are poorly maintained irrigation systems and almost universal lack of good extension services. So, this project is also meant for control the irrigation activities. Through this design the required water supply is automatically monitored and used whenever needed. Some farmers areunaware of the technological developments in farming. Increase the yields which not only alleviated the level of hunger, but simultaneously released the human population from the nutritional obstacle to further growth. So continued growth demands even more agricultural improvements. So, we must find whatever ways to improve the agriculture and make our country a better place to live. India is an agricultural country. Nearly seventy percent of our population depends on agriculture. Agriculture yields one-third of our National income. Our economy is based on agriculture. The development of agriculture has a great impact on the economic welfare of our country. Our agriculture was in underdeveloped condition for a long time. We didn't produce enough food for our people. Our country had to buy vegetables and grains from foreign countries, but the things are changing now. India is producing more foodgrains than its needs. Some food-grains are being sent to other countries. Great improvements have been made in. agriculture through our five-year plans. Green Revolution has been brought in the agricultural sector. Now our country is self-sufficient in food-grains. It is now able to export surplus food-grains and some other agricultural products to other countries. Our country ranks first in the world in the production of tea and groundnuts. It ranks second in the world in the production of rice, sugarcane, jute and oil seed.

CHAPTER-2

LITERATURE SURVEY AND EXISTING METHODS

2.1 Literature survey:

The seeding and fertilizing agriculture robot using microcontroller . The point of composed framework for seeding, treating, moisture level calculation, temperature checking and humidity level detection.The robot system is monitored by remotely. Composed framework includes the route of robot to the goal effectively and does the predetermined functions. The robotic system and the remote framework are associated via networking framework. DC motors are utilized for route of the robotic system. DC motors speed can be managed by utilizing controller. The solenoid is utilized to control the seeding operation and treating.

Seed Sowing Using Robotics Technology. This framework presents a control system which drop the seed at specific location which determined separation in two seeds that is row and column while sowing. They utilizes mechanical autonomy innovation in the cultivating framework to decrease the effort of agriculturists and also to reduce time, energy and required cost for the seeding. The issues of the existing system in seeding operation can be removed completely in this proposed system.

“Multi-purpose agriculture machine”. This paper proposed the concept of usage of solar energy in the agricultural fields. Solar energy is the very power full energy, it plays very significant role in agriculture. This can be used in irrigation purpose for drawing water in the well remotely to the towns without power supply. Mechanism includes the utilization of a hybrid energy device between the power source and the work. project is to develop the cost-effective system. So, disable can able to grow in their respective field. Every Normal human being sees, listens and then reacts to the situations by speaking himself out. But there are some human beings those who are not able to speak or listen, but they try to react through actions most of time normal people are not able to understand what they want to say. This application will help for both of them to communicate with each other. It consists ofseveral parts, in part one This system deals with multi- operational irrigation machine for seeding, fungicides, splashing pesticides & fertilizers and plant cutting. Consequently, it is more efficient, economic and multi-operational equipment for agriculturist.

Mahesh R. Pundkar contemplated the execution of seeding, ploughing and plant cutting machine using algorithm of image processing by flash magic. They also declared the impact in depth of seed, spacing between the seed, ratio of miss seeding and proportion and execution seeding gadget on germination of seed and productivity of yield trim.

2.2 Existing systems:

Agriculture is main stay of many nations across the globe. There are multiple interventions by many scholars to utilize technology and mechanical intervention into agriculture. In the early 1920’s itself there were attempts to implement robotics to help improve agriculture. These were primitive models that required use of cable connection to operate the machine. Revolutionary attempts in the field of applying robots to agriculture continued to develop especially after 1980s as technological advancement in the field of computer science and engineering made machine vision (MV) guidance possible.Machine vision is nothing but integration of many technologies, software, hardware products, systems engineering, applied mechanical engineering actions,methods and expertise. It is integration of these technological advancements to resolve real world issues and problems on hand. Our design and structure of Agricultural Robot is one such attempt to help the farmers reap the benefits of technological advancement and to get benefits.

An “agricultural robot” is a robot deployed for agricultural purposes. It utilizes MV applications to identify required tasks assigned and shall do as per given schema to achieve results. Robots are mainly applied in the field of agriculture in weed control, seeding, and harvesting. Soil analysis and environmental monitoring are other areas where Robotics and AI are helping the farmers.

In the coming years there is an expectation of exponentially increased market for robotics in agricultureMany of the versions that are available in market or are being developed elsewhere globally have one limitation that they all are making robots to work in predefined way of work. So, specific tasks that are often repeatable and can be programmed were only being addressed. Most of these machines require electricity or battery or solar power to operate. Our robot is designed solely to minimize the labor of farmers in addition to increasing the speed and accuracy of the work. It performs the elementary functions involved in farming i.e. ploughing the field, sowing of seeds and covering the seeds with soil, spray fertilizer, sprinkle water etc.

(1) NURSERY

The Agri robot is used in nursery for row to row spacing can be adjusted, Required seed spacing can be achieved as well as variety of seed can be sowing by this robot.

(2) FARM

In farm Agri robot is used for planted seeds in all the rows of the farming plot are watered and fertilizer is sprayed on all the plants. Some crops need fertilizers when the seed germinates and the plant begins to grow

(3) AUTOMATIC GRASS CUTTER

Using the electrical energy without the help of manpower it removes all the unwanted grass in the fields.

(4) AUTOMATIC SEEDLING

To reduce the manpower robots are designed for the purpose of sowing seeds.

(5)SPRINKLING FERTILIZER

Using the robot it sprinkles the fertilizers in the fields it consumes less time saves manpower.

(6) SOLAR POWER OPERATED

Using solar energy it can perform task i.e., sowing seed Solar power operated.

(7) MULTI PURPOSE ROBOT

Using solar energy it can perform wide variety of tasks such as grass cutting ,soil testing, sowing.

(8) AUTOMATED WITH MOBILE PHONE

In this we operated the given tasks using mobile phone technology.

CHAPTER-3

BLOCK DIAGRAM

3.1 Block diagram

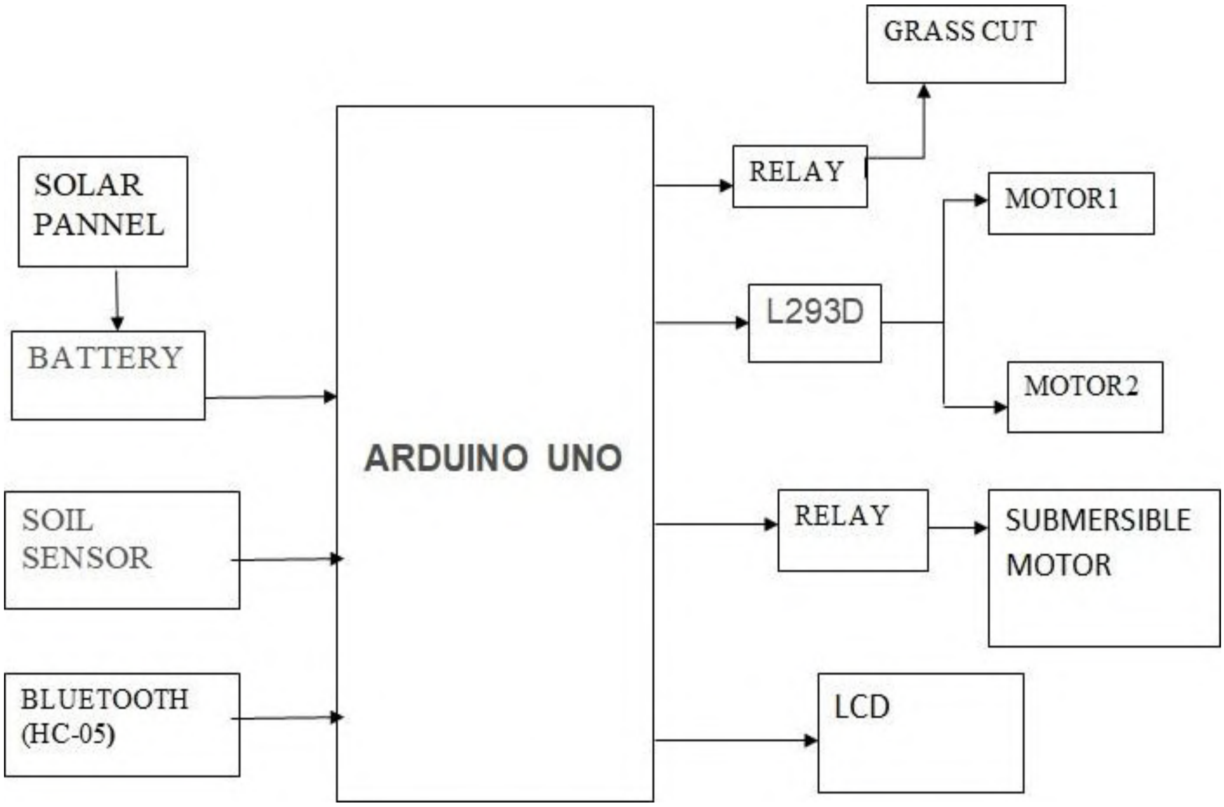


Fig : 3.1 Block diagram

The diagram 3.1 shows the block diagram of solar powered automated multitasking agriculture robot using IOT

The blocks of the circuit are

1. Battery
2. LCD 3.

L293D

4. Motors

5. Soil sensor
6. Arduino UNO
7. 2-Relay
8. Submersible Motor
9. Bluetooth (HC-05)

- Firstly the soil moisture senses whether the soil is in dry or wet condition and the information will be provided to the Arduino.
- The Arduino receives the information from the soil sensor.
- The Arduino output is given to the relay to on/off the submersible motor.
- The command will be displayed in the LCD.

3.2 COMPONENT DESCRIPTION:

3.2.1 ARDUINO UNO:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

3.2.2 Soil Sensor:

Soil moisture sensors measure the volumetric water content in soil. Since the direct gravimetric measurement of free soil moisture requires removing, drying, and weighing of a sample, soil moisture sensors measure the volumetric water content indirectly by using some other property of the soil, such as electrical resistance, dielectric constant, or interaction with neutrons, as a proxy for the moisture content.

3.2.3 LCD:

A liquid crystal display (LCD) is a thin, flat display device made up of any number of colour or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists to allow it to pass through other.

3.2.4 RELAY:

A relay is an electrically operated switch. These are remote control electrical switches that are controlled by another switch, such as a horn switch or a computer as in a power train control module, devices in industries, home based applications. Relays allow a small current pin, 4-pin, 5-pin, and 6-pin, single switch or dual switches. Relays are used throughout the automobile. Relays which come in assorted sizes, ratings, and applications, are used as remote control switches. A typical vehicle can have 20 relays or more.

3.2.5 BLUETOOTH:

Bluetooth is a short-range wireless technology standard used for exchanging data between fixed and mobile devices over short distances using UHF radio waves in the ISM bands, from 2.402 GHz to 2.48 GHz, and building personal area networks (PANs). It was originally conceived as a wireless alternative to RS-232 data cables. It is mainly used as an alternative to wire connections, to exchange files between nearby portable devices and connect cell phones and music players with wireless headphones. In the most widely used mode, transmission power is limited to 2.5 milli watts, giving it a very short range of up to 10 meters (30 feet).

3.2.6 L293D:

The L293D is quadruple high-current half-H drivers. It is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications. All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo- Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

3.2.7 BATTERY:

A rechargeable battery, storage battery, or secondary cell, (or archaically accumulator) is a type of electrical battery which can be charged, discharged into a load, and recharged many times, as opposed to a disposable or primary battery, which is supplied fully charged and discarded after use. It is composed of one or more electrochemical cells. The term "accumulator" is used as it accumulates and stores energy through a reversible electrochemical reaction. Rechargeable batteries are produced in many different shapes and sizes, ranging from button cells to megawatt systems connected to stabilize an electrical distribution network. Several different combinations of electrode materials and electrolytes are used, including lead–acid, zinc–air, nickel– cadmium (NiCd), nickel–metalhydride (NiMH), lithium-ion (Li-ion), lithium connected iron phosphate (LiFePO4), and lithium-ion polymer (Li-ion polymer).

3.2.8 SUBMERSIBLE MOTOR PUMP:

A submersible pump (or sub pump, electric submersible pump (ESP)) is a device which has a hermetically sealed motor close-coupled to the pump body. The whole assemblyis submerged in the fluid to be pumped. The main advantage of this type of pump is that it prevents pump cavitation, a problem associated with a high elevation difference between the pump and the fluid surface. Submersible pumps push fluid to the surface, ratherthan jet pumps, which create a vacuum and rely upon atmospheric pressure. Submersibles use pressurized fluid from the surface to drive a hydraulic motor downwhole, rather than an electric motor, and are used in heavy oil applications with heated water as the motive fluid.

CHAPTER 4

SOFTWARE AND HARDWARE COMPONENTS

4.1 Proteus:

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

4.1.1 About Proteus

It is a software suite containing schematic, simulation as well as PCB designing.

ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.

The designer can also develop 2D drawings for the product.

4.1.2 Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through whole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

4.2 Starting New Design

Step 1: Open ISIS software and select new design in File menu

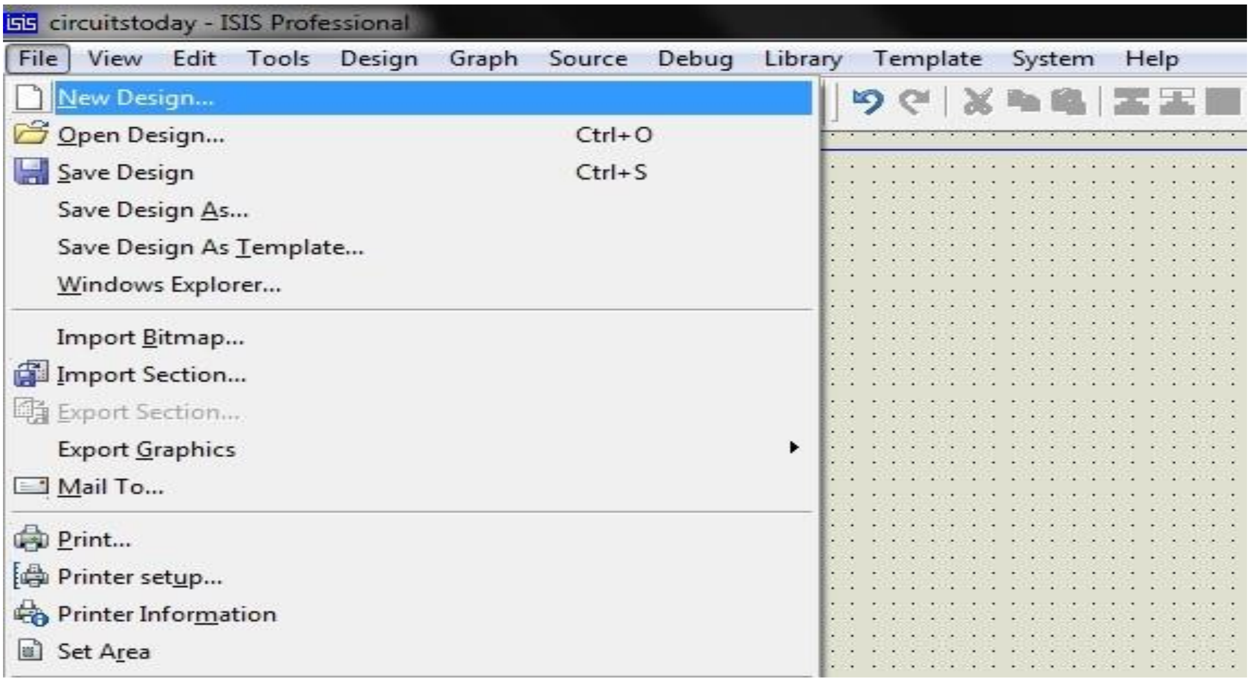


Figure 4.2.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

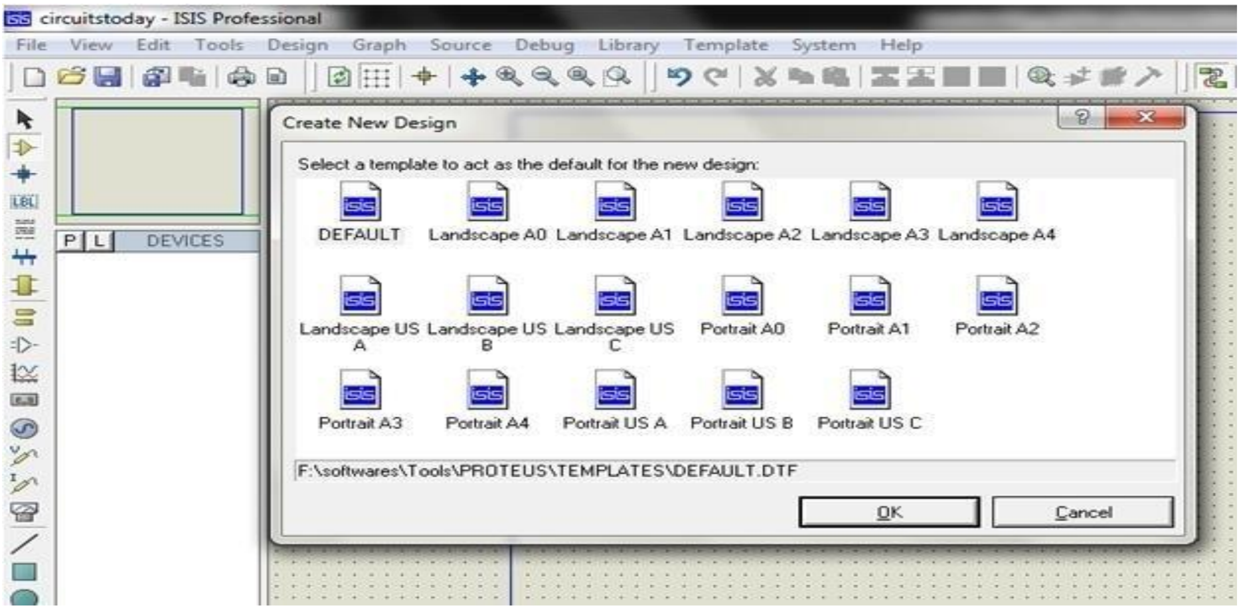


Figure 4.2.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

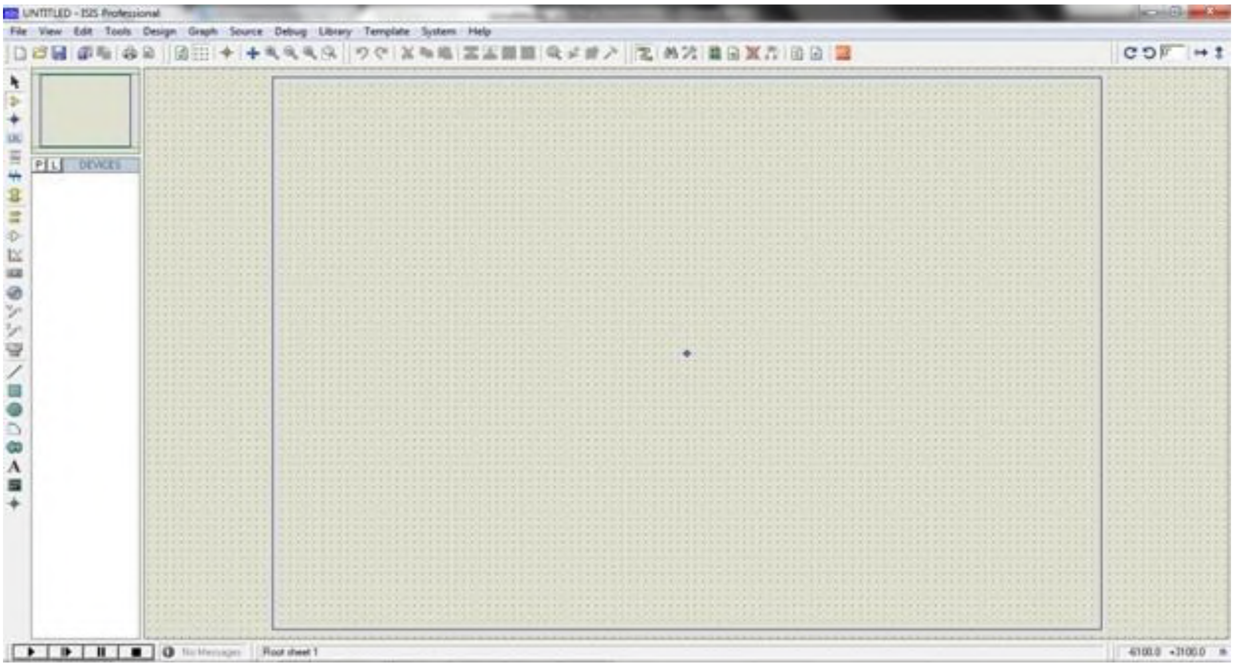


Figure 4.2.3 Proteus Design Sheet

Step 4: To select components, Click on the component mode button.

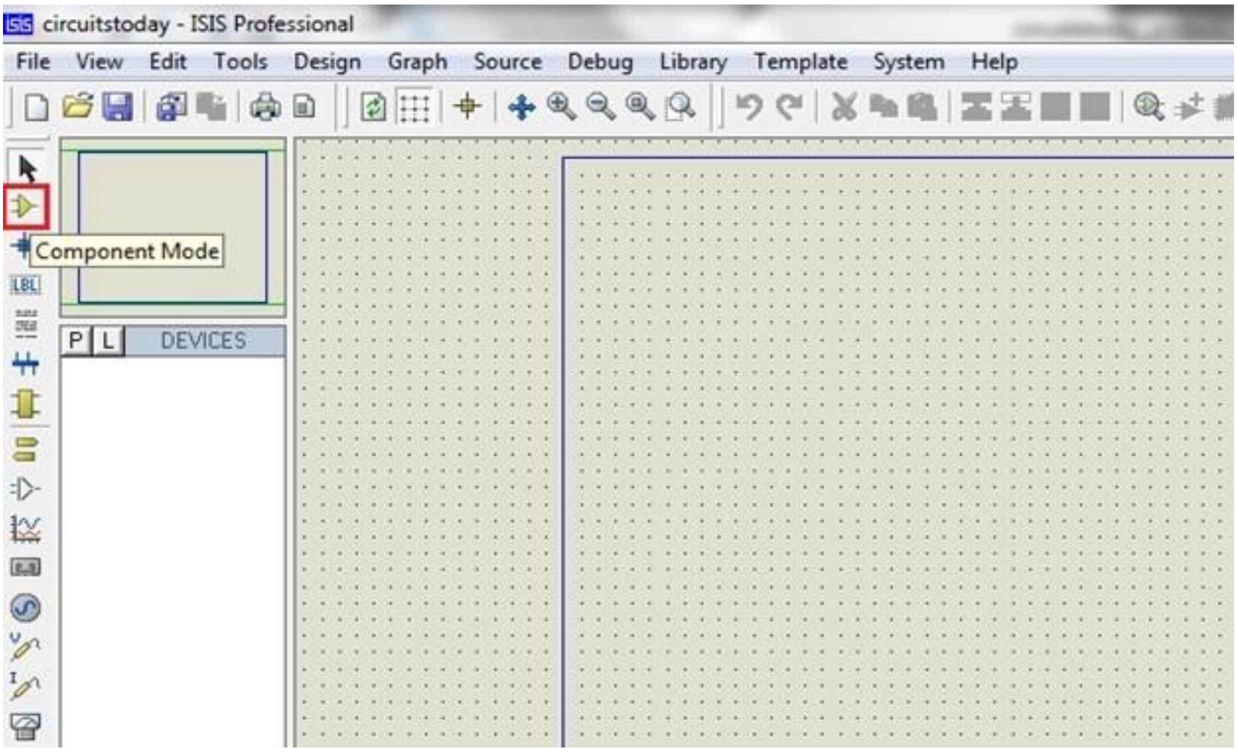


Figure 4.2.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

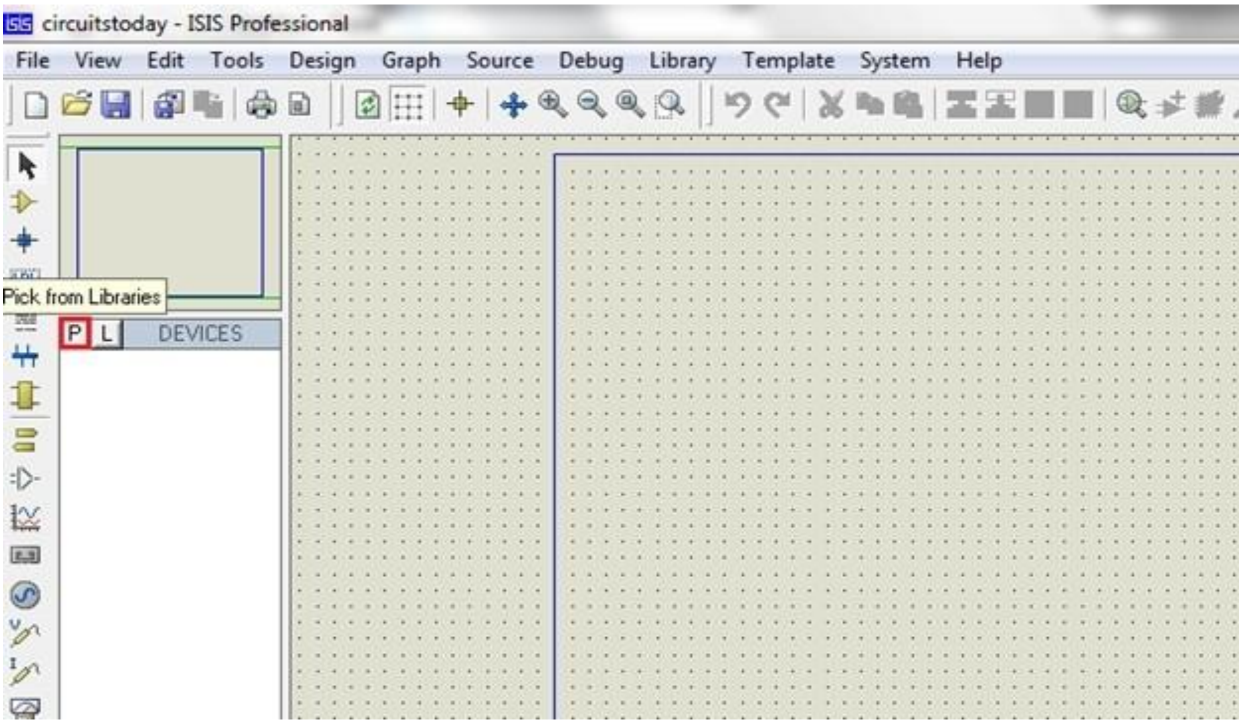


Figure 4.2.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

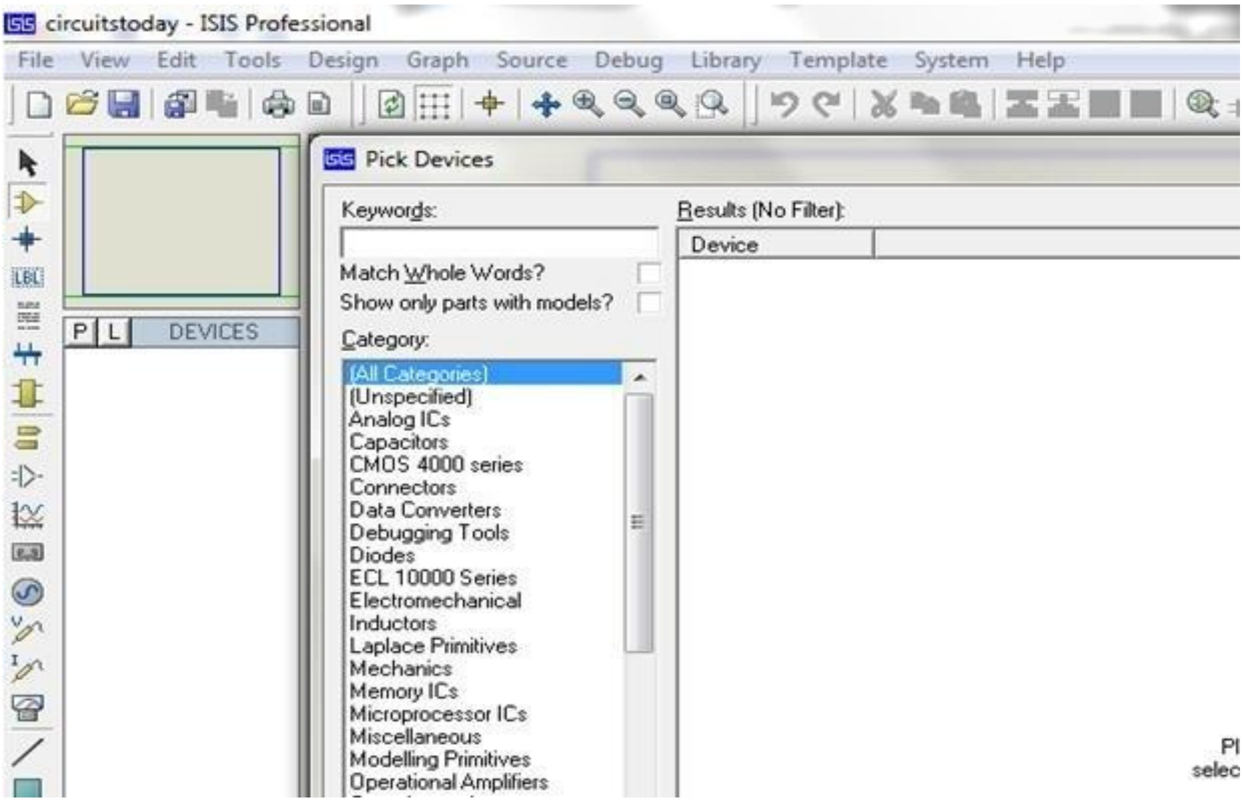


Figure 4.2.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

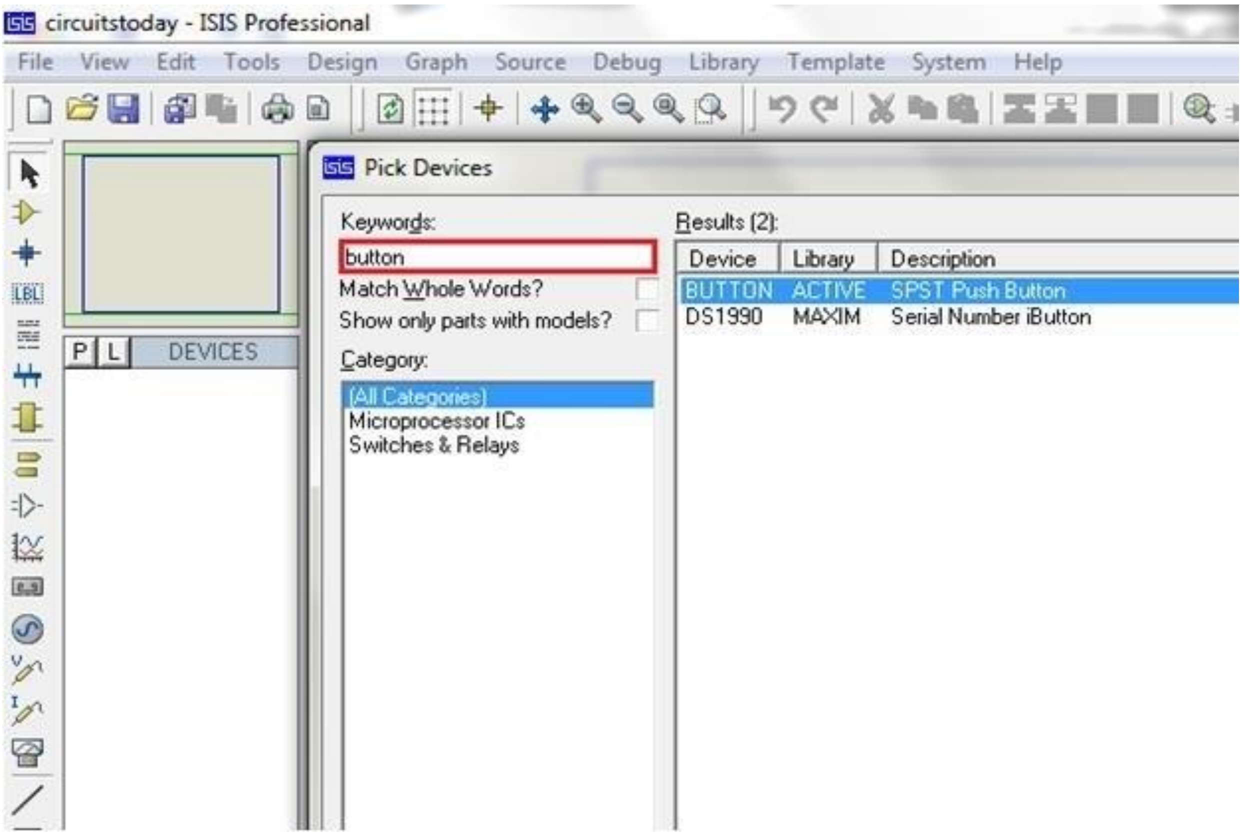


Figure 4.2.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

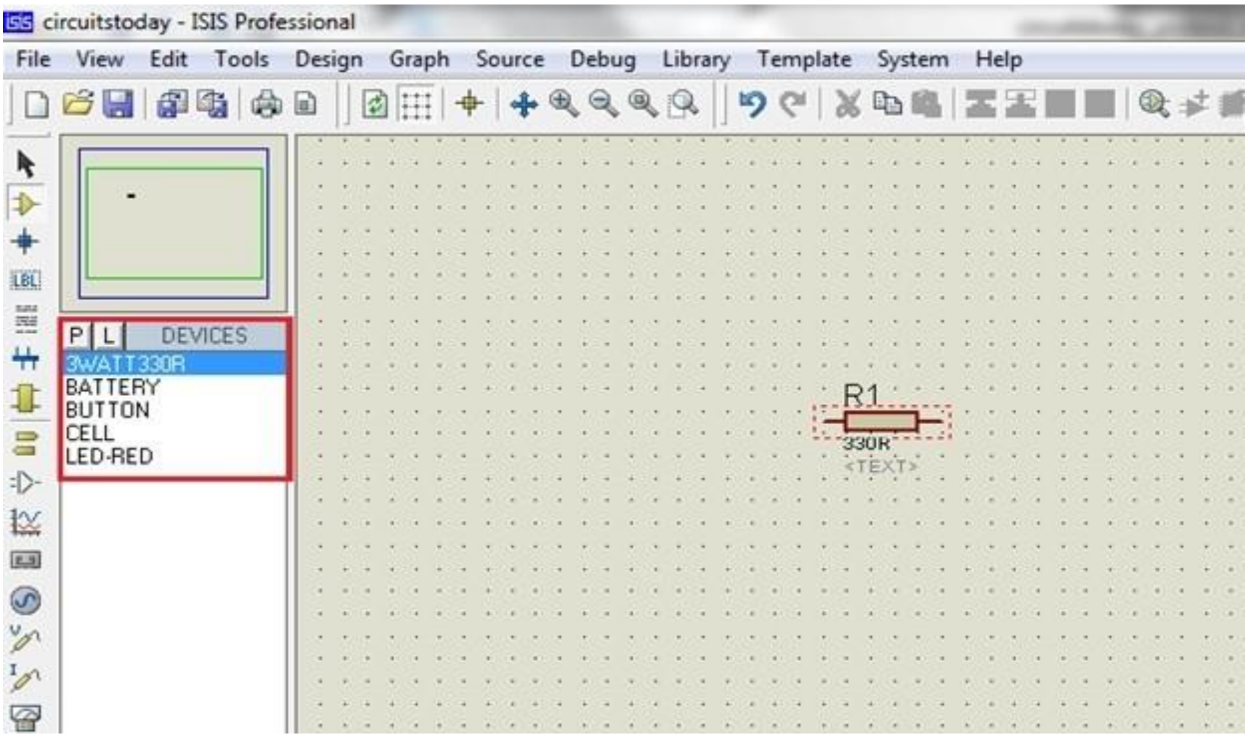


Figure 4.2.8 Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

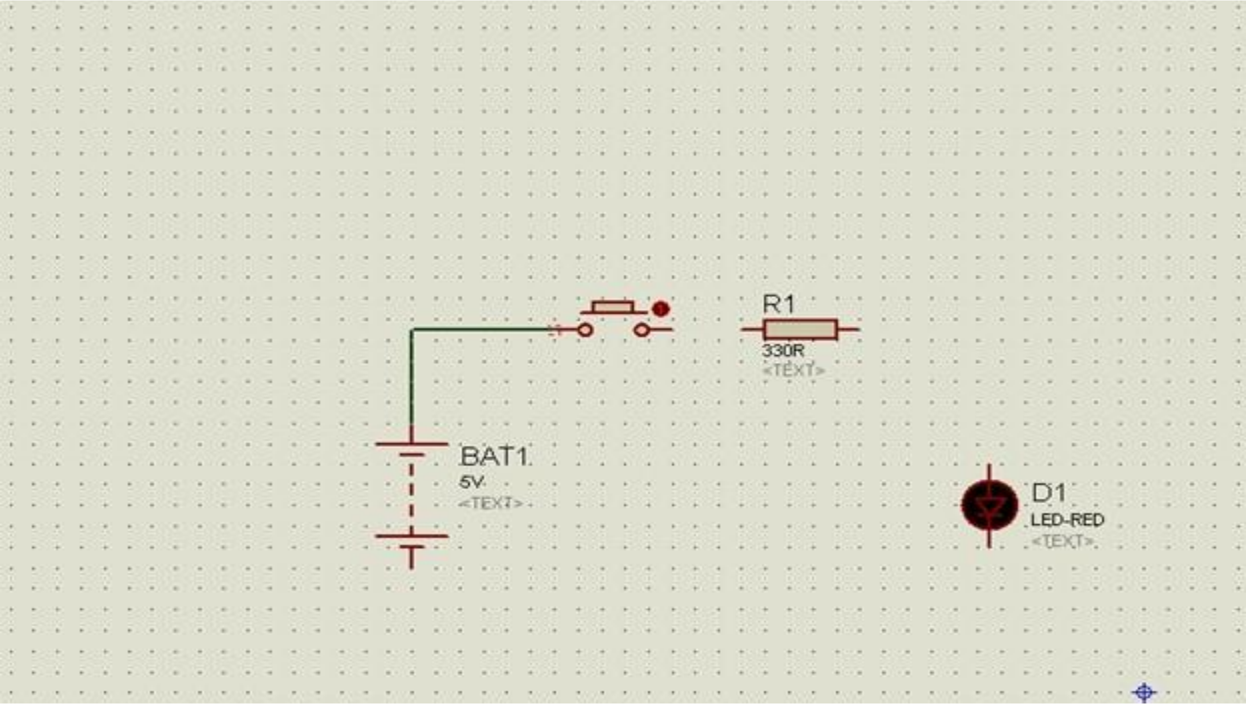


Figure 4.2.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

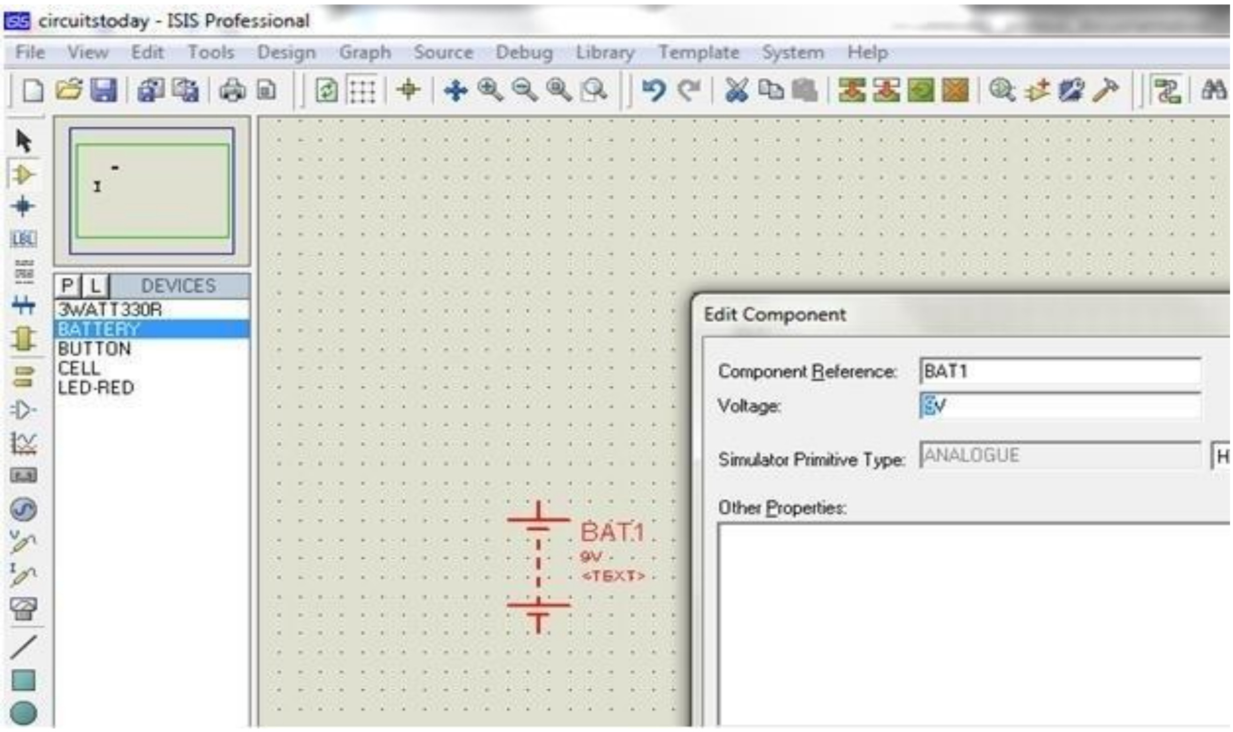


Figure 4.2.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

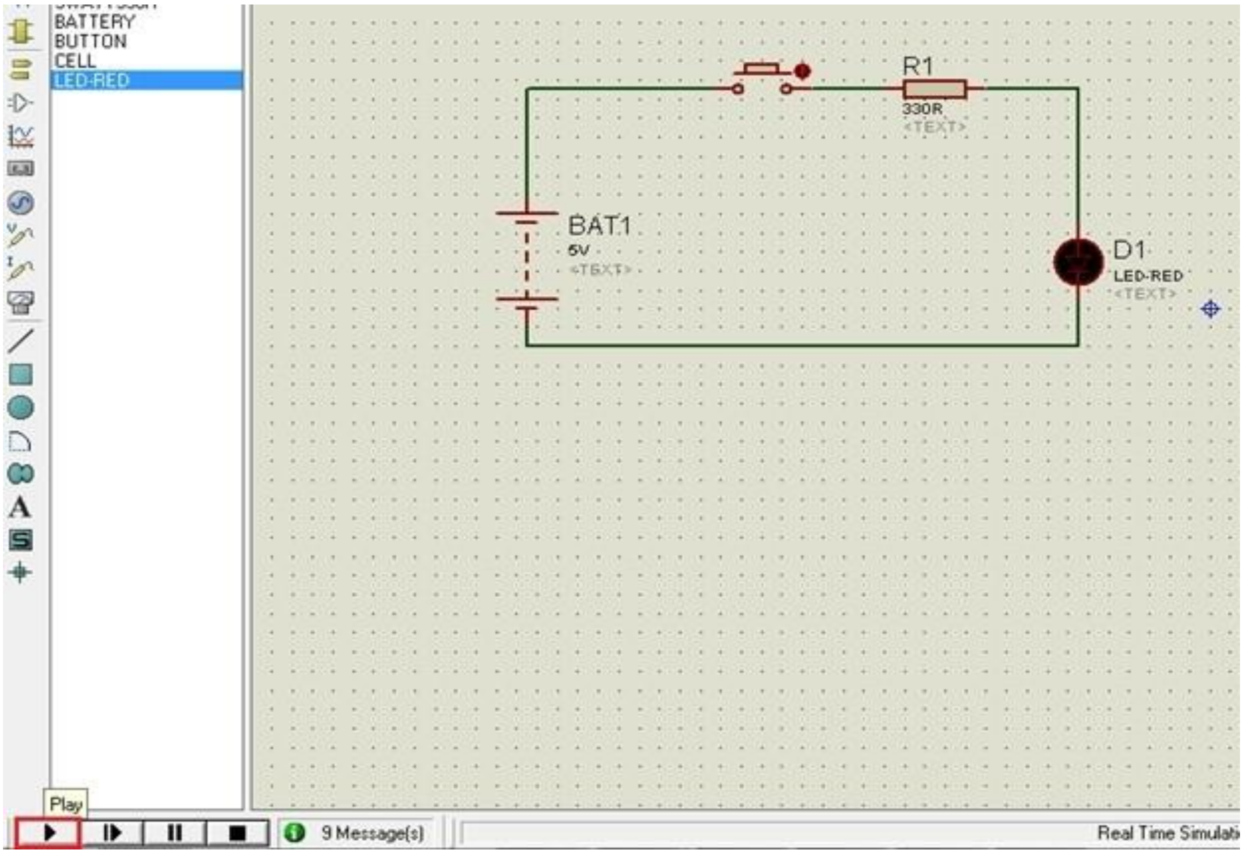


Figure 4.2.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

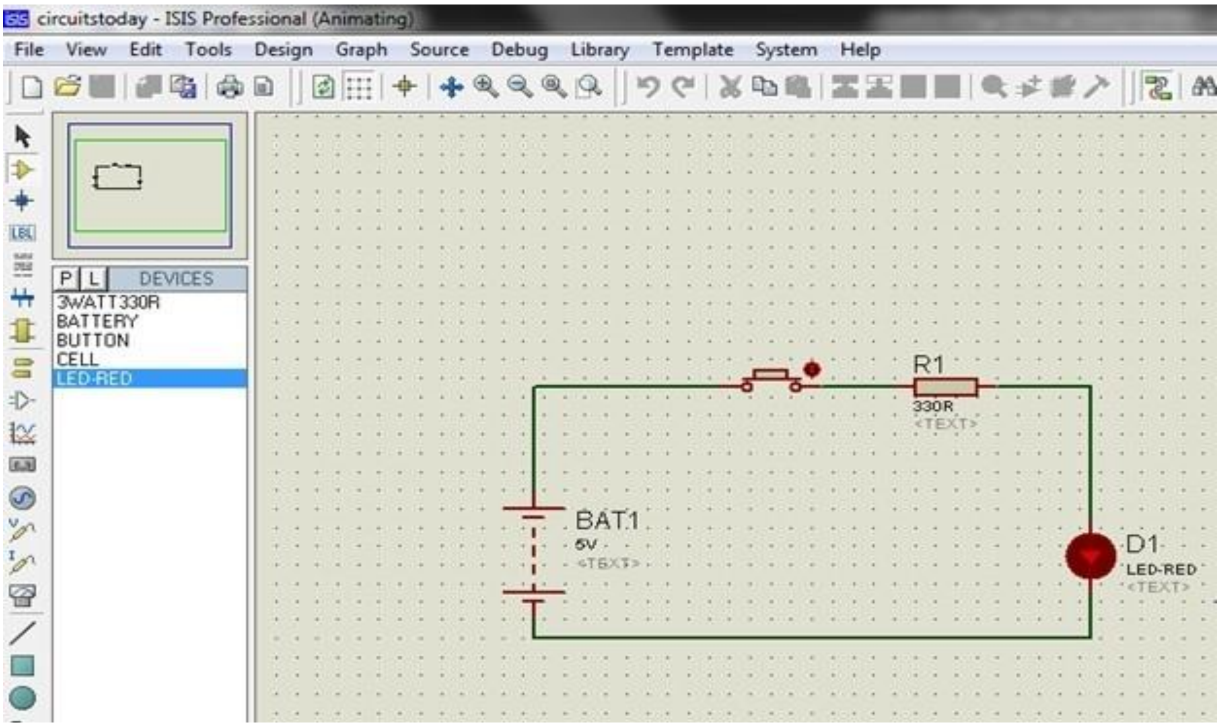


Figure 4.2.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

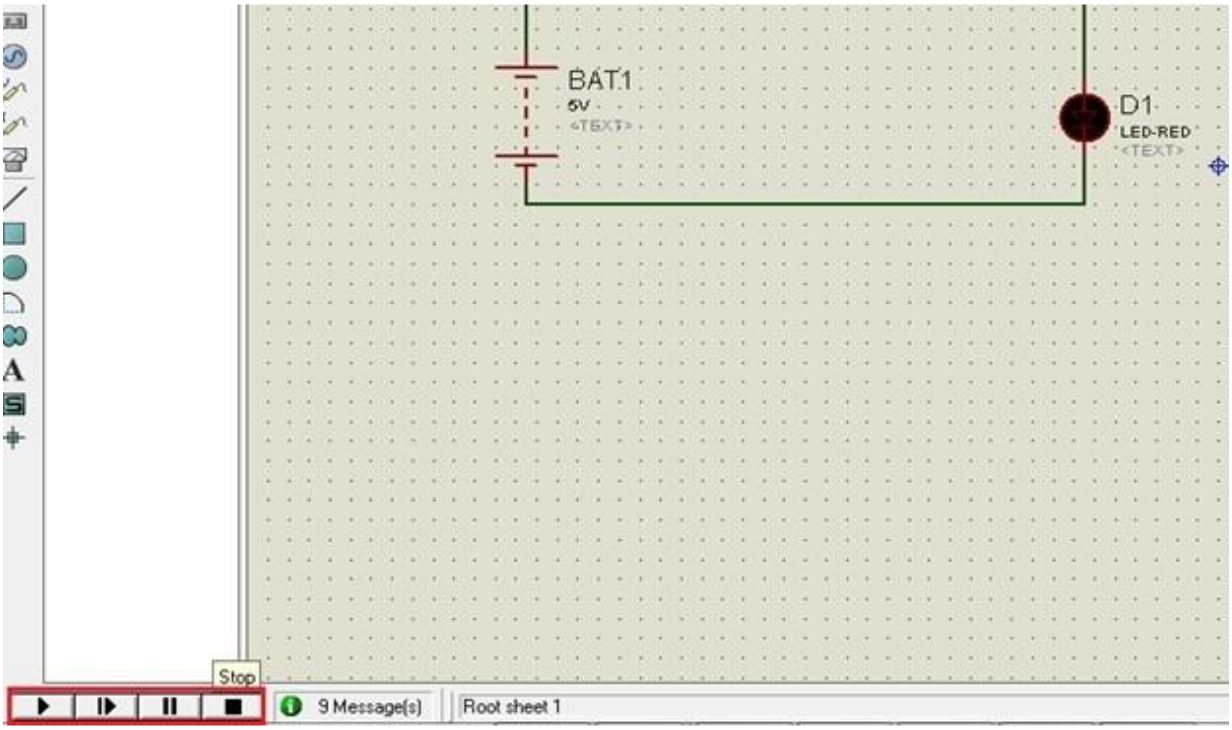


Figure 4.2.13 Simulation Step-Pause-Stop Buttons

4.3 Arduino IDE

The **Arduino Integrated Development Environment (IDE)** is a cross- platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Arduino Pro IDE	
Developer(s)	Arduino Software
Preview release	v0.0.2 / 28 October 2019; 3 months ago[8]
Repository	github.com/arduino/Arduino
Written in	C, C++
Operating system	Windows, macOS, Linux
Platform	IA-32, x86-64, ARM
Type	Integrated development environment
License	LGPL or GPL license
Website	blog.arduino.cc/2019/10/18/arduino-pro-ide-alpha-preview-with-advanced-features/

In October 2019 the Arduino organization began providing early access to a new Arduino Pro IDE with debugging [9] and other advanced features.

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

4.3.1 Installation

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Figure 4.3.1 USB Cable

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

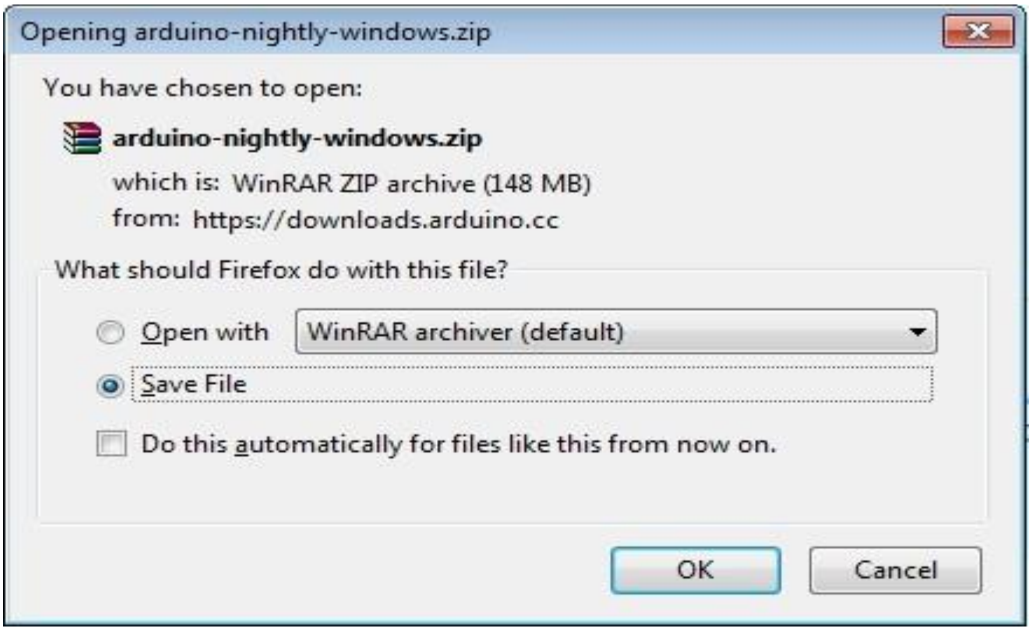


Figure 4.3.2 Downloading Arduino IDE

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece ofplastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labelled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

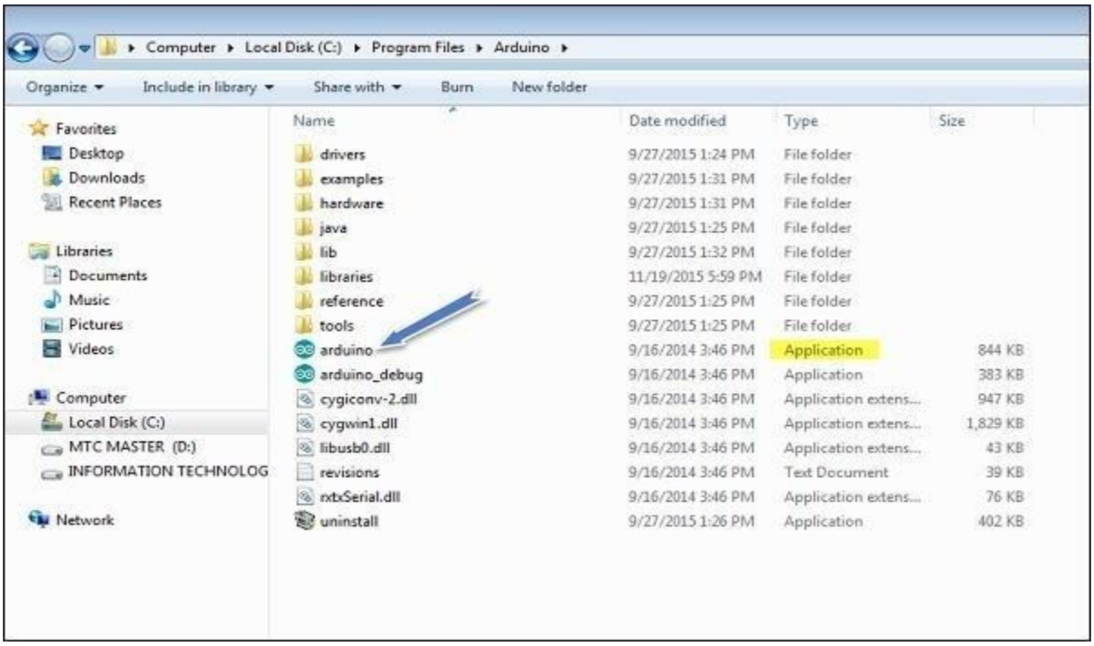


Figure 4.3.3Launching Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.

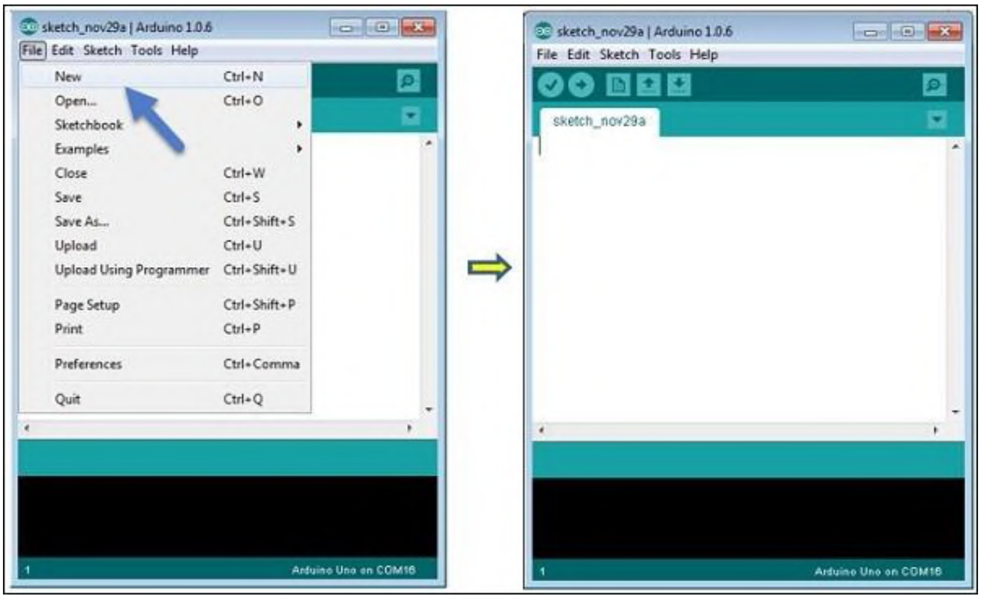


Figure 4.3.4Opening first project

To open an existing project example, select File → Example → Basics → Blink.

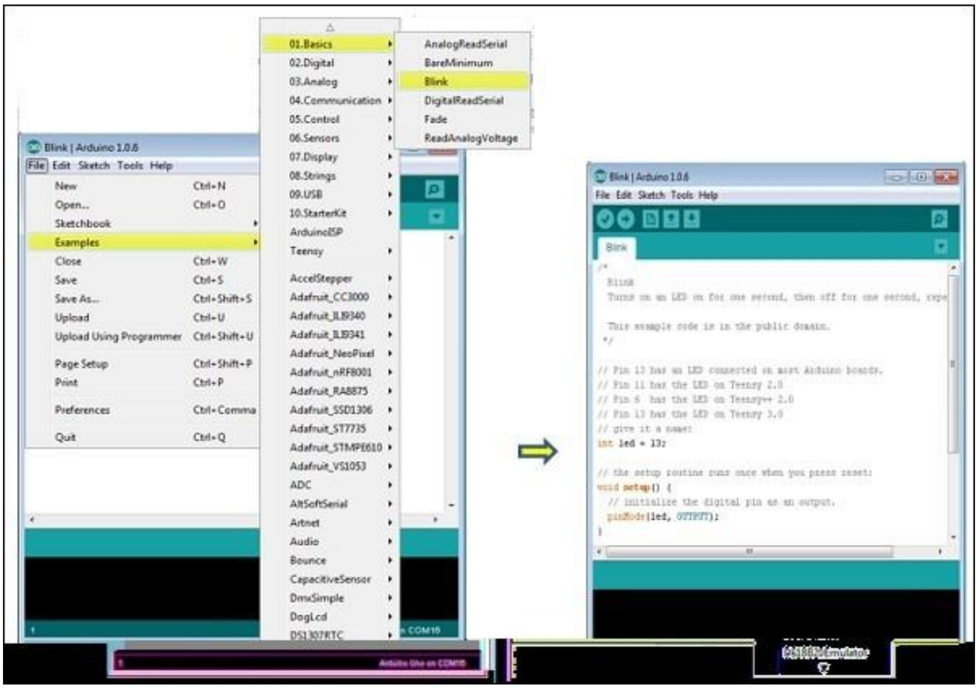


Figure 4.3.4.1Opening first project

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

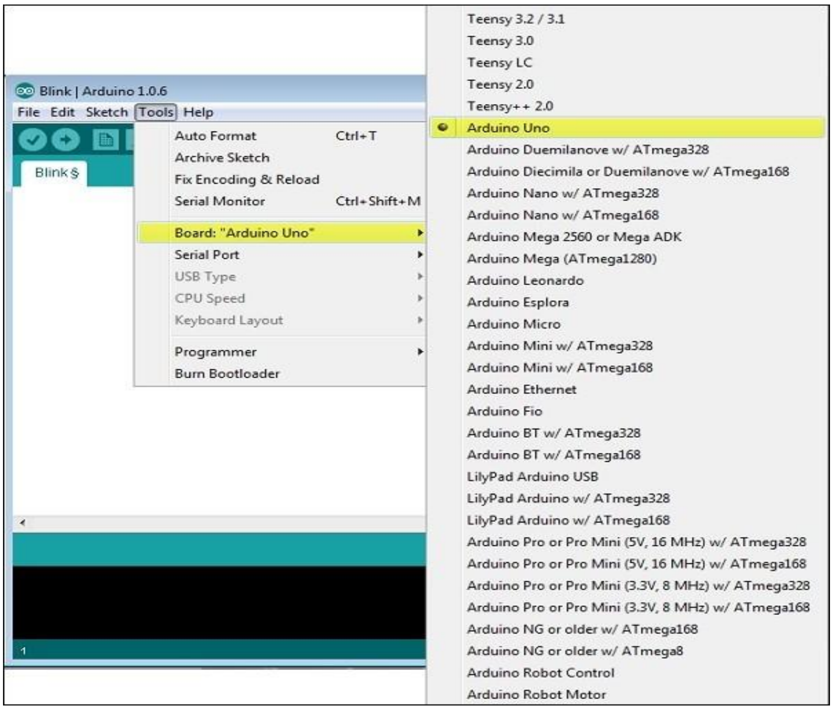


Figure 4.3.5 Selecting Arduino board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

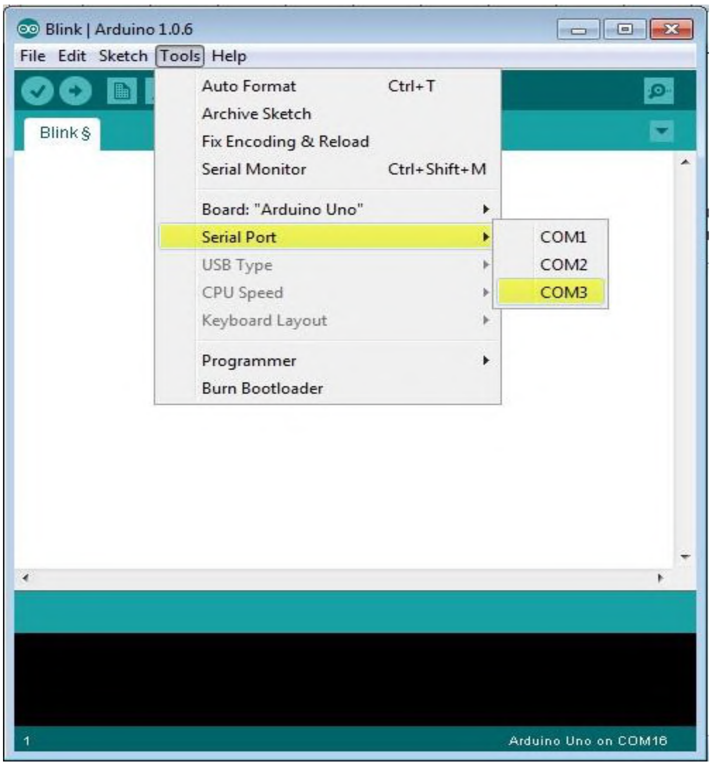


Figure 4.3.6 Selecting serial port

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

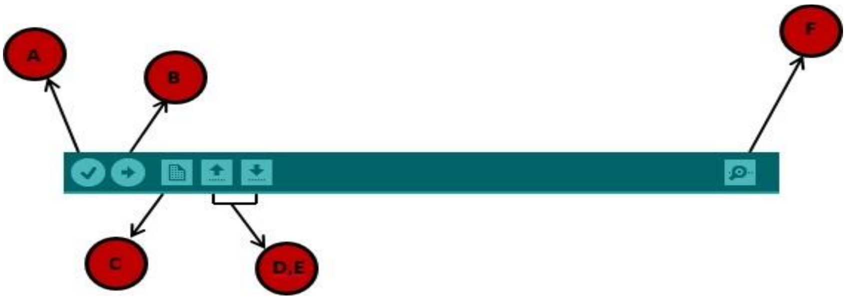


Figure: 4.3.7 Uploading the program to board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

4.4 Arduino

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board. The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

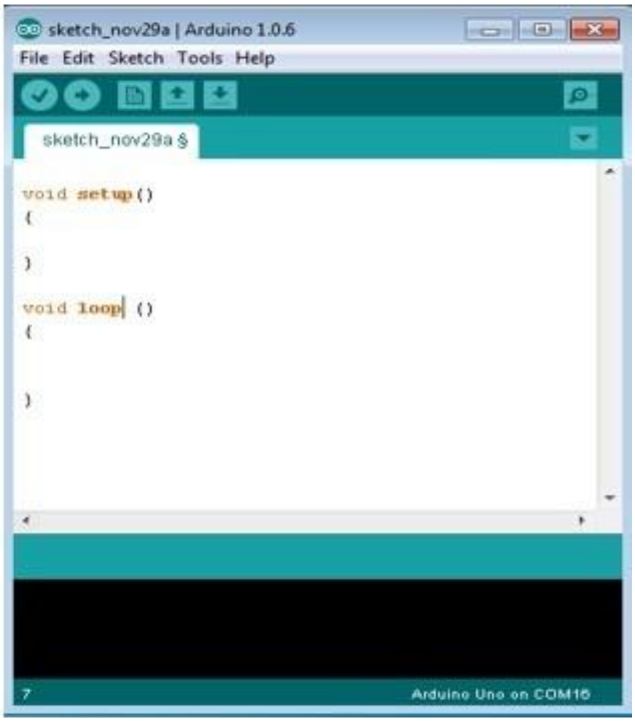


Figure 4.4 Arduino Platform

4.4.1 Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Table 4.4.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI

Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Lily Pad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Lily Pad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 4.4.2Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB

Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
Lily Pad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 4.4.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

4.4.4 Arduino boards based on AT91SAM3X8E microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

4.5 Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

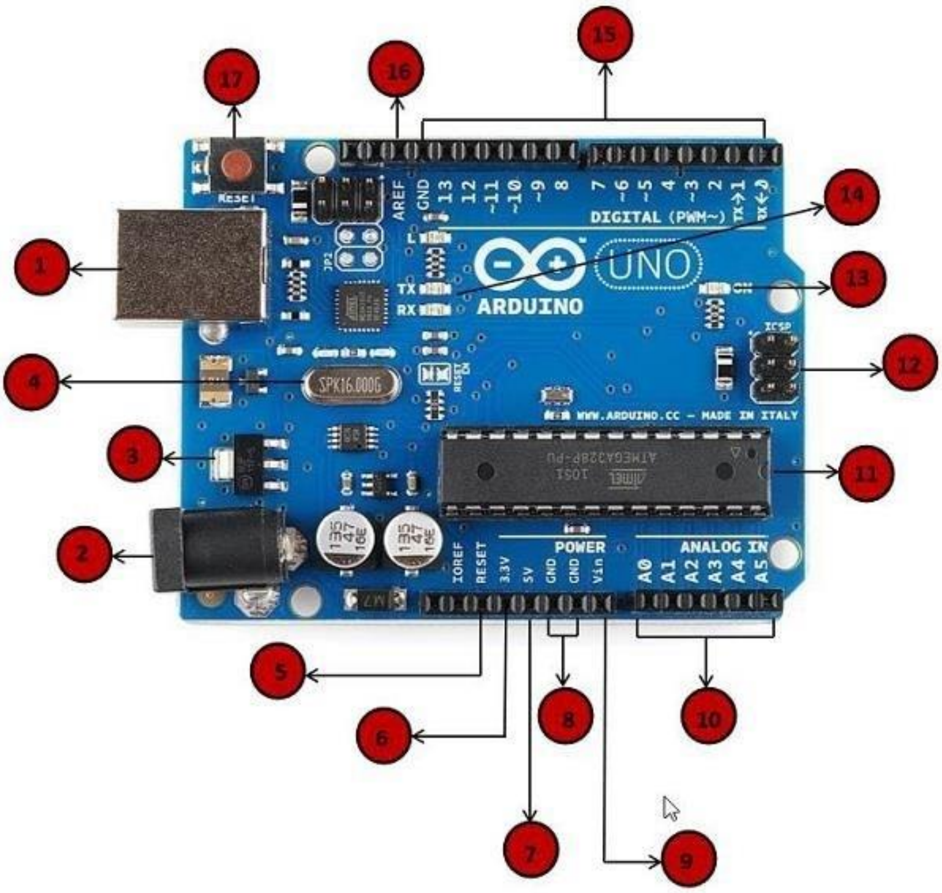


Fig: 4.5 Arduino pins

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz’s

5,17	**Arduino Reset** You can reset your Arduino board, i.e., starts your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data.The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labelled "~" can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Fig: 4.5.1 Pin description

4.6 Liquid Cristal Display

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to a controller is an LCD display. Some of the most common LCDs connected to the controllers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

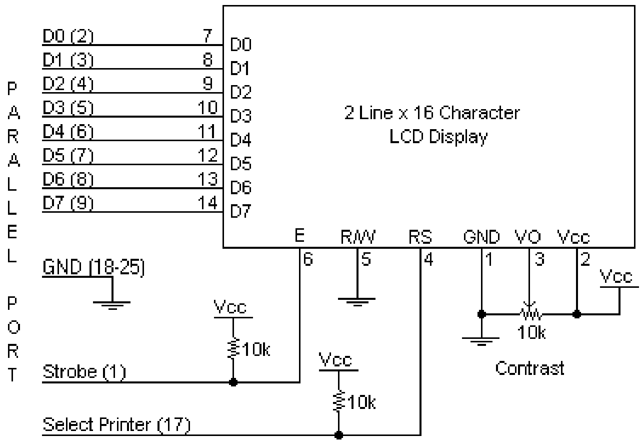
Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5

D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines(RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

4.6.1 Description Of 16x2:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all-Parallel Ports. It however doesn't show the use of the Status Port as an input. So what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

Schematic Diagram



s

Figure 4.6.1Description Of 16x2

- Above is the quite simple schematic. The LCD panel's *Enable* and *Register Select* is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there is a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.
- We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

- The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. You can use a bench power supply set to 5v or use a on board +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.

4.6.2 16 x 2 Alphanumeric LCD Module Features:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM prov
- iding simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

4.6.3 16 x 2 Alphanumeric LCD Module Specifications:

Pin	Symbol	Level	Function
1	V_{SS}	-	Power, GND
2	V_{DD}	-	Power, 5V
3	Vo	-	Power, for LCD Drive
4	RS	H/L	Register Select Signal H: Data Input L: Instruction Input
5	R/W	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)
6	E	H,H->L	Enable
7-14	DB0-DB7	H/L	Data Bus; Software selectable 4- or 8-bit mode
15	NC	-	**NOT CONNECTED**
16	NC	-	**NOT CONNECTED**

4.6.4 FEATURES:

• 5 x 8 dots with cursor

• Built-in controller (KS 0066 or Equivalent)

• + 5V power supply (Also available for + 3V)

• 1/16 duty cycle

• B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)

• N.V. optional for + 3V power supply

Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are:

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

4.6.4.1 Figure Address locations for a 1x16 line LCD

Even limited to character based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8,16,20,24,32 and 40 characters are all standard, in one, two and four line versions.

Several different LC technologies exist. “Super twist” types, for example, offer improved contrast and viewing angle over the older “twisted nematic” types. Some modules are available with back lighting, so that they can be viewed in dimly-lit conditions. The back lighting may be either “electro-luminescent”, requiring a high voltage inverter circuit, or simple LED illumination.

4.6.5 PIN DESCRIPTION:

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections).

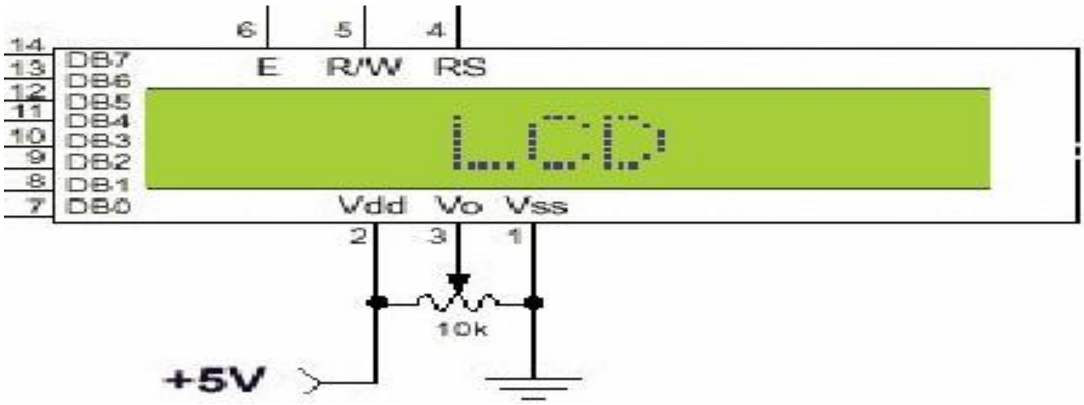


Figure 4.6.5 Pin diagram of 1x16 lines LCD

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Figure 4.6.6 Pin specifications

4.7 CONTROL LINES:

EN: Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS: Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW: Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands, so RW will almost always be low. Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Logic status on control lines:

• E - 0 Access to LCD disabled

- 1 Access to LCD enabled

• R/W - 0 Writing data to LCD

- 1 Reading data from LCD

• RS - 0 Instructions

- 1 Character

Writing data to the LCD:

1) Set R/W bit to low
2) Set RS bit to logic 0 or 1 (instruction or character)
3) Set data to data lines (if it is writing)
4) Set E line to high
5) Set E line to low

Read data from data lines (if it is reading) on LCD:

1) Set R/W bit to high
2) Set RS bit to logic 0 or 1 (instruction or character)
3) Set data to data lines (if it is writing)
4) Set E line to high
5) Set E line to low

Entering Text:

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the d.i.l. switch pack with a of re-wiring is necessary.

4.7.1 LCD Commands:

There are some present commands instructions in LCD, which we need to send to LCD through some microcontroller. Some important command instructions are given below:

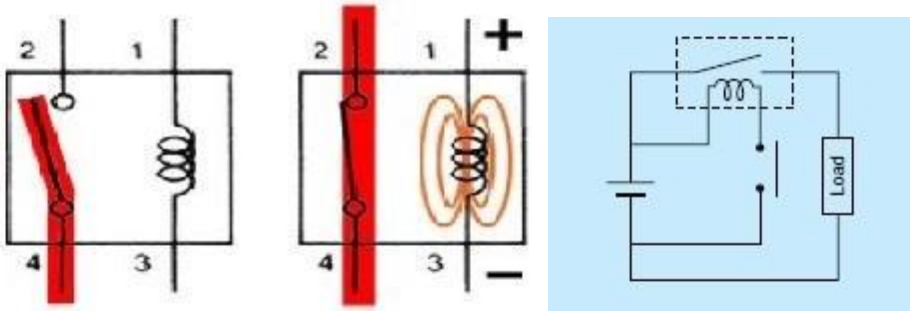
Table 4.7.1 Commands

Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home

04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left
0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

4.8 Relays

A relay is an electrically operated switch. These are remote control electrical switches that are controlled by another switch, such as a horn switch or a computer as in a power train control module, devices in industries, home based applications. Relays allow a small current pin, 4-pin, 5-pin, and 6-pin, single switch or dual switches. Relays are used throughout the automobile. Relays which come in assorted sizes, ratings, and applications, are used as remote control switches. A typical vehicle can have 20 relays or more.



4.8.1 Basics on Relay Handling

- To maintain initial performance, care should be taken to avoid dropping or hitting the relay.

- Under normal use, the relay is designed so that the case will not detach. To maintain initial performance, the case should not be removed. Relay characteristics cannot be guaranteed if the case is removed.
- Use of the relay in an atmosphere at standard temperature and humidity with minimal amounts of dust, SO 2, H 2 S, or organic gases is recommended.
- Please avoid the use of silicon-based resins near the relay, because doing so may result in contact failure. (This applies to plastic sealed type relays, too.)
- Care should be taken to observe correct coil polarity (+, –) for polarized relays.
- Proper usage requires that the rated voltage be impressed on the coil. Use rectangular waves for DC coils and sine waves for AC coils.
- Be sure the coil impressed voltage does not continuously exceed the maximum allowable voltage.
- Absolutely avoid using switching voltages and currents that exceed the designated values.
- The rated switching power and life are given only as guides. The physical phenomena at the contacts and contact life greatly vary depending on the type of load and the operating conditions. Therefore, be sure to carefully check the
- Type of load and operating conditions before use.
- Do not exceed the usable ambient temperature values listed in the catalog.
- Use the flux-resistant type or sealed type if automatic soldering is to be used.
- Use alcohol based cleaning solvents when cleaning is to be performed using a
  sealed type relay.
- Avoid ultrasonic cleaning of all types of relays.
- Avoid bending terminals, because it may cause malfunction.
- As a guide, use a Fasten mounting pressure of 40 to 70N {4 to 7kgf} for relays
  with tab terminals.

4.8.2 Specification

- Number and type of contacts – normally open, normally closed, (double-throw)
- Contact sequence – "Make before Break" or "Break before Make". For example, the old style telephone exchanges required Make-before-break so that the connection didn't get dropped while dialing the number.
- Rating of contacts – small relays switch a few amperes, large contactors are rated for up to 3000 amperes, alternating or direct current

- Voltage rating of contacts – typical control relays rated 300 VAC or 600 VAC, automotive types to 50 VDC, special high-voltage relays to about 15 000 V
- Coil voltage – machine-tool relays usually 24 VAC, 120 or 250 VAC, relays for switchgear may have 125 V or 250 VDC coils, "sensitive" relays operate on a few milli-amperes

4.8.3 Applications:

Relays are used:

- To control a high-voltage circuit with a low-voltage signal, as in some types of modems,
- To control a high-current circuit with a low-current signal, as in the starter solenoid of an automobile,
- To detect and isolate faults on transmission and distribution lines by opening and closing circuit breakers (protection relays),
- To isolate the controlling circuit from the controlled circuit when the two are at different potentials, for example when controlling a mains-powered device from a low-voltage switch. The latter is often applied to control office lighting as the low voltage wires are easily installed in partitions, which may be often moved as needs change. They may also be controlled by room occupancy detectors in an effort to conserve energy,
- To perform logic functions. For example, the Boolean AND function is realized by connecting relay contacts in series, the OR function by connecting contacts in parallel. Due to the failure modes of a relay compared with a semiconductor, they are widely used in safety critical logic, such as the control panels of radioactive waste handling machinery.
- As oscillators, also called vibrators. The coil is wired in series with the normally closed contacts. When a current is passed through the relay coil, the relay operates and opens the contacts that carry the supply current. This stops the current and causes the contacts to close again. The cycle repeats continuously, causing the relay to open and close rapidly. Vibrators are used to generate pulsed current.
- To generate sound. A vibrator, described above, creates a buzzing sound because of the rapid oscillation of the armature. This is the basis of the electric bell, which consistsof a vibrator with a hammer attached to the armature so it can repeatedly strike a bell.

- To perform time delay functions. Relays can be used to act as a mechanical time delay device by controlling the release time by using the effect of residual magnetism by means of a inserting copper disk between the armature and moving blade assembly.

Figure 4.8.3 Relay

4.9 PIN DESCRIPTION OF ATMEGA328

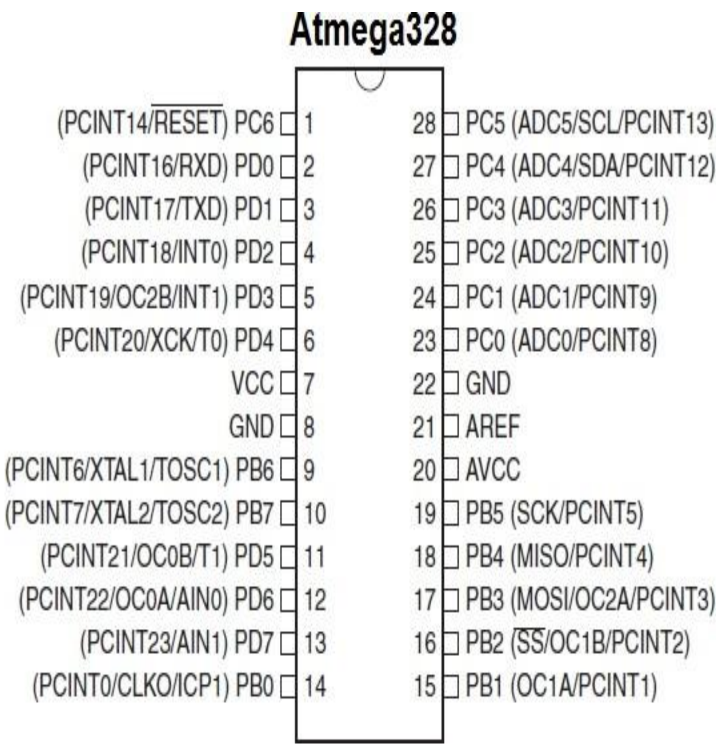


Fig.4.9: Pin description of ATMEGA328

ADVANTAGES OF ARDUINO

- It is cheap
- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
- For beginners, it is very simple to use.

APPLICATIONS

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

4.10 SOLAR PANNEL

The term solar panel is used colloquially for a photo-voltaic (PV) module.

A PV module is an assembly of photo-voltaic cells mounted in a frame work for installation. Photo-voltaic cells use sunlight as a source of energy and generate direct current electricity. A collection of PV modules is called a PV Panel, and a system of Panels is an Array. Arrays of a photovoltaic system supply solar electricity to electrical equipment.

The most common application of solar energy collection outside agriculture is solar water heating systems.

Fig 4.10 : Solar pannel

Photovoltaic modules use light energy (photons) from the Sun to generate electricity through the photovoltaic effect. Most modules use wafer-based crystalline silicon cells or thin-film cells. The structural (load carrying) member of a module can be either the top layer or the back layer. Cells must be protected from mechanical damage and moisture. Most modules are rigid, but semi-flexible ones based on thin-film cells are alsoavailable. The cells are connected electrically in series, one to another to a desired voltage, and then in parallel to increase amperage. The wattage of the module is the mathematical product of the voltage and the amperage of the module.

A PV junction box is attached to the back of the solar panel and functions as its output interface. External connections for most photovoltaic modules use MC4 connectors to facilitate easy weatherproof connections to the rest of the system. A USB power interface can also be used.

Module electrical connections are made in series to achieve a desired output voltage or in parallel to provide a desired current capability (amperes) of the solar panel or the PV system. The conducting wires that take the current off the modules are sized according to the ampacity and may contain silver, copper or other non-magnetic conductive transition metals. Bypass diodes may be incorporated or used externally, in case of partial module shading, to maximize the output of module sections still illuminated.

Some special solar PV modules include concentrators in which light is focused by lenses or mirrors onto smaller cells. This enables the use of cells with a high cost per unit area (such as gallium arsenide) in a cost-effective way.

Solar panels also use metal frames consisting of racking components, brackets, reflector shapes, and troughs to better support the panel structure.

4.11 BATTERY:

Batteries are a collection of one or more cells whose chemical reactions create a flow of electrons in a circuit. All batteries are made up of three basic components: an anode (the '-' side), a cathode (the '+' side), and some kind of electrolyte (a substance that chemically reacts with the anode and cathode).

When the anode and cathode of a battery is connected to a circuit, a chemical reaction takes place between the anode and the electrolyte. This reaction causes electrons to flow through the circuit and back into the cathode where another chemical reaction takes place. When the material in the cathode or anode is consumed or no longer able to be used in the reaction, the battery is unable to produce electricity. At that point, your battery is "dead."

Batteries that must be thrown away after use are known as **primary batteries**. Batteries that can be recharged are called **secondary batteries**.

Fig 4.11(A) : Lithium Battery

Lithium polymer batteries, for example, can be recharged without batteries, your quadcopter would have to be tethered to the wall, you would have to hand crank your car, and your Xbox controller would have to plugged in all the time (like in the good old days). Batteries offer a way to store electrical potential energy in a portable container.

Fig4.11(B) : Batteries come in a variety of shapes, sizes, and chemistries.

4.12 History of battery

The Term Battery

Historically, the word "battery" was used to describe a "series of similar objects grouped together to perform a function," as in a battery of artillery. In 1749, Benjamin Franklin first used the term to describe a series of capacitors he had linked together for his electricity experiments. Later, the term would be used for any electrochemical cells linked together for the purpose of providing electric power.

Fig 4.12(A) : Battery of Leyden Jar

Battery of Leyden Jar "capacitors" linked together (Image courtesy of Alvinrune of Wikimedia Commons)

4.12.1 Invention of the Battery

One fateful day in 1780, Italian physicist, physician, biologist, and philosopher, Luigi Galvani, was dissecting a frog attached to a brass hook. As he touched the frog's leg with an iron scapel, the leg twitched. Galvani theorized that the energy came from the leg itself, but his fellow scientist, Alessandro Volta, believed otherwise.

Volta hypothesized that the frog's leg impulses were actually caused by different metals soaked in a liquid. He repeated the experiment using cloth soaked in brine instead of a frog corpse, which resulted in a similar voltage. Volta published his findings in 1791 and later created the first battery, the voltaic pile, in 1800.

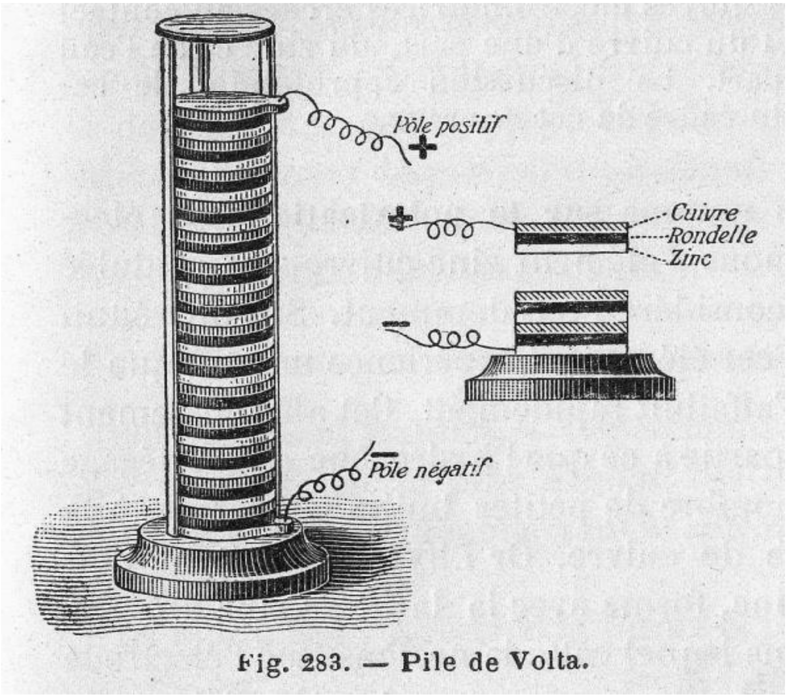


Fig 4.12.1.1: pile de volta

The voltaic pile consisted of a stack of zinc and copper plates separated by cloth soaked in brine. Volta's pile was plagued by two major issues: the weight of the stack caused the electrolyte to leak out of the cloth, and the particular chemical properties of the components resulted in a very short life span (about an hour). The next two hundred years would be spent perfecting Volta's design and solving these issues.

4.12.2 Fixes to the Voltaic Pile

William Cruickshank of Scotland solved the leakage problem by laying the voltaic pile on its side to form the "trough battery."

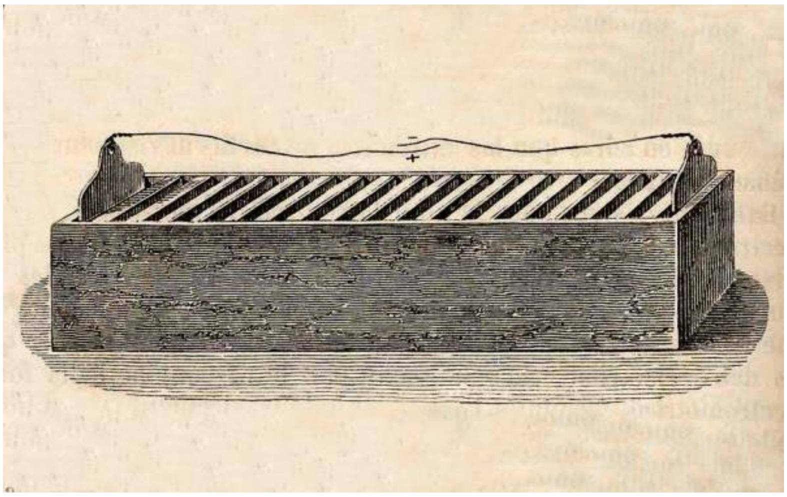


Fig4.12.2.1 : The trough battery solved the leakage problem of the voltaic pile

The second problem, short life span, was caused by the degradation of the zinc due to impurities and a build-up of hydrogen bubbles on the copper. In 1835, William Sturgeon discovered that treating the zinc with mercury would prevent degradation.

The British chemist John Frederic Daniell used a second electrolyte that reacted with the hydrogen, preventing buildup on the copper cathode. Daniell's two-electrolyte battery, known as the "Daniell cell," would become a very popular solution to providing power to the budding telegraph networks.

Fig4.12.2.2 : A collection of Daniell cells from 1836

4.12.3 The First Rechargeable Battery

In 1859, the French physicist Gaston Planté created a battery using two rolled sheets of lead submerged in sulfuric acid. By reversing the electrical current through the battery, the chemistry would return to its original state, thus creating the first rechargeable battery.

Later, in 1881, Camille Alphonse Faure improved Planté's design by forming the lead sheets into plates. This new design made the batteries easier to manufacture, and the lead acid battery saw wide-spread use in automobiles.

Fig 4.12.3.1 : Rechargeable Battery

-> The design for the common "car battery" has been around for more than 100 years (Image courtesy of Emilian Robert Vicol of Wikimedia Commons) <-

4.12.4 The Dry Cell

Up until the late 1800s, the electrolyte in batteries was in a liquid state. This made battery transportation a very careful endeavor, and most batteries were never intended to be moved once attached to the circuit.

In 1866, Georges Leclanché created a battery using a zinc anode, a manganese dioxide cathode, and an ammonium chloride solution for the electrolyte. While the electrolyte in the Leclanché cell was still a liquid, the battery's chemistry proved to be an important step for the invention of the dry cell.

Carl Gassner figured out how to create an electrolyte paste out of ammonium chloride and Plaster of Paris. He patented the new "dry cell" battery in 1886 in Germany. These new dry cells, commonly called "zinc-carbon batteries," were massed produced and proved hugely popular until the late 1950s. While carbon is not used in the chemical reaction, it performs an important role as an electrical conductor in the zinc-carbon battery.

Fig 4.12.4.1: Dry cell

-> 3V zinc-carbon battery from the 1960s (Image courtesy of PhFabre of Wikimedia Commons)

<- In the 1950s, Lewis Urry, Paul Marsal, and Karl Kordesch of the Union Carbide company (later known as "Eveready" and then "Energizer") replaced the ammonium chloride electrolyte with an alkaline substance, based on the battery chemistry formulated by Waldemar Jungner in 1899. Alkaline dry cell batteries could hold more energy than zinc carbon batteries of the same size and had a longer shelf life.

Alkaline batteries rose in popularity in the 1960s, overtook zinc-carbon batteries, and have since become the standard primary cell for consumer use.

Fig 4.12.4.2 batteries

-> Alkaline batteries come in many shapes and sizes (Image courtesy of Aney~commonswiki of Wikimedia Commons) <-

4.12.5 20th Century Rechargeable Batteries

In the 1970s, COMSAT developed the nickel-hydrogen battery for use in communication satellites. These batteries store hydrogen in a pressurized, gaseous form. Many man-made satellites, like the International Space Station, still rely on nickel-hydrogen batteries.

The research of several companies since the late 1960s resulted in the creation of the nickel-metal hydride (NiMH) battery. NiMH batteries were released to the consumer market in 1989, and provided a smaller, cheaper alternative to the rechargeable nickel-hydrogen cells.

Asahi Chemical of Japan built the first lithium-ion battery in 1985, and Sony created the first commercial lithium-ion battery in 1991. In the late 1990s, a soft, flexible casing was created for lithium-ion batteries and gave rise to the "lithium polymer" or "LiPo" battery.

Fig 4.12.5.1 : lithium-ion battery

The chemical reactions in the lithium polymer battery are essentially the same as those in the lithium-ion battery. Obviously, many more battery chemistries have been invented, manufactured, and become obsolete. If you would like to read more about modern, popular battery technologies, check out our Battery Technologies tutorial.

4.12.6 Components

Batteries are made up of three basic components: an **anode**, a **cathode**, and an **electrolyte**. A **separator** is often used to prevent the anode and cathode from touching, if the electrolyte is not sufficient. In order to store these components, batteries usually have some kind of **casing**.

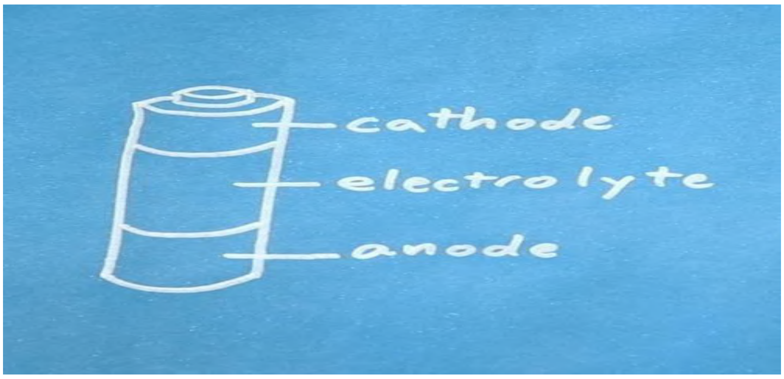


Fig 4.12.6.1 : Electrode

OK, most batteries are not actually divided up in three equal sections, but you get the idea. A better cross-section of an alkaline cell can be found on Wikipedia.

Both the anode and cathode are types of **electrodes**. Electrodes are conductors through which electricity enters or leaves a component in a circuit.

4.12.6(A) Anode

Electrons flow out from the anode in a device connected to a circuit. This means that conventional "current" flows into an anode.

Fig4.12.6(A).1: On batteries, the anode is marked as the negative (-) terminal

In a battery, the chemical reaction between the anode and electrolyte causes a build up of electrons in the anode. These electrons want to move to the cathode, but cannot pass through the electrolyte or separator.

4.12.6(B) Cathode

Electrons flow into the cathode in a device connected to a circuit. This means that conventional "current" flows out from a cathode.

Fig4.12.6(B).1: On batteries, the anode is marked as the negative (+) terminal

In batteries, the chemical reaction in or around the cathode uses the electrons produced in the anode. The only way for the electrons to get to the cathode is through a circuit, external to the battery.

4.12.6(C) Electrolyte

The electrolyte is the substance, often a liquid or gel, that is capable of transporting ions between the chemical reactions that happen at the anode and cathode. The electrolyte also inhibits the flow of electrons between the anode and cathode so that the electrons more easily flow through the external circuit rather than through the electrolyte.

Fig4.12.6(C).1: Alkaline batteries

-> Alkaline batteries can leak their electrolyte, potassium hydroxide, if subjected to high heat or reverse voltage (Image courtesy of Wiliam Davies of Wikimedia Commons)

<- The electrolyte is crucial in the operation of a battery. Because electrons cannot pass through it, they are forced to travel through electrical conductors in the form of a circuit that connect the anode to the cathode.

4.12.6(D) Separator

Separators are porous materials that prevent the anode and cathode from touching, which would cause a short circuit in the battery. Separators can be made from a variety of materials, including cotton, nylon, polyester, cardboard, and synthetic polymer films. Separators do not chemically react with either the anode, cathode, or electrolyte.

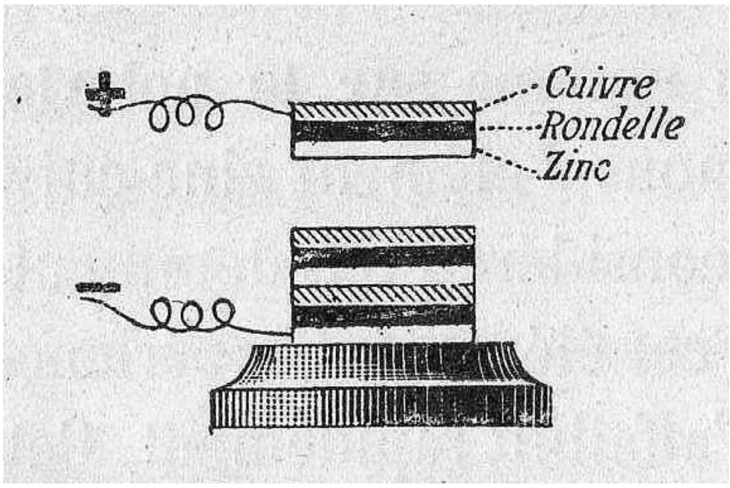


Fig 4.12.6(D).1: separator

The voltaic pile used cloth or cardboard (separator) soaked in brine (electrolyte) to keep the electrodes apart. Ions in the electrolyte can be positively charged, negatively charged, and can come in a variety of sizes. Special separators can be manufactured that allow some ions to pass but not others.

4.12.7 Casing

Most batteries need a way to contain their chemical components. Casings, otherwise known as "housings" or "shells," are simply mechanical structures meant to hold the battery's internals.

Fig4.12.7.1 :This lead-acid battery has a plastic casing

Battery casings can be made of almost anything: plastic, steel, soft polymer laminate pouches, and so on. Some batteries use a conducting steel casing that is electrically connected to one of the electrodes. In the case of the common AA alkaline cell, the steel casing is connected to the cathode.

4.12.8 Operation

Batteries generally require several chemical reactions in order to operate. At least one reaction occurs in or around the anode and one or more reactions occur in or around the cathode. In all cases, the reaction at the anode produces extra electrons in a process called **oxidation**, and the reaction at the cathode uses the extra electrons during a process known as **reduction**.

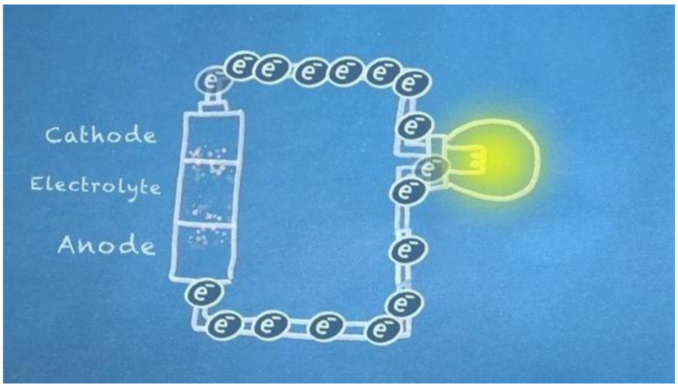


Fig 4.12.8.1: Electron flows

When the switch is closed, the circuit is complete, and electrons can flow from the anode to the cathode. These electrons enable the chemical reations at the anode and cathode.

In essence, we are separating a certain kind of chemical reaction, a reduction-oxidation reaction or redox reaction, into two separate parts. Redox reactions occur when electrons are transferred between chemicals. We can harness the movement of electrons in this reaction to flow outside the battery to power our circuit.

4.13 L293D

L293D is basically a high current dual motor driver/controller Integrated Circuit (IC). It is able to drive load having current up to 1A at the voltage ranging from 4.5V to 36V. Motor driver usually act as current amplifier because they receive a low current signal as an input and provides high current signal at the output.

Motors usually operates on this higher current. L-293D has to builtin H-Bridge driver circuits and is able to control two DC motors at a time in both clockwise and counter clockwise direction. It has two enable pins and they should be kept high in order to control the motor. By changing the polarity of applied signal motor can be rotated in either clockwise or counter clockwise direction. If L 293D enable pin is high, its corresponding driver will provide the desired out. If the enable pin is low, there will be no output. L-293D has different features including internal ESD protection, large voltage supply range, large output current per channel, high noise immunity input etc. L 293D plays a vital role in electronics era and has several different applications e.g relay drivers, DC motor drivers stepping motor drivers etc. The further detail about L 293D motor driver/controller will be given later in this tutorial.

4.13.1 Introduction to L293D

L293D is basically a motor driver or controller. It has two builtin H-bridge circuits which are able to control two DC motors simultaneously in both clockwise and counter clockwise direction. It acts as an current high amplifier because it take low current signal at its input and provides higher current signal at the output in order to drives different load e.g stepper motor & DC motors. Its features include large input voltage supply range, large output current high noise immunity input signals etc. Its common real life applications include stepping motor drivers, relay drivers, DC motor drivers etc. L-293D motor driver/controller is shown in the figure given below.

Fig 4.13.1(A) : L293D Motor Driver

1. L293D Pins

- L-293D has sixteen (16) pins, having different individual functions.
- All of these sixteen pins along with their serial no, are given in the table shown below.

2. L293D Pin Functions

- L 293D each pin has different tasks to perform when they are in working condition.
- The tasks assigned to each pin are given in the table shown below.

L293D Pins	
Pin. No	Pin Name
1	Enable 1,2
2	Input 1
3	Output 1
4	GND
5	GND
6	Output 2
7	Input 2
8	Vcc2
9	Enable 3,4
10	Input 3
11	Output 3
12	GND
13	GND
14	Output put 4
15	Input 4
16	Vcc1

Fig 4.13.1(B) : L293D Pins

3. L293D Pinout

- Pinout diagram of any device presents the pins configuration through a completely labelled diagram.
- L293D pinout diagram is shown in the figure given below.

Fig 4.13.1(C) L293D Pins OUT

4.14 Soil Sensor

Soil moisture sensors measure the volumetric water content in soil. Since the direct gravimetric measurement of free soil moisture requires removing, drying, and weighing of a sample, soil moisture sensors measure the volumetric water content indirectly by using some other property of the soil, such as electrical resistance, dielectric constant, or interaction with neutrons, as a proxy for the moisture content.

The relation between the measured property and soil moisture must be calibrated and may vary depending on environmental factors such as soil type, temperature, or electric conductivity. Reflected microwave radiation is affected by the soil moisture and is used for remote sensing in hydrology and agriculture. Portable probe instruments can be used by farmers or gardeners.

Soil moisture sensors typically refer to sensors that estimate volumetric water content. Another class of sensors measure another property of moisture in soils called water potential; these sensors are usually referred to as soil water potential sensors and include tensiometers and gypsum blocks.

In the waveform analysis a sensor (usually a probe) is placed in the material to be tested. The sensor contains a waveguide consisting of two, three, or more parallel wires which is connected via a coaxial cable to a voltage pulse generator which sends precisely defined voltage pulses into the sensor. As the pulse travels along the wave guide its progress varies depending on the moisture content of the material being examined. When the pulse reaches the end of the wave guide it is reflected. This reflection is visualized in a TDR waveform using an oscilloscope connected to the sensor. The rate of travel of the pulse in the probe is measured and related to moisture content, with slower travel indicating an increase of moisture. By measuring the time from the initial pulse until the reflection is received the average moisture content and relative permitivity of the sample can be calculated by using an equivalent circuit as a reference.

Standard waveform analysis can be used either manually (hand held instruments) or automatically for monitoring moisture content in several areas such as hydrology, agriculture and construction.

Measurement usually involves inserting a sensor into the substance to be tested and then applying either Standard Waveform Analysis to determine the average moisture content along the sensor or Profile Analysis to provide moisture content at discrete points along the sensor. A spatial location can be achieved by appropriate installation of several sensors.

Moisture may be present as adsorbed moisture at internal surfaces and as capillary condensed water in small pores. At low relative humidities, moisture consists mainly of adsorbed water. At higher relative humidities, liquid water becomes more and more important, depending or not depending on the pore size can also be an influence of volume

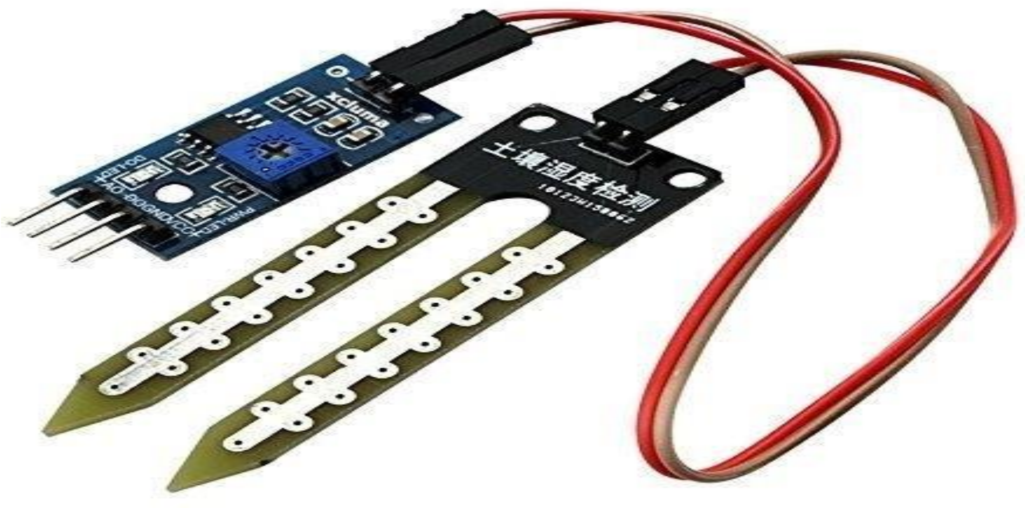


Fig 4.14(A) : Soil Moisture Sensor

4.14.1 Soil Sensor Pin Configuration

Pin Name	Description
VCC	The VCC **pin** powers the module, typically with +5V
GND	Power Supply Ground
DO	Digital Out **Pin** for Digital Output.
AO	Analog Out **Pin** for Analog Output

4.14.2 Moisture Sensor Module Features & Specifications

- Operating Voltage: 3.3V to 5V DC.
- Operating Current: 15mA.
- Output Digital - 0V to 5V, Adjustable trigger level from preset.
- Output Analog - 0V to 5V based on infrared radiation from fire flame falling on the **sensor**.
- LEDs indicating output and power.
- PCB Size: 3.2cm x 1.4cm.

4.15 POWER SUPPLY

4.15.1 INTRODUCTION:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as "Regulated D.C Power Supply".

4.15.2 BLOCK DIAGRAM OF POWER SUPPLY:

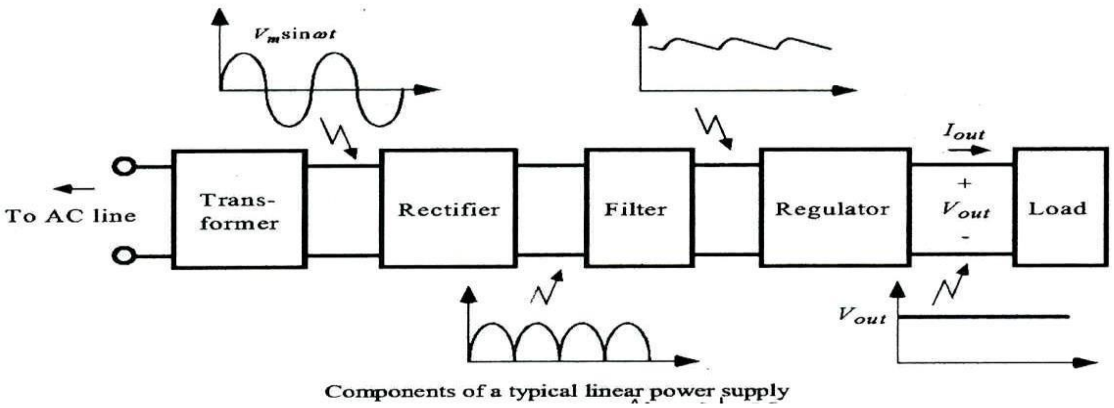


Fig.4.15.2.1: Block Diagram of Power Supply

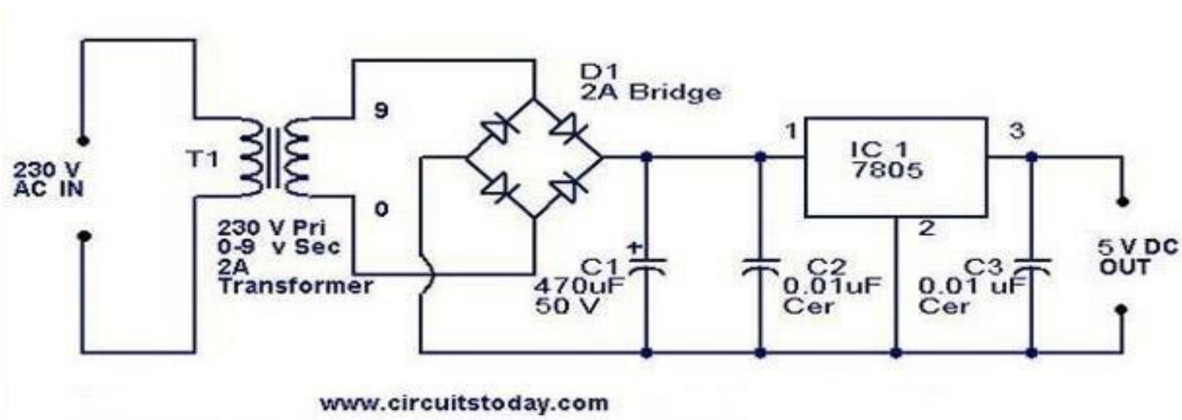


Fig.4.15.2.2: Schematic Diagram of Power Supply

4.15.3 DESCRIPTION OF POWER SUPPLY:

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a a different form of energy – such as solar, mechanical, or chemical - into electrical energy.

A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered. For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

4.15.4 TRANSFORMER:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current inthe first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

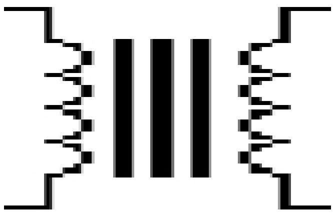


Fig.4.15.4.1: Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

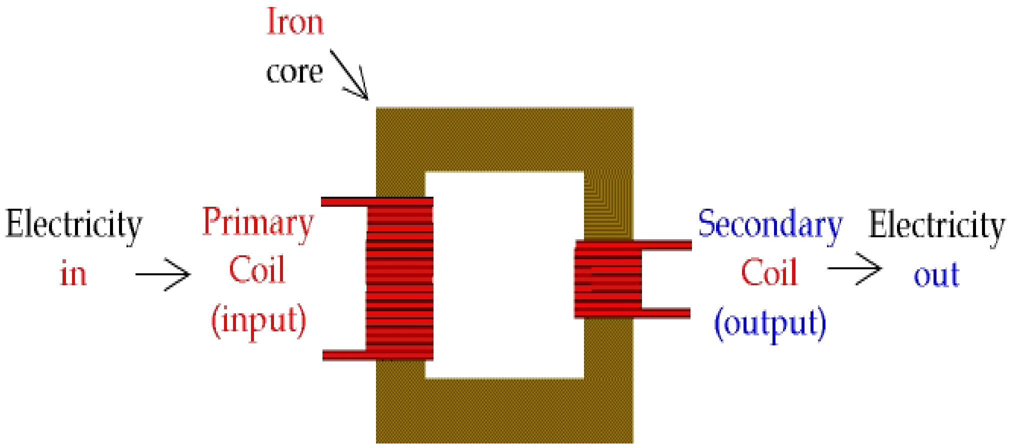
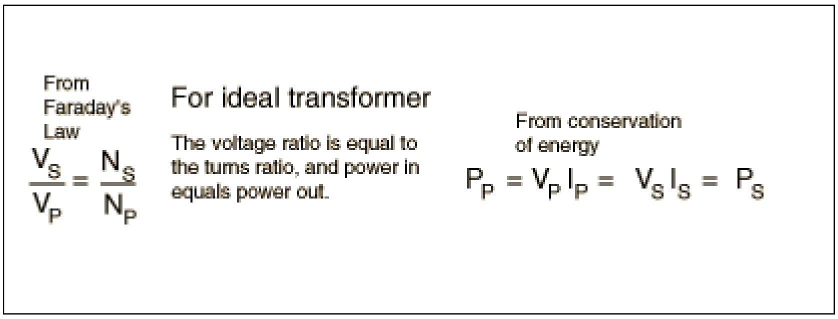


Fig.4.15.4.2: Transformer

4.15.4(A): Basic Principle of Transformer:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.



4.15.4(B) Working of Transformer:

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

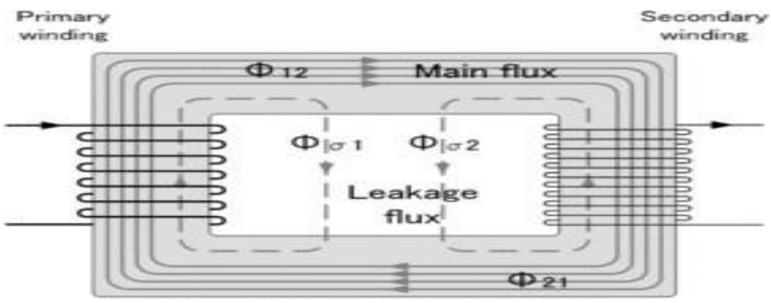


Fig.4.15.4.1(B): Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f. The iron core is laminated to prevent 'eddy currents' flowing in the core.

These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

4.15.4.1 Classification of Transformer:

- Step-Up Transformer
- Step-Down Transformer

4.15.4.1(A) Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn's ratio of the two sets of windings determines the amount of voltage transformation.

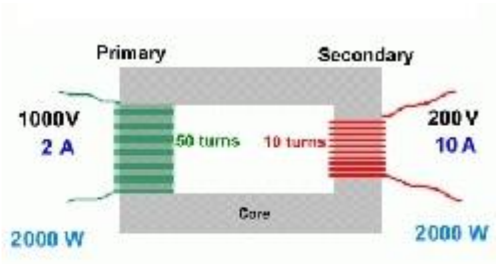


Fig.4.15.4.1(A).1: Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn’s ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn’s ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single phase steps down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase steps up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

4.15.4.1(B) Step-Up Transformer:

A step up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply. A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

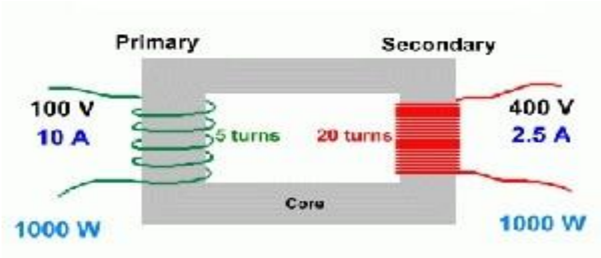


Fig.4.15.4.1(B).1: Step-Up Transformer

4.16 VOLTAGE REGULATOR:

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx'indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx'indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consist the three pins there are:

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

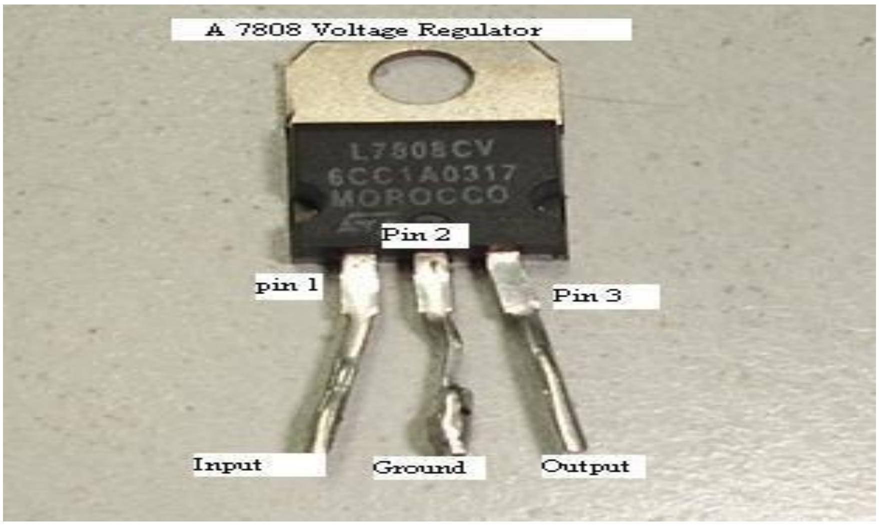


Fig.4.16.1: Regulator

4.17 Bluetooth :

Generally, the HC-05 Bluetooth Module, or the HC-05 Sub Module, to be precise, comes with the BC417 IC along with a flash memory. Such Modules come as surface mount board and several third-party manufacturers use these board to build a more complete system with necessary pins and components.

Bluetooth is a short-range wireless technology standard that is used for exchanging data between fixed and mobile devices over short distances using UHF radio waves in the ISM bands, from 2.402 GHz to 2.48 GHz, and building personal area networks.

Bluetooth is managed by the Bluetooth Special Interest Group (SIG), which has more than 35,000 member companies in the areas of telecommunication, computing, networking, and consumer electronics. The IEEE standardized Bluetooth as IEEE 802.15.1, but no longer maintains the standard. The Bluetooth SIG oversees development of the specification, manages the qualification program, and protects the trademarks. A manufacturer must meet Bluetooth SIG standards to market it as a Bluetooth device.

The following image shows one such HC-05 Bluetooth Sub Module (the green board mounted on the blue board) being used as a part of a complete Bluetooth Module.

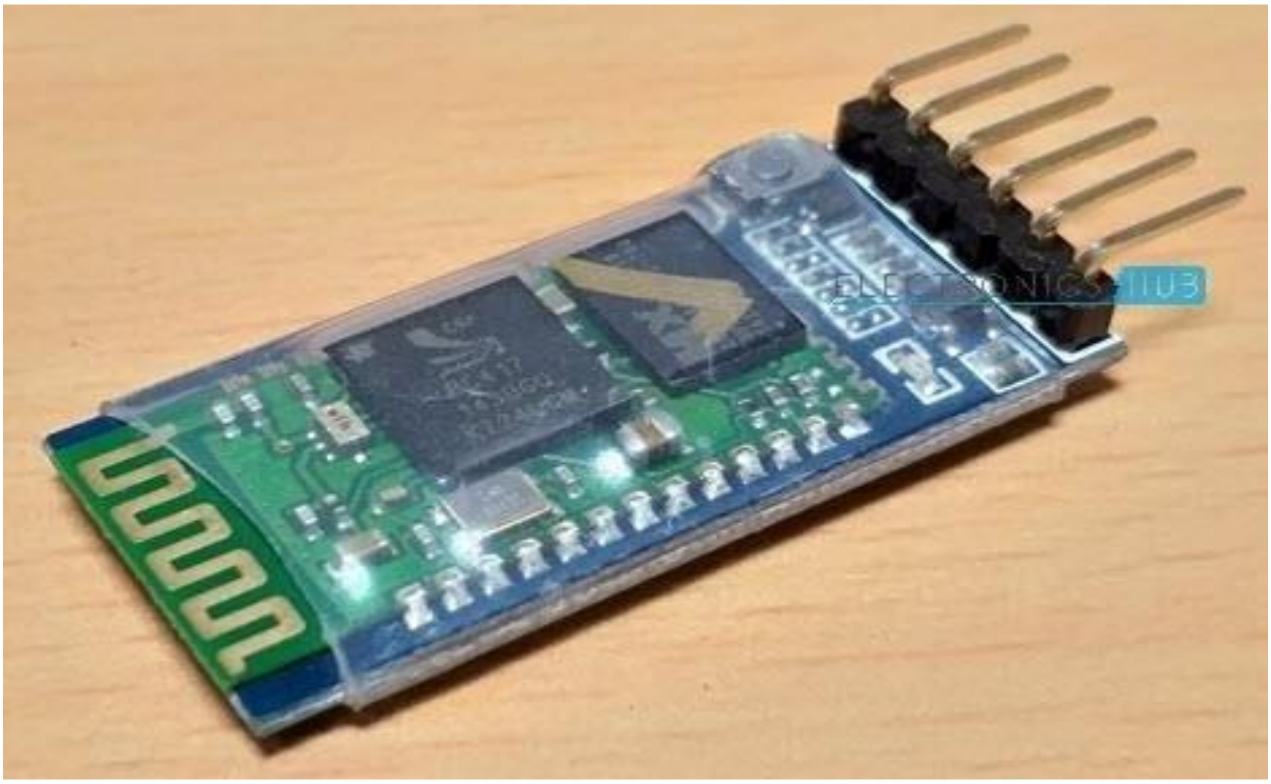


Fig 4.17(A): Bluetooth

4.17.1 Pins of HC-05 Bluetooth Module

The HC-05 Module supports for UART, USB as well as SPI communication and depending on the application, necessary pins can be used. In my case, the board uses the UART communication.

Coming to the pins of the Bluetooth Module, generally, four pins are sufficient for successfully enabling a wireless communication link but the modules produced now-a-days come with six pins namely: VCC, GND, TX, RX, EN and STATE.

Image below shows the pins and other components on a typical HC-05 Bluetooth Module. An important point to remember is the HC-05 Bluetooth Module works on a logic level of 3.3V. Hence, a 3.3V Regulator is used on the board.

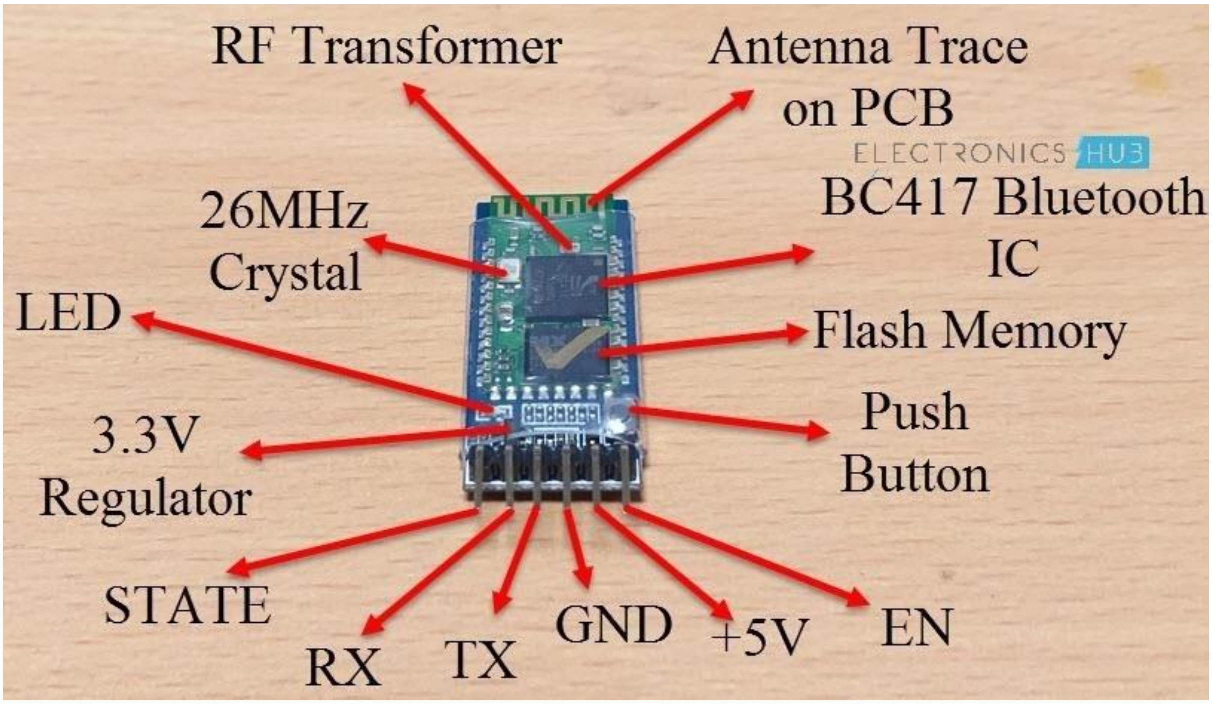


Fig 4.17.1(A) Bluetooth pins.

NOTE: The button present on the board is used to configure the Bluetooth Module in AT Command Mode. This part of the Module is not discussed in this project.

4.17.2 Pin Description

- **EN:** It is the enable pin. When this pin is floating or connected to 3.3V, the module is enabled. If this pin is connected to GND, the module is disabled.
- **+5V:** This is the supply pin for connecting +5V. As the Module has on-board 3.3V regulator, you can provide +5V supply.
- **GND:** It is the ground pin.
- **TX:** It is the Transmitter pin of the UART Communication.
- **RX:** It is the Receive Pin of UART.
- **STATE:** This is a status indicator pin. This pin goes LOW when the module is not connected to any device. When the module is paired with any device, this pin goes HIGH.

NOTE: The on-board LED is used to indicate the status of the connection. When the module is not paired, the LED blinks or flashes repeatedly. Once the module is paired, the LED blinks at a constant delay of 2 seconds.

4.17.3 Modes of Operation

The HC-05 Bluetooth Module can be configured in two modes of operation: Command Mode and Data Mode.

In Command Mode, you can communicate with the Bluetooth module through AT Commands for configuring various settings and parameters of the Module like get the firmware information, change UART Baud Rate, change module name, set it as either Master or slave etc.

An important point about HC-05 Module is that it can be configured as Master or Slave in a communication pair. In order to select either of the modes, you need to activate the Command Mode and sent appropriate AT Commands.

Coming to the Data Mode, in this mode, the module is used for communicating with other Bluetooth device i.e. data transfer happens in this mode.

4.17.4 Default Settings of HC-05 Bluetooth Module

The following is a list of few of the default settings of the HC-05 Bluetooth Module.

- Name: HC-05
- Password: 1234 (or 0000)
- Type: Slave
- Mode: Data
- Baud Rate: 9600 with 8 data bits, no parity and 1 stop bit

4.18 MOTORS

Industrial motors come in a variety of basic types. These variations are suitable for many different applications. Naturally, some types of motors are more suited for certain applications than other motor types are. This document will hopefully give some guidance in selecting these motors.

4.18.1 AC Motors

The most common and simple industrial motor is the three phase AC induction motor, sometimes known as the "squirrel cage" motor. Substantial information can be found about any motor by checking its (nameplate).

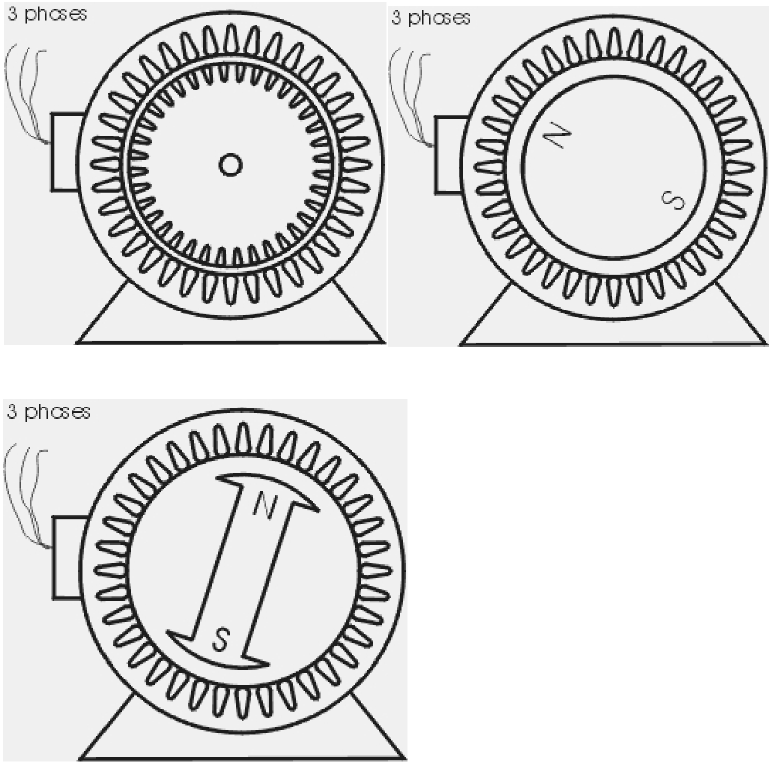


Fig 4.18.1(A) : three figures of AC motor 3 phases

Advantages

- Simple Design
- Low Cost
- Reliable Operation
- Easily Found Replacements
- Variety of Mounting Styles
- Many Different Environmental Enclosures

Simple Design

The simple design of the AC motor -- simply a series of three windings in the exterior (stator) section with a simple rotating section (rotor). The changing field caused by the 50 or 60 Hertz AC line voltage causes the rotor to rotate around the axis of the motor.

The speed of the AC motor depends only on three variables:

1. The fixed number of winding sets (known as poles) built into the motor, which determines the motor's base speed.

2. The frequency of the AC line voltage. Variable speed drives change this frequency to change the speed of the motor.
3. The amount of torque loading on the motor, which causes slip.

Low Cost

The AC motor has the advantage of being the lowest cost motor for applications requiring more than about 1/2 hp (325 watts) of power. This is due to the simple design of the motor. For this reason, AC motors are overwhelmingly preferred for fixed speed applications in industrial applications and for commercial and domestic applications where AC line power can be easily attached. Over 90% of all motors are AC induction motors. They are found in air conditioners, washers, dryers, industrial machinery, fans, blowers, vacuum cleaners, and many, many other applications.

Reliable Operation

The simple design of the AC motor results in extremely reliable, low maintenance operation. Unlike the DC motor, there are no brushes to replace. If run in the appropriate environment for its enclosure, the AC motor can expect to need new bearings after several years of operation. If the application is well designed, an AC motor may not need new bearings for more than a decade.

Easily Found Replacements

The wide use of the AC motor has resulted in easily found replacements. Many manufacturers adhere to either European (metric) or American (NEMA) standards. (For Replacement Motors)

Variety of Mounting Styles

AC Motors are available in many different mounting styles such as:

- Foot Mount
- C-Face
- Large Flange
- Vertical

- Specialty

4.18.2 DC Motors

The brushed DC motor is one of the earliest motor designs. Today, it is the motor of choice in the majority of variable speed and torque control applications.

Advantages

- Easy to understand design
- Easy to control speed
- Easy to control torque
- Simple, cheap drive design

Easy to understand design

The design of the brushed DC motor is quite simple. A permanent magnetic field is created in the stator by either of two means:

- Permanent magnets
- Electro-magnetic windings

If the field is created by permanent magnets, the motor is said to be a "permanent magnet DC motor" (PMDC). If created by electromagnetic windings, the motor is often said to be a "shunt wound DC motor" (SWDC). Today, because of cost-effectiveness and reliability, the PMDC motor is the motor of choice for applications involving fractional horsepower DC motors, as well as most applications up to about three horsepower.

At five horsepower and greater, various forms of the shunt wound DC motor are most commonly used. This is because the electromagnetic windings are more cost effective than permanent magnets in this power range.

Caution: If a DC motor suffers a loss of field (if for example, the field power connections are broken), the DC motor will immediately begin to accelerate to the top speed which the loading will allow. This can result in the motor flying apart if the motor is lightly loaded. The possible loss of field must be accounted for, particularly with shunt wound DC motors.

Opposing the stator field is the armature field, which is generated by a changing electromagnetic flux coming from windings located on the rotor. The magnetic poles of the armature field will attempt to line up with the opposite magnetic poles generated by the stator field. If we stopped the design at this point, the motor would spin until the poles were opposite one another, settle into place, and then stop -- which would make a pretty useless motor!

However, we are smarter than that. The section of the rotor where the electricity enters the rotor windings is called the commutator. The electricity is carried between the rotor and the stator by conductive graphite-copper brushes (mounted on the rotor) which contact rings on stator. Imagine power is supplied:

The motor rotates toward the pole alignment point. Just as the motor would get to this point, the brushes jump across a gap in the stator rings. Momentum carries the motor forward over this gap. When the brushes get to the other side of the gap, they contact the stator rings again and -- the polarity of the voltage is reversed in this set of rings! The motor begins accelerating again, this time trying to get to the opposite set of poles. (The momentum has carried the motor past the original pole alignment point.) This continues as the motor rotates.

In most DC motors, several sets of windings or permanent magnets are present to smooth out the motion.

Easy to control speed

Controlling the speed of a brushed DC motor is simple. The higher the armature voltage, the faster the rotation. This relationship is linear to the motor's maximum speed.

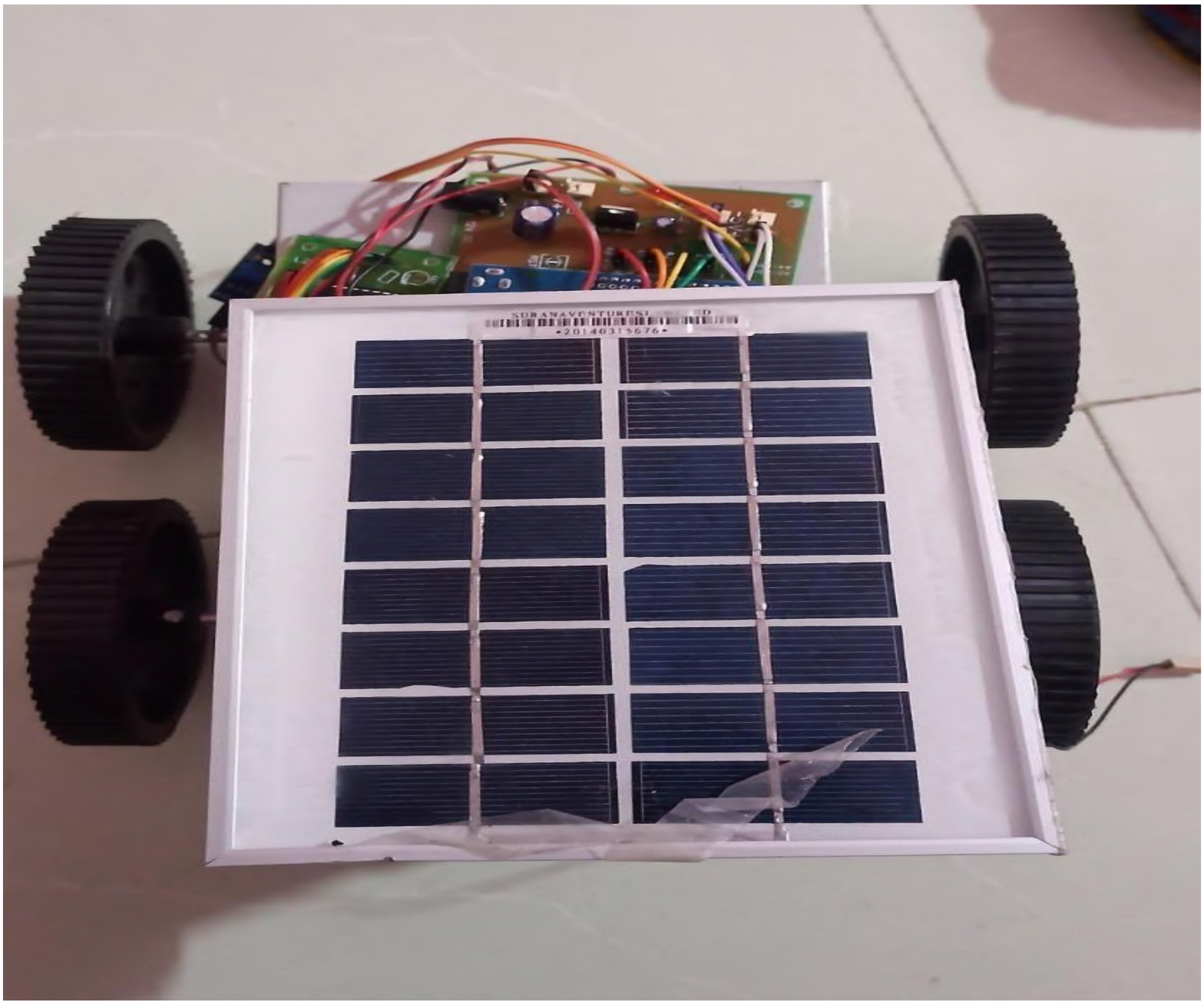
The maximum armature voltage which corresponds to a motor's rated speed (these motors are usually given a rated speed and a maximum speed, such as 1750/2000 rpm) are available in certain standard voltages, which roughly increase in conjunction with horsepower. Thus, the smallest industrial motors are rated 90 VDC and 180 VDC. Larger units are rated at 250 VDC and sometimes higher.

Specialty motors for use in mobile applications are rated 12, 24, or 48 VDC. Other tiny motors may be rated 5 VDC.

Most industrial DC motors will operate reliably over a speed range of about 20:1 -- down to about 5-7% of base speed. This is much better performance than the comparible AC motor. This is partly due to the simplicity of control, but is also partly due to the fact that most industrial DC motors are designed with variable speed operation in mind, and have added heat dissipation features which allow lower operating speeds.

CHAPTER-5

RESULTS



5.1 Results

Fig 5.1.1 : Robot outer view

The outer part of the agriculture robot is covered with a solar pannel. Here solar pannel is generating power supply required to work all the components in the project

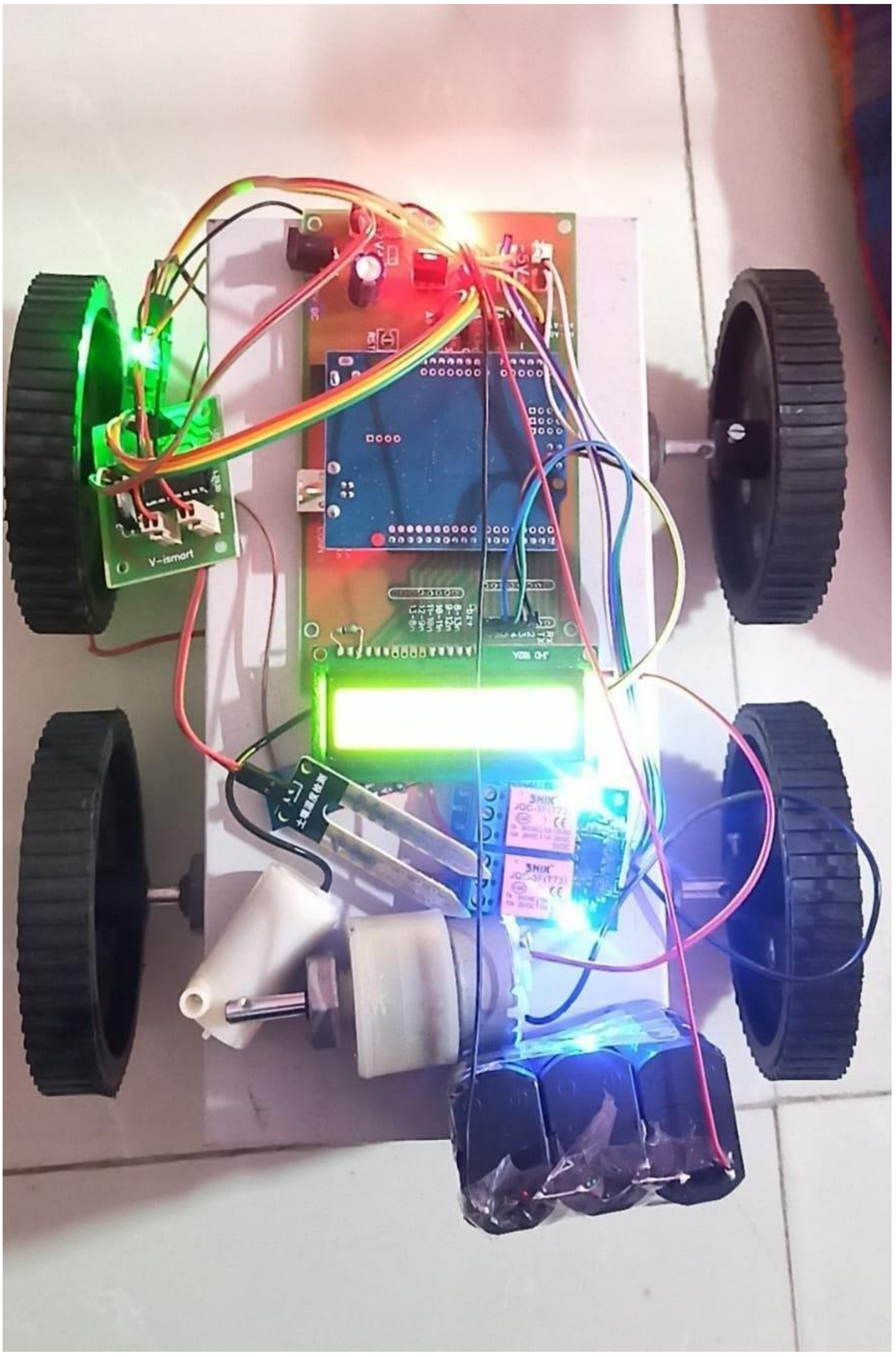


Fig 5.1.2: Internal structure of robot

Fig 5.1.3 : Soil sensor operation

The designed robot gets power from solar panel which converts sunlight into electricity. This electrical energy is given to the charging circuit in order to charge the battery to 12V. This battery gives power to controller, motor driver and other mechanisms. The entire prototype of solar powered multifunction agricultural robot is shown in fig: 5.1.3.

Firstly it paired app with the HC-05 module. The movement of robot controlled by following commands #: Stop, *: Forward movement, @: Backward movement, %: Right turn, &: Left turn, c: Grass cutting

When we send ‘c’ commands the robot starts Grass Cutting . Grass cutting mechanism has a circular cutter provided with sharp edges. Weeds were successfully cut by this cutter. ‘#’ is sent to stop the function of the robot. If the Soil Sensor receive the command dry detector then the Submersible motor turns on and sprinkle water on field.

CHAPTER-6

ADVANTAGES

6.1 ADVANTAGES OF THE PROJECT:

- No extra cost is required for fueling.
- Reduces the labour cost.
- The use of machineries in this field save time.
- Increase efficiency and indirectly increase the production in farms.
- No external power consumption is required.

6.2 Draw backs of the project :

- It can be controlled up to short range itself.

CHAPTER – 7

CONCLUSION & FUTURE SCOPE

7.1 Conclusion:

Multipurpose farming robot has effectively actualized and tried for operations like grass cutting and water sprinkling. An underlying result of this examination shows that the greater part of these frameworks that work with self-governing, are more adaptable than customary frameworks. It was developed for integrating agricultural robot using programming. The advantages of Multi-tasking agricultural robots are reducing human effort, ensuring proper irrigation and efficient utilization of all resources. The upsides of multipurpose horticultural robots are lessening human intercession, guaranteeing appropriate water system and proficient use of assets. the different parameters like soil condition, region secured by the robot and leveling.

7.2 Future scope:

Robots will replace the human labor and, in every aspect, which will make agricultural sector much more efficient than today’s time. Artificial intelligence technologies, especially in machine learning, are expected to play a serious role in most of the above technology areas, and can be essential enablers for agricultural robots. Agricultural environments are subject to changes throughout the lifetime of a robotic system. For example, there could also be new crop varieties, weeds, pests, diseases, treatments, legislation, temperature change, etc., as well as new robotic technologies. In AI terms this suggests managing an open world, so techniques to enable adaptation during operation rather than at the planning phase are going to be crucial. Techniques that allow robots to find out from experience include reinforcement learning, learning from demonstration, and transfer learning to exploit prior knowledge, e.g. from another domain or task. Ongoing research is investigating deep learning methods, especially in perception-related tasks involving the interpretation of sensor data, including recognition and segmentation tasks in automated weeding and fruit picking. Robots also will have to leverage human knowledge, especially when facing situations that weren't foreseen at design time.

REFERENCES

[1] S. Umarkar and A. Karwankar, “Automated Seed Sowing Agribot using Arduino", IEEE Conference on Communication and Signal Processing, April 2016, pp.1379-1383.

[2] M.D.I. Sujon, R. Nasir, M.M.I. Habib, M.I. Nomaan J. Baidya and M.R. Islam “Agribot: Arduino Controlled Autonomous Multipurpose Farm Machinery Robot for Small to medium scale cultivation", IEEE conference on intelligent autonomous systems,March 2018, pp.155- 159.

[3] P.V.S. Jayakrisna, M.S. Reddy, N.J. Sai, N. Susheel and K.P. Peeyush, “Autonomous Seed Sowing Agricultural Robot”, in IEEE Conference on advances in computing, communications and informatics (ICACCI),2018, pp.2332-2336.

[4] S. Kareemulla, E. Prajwal, B. Sujeshkumar, B. Mahesh, and V Reddy, “GPS based Autonomous Agriculture Robot”, IEEE International conference on design innovations for 3Cs compute communicate control, 2018, pp.100-105.

[5] Ranjitha B., Nikhitha M. N. and Aruna K, “Solar Powered Autonomous Multipurpose Agricultural Robot Using Bluetooth/Android App”, IEEE Conference on Electronics Communication and Aerospace Technology [ICECA], June 2019, pp.872-877.

[6] C.M. Barber, R.J. Shucksmith, B.M. Donald and B.C Wunsche, “Sketch-based robot programming,” in IEEE International conference of image and vision computing newzealand, 2010, pp. 1-8.

[7] N.S. Naik, V.V.Shete and S.R.Danve, “Precision agricultural robot for seed function”, in IEEE International conference on inventive computational technologies (ICICT), 2016, pp.15-19.

[8] P.V. Santhi, N. Kapileshwae, V. K. R. Chenchela and C. H. V. S. Prasad, “Sensor and vision based autonomous agribot for sowing seeds”, in IEEE International conference on energy communication, data analysis and soft computing (ICECDS), 2017, pp. 242-245.

A

PROJECT REPORT

On

DEEP LEARNING FOR FACE RECOGNITION UNDER COMPLEX ILLUMINATION CONDITIONS BASED ON LOG-GABOR AND LBP

Submitted by

P. Nihal Reddy (17K81A04G6)

P. Hrithik (17K81A04G7)

S. Pramodini (17K81A04H0)

in partial fulfilment of the requirement forthe award of degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS & COMMUNICATION ENGINEERING

Under the Guidance of

Mr. Basava Dhanne

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

St. MARTIN'S ENGINEERING COLLEGE

(AUTONOMOUS INSTITUTE)

Dhulapally, Secunderabad-500 100.

ST.MARTIN'S ENGINEERING COLLEGE

(An UGC Autonomous Institute)

Dhulapally, Secunderabad-500 100

NBA & NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the project report entitled **"DEEP LEARNING FOR FACE RECOGNITION UNDER COMPLEX ILLUMINATION CONDITIONS BASED ON LOG-GABOR AND LBP",** is being submitted by **P. Nihal Reddy (17K81A04G6) P. Hrithik (17K81A04G7) S. Pramodini (17K81A04H0)** in partial fulfilment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS & COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in the report have been verified and found satisfactory.

INTERNAL GUIDE
Basava Dhanne
Assistant Professor
Department of ECE.

HEAD OF DEPARTMENT
Dr.B.Hari Krishna,
Professor.

INTERNAL EXAMINER

EXTERNAL EXAMINER

TUESDAY, 15 JUNE 2021

<u>INTERNSHIP CERTIFICATE</u>

THIS IS TO CERTIFY THAT **P.HRITHIK** WITH ROLL NO.**17K81A04G7, P.NIHAL REDDY** WITH ROLL NO.**17K81A04G6**, **PRAMODINI SABINKAR** WITH ROLL NO.**17K81A04H0,** OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**DEEP LEARNING FOR FACE RECOGNITION UNDER COMPLEX ILLUMINATION CONDITION BASED ON LOG-GABOR AND LBP**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

Place:

Date:

We the students of Bachelor of Technology in department of Electronics and Communication Engineering, session: 2017-2021, St. Martin's Engineering College, Dhulapally, Kompally, Project work entitled Deep Learning For Face Recognition Under Complex Illumination Conditions Based On Log-Gabor And LBP is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

P. Nihal Reddy
17K81A04G6

P. Hrithik
17K81A04G7

S. Pramodini
17K81A04H0

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to our **College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr. Santosh Kumar Patra, our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna, Head and Professor, the department of Electronics and Communication** for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our **department project coordinator Dr.A.Anand** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

Finally, we are highly grateful and obliged to our **Internal Project Guide Mr. Basava Dhanne** , Assistant Professor in ECE for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

P. NIHAL REDDY.

P. HRITHIK.

S. PRAMODINI.

ABSTRACT

Complex illumination condition is one of the most critical challenging problems for practical face recognition. In this paper, we propose a novel method based on deep learning to solve the adverse impact imposed by illumination variation in the face recognition process. Firstly, illumination preprocessing is applied to improve the adverse effects of intense illumination changes on face images. Secondly, the Log-Gabor filter is used to obtain the Log-Gabor feature images of different scales and directions, then, LBP (Local Binary Pattern) features of images subblock is extracted. Lastly, texture feature histograms are formed and input into the deep belief network (DBN) visual layer, then classification and recognition are completed through deep learning in DBN. Experimental results show that superior performance can be obtained in the developed approach by comparisons with some state-of-the-arts.

CONTENTS

TABLE OF FIGURES

1 INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

Complex illumination condition is one of the most critical challenging problems for practical face recognition. This project presents an effective method to extract the robust deep features of face images under severe illumination conditions. We propose a novel method based on deep learning to solve the adverse impact imposed by illumination variation in the face recognition process. Firstly, illumination preprocessing (analyzing the pixel values, size of images, background, foreground, intensities, contrast and so on of images) is applied to improve the adverse effects of intense illumination changes on face images. Secondly, the Log-Gabor filter (a low pass filter) is used to obtain the Log-Gabor feature images of different scales and directions, then, LBP (Local Binary Pattern) features of image subblocks is extracted. Deep learning simulates the deep organizational structure of brain groups, and it forms more abstract and effective high-level representation by combining low-level features. Deep belief network (DBN) is a typical deep learning method. DBN automatically learns the abstract features of different levels from bottom to top, and finally obtains the nonlinear description of features. An automatic feature extraction process without artificial choice is presented. DBN has been successfully applied to handwritten numeral recognition, dynamic human detection, and many other fields.

However, DBN may ignore the local structure of the image and it is difficult to learn the local features of the face image . At the same time, the network will learn the unfavorable feature expression for the influence of illumination and other factors when the pixel-level face features are used as the input of DBN. It uses LBP feature as the input of the deep learning network. It improves the performance of LBP and deep learning algorithm respectively. The texture feature histograms are formed and input into the deep belief network (DBN) visual layer, then face classification and recognition are completed through deep learning in DBN. Experimental results show that superior performance can be obtained in the developed approach by comparisons with some state-of-the-arts.

1.2 STATEMENT OF THE PROBLEM

Face Recognition has gained much attention due to its wide range of applications, such as biometric, identity authentication, law enforcement, and video surveillance. Even though human eye can detect and identify faces in a scene easily, building an automatic system that accomplishes such a task is still very challenging. Due to the dynamic nature of face images, a face recognition system encounters various problems during the recognition process. It is possible to classify a face recognition system as either "robust" or "weak" based on its recognition performances.

Robust face recognition requires the ability to recognize identity despite many variations in appearance that the face can have in a scene. The face is a object which is illuminated from a variety of light sources and surrounded by arbitrary background data (including other faces). Therefore, the appearance a face has when projected onto a image can vary tremendously. The challenges are mainly from large variations of illumination, expression, pose, corruption, and occlusion which can cause sharp decline in recognition rate.

1.3 AIMS AND OBJECTIVES OF THE STUDY

Face recognition presents a challenging problem in the field of image analysis and computer vision, and as such has received a great deal of attention over the last few years because of its many applications in various domains. Face recognition techniques can be broadly divided into categories based on the face data acquisition methodology: methods that operate on intensity images; In the language of information theory, the objective is to extract the relevant information in a face image, encode it as efficiently as possible, and compare one face encoding with a database of models encoded in the same way.

In mathematical terms, the objective is to find the principal components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of face images or eigen vectors. The set of eigen vectors is termed as eigen faces. The aim of the study is to recognize face images under worst/complex light illumination conditions by extracting robust deep features of the face images for robust Face Recognition.

1.4 SCOPE OF THE STUDY

As an important branch of biometric identification technology, face recognition technology has the characteristics of convenient acquisition and high reliability that has widely used in the fields of information security, national security and traffic monitoring. Face recognition appears to offer several advantages over other biometric methods, a few of which are outlined here. Almost all these technologies require some voluntary action by the user,for examples, the user needs to place his hand on a hand-rest for fingerprinting or hand geometrydetection and has to stand in a fixed position in front of a camera for iris or retina identification. However, face recognition can be done passively without any explicit action or participation on the part of the user since face images can be acquired from a distance by a camera. This is particularly beneficial for security and surveillance purposes.

In security system, many types of passwords are used to access the private and confidential data. Such password can be as insert characters (key in pin) and touch smart card using RFID technology. Passwords and PINs are hard to remember and can be stolen or guessed; cards, tokens, keys and the like can be misplaced, forgotten, purloined or duplicated; magnetic cards can become corrupted and unreadable. By developed face recognition it more secure because facial image had been used as the ID. It also helps to avoid any duplicated identification. Other problem is to identify certain criminals especially in identification technique used by the police. Face recognition helps to recognize the facial image in more efficient and accurate in order to match with the identity stored in the database.

1.5 SYSTEM REQUIREMENTS

MATLAB - A high-performance language for technical computing.

Typical uses of MATLAB include :

-Math and computation.

-Algorithm development.

-Modeling, simulation, and prototyping.

DIGITAL IMAGE PROCESSING - an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An important characteristic underlying the design of image processing systems is the significant level of testing and experimentation that normally is required before arriving at an acceptable solution.

1.6 ORGANIZATION OF CHAPTERS

The paper is organized as follows, Chapter 1 presents Introduction of the project including the overview of the project, the statement of the problem, objective of the project, the scope of study, and the system requirements of the project.

Chapter 2 presents the Literature Survey ,the literature review on research area of related existing methods, review on research of proposed project, and conclusions on reviews.

Chapter 3 presents Research Methodology which includes the system requirements of the project, analyzing the problem of existing methods, defining a solution of the problem through the proposed system, , the architectural design related block diagram of project defining the independent units and explaining their functionalities

Chapter 4 presents the Project Implementation, the implementation stages of project ,the activities involved in implementation, the testing of the project and results of the project.

Chapter 5 presents the Project Testing taking into account two standard databases as training and testing datasets and comparing the efficiency of the recognition rate with various other face recognition methods.

Chapter 6 presents the Conclusion and Future Enhancement of the project.

2 LITERATURE SURVEY

As one of the most visible and challenging problems in computer vision and pattern recognition, face recognition (FR) has been extensively studied in the past two decades and many representative methods, such as Eigenface, Fisher face and SVM, have been proposed. Moreover, to deal with the challenges in practical FR system, active shape model and active appearance model were developed for face alignment; LBP and its variants were used to deal with illumination changes; and Eigenimages and probabilistic local approach were proposed for FR with occlusion.

Although much progress have been made, robust FR to occlusion/corruption is still a challenging issue because of the variations of occlusion, such as disguise, continuous or pixel-wise occlusion, randomness of occlusion position and the intensity of occluded pixels. The recognition of a query face image is usually accomplished by classifying the features extracted from this image. The most popular classifier for FR may be the nearest neighbor (NN) classifier due to its simplicity and efficiency. In order to overcome NN's limitation that only one training sample is used to represent the query face image, Li and Lu proposed the nearest feature line (NFL) classifier, which uses two training samples for each class to represent the query face. Chien and Wu then proposed the nearest feature plane (NSP) classifier, which uses three samples to represent the test image. Lateron, classifiers using more training samples for face representation were proposed, such as the local subspace classifier (LSC) and the nearest subspace (NS) classifiers, which represent the query sample by all the training samples of each class. Though NFL, NSP, LSC and NS achieve better performance than NN, all these methods with holistic face features are not robust to face occlusion.

Generally speaking, these nearest classifiers, including NN, NFL, NFP, LSC and NS, aim to find a suitable representation of the query face image, and classify it by checking which class can give a better representation than other classes. Nonetheless, how to formulate the representation model for classification tasks such as FR is still a challenging problem.

Wright et al. proposed a novel face recognition method called sparse representation classification (SRC). In this method, a test face sample can be represented by the linear combination of part of training face samples from all subjects with sparsity regularization on the representation vector Wright *et al.* applied sparse coding to FR and proposed the sparse coding classification (SRC) scheme. The SRC has close relationship to the nearest classifiers. Like NN, NFL, NFP, LSC and NS classifiers, SRC also represents the query sample as the linear combination of training samples however, it forces the representation coefficients being sparse (instead of presetting the number of non-zero representation be from samples of different classes). SRC could be seen as a more general model than the previous nearest classifiers, and it uses the samples from all classes to collaboratively represent the query sample to overcome the small-sample-size problem in FR. In addition, different from the methods such as which use local region features, color features or gradient information to handle some special occlusion SRC shows interesting results in dealing with occlusion by assuming a sparse coding.

Recently, Hinton and Salakhutdinov proposed a deep learning (DL) method to train a deep neural network, which is also successfully applied for the face recognition task. Since the trained deep network can achieve a better representation of the face data, the DL based face recognition methods obtain very good performances. We propose a regularized robust coding (RRC) model in our project. A special case of RRC, namely robust sparse coding (RSC), has been presented in our previous work by assuming that the coding coefficients are sparse. Our extensive experiments in benchmark face databases show that RRC achieves much better performance than existing sparse representation based FR methods, especially when there are complicated variations, such as face occlusions, corruptions etc. problem involves face detection, feature extraction from the face regions and recognition.

The key procedure in PCA is based on Karhumen-Loeve transformation. If the image elements are considered to be random variables, the image may be seen as a sample of a stochastic process. The focus of the research is to find the accuracy of eigenfaces method in face recognition. Eigenface approach seemed to be an adequate method to be used in face recognition due to its simplicity, speed and learning capability.

3 RESEARCH METHODOLOGY

3.1 SYSTEM REQUIREMENTS

3.1.1 MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems along withsolutions are expressed in familiar mathematical notation. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for *matrix laboratory*. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high- productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called *toolboxes*. Very important to most users of MATLAB, toolboxes allow you to *learn* and *apply* technology.

Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

Typical uses of MATLAB include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

The MATLAB system consists of five main parts

Development Environment

This is the set of tools and facilities that help you use MATLAB functions and files. Manyof these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

The MATLAB Mathematical Function

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigen values, Bessel functions, and fast Fourier transforms.

The MATLAB Language

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well asannotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

The MATLAB Application Program Interface (API**)**

This is a library that allows you to write C and Fortran programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

3.1.2 MATLAB WORKING ENVIRONMENT

MATLAB DESKTOP

Matlab Desktop is the main Matlab application window. The desktop contains five sub windows, the command window, the workspace browser, the current directory window, the command history window, and one or more figure windows, which are shown only when the user displays a graphic.

The command window is where the user types MATLAB commands and expressions atthe prompt (>>) and where the output of those commands is displayed. MATLAB defines the workspace as the set of variables that the user creates in a work session. The workspace browser shows these variables and some information about them. Double clicking on a variable in the workspace browser launches the Array Editor, which can be used to obtain information and income instances edit certain properties of the variable.

The current Directory tab above the workspace tab shows the contents of the current directory, whose path is shown in the current directory window. For example, in the windows operating system the path might be as follows: C:\MATLAB\Work, indicating that directory “work” is a subdirectory of the main directory “MATLAB”; WHICH IS INSTALLED IN DRIVE C. clicking on the arrow in the current directory window shows a list of recently used paths. Clicking on the button to the right of the window allows the user to change the current directory.

MATLAB uses a search path to find M-files and other MATLAB related files, which are organize in directories in the computer file system. Any file run in MATLAB must reside in the current directory or in a directory that is on search path. By default, the files supplied with MATLAB and math works toolboxes are included in the search path. The easiest way to see which directories are on the search path. The easiest way to see which directories are soon the search path, or to add or modify a search path, is to select set path from the File menu the desktop, and then use the set path dialog box. It is good practice to add any commonly used directories to the search path to avoid repeatedly having the change the current directory.

The Command History Window contains a record of the commands a user has entered in the command window, including both current and previous MATLAB sessions. Previously entered MATLAB commands can be selected and re-executed from the command history window by right clicking on a command or sequence of commands. This action launches a menu from which to select various options in additionto executing the commands. This is useful to select various options in addition to executing the commands. This is a useful feature when experimenting with various commands in a work session.

Using the MATLAB Editor to create M-Files

The MATLAB editor is both a text editor specialized for creating M-files and a graphical MATLAB debugger. The editor can appear in a window by itself, or it can be a sub window in the desktop. M-files are denoted by the extension .m, as in pixel up.m. The MATLAB editor window has numerous pull-down menus for tasks such as saving, viewing, and debugging files.

Because it performs some simple checks and also uses color to differentiate between various elements of code, this text editor is recommended as the tool of choice for writing and editing M-functions. To open the editor window, type edit at the prompt opens the M-file file name.m in an editor window, ready for editing. As noted earlier, the file must be in the current directory, or in a directory in the search path.

Getting Help from MATLAB

The principal way to get help online is to use the MATLAB help browser, opened as a separate window either by clicking on the question mark symbol (?) on the desktop toolbar, or bytyping help browser at the prompt in the command window. The help Browser is a web browser integrated into the MATLAB desktop that displays a Hypertext Markup language (HTML) documents. The Help Browser consists of two panes, the help navigator pane, used to find information, and the display pane, used to view the information. Self-explanatory tabs other than navigator pane are used to perform a search.

3.2 DIGITAL IMAGE PROCESSING

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An important characteristic underlying the design of image processing systems is the significant level of testing and experimentation that normally is required before arriving at an acceptable solution. This characteristic implies that the ability to formulate approaches and quickly prototype candidate solutions generally plays a major role in reducing the cost and time required to arrive at a viable system implementation.

An image may be defined as a two-dimensional function f(x, y), where x & y are spatial coordinates, & the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital image by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensor, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to thevisual band of the EM spectrum imaging machines cover almost the entire EM spectrum, rangingfrom gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis& computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting & somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low-level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low- level process is characterized by the fact that both its inputs & outputs are images.

Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images butits outputs are attributes extracted from those images. Finally higher- level processing involves "Making sense" of an ensemble of recognized objects, as in image analysis & at the far end ofthe continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value. An image is represented as a two dimensional function f(x, y) where x and y are spatial co-ordinates and the amplitude of 'f' at any pair of coordinates (x, y) is called the intensity of theimage at that point.

Gray scale image

A grayscale image is a function I (xylem) of the two spatial coordinates of the image.

I(x, y) is the intensity of the image at the point (x, y) on the image plane.

I(xylem) takes non-negative values assume the image is bounded by a rectangle [0, a] $\times$[0, b]I:[0, a] $\times$ [0, b] $\rightarrow$ [0, info)

Color image

It can be represented by three functions, R (xylem) for red, G (xylem) for green *and* B (xylem) for blue. An image may be continuous with respect to the x and y coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate's values is called sampling. Digitizing the amplitude values is called quantization.

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image f(x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size M X N. The values of the coordinates (xylem) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates. In many image processing books, the image origin is defined to be at (xylem)=(0,0).The next coordinate values along the first row of the image are (xylem)=(0,1).It is important to keep in mind that the notation (0,1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Following figure shows the coordinate convention. Note that x ranges from 0 to M-1 and y from 0 to N-1 in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (xylem) the toolbox uses the notation (race) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate topples, (alb), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentationrefers to the coordinates. Less frequently the toolboxalso employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refers torows. This is the opposite of our use of variables x and y.

Image as Matrices

The preceding discussion leads to the following representation for a digitized image function:

f (0,0) f(0,1) ……….. f(0,N-1)

f (1,0) f(1,1) ………… f(1,N-1)

f (xylem)=f (M-1,0) f(M-1,1) f(M-1,N-1)

Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements.

A digital image can be represented naturally as a MATLAB matrix:

f (1,1) f(1,2)……….f(1,N)

f (2,1) f(2,2)……….f (2,N)

f (M,1) f(M,2)……….f(M,N)

Where f (1,1) = f(0,0), Clearly the two representations are identical, except for the shift in origin. The notation f(p ,q) denotes the element located in row p and the column q.

For example f(6,2) is the element in the sixth row and second column of the matrix f. Typically we use the letters M and N respectivelyto denote the number of rows and columns in a matrix. A 1xN matrix is called a row vector whereas an Mx1 matrix is called a column vector. A 1x1 matrix is a scalar. Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. All MATLAB quantities are written using mono-scope characters. We use conventional Roman, italic notation such as f(x ,y), for mathematical expressions

Reading Images:

Images are read into the MATLAB environment using function imread whose syntax

is imread (‘filename’)

Use of single quotes (‘) to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output If a semicolon is not included. MATLAB displays the resultsof the operation(s) specified in that line. The prompt symbol (>>) designates the beginning of a command line, as it appears in the MATLAB command window.

Data Classes

Although we work with integers coordinates the values of pixels themselves are not restricted to be integers in MATLAB. Table above list various data classes supported by MATLAB and IPT are representing pixels values. The first eight entries in the table are refers to as numeric data classes. The ninth entry is the char class and, as shown, the last entry is referred to as logical data class.

All numeric computations in MATLAB are done in double quantities, so this is also a frequent data class encounter in image processing applications. Class unit 8 also is encountered frequently, especially when reading data from storages devices, as 8 bit images are most common representations found in practice. These two data classes, classes logical, and, to a lesser degree, class unit 16 constitute the primary data classes on which we focus. Many ipt functions however support all the data classes listed in table. Data class double requires 8 bytesto represent a number uint8 and int 8 require one byte each, uint16 and int16 requires 2bytes and unit 32.

Int 32 and single required 4 bytes each. The char data class holds characters in Unicode representation. A character string is merely a 1*n array of characters logical array contains only the values 0 to 1 with each element being stored in memory using function logical or by using relational operators.

Image Types

The toolbox supports four types of images:

- Intensity images.
- Binary images.
- Indexed images.
- R G B images.

Most monochrome image processing operations are carried out using binary or intensity images, so our initial focus is on these two image types. Indexed and RGB colour images.

Intensity Images

An intensity image is a data matrix whose values have been scaled to represent intentions. When the elements of an intensity image are of class unit 8, or class unit 16, they have integer values in the range [0,255] and [0,65535], respectively. If the image is of class double, the values are floating point numbers. Values of scaled, double intensity images are in the range [0, 1] by convention.

Binary Images

Binary images have a very specific meaning in MATLAB.A binary image is a logical array 0s and1s.Thus, an array of 0s and 1s whose values are of data class, say unit8, is not considered as a binary image in MATLAB. A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0's and 1's, we create an array B using the statement.

B=logical (A)

If A contains elements other than 0s and 1s.Use of the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.

Using relational and logical operators also creates logical arrays.

To test if an array is logical we use the I logical function is logical(c).

If c is a logical array, this function returns a 1. Otherwise returns a 0. Logical array can be converted to numeric arrays using the data class conversion functions.

Indexed Image

An indexed image has two components

A data matrix integer, xA color map matrix.

Matrix map is an m*3 arrays of class double containing floating point values in the range [0, 1].The length m of the map are equal to the number of colors it defines. Each row of map specifies the red, green and blue components of a single color. An indexed images uses “direct mapping” of pixel intensity values color map values. The color of each pixel is determined by using the corresponding value the integer matrix x as a pointer in to map. If x is of class double then all of its components with values less than or equal to 1point to the first row in map, all components with value 2 point to the second row and soon. If x is of class units or unit 16, then all components value 0 point to the first row in map, all components with value 1 point to the second and so on.

RGB Image

A RGB color image is an M*N*3 array of color pixels where each color pixel is triplet corresponding to the red, green and blue components of an RGB image, at a specific spatial location. An RGB image may be viewed as “stack” of three gray scale images that when fed in tothe red, green and blue inputs of a color monitor

Produce a color image on the screen. Convention the three images forming an RGB colorimage are referred to as the red, green and blue components images. The data class of the components images determines their range of values. If an RGB image is of class double the range of values is [0, 1].

Similarly the range of values is [0,255] or [0, 65535]. For RGB images of class units or unit 16 . The number of bits use to represents the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8-bit image, the corresponding RGB image is said to be 24-bits deep.

Generally, the number of bits in all component images is the same. In this case the number of possible color in an RGB image is (2^b) ^3, where b is a number of bits

3.3 ANALYSIS OF THE PROBLEM

With the aggravation of social public security problems, the face recognition without occluded parts can no longer meet the needs of modern society. All kinds of face recognitions in complex environment need to be realized in the real situations. Face images are disturbed by occlusion factors, which results in the loss or alienation of the recognized image pixels, thus resulting in the recognition errors of traditional models.

Even though human eye can detect and identify faces in a scene easily, building an automatic system that accomplishes such a task is still very challenging. Due to the dynamic nature of face images, a face recognition system encounters various problems during the recognition process.

Robust face recognition requires the ability to recognize identity despite many variations in appearance that the face can have in a scene.The appearance a face when projected onto image can vary tremendously. The challenges are mainly from large variations of illumination, expression, pose, corruption, and occlusion which can cause sharp decline in recognition rate. At present, researchers have proposed numerous illumination processing algorithms for face images under complex illumination conditions, and achieved relatively good experimental results. attitude and illumination in environmental factors.

The algorithms based on 2-D model are the research hotspots Deep learning algorithm has the characteristics of high recognition rate and strong robustness. The Deep Learning based algorithm uses an effective method of extracting the ROBUST deep features of face images under severe illumination conditions based on the combination of Log-Gabor filter and LBP (an effective local texture descriptor. It is widely used in face recognition because of its advantages in image texture description) with the Deep Belief Network (DBN). The description of LBP (local binary pattern) and combine it with Gabor wavelet to obtain the best representation of face image features under complex illumination conditions The Deep Learning based face recognition methods obtain very good performances and it obtains accurate recognition rate.

3.4 BLOCK DIAGRAM/DESIGN OF PROJECT

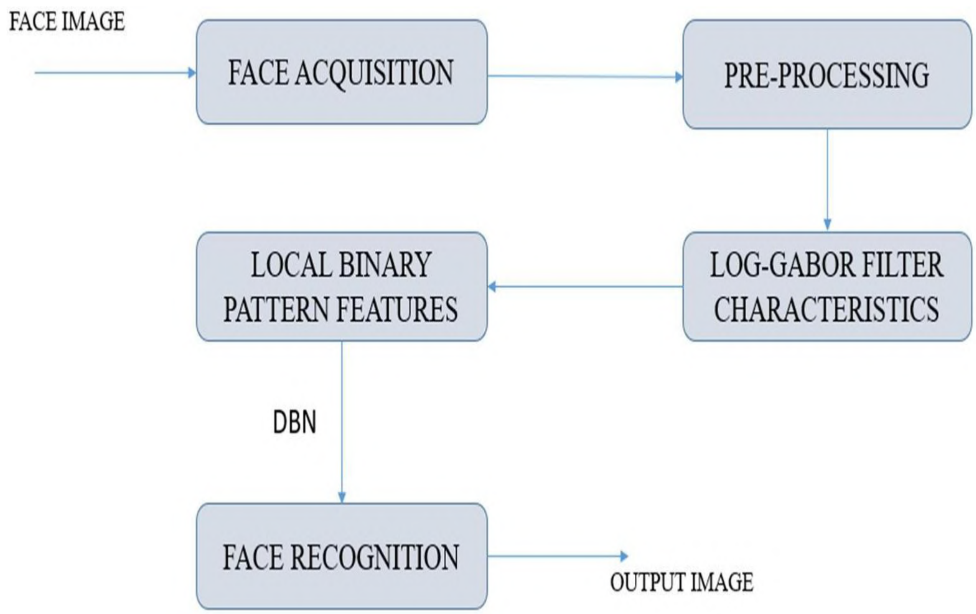


Figure 3-1 Block diagram

3.5 MODULES AND THEIR FUNCTIONALITIES

The project involves the following standard units or blocks for its implementation

PREPROCESSING

Illumination preprocessing is applied to improve the adverse effects of intense illumination changes on face images. The pre-processing is done so as to analyze the face images based on the required needs it involves the resizing of images, minimum pixels , maximum pixels, adjusting the contrast, brightness, the foreground and background of each image for the required algorithm.

LOG-GABOR

Log-Gabor function is an alternative to the Gabor function.log-Gabor functions, by definition, always have no DC component, therefore, the influence of illumination conditions on image processing is relatively small, which can overcome the adverse effects of illumination on face recognition to a certain extent. It acts as a low pass filter to get rid of high frequency components. It is a filter for extracting the log-gabor features of an image.

LOCAL BINARY PATTERN

The local binary pattern operator is used to extract the sub-blocks of the image. LBP has the advantages of gray translation invariance, rotation invariance and simple calculation. The texture features of LBP have been successfully applied in texture classification, face recognition, image analysis, background modeling and other fields. It shows superior performance in texture classification of a 2-D image .

DEEP BELIEF NETWORK

DBN is a typical deep learning network. Its prototype is similar to human brain tissue structure. The DBN extracts the input data features from simple to complex and from low-level to high-level. The DBN does not rely on the manual selection, and it learns the input data actively and digs out rich information hidden in the known data automatically. Using the deep learning and DBN the data is sliced, layered and forms histograms and recognition is done effectively by combining low level features.

4 PROJECT IMPLEMENTATION

4.1 FLOWCHART OF PROPOSED PROJECT

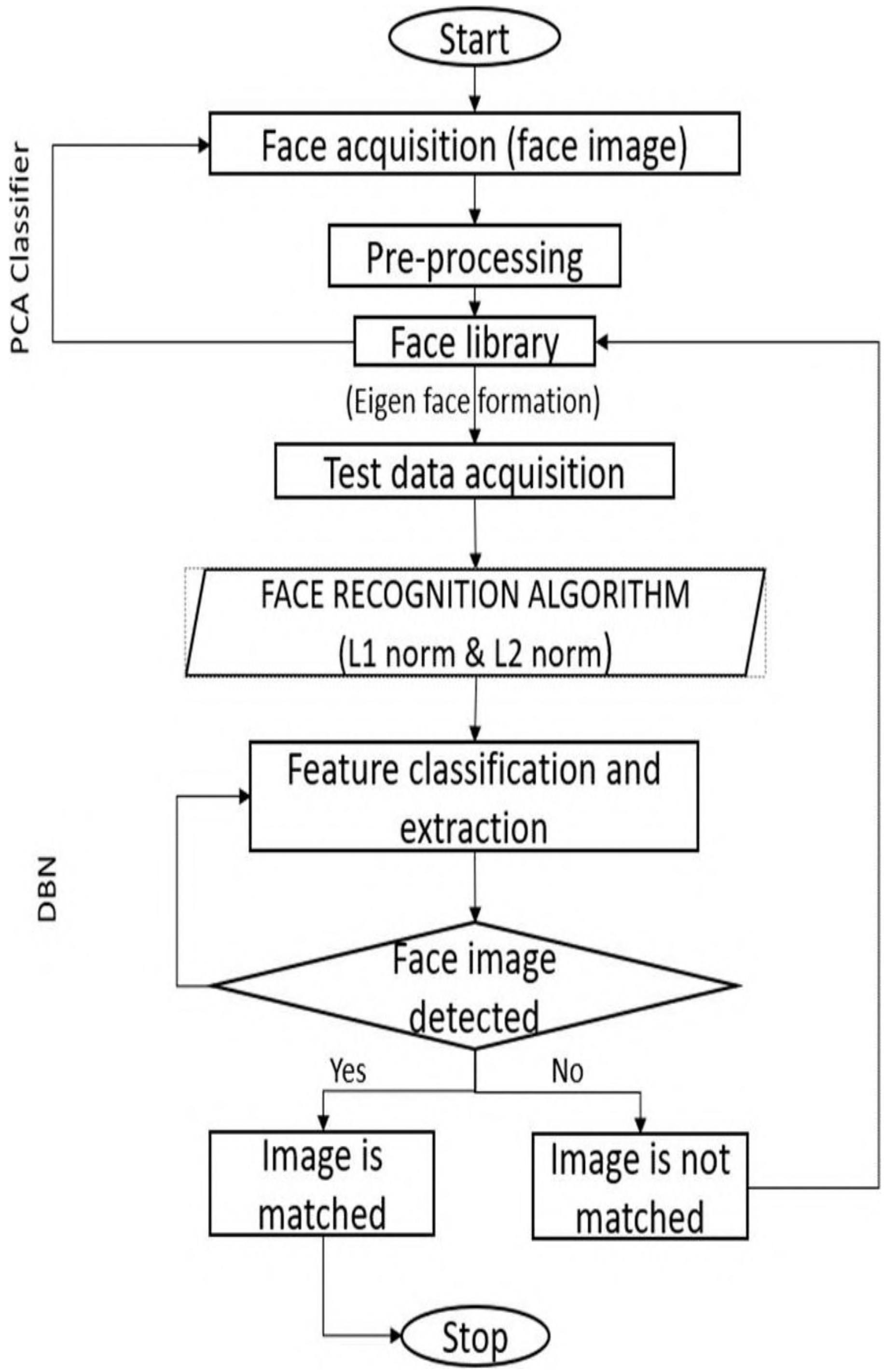


Figure 4-1 Flowchart of project.

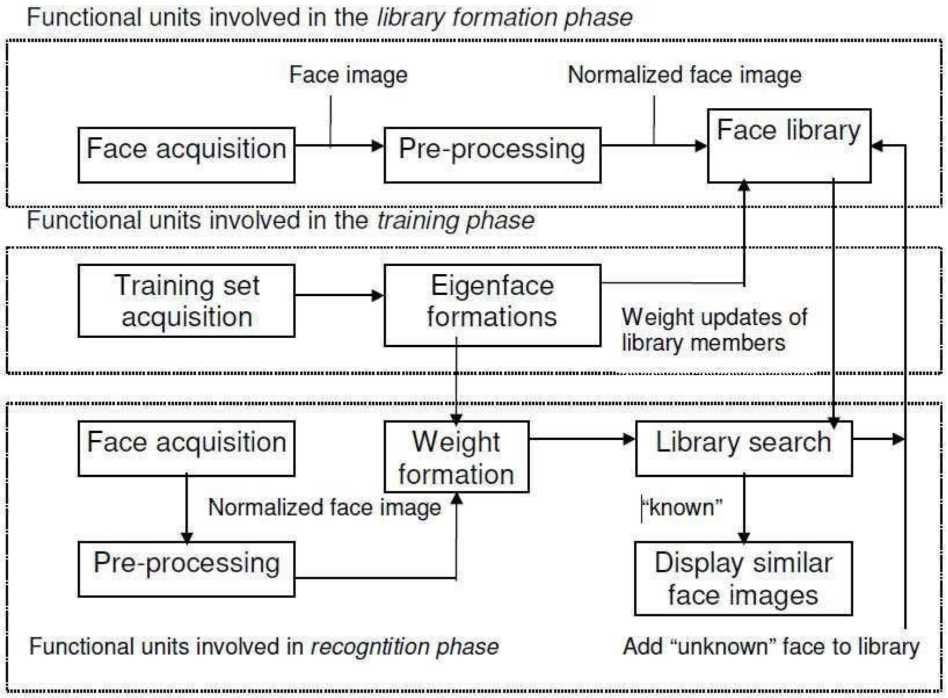


Figure 4-2 Functional block diagram.

The above block diagram represents the functional units involved and the phases involved in the face recognition process

4.2 IMPLEMENTATION STAGES

The proposed face recognition system passes through three main phases/stages during a face recognition process. Three major functional units are involved in these phases and they are defined. The characteristics of these phases in conjunction with the three functional units are given below:

4.2.1 Face Library Formation Phase

In this phase, the acquisition and the preprocessing of the face images that are going to be added to the face library are performed. Face images are stored in a face library in the system. We call this face database a "face library" because at the moment, it does not have the properties of a relational database. Every action such as 8 training set or eigenface formation is performed on this face library. Face library is initially empty. In order to start the face recognition process, this initially empty face library has to be filled with face images. In order to perform image size conversions and enhancements on face images, there exists the "pre-processing" module. This module automatically converts everyface image to necessary size based on user request, it can modify the dynamic range of face images (histogram equalization) in order to improve face recognition performance. After acquisition and pre- processing, face image under consideration is added to the face library. Each face is represented by two entries in the face library: One entry corresponds to the face image itself and the other corresponds to the weight vector associated for that face image. Weight vectors of the face library members are empty until a training set is chosen and eigenfaces are formed.

4.2.2 Training Phase

After adding face images to the initially empty face library, the system is ready to perform training set and eigenface formations. Those face images that are going to be in the training set are chosen from the entire face library. Because that the face library entries are normalized, no further pre-processing is necessary at this step. After choosing the training set, eigenfaces are formed and stored for later use. Eigenfaces arecalculated from the training set, keeping only the M images that correspond to thehighest eigen values. These M eigenfaces define the M-dimensional "face space". As new faces are experienced, the eigen faces can be updated or recalculated. The corresponding distribution in the M-dimensional weight space is calculated for each face library member, by projecting its face image onto 9 the "face space" spanned by the eigen faces. Now the corresponding weight vector of each face library member has been updated which were initially empty. The system is now ready for the recognition process.

4.2.3 Recognition and Learning Phase

Now the corresponding weight vector of each face library member has been updated which were initially empty. The system is now ready for the recognition process. User initiates the recognition process through selecting a face image. Based on the user request and the acquired image size, pre-processing is applied to normalize this acquired image to face library specifications (if necessary). Once the image is normalized, its weight vector is constructed with the help of the eigenfaces that were already stored during the training phase. After obtaining the weight vector, it is compared with the weight vector of every face library member within a user defined "threshold".If there exists at least one face library member that is similar to the acquired image within the threshold then, the face image is classified as "known". Otherwise, a miss has occurred and the face image is classified as "unknown". After being classified as unknown, this new face imagecan be added to the face library with its corresponding weight vector for later use

4.3 EXPERIMENTAL RESULTS

4.3.1 TEST DATA COLLECTED

Figure 4-3 Train Database

4.3.2 TRAIN DATA COLLECTED

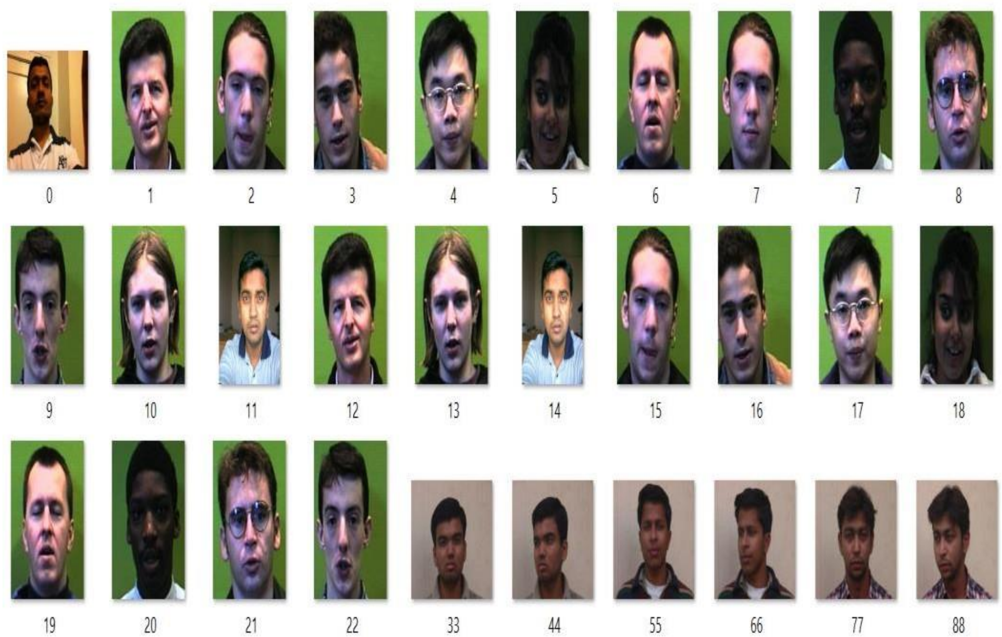


Figure 4-4 Test Database

The database of images shown above are both training dataset and test dataset. These are taken as input images for Robust Face Recognition.

When the program is executed the following is shown , it displays Train and Test data uploading.

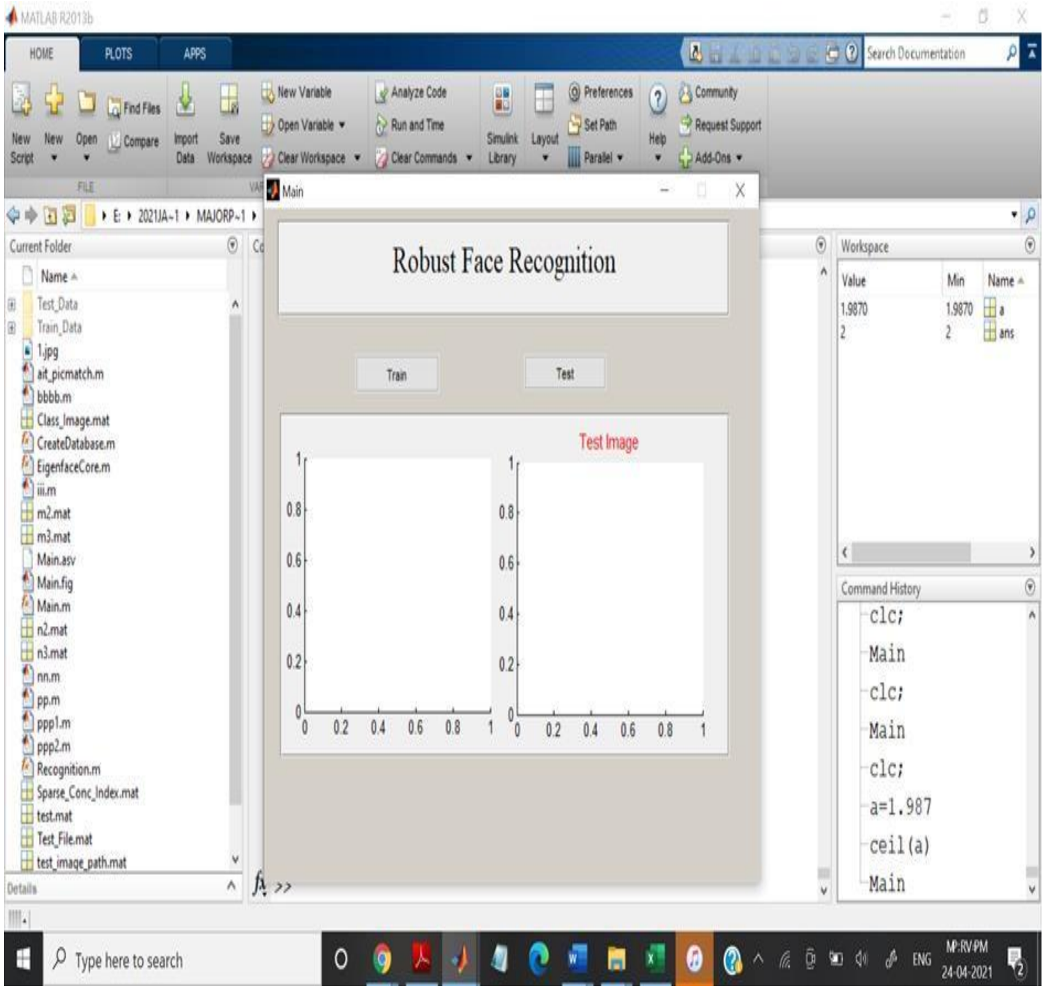


Figure 4-5 Uploading Train and Test database

Use the images above and Upload them into train and test database, select the test image and crop it and upload to test database. The following outputs are displayed.

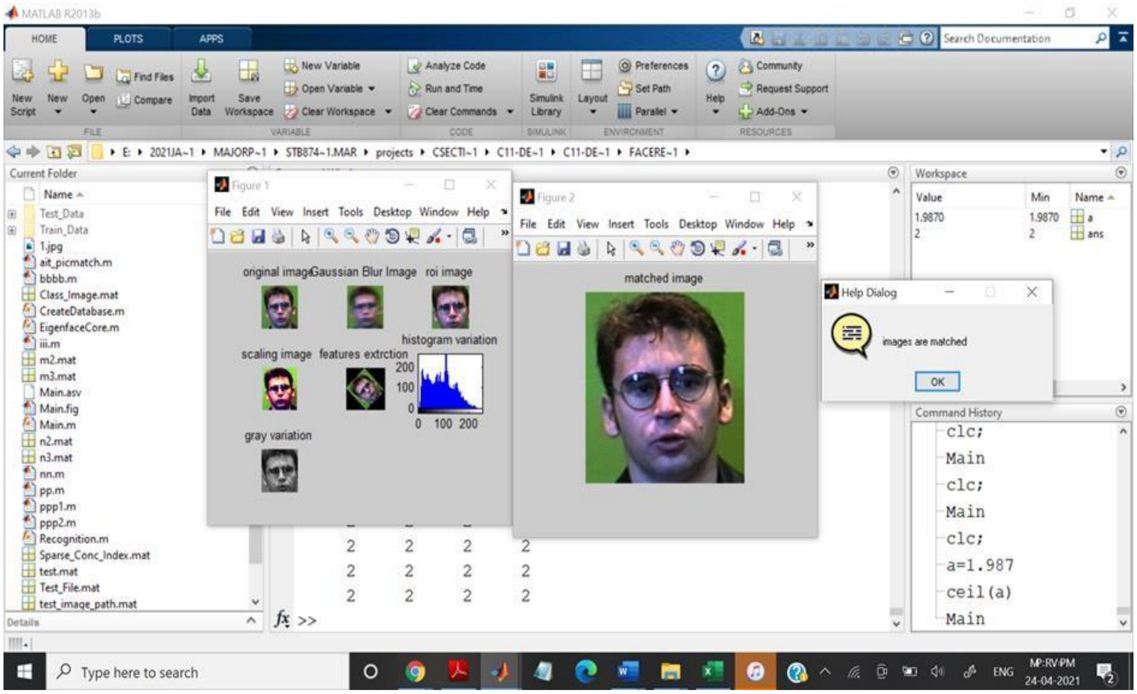


Figure 4-6 Result displaying Images are Matched.

The image shown is a result displaying “**Images are matched**.”

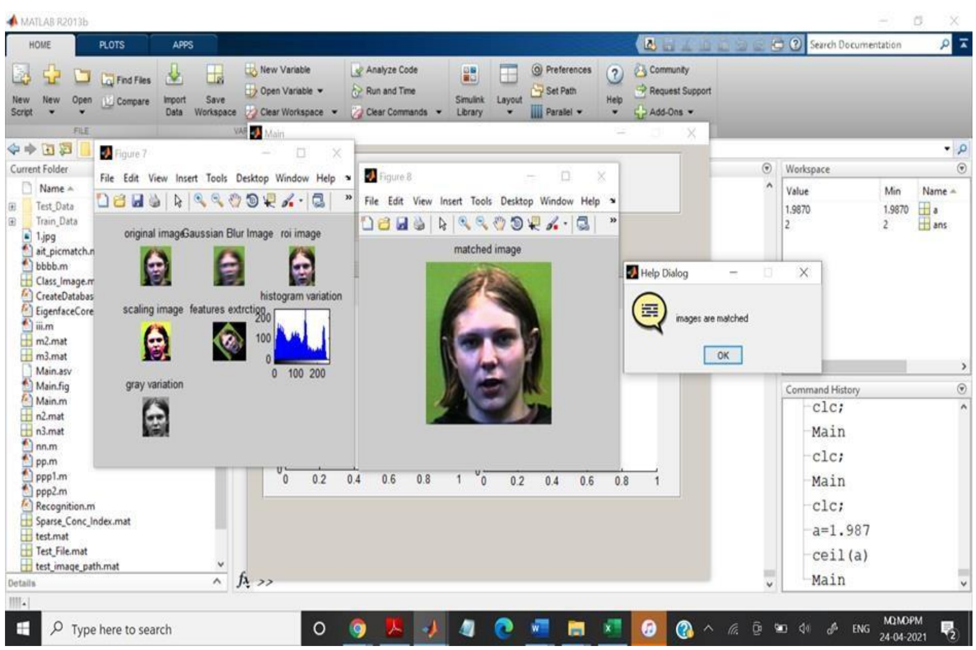


Figure 4-7 Image displaying images are matched.

The following.displays result showing “**Images are not matched.**”

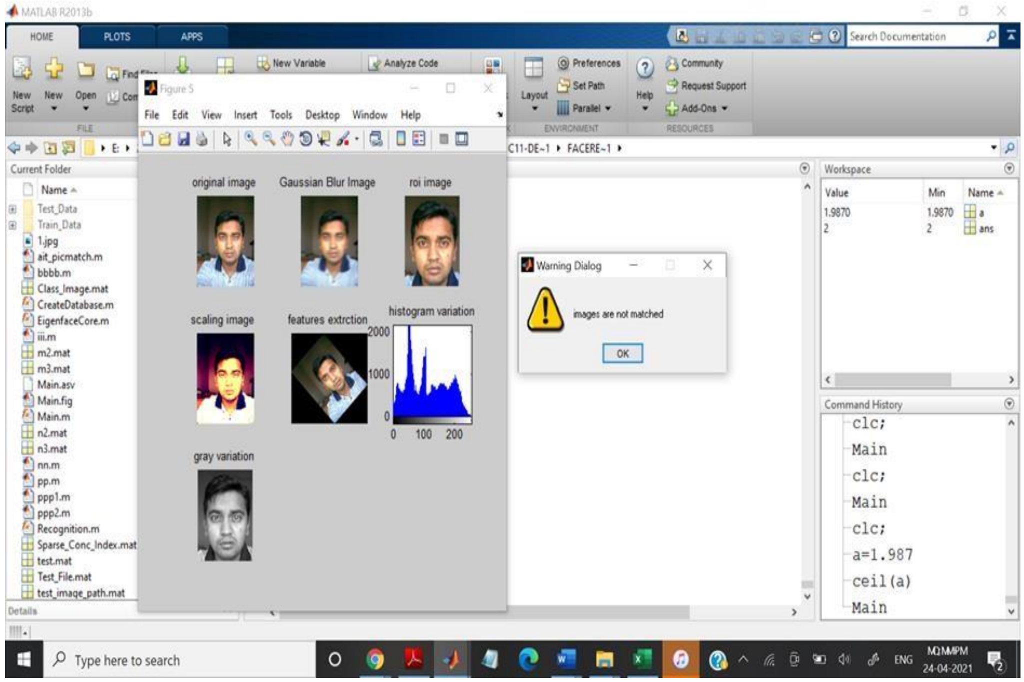


Figure 4-8 Result displaying Images are not Matched

5 PROJECT TESTING

5.1 OVERVIEW OF TESTING METHODS

Yale B+, CMU-PIE databases are selected to evaluate the performance of the method introduced in this project. Face recognition on those databases is still a challenging task owing to their complex lighting conditions. Yale B+ consists of 38 subjects under 64 lighting conditions. The CMU-PIE includes 68 subjects under 21 illumination conditions with a little hair and background in the face images.

Methods	CMU-PIE	Yale B+
PCA+SVM	72.53	85.78
LBP+SVM	80.52	92.16
DBN	84.36	93.61
LBP+DBN	90.12	95.17
PBLGLD	95.80	98.89

Figure 5-1 Overview of testing methods

The recognition rate of Proposed method Based on Log-Gabor, Local binary pattern and Deep belief network (PBLGLBD) is compared with the recognition rate of the methods based on LBP- DBN, DBN, LBP and SVM, PCA and SVM. The experimental selections of training sets and testing sets on Yale B+, and CMU-PIE are exactly the same as the previous experimental selections. The experiments of the PBLGLD and the LBP - DBN are under situations with the best partitioning way and the best hidden unit numbers. For these experimental methods, entire face images need to be directly input. The face recognition experiments are performed 20 times by each face recognition method, and the average of recognition rates

As shown in table above, for these experimental databases, Yale B+, and CMU-PIE, the proposed PBLGLD method can achieve the highest recognition rate on the face databases. The PBLGLD recognition rate is up to 95.80% on CMU-PIE and 98.89% on Yale B+ database. The proposed method has superior performance in face recognition by comparing with that of other methods. The proposed method has superior performance in face recognition by comparing with some state-of-the-arts.

6 CONCLUSION AND FUTURE ENHANCEMENT

This project presented a novel Robust regularized coding (RRC) model and an associated effective iteratively reweighted regularized robust coding (IR3C) algorithm for robust face recognition (FR). One important advantage of RRC is its robustness to various types of outliers (e.g., occlusion, corruption, expression, etc.) by seeking for an approximate MAP (maximum a posterior estimation) solution of the coding problem. By assigning adaptively and iteratively the weights to the pixels according to their coding residuals, the IR3C algorithm could robustly identify the outliers and reduce their effects on the coding process. Meanwhile, we have shown that the $l2$-norm regularization is as powerful as $l1$-norm regularization in RRC but the former has much lower computational cost.

The proposed RRC methods were extensively evaluated on FR with different conditions, including variations of illumination, expression, occlusion, corruption, and face validation. The experimental results clearly demonstrated that RRC outperforms significantly previous state-of-the-art methods, such as Sparse Representation Coding(SRC), correntropy based sparse representation (CESR) and Gradient-based sparse representation (GSRC). In particular, RRC with $l2$-norm regularization could achieve very high recognition rate but with low computational cost, which makes it a very good candidate scheme for practical robust Face Recognition systems.

It can be concluded that, compared with other algorithms, in the case of occlusion factors interfering with the model, the model does not lose stability, and can still distinguish the faces of different people very well, and the recognition rate is higher. In conclusion, compared with traditional algorithms, the algorithm proposed in the paper is more practical in face recognition with occlusion.

PUBLICATION

TITLE OF THE PAPER - Deep Learning For Face Recognition Under Complex Illumination Based On Log-gabor And Lbp

Paper ID - ICSMEC21-0040

Paper Accepted.

On Thu, 27 May 2021 at 8:18 AM, ic smec ece <icsmec21@smec.ac.in> wrote:

Dear Author(s),

Warm Greetings from St.Martins Engineering College.

With heartiest congratulations we are pleased to inform you that based on the recommendations of the expert reviewers, your paper has been accepted for journal publication and oral presentation in the online mega International Conference on "Smart Modernistic in Electronics and Communication" (ICSMEC-21).

Paper ID:ICSMEC21-0040
Paper Title:Deep Learning for Face Recognition under Complex Illumination Conditions Based on Log-Gabor and LBP

Paper Accepted.

Please Send the final camera ready paper Paper After Registration

REFERENCES

[1] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, vol. 3, no.1, pp. 71–86, 1991.

[2] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriengman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[3] B. Heisele, P. Ho, and T. Poggio, "Face recognition with support vector machine: Global versus component-based approach," *Proc. IEEE Int'l Conf. Computer Vision*, 2001.

[4] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp.2037–2041, 2006.

[5] A. Leonardis and H. Bischof, "Robust recognition using eigenimages," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 99-118, 2000.

[6] A.M. Martinez, "Recognizing Imprecisely localized, partially occluded, and expression variant faces from a single sample per class," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 748-763, 2002.

[7] J.T. Chien, and C.C. Wu, "Discriminant waveletfaces and nearest feature classifiers for face recognition," *IEEETrans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1644-1649, 2002.

[8] J. Laaksonen, "Local subspace classifier", *Proc. Int'l Conf. Artificial Neural Networks*, 1997.

[9] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, and Y. Ma, "Robust face recognition viasparse representation," *IEEETrans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[10] M. Yang and L. Zhang, "Gabor Feature based Sparse Representation for Face Recognition with Gabor Occlusion Dictionary," *Proc. European Conf. Computer*

Vision,2010.

[11] D. Donoho, “For Most Large Underdetermined Systems of Linear Equations the Minimal *l*1-Norm Solution is also the Sparsest Solution,” *Comm. Pure and Applied Math.*, vol. 59, no. 6, pp. 797–829, 2006.

[12] L. Zhang, M. Yang and X.C. Feng, “Sparse representation or collaborative representation: which helps face recognition?” *Proc. IEEE Int’l Conf. Computer Vision*, 2011.

[13]A. Wagner, J. Wright, A. Ganesh, Z.H. Zhou, and Y. Ma, “Towards a Practical Face Recognition System: RobustRegistration and Illumination by Sparse Representation,” *Proc. IEEE Conf. Computer Vision andPattern Recognition*, 2009.

[14] R. He, W.S. Zheng, B.G. Hu, and X.W. Kong, “A regularized correntropy framework forrobust pattern recognition,” *Neural Computation*, vol. 23, pp. 2074-2100, 2011.

[15] M. Yang, L. Zhang, J. Yang and D. Zhang, “Robust sparse coding for face recognition,”*Proc. IEEE Conf. ComputerVision and Pattern Recognition*, 2011.

[16] J. Yang and Y. Zhang, “*Alternating direction algorithms for l1-problems in compressivesensing*,” Technical report,Rice University, 2009.

[17] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistica Learning.* SpringerSeries in Statistics, 2003.

[18] P. Viola and M.J. Jones, “Robust real-time face detection,” *Int’l J. Computer Vision*, vol.57, pp. 137-154, 2004.

[19] R. Rigamonti, M. Brown, V. Lepetit, “Are Sparse Representations Really Relevant forImage Classification?” *Proc.IEEE Conf. Computer Vision and Pattern Recognition*, 2011..

[20] E.J. Candès, M.B. Wakin, and S.P. Boyd, “Enhancing Sparsity by reweighted *l*1Minimization,” *J. Fourier Anal.Appl.*, vol.14, pp. 877-905, 2008.

[21] Z.Y. Zhang, "Parameter Estimation Techniques: A Tutorial with Application to ConicFitting," *Image and Vision Computing*, vol.15, no.1, pp. 59-76, 1997.

[22] J. Ruiz-del-Solar, R. Verschae, and M. Correa, "Recognition of Facesin Unconstrained Environments: A Comparative Study," *EURASIP Journal on Advances in Signal Processing (Recent Advances inBiometric Systems:A Signal Processing Perspective)*, Vol. 2009, Article ID 184617.

[23] L. Wolf, T. Hassner, and Y. Taigman, "Effective Face Recognition by Combining MultipleDescriptors and Learned Background Statistic," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 10,pp. 1978-1990, 2011.

[24] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Principal Component Analysis of ImageGradient Orientations forFace Recognition", *Proc. IEEE Conf. Automatic Face & Gesture Recognition and Workshops*,2011.

BIBLIOGRAPHY

[1] G. Wu, J. Tao and X. Xu, "Occluded Face Recognition Based on the Deep Learning," 2019 Chinese Control And Decision Conference (CCDC), 2019 <https://ieeexplore.ieee.org/Xplore/home.jsp >

[2] L. Zhuang and Y. Guan, "Deep Learning for Face Recognition under Complex Illumination Conditions Based on Log-Gabor and LBP," 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019 < https://ieeexplore.ieee.org/Xplore/home.jsp>

[3] O. A. Aghdam and H. K. Ekenel, "Robust deep learning features for face recognition under mismatched conditions," 2018 26th Signal Processing and CommunicationsApplicationsConference (SIU), 2018 <https://ieeexplore.ieee.org/Xplore/home.jsp>

[4] A. Georghiades, P. Belhumeur, and D. Kriegman, “From few to many: Illumination conemodels for face recognition under variable lighting and pose,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.23, no. 6, pp. 643–660, 2001 < https://ieeexplore.ieee.org/Xplore/home.jsp >

[5]S. Fidler, D. Skocaj, and A. Leonardis, “Combining Reconstructive and DiscriminativeSubspace Methods for Robust Classification and Regression by Subsampling,” *IEEE Trans. Pattern Analysis andMachine Intelligence*, vol. 28, no. 3, pp. 337-350, 2006.< https://ieeexplore.ieee.org/Xplore/home.jsp >

APPENDICES

```
%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%Face
Recognition
%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%
%
function varargout = Main(varargin)
% MAIN MATLAB code for Main.fig
%MAIN, by itself, creates a new MAIN or raises the existing
%singleton*.
%
%H = MAIN returns the handle to a new MAIN or the handle to
%the existing singleton*.
%
%MAIN('CALLBACK',hObject,eventData,handles,...) calls the local
%function named CALLBACK in MAIN.M with the given input
arguments.
%
%MAIN('Property','Value',...) creates a new MAIN or raises the
%existing singleton*.  Starting from the left, property value pairs are
%applied to the GUI before Main_OpeningFcn gets called.  An
%unrecognized property name or invalid value makes property application
%stop.  All inputs are passed to Main_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help Main
% Last Modified by GUIDE v2.5 13-Mar-2021 15:57:50
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @Main_OpeningFcn, ...
'gui_OutputFcn', @Main_OutputFcn, ...
'gui_LayoutFcn', [] , ...
'gui_Callback',   []);
if nargin && ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end
```

```
if nargout
[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
Else
gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before Main is made visible.
function Main_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
% --- Outputs from this function are returned to the command line.
function varargout = Main_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%%%%%%%%%%%%%%%%%%%%     $ Loading Test and Train
Databases                                          $
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
global A m1 n1 No_Files_In_Class_Folder Class_Count
Training_Set_Folder
Training_Set_Folder                                =
[uigetdir('FaceDatabase\Database1\Train_Data'),'\'];
m1 = 6;
n1 = 3;
helpdlg('Train Database Loaded');
%%%%%%%%%%%%%    Loading    Train
Database%%%%%%%%%%%%%%%%
Train_Database = dir(Training_Set_Folder);
No_Folders_In_Train_Database = length(Train_Database);
File_Count = 1;
Class_Count = 1;
h = waitbar(0,'Reading Train Images,Please wait...');
for k = 3:No_Folders_In_Train_Database
waitbar(k/(No_Folders_In_Train_Database-2))
Class_Folder = [Training_Set_Folder '\' Train_Database(k).name,'\'];
imgs_Train = dir(Class_Folder)
No_Files_In_Class_Folder(Class_Count) = length(imgs_Train)-
2; drawnow;
for p = 3:No_Files_In_Class_Folder(Class_Count)+2
Tmp_Image_Path = Class_Folder;
Tmp_Image_Name = imgs_Train(p).name;
Tmp_Image_Path_Name = [Tmp_Image_Path,Tmp_Image_Name];
if strcmp(Tmp_Image_Name,'Thumbs.db')
Break
```

```
end
test = imread(Tmp_Image_Path_Name);
if length(size(test))==3
Tmp_Image = rgb2gray(test);
else
Tmp_Image = test;
end
Tmp_Image_Down_Sampled = double(imresize(Tmp_Image,[m1 n1]));
Image_Data_Matrix(:,File_Count) = Tmp_Image_Down_Sampled(:);
File_Count = File_Count+1;
end
Class_Count = Class_Count+1;
end
close(h)
A = Image_Data_Matrix;
A =
A/(diag(sqrt(diag(A'*A))));%%%%%%%Covaraince
Matrix%%%%%%%%%
function pushbutton2_Callback(hObject, eventdata, handles)
global A m1 n1 No_Files_In_Class_Folder Class_Count
Training_Set_Folder
global ID
warning off;
[Test_File  Test_File_Path]
=
uigetfile('FaceDatabase\Da
tabase1\*.jpg;*.pgm;*.png;
*.tif','Select a Test Image');
test_image_path = [Test_File_Path Test_File];
% axes(handles.axes1)
cla
% axes(handles.axes2)
cla
% axes(handles.axes1)
% imshow(test_image_path);
% imwrite(test_image_path,'1.jpg');
save test_image_path
title('Test Image','Color','red','FontSize',15);
save Test_File
% LEN = 31;
% THETA = 11;
% PSF = fspecial('motion',LEN,THETA);
% Blurred = imfilter(Test_File,PSF,'circular','conv');% figure;
imshow(Blurred);title('Blurred Image');
 iii
drawnow;
```

```
Test_File = [Test_File_Path Test_File];
test = imread(Test_File);
if length(size(test))==3
   Test_Image = rgb2gray(test);
else
   Test_Image = test;
end
save test
save Test_File
 %%%%%%%%%%%%%%%%%%%%Robust face recognisation ALGORITHM
BEGINS%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
Test_Image_Down_Sampled = double(imresize(Test_Image,[m1 n1]));
y = Test_Image_Down_Sampled;
n = size(A,2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%Reguralized Robust Coding%%%%%%
%%%%%%%%%%%%%%%%%%%%%% for l1 Norm (Top Rows)
%%%%%%%%%%%%%%%%%%%%%%%%
 residual =(Test_Image_Down_Sampled-y).^2;
residual_sort = sort(residual);
svm = residual_sort(ceil(length(residual))); beta = svm;
w = 1./(1+1./exp(-beta*(residual-svm)));
weight_pref = w;
norm_y_D = norm(y);
y = y./norm(y);
nIter=svm;
for nit = 1: nIter
tem_w = w./max(w);
index_w = find(tem_w>1e-3);
% remove the pixels with very small weight
W_y = w(index_w).*y(index_w);
W_D = repmat(w(index_w),[1 size(A,2)]).*A(index_w,:);
%---------------Internal sparse coding begin------------- %
 x_i =  zeros(A,1);
w_i =  ones(A,1);
x = ones(A,1);
Kratio =  0.01;
innerit  =  0;
yupu_pref  =  10;
WDWD  =  W_D'*W_D;
WDWy  =  W_D'*W_y;
newlambda =  lambda*norm(W_y);
while norm(x-x_i,2)/norm(x,2) > 1e-2 && innerit <=10
x_i  = x;
```

```
 w_l =  repmat(w_i,[1 A]);
w_r =  w_l';
 z =  (WDWD.*w_r+newlambda*eye_M)\WDWy;
x= w_i.*z;
 x_sort  =  sort(abs(x));
yupu  =  abs(x_sort(ceil(Kratio*dic_l)));
yupu = min(yupu/dic_l,yupu_pref);
yupu_pref  = yupu;
 w_i =  sqrt(x.^2+yupu.^2);
innerit =  innerit + 1;
end
temp_s  =  x;
save temp_s
residual = norm_y_D^2.*(y-A*temp_s).^2;
residual_sort = sort(residual);
svm = residual_sort(ceil(median_a*length(residual))); beta = beta_a/svm;
w = 1./(1+1./exp(-beta*(residual-svm)));
weight_g  = norm(w-weight_pref,2)/norm(weight_pref,2);
weight_pref = w;
weight_gap = [weight_gap weight_g];
rec_sample = temp_s;
end
%%%%%%%%%%%%%%%%%%%%%%%            Checking
Identity  %%%%%%%%%%%%%%%%%%%%%%
n = size(A,2);
f=ones(2*n,1);
Aeq=[A -A];
lb=zeros(2*n,1);
rec_sample=f;
x1 = linprog(rec_sample,[],[],Aeq,y,lb,[],[],[]);
x1 = x1(1:n)-x1(n+1:2*n);
nn = No_Files_In_Class_Folder;
nn = cumsum(nn);
tmp_var = 0;
k1 = Class_Count-1;
for i = 1:k1
delta_xi = zeros(length(x1),1);
if i == 1
else
tmp_var = tmp_var + nn(i-1);
begs = nn(i-1)+1;
ends = nn(i);
delta_xi(begs:ends) = x1(begs:ends);
end
tmp(i) = norm(A*delta_xi,2);
tmp1(i) = norm(delta_xi,1)/norm(x1,1);
```

```
end
TotImg=Test_Image+1;
Sparse_Conc_Index(TotImg) = (k1*max(tmp1)-1)/(k1-1);
save Sparse_Conc_Index
clss = find(tmp==min(tmp));
% figure,plot(tmp)
ssttrr = sprintf('The Test Image Corresponds to Class: %d',clss);
cccc = dir([Training_Set_Folder]);
Which_Folder = dir([Training_Set_Folder,cccc(clss+2).name,'\']);
Which_Image = randsample(3:length(Which_Folder),1);
Image_Path =
[Training_Set_Folder,cccc(clss+2).name,'\',Which_Folder(Which_Image
).name];
 Class_Image = (Image_Path);
%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%% L2 Norm
(Bottom
Rows)%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%=====================
begin============================%
residual = (Test_Image_Down_Sampled-y).^2;
residual_sort = sort(residual);
svm = residual_sort(ceil(length(residual))); beta = svm;
w = 1./(1+1./exp(-beta*(residual-svm)));
weight_pref = w;
norm_y_D = norm(y);
y = y./norm(y);
   %%%%%%%%%%%%%%%%%%%%%%%%%%%% via
Iteratively Reweighting
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for nit = 1: nIter
tem_w = w./max(w);
index_w = find(tem_w>1e-3);
% remove the pixels with very small weight
W_y = w(index_w).*y(index_w);
W_D = repmat(w(index_w),[1 size(D,2)]).*D(index_w,:);
%temp_s = inv(W_D'*W_D+lambda*norm(W_y)*eye_M)*W_D'*W_y;
temp_s = (W_D'*W_D+lambda*norm(W_y)*eye_M)\(W_D'*W_y);
residual = norm_y_D^2.*(y-D*temp_s).^2;
residual_sort = sort(residual);
svm = residual_sort(ceil(median_a*length(residual))); beta = beta_a/svm;
w = 1./(1+1./exp(-beta*(residual-svm)));
weight_g  = norm(w-weight_pref,2)/norm(weight_pref,2);
```

```
weight_pref = w;
weight_gap = [weight_gap weight_g];
if weight_g < weight_thre
nit
break;
end
s_pref = temp_s;
end
    %==========================checking
feature
classification============================%
for class = 1:Class_Count
s = (Image_Path == class);
end
ID = s;
%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%
%%     Comparision with Test Database train
%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
f=ones(2*n,1);
Aeq=[A -A];
lb=zeros(2*n,1);
ID=f;
x1 = linprog(ID,[],[],Aeq,y,lb,[],[],[]);
x1 = x1(1:n)-x1(n+1:2*n);
nn = No_Files_In_Class_Folder;
nn = cumsum(nn);
tmp_var = 0;
k1 = Class_Count-1;
for i = 1:k1
delta_xi = zeros(length(x1),1);
if i == 1
else
tmp_var = tmp_var + nn(i-1);
begs = nn(i-1)+1;
ends = nn(i);
delta_xi(begs:ends) = x1(begs:ends);
 end
tmp1(i) = norm(delta_xi,1)/norm(x1,1);
end
Sparse_Conc_Index = (k1*max(tmp1)-1)/(k1-1);
clss = find(tmp==min(tmp));
cccc = dir([Training_Set_Folder]);
```

```
Which_Folder = dir([Training_Set_Folder,cccc(clss+2).name,'\']);
Which_Image = randsample(3:length(Which_Folder),1);
Image_Path =
[Training_Set_Folder,cccc(clss+2).name,'\',Which_Folder(Which_Image
).name];
Class_Image = (Image_Path);
%axes(handles.axes2);
%imshow(Class_Image);
%title('Detected Image','Color','black','FontSize',15);
save Class_Image
ppp1
ppp2
%pic1 = imread(test_image_path);
%pic2 = imread(Class_Image);
%    ait_picmatch(pic1,pic2);
%TrainDatabasePath = uigetdir('C:\Documents and
Settings\phani.TAKEOFFPROJECTS.001\Desktop\Face Recog RRC
Code\PCAbased eigen attend\PCA_ based eigen Face Recognition
System', 'Select training database path' );
%load test
%T = CreateDatabase(TrainDatabasePath);
%[m, A, Eigenfaces] = EigenfaceCore(T);
%OutputName = Recognition(test, m, A, Eigenfaces);
function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a
double
 --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes on button press in togglebutton1.
function togglebutton1_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of togglebutton1
% --- Executes on key press with focus on togglebutton1 and none of its
controls.
function togglebutton1_KeyPressFcn(hObject, eventdata, handles)
% hObject    handle to togglebutton1 (see GCBO)
% eventdata  structure with the following fields (see UICONTROL)
%       Key: name of the key that was pressed, in lower case
%       Character: character interpretation of the key(s) that was pressed
%       Modifier: name(s) of the modifier key(s) (i.e., control, shift)
pressed
% handles    structure with handles and user data (see GUIDATA)
set(handles.togglebutton1,'value',1)
disp('Images are same');

% --- Executes on button press in togglebutton2.
function togglebutton2_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of togglebutton2
% --- Executes on key press with focus on togglebutton2 and none of its
controls.
function togglebutton2_KeyPressFcn(hObject, eventdata, handles)
% hObject    handle to togglebutton2 (see GCBO)
% eventdata  structure with the following fields (see UICONTROL)
%       Key: name of the key that was pressed, in lower case
%       Character: character interpretation of the key(s) that was pressed
%       Modifier: name(s) of the modifier key(s) (i.e., control, shift)
pressed
% handles    structure with handles and user data (see GUIDATA)

set(handles.togglebutton2,'value',1);
disp('Images are not same');
```

A

PROJECT REPORT

On

MOBILE CHARGING ON COIN INSERTION

Submitted by

1)Mr. E.PRANEETH REDDY(17K81A04E4)

2)Mr. B.BALAJI (18K85A04E4)

3)Mr.B.VAMSHI KRISHNA(17K81A04D6)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Mr.G.RAMESH

Assistant professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST.MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

St.MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA&NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the project entitled **Mobile Charging On Coin Insertion**, is being submitted by **Mr.E.PRANEETH REDDY-17K81A04E4, Mr.B.BALAJI-18K85A0427, Mr.B.VAMSHI KRISHNA-17K81A04D6** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICSAND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Internal Guide	Head of the Department
Mr. GARLA RAMESH	Dr. B. HARI KRISHNA
Assistant Professor, ECE	Professor & HOD, ECE

Internal Examiner External Examiner

Place:

Date:

LASYA INFOTECH

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **E.PRANEETH REDDY** WITH ROLL NO.**17K81A04E4, B.BALAJI** WITH ROLL NO.**18K85A0427, B.VAMSHI KRISHNA** WITH ROLL NO.**17K81A04D6**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**MOBILE CHARGING ON COIN INSERTION**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT

DIRECTOR

TRAININGS & PLACEMENTS LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014 Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of **'Bachelor of Technology in Department of Electronics and Communication Engineering'**, session: 2017 – 2021, **St. Martin's Engineering College**, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled **'MOBILE CHARGING ON COIN INSERTION'** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

E.PRANEETH REDDY(17K81A04J0)
B.BALAJI (18K85A0427)
B.VAMSHI KRISHNA (17K81A04D6)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr. B. HARI KRISHNA**, **Head &Professor**, the Dept. of Electronics and communication engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our dept. project **coordinator Dr.A.ANAND, Professor** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide, **Mr. G.RAMESH,** Assistant Professor, Department of ECE and our calss project coordinator **Mr.K.NISHKAR**, Professor, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.A.ANAND**, **Professor and Dr. R. SANTHOSH KUMAR,** Associate professor **in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1) E.PRANEETH REDDY

2) B.BALAJI

3) B.VAMSHI KRISHNA

ABSTRACT

This is the smart coin based mobile charging system that charges your mobile for particular amount of time on inserting a coin. The system is to be used by shop owners, public places like railway stations to provide mobile charging facility. So the system consists of a coin recognition module that recognizes valid coins and then signals the microcontroller for further action. If a valid coin is found it signals the microcontroller and microcontroller then starts the mobile charging mechanism providing a 5V supply through a power supply section to the mobile phone, now system also needs to monitor the amount of charging to be provided. So the microcontroller starts a reverse countdown timer to display the charging time for that mobile phone. Now if the user inserts another coin in that time, the microcontroller adds the time to currently remaining charging time and starts the reverse countdown. So the system can be used for smart mobile charging at public places.

TABLE OF CONTENTS

LIST OF FIGURES

TABLE OF TABLES

CHAPTER-1

INTRODUCTION

1.1. Introduction to the project:

The mobile phone market is a vast industry, and has spread into rural areas as an essential means of communication. While the urban populations use more sophisticated mobiles with good power batteries lasting for several days, the rural populations buy the pre-owned mobile phones that require charging frequently. Many times battery becomes flat in the middle of conversation particularly at inconvenient times when access to a standard charger isn't possible. The coin-based mobile battery chargers are designed to solve this problem. The user has to plug the mobile phone into one of the adapters and insert a coin; the phone will then be given a micro pulse for charging. It does not bring a mobile from 'dead' to fully charged state. The charging capacity of the mobile is designed with the help of pre-defined values. It is of course, possible to continue charging the mobile by inserting more coins. This compact and lightweight product is designed to cater for the growing number of rural mobile users worldwide. A suitable microcontroller is programmed for all the controlling applications. The source for charging is obtained from direct power grid.

CHAPTER-2

INTRODUCTION

2.1 INTRODUCTION ABOUT EMBEDDED SYSTEMS:

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.

The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes.

The applications software on such processors is sometimes referred to as firmware.

The simplest devices consist of a single microprocessor (often called a "chip”), which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine.

As the embedded system is the combination of both software and hardware

2.1.1 Block Diagram of Embedded System:

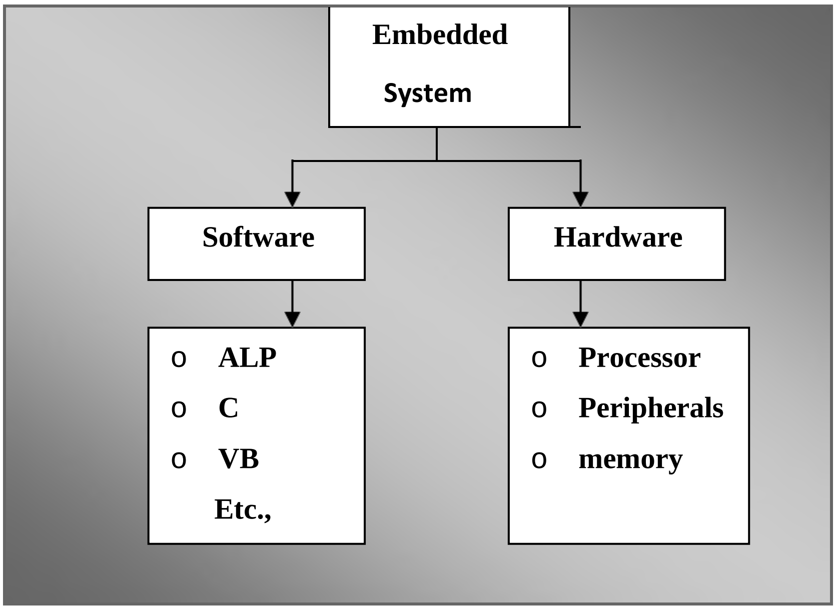


Fig.2.1: Block Diagram of Embedded System

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory

2.1.2 Applications of Embedded Systems:

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

2.2 MICRO PROCESSOR (µP):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Processors are classified into four types like:

- Micro Processor (µp)
- Micro controller (µc)
- Digital Signal Processo1r (DSP)
- Application Specific Integrated Circuits (ASIC)

2.2.1 Three Basic Elements of A Microprocessor:

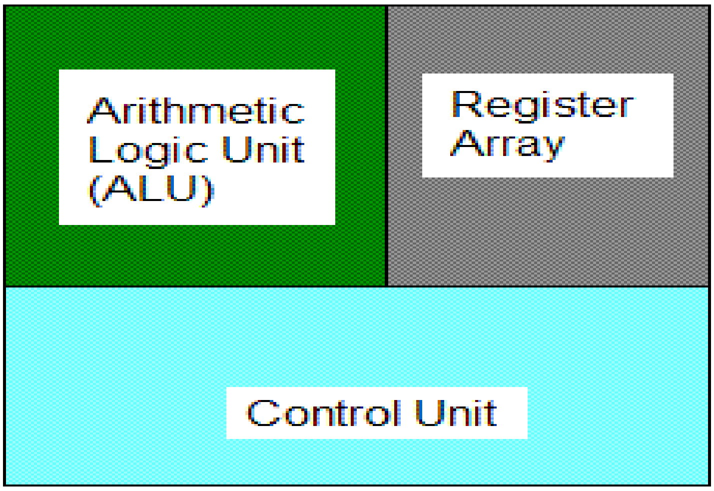


Fig.2.2: Three Basic Elements of a Microprocessor

2.3 HARVARD ARCHITECTURE:

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data.

The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

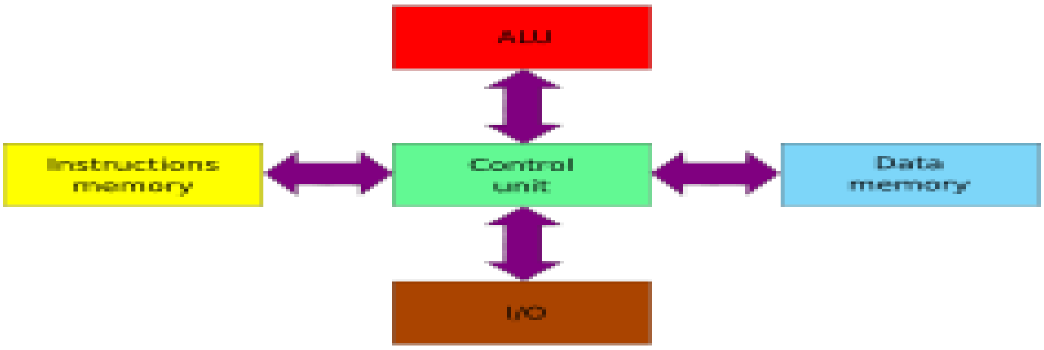


Fig.2.3: Harvard Architecture

2.3.1 Uses of The Harvard Architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

- Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behaviour must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.

- Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access.
- The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

2.4 VON-NEUMANN ARCHITECTURE

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. They cannot be performed at the same time. The von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A stored-program digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

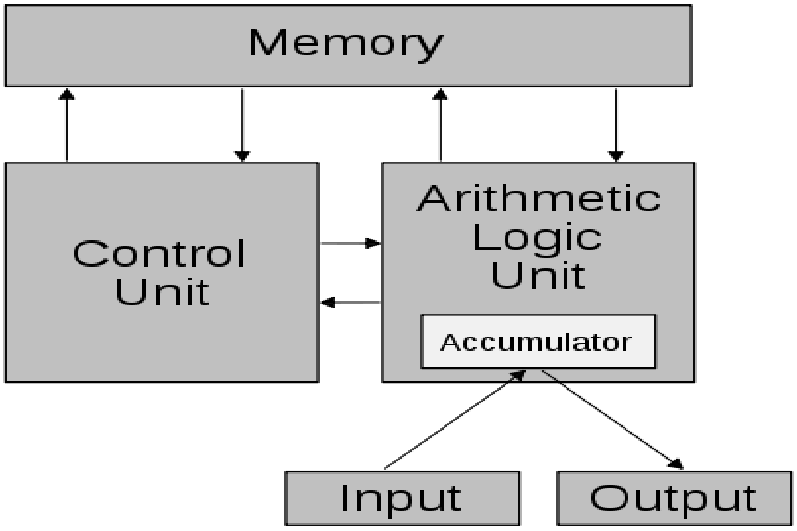


Fig.2.4: Schematic of the Von-Neumann Architecture

CHAPTER-3

LITERATURE SURVEY AND EXISTING METHODS

3.1 LITERATURE SURVEY

In 2017, Dhara G. Rangani, Nikunj V. Tahilramani [1] have presented mobile charging using coin in which their main focus was coin detection for which they have used cantilever type sensor for coin detection. Cantilever type sensor detects weight of 5 rupee coin and gives digital signal to ADC. Using this controller check whether coin is original or duplicate. They have also used solar power for charging the mobile station battery and used greed power when solar power is not available.

In May 2017, Mr. C V Raja Reddy, Uzoigwe Daniel, Rupesh Rai, Balaji R [2] have proposed coin based mobile charging with solar tracking in which their main focus was solar tracking for which they have used LDRs so according to the sunlight intensity LDR resistance will be varied. When the sun intensity is high then LDR offers less resistance, the voltage across each LDR is given to the ADC, then controller checks in accordance with the algorithm designed and rotates the motor in specified direction.

In 2015, Nupur Khera [3] have presented a solution to improve the charging and discharging control of battery. The solar charge controller will prevent the overcharging of the battery thus it will help to increase the life of battery. Solar charge controller will also prevent reverse flow of current from the batteries to the solar panels at night.

In 2013, S. B. Sridevi [4] have provided a solution to solar tracking. As sun rises in the east and sets in the west, So if the system cannot change the direction towards sun then it will not be able to consume maximum sunlight this makes the system inefficient.

3.2 EXISTING METHODS

The existing system in now a days is a payphone (alternative spelling: pay phone) is typically a coin-operated public telephone, often located in a telephone booth or in high-traffic outdoor areas, with pre-payment by inserting money (usually coins) or by billing a credit or debit card, or a telephone card. Prepaid calling cards also facilitate establishing a call by first calling the provided toll-free telephone number, entering the card account number and PIN, then the desired telephone number. An equipment usage fee may be charged as additional units, minutes or tariff fee to the collect/third-party, debit, credit, telephone or prepaid calling card when used at payphones. By agreement with the landlord, either the phone company pays rent for the location and keeps the revenue, or the landlord pays rent for the phone and shares the revenue.

Payphones are often found in public places to contribute to the notion of universal access to basic communication services.In the late 1920's the cost of a pay phone call was a paltry two cents. The 1930's calls were a nickel. By the time pay phones became extinct, the price of a call was fifty cents. [1] One thesis, written as early as 2003, recognised this as a digital divide problem.

In the 20th century, payphones in some countries, such as Spain, used token coins, available for sale at a local retailer, to activate pay phones, instead of legal tender coins. In some cases, these were upgraded to use magnetic cards or credit card readers over the years.

In the past, payphones were ubiquitous worldwide, but their prevalence has decreased significantly over the years due to the increasing availability of mobile phones.

3.3 PROPOSED SYSTEM

This proposed system will charge the mobile phone by inserting coin. For coin insertion we are using coin insertion module which will detect the inserted coin, if appropriate coin is detected then it will give the digital signal to the microcontroller otherwise it will refund the coin. Then the microcontroller will ON the respective relay for specific time interval according to the algorithm. This proposed system will not charge the mobile full from 0% to 100% but it will charge according to number of coins inserted. If we add more coins then time will also get added to the charging and reverse count down time will get displayed on LCD. This proposed system will also use the solar energy for charging the battery at mobile charging station and uses AC power supply as secondary source if sun is not available. We use solar tracking system for maximum utilization of solar energy using LDRs. As we are using 12V battery for storing the solar energy so to protect the battery from over voltage we are using charge controller circuit. If battery voltage exceeds 13.5V then charge controller will automatically cut off the supply to the battery and hence battery will get saved from damage to over voltage.

CHAPTER-4

PROJECT DISCRIPTION

4.1Block diagram:

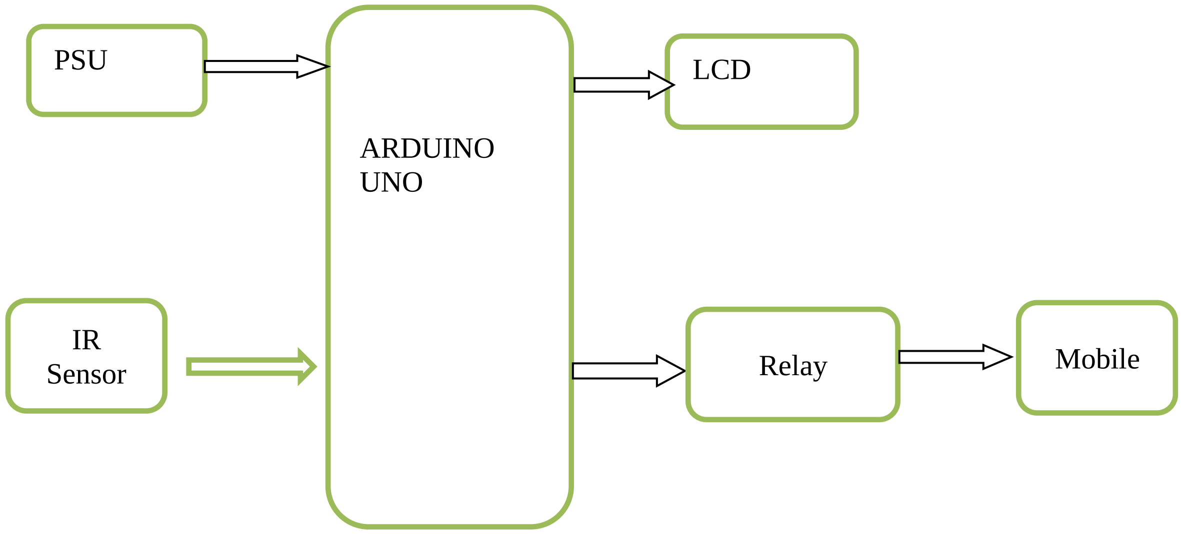


Fig.4.1: Block diagram of the project

4.1.1Components Required

Hardware Requirement:

- Arduino Uno
- Coin Sensor
- DC Regulation
- Coin Module
- LCD
- Power Supply

Software Components:

- Arduino IDE
- Proteous

Arduino Uno

- Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software.

- It consists of a circuit board, which can be programed (referred to as a microcontroller) in a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.
- An Arduino has 17 different kind of pins which have their own operation and every pin has its own importance equally with others.

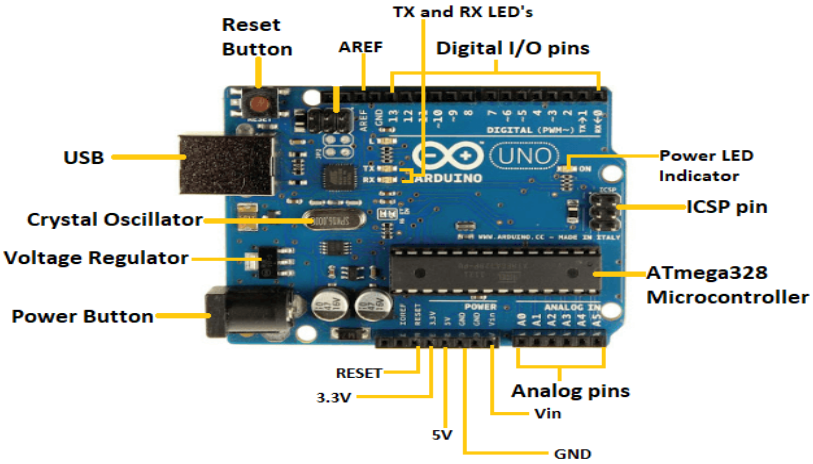


Fig.4.2: Aurdino Uno

Coin Module

- The coin acceptor device accepts the coin through coin insertion slot.
- The sensor inside the coin acceptor device will recognizeand validate the coin based on the coin diameter.
- When the inserted coin is validated, it sends signal to the Arduino for powersupply otherwise the coin will be returned back.

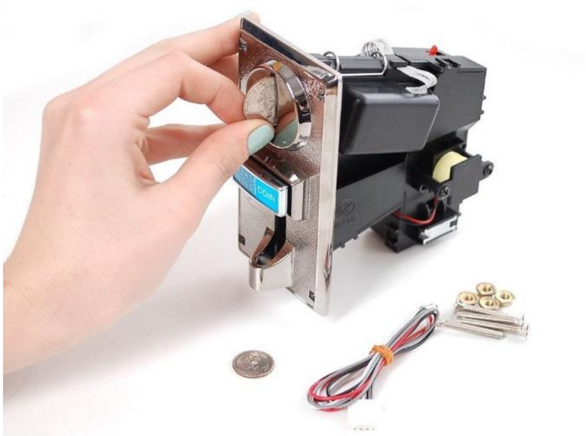


Fig.4.3: Coin Module

IR Sensor

- An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment.
- When an object comes close to the sensor, the infrared light from the LED reflects off of the object and is detected by the receiver

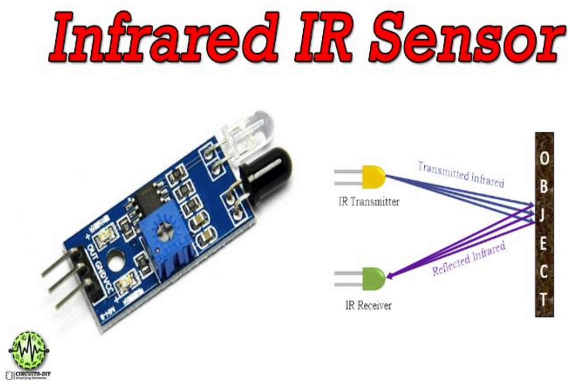


Fig.4.4: IR Sensor

Relay

- A relay is an electrically operated switch.
- Relay controlone electrical circuit by opening and closing contacts in another circuit.
- As relay diagrams show, when a relay contact is normally open (NO), there is an open contact then the relay is not energized.
- When a relay contact is Normally Closed (NC), there is a closed contact then the relay is energized.
- Relays are used throughout the automobile.
- Applications of relay are used as remote control switches.
- A typical vehicle can have 20 relays or more

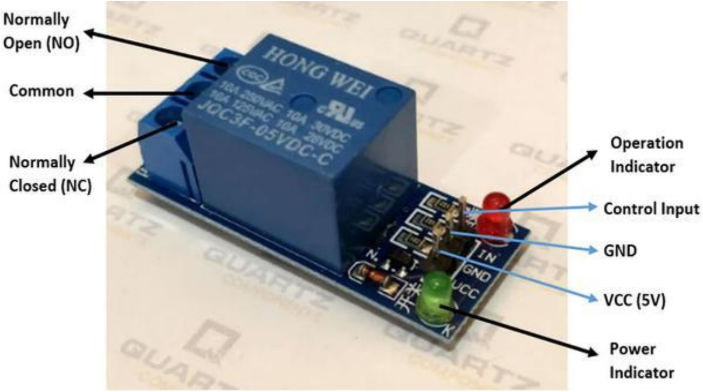


Fig.4.5: Relay

Liquid Crystal Display(LCD)

- A liquid-crystal display is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers.
- Liquid crystals do not emit light directly,instead using a backlight or reflector to produce images in color .

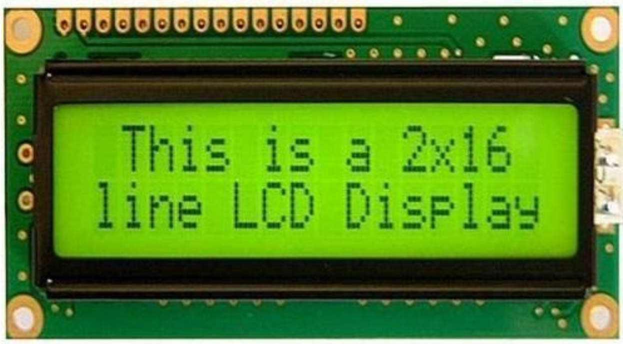


Fig.4.6: Liquid Crystal Display(LCD)

Buzzar

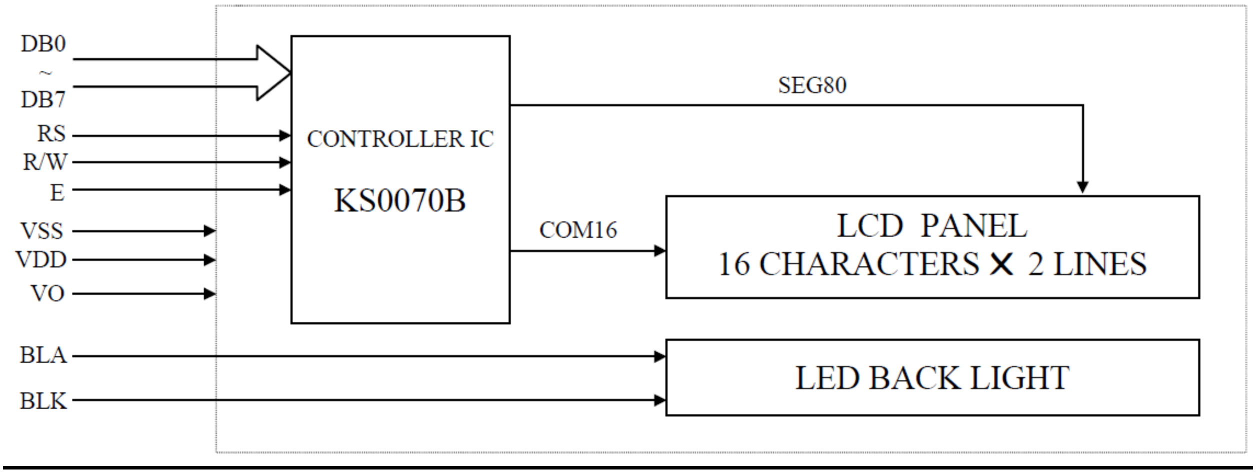
- A buzzer or beeper is an audio signalling device.
- which may be mechanical, electromechanical, or piezoelectric.
- Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

4.2 16 * 2 Alphanumeric LCD

4.2.1 Description

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

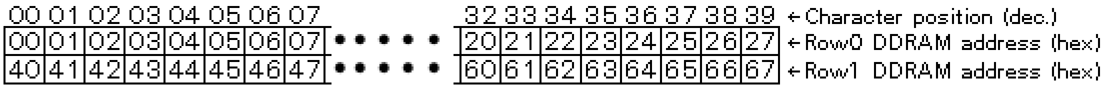
4.2.2 Block Diagram



4.2.3 Display Data RAM

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.



DDRAM Address for 2 Line LCD

4.2.4 Character Generator ROM

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. Userdefined character patterns are also available by mask-programmed ROM.

4.2.5 Busy Flag

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact ammount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

4.2.6Instruction Register(IR) and Data Register(DR)

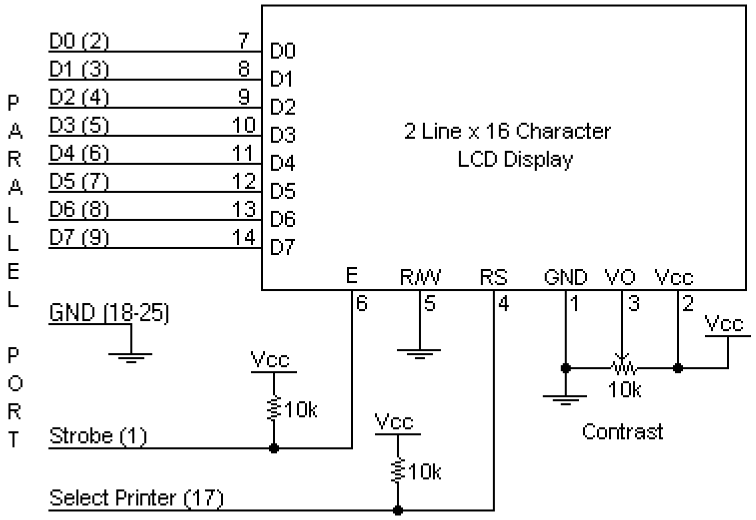
There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD. Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

4.2.7 16 x 2 Alphanumeric LCD Module Features

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line

- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

4.3 Schematic

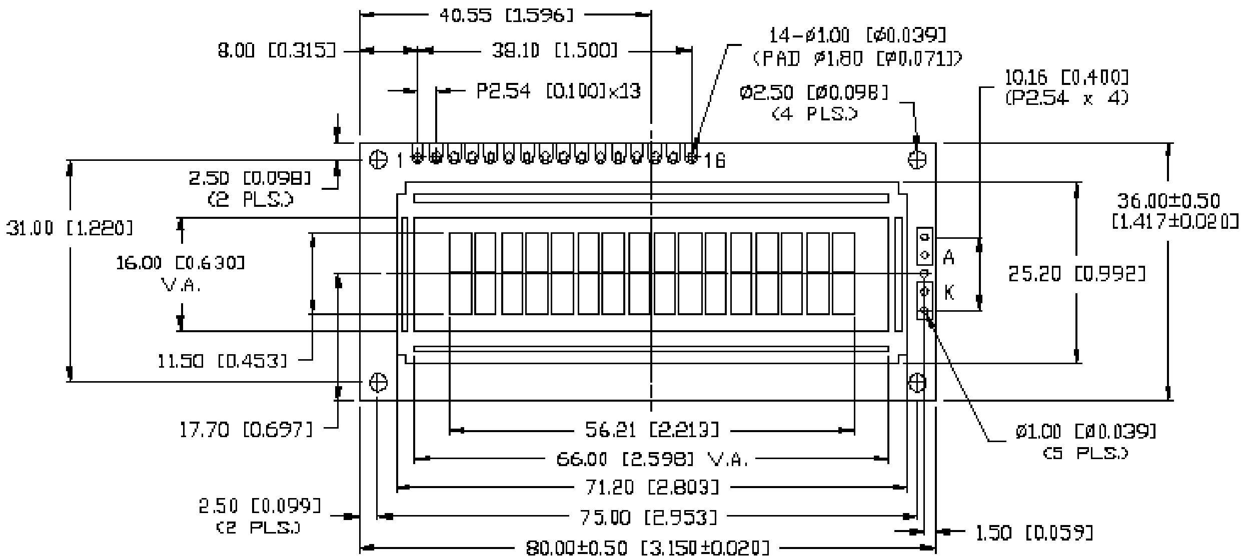


4.4 Specifications

Connector Pin Assignment:

Pin	*Symbol*	*Function*	*Pin*	*Symbol*	*Function*
14	DB7	Data Bus Line	6	E	Enable
13	DB6	Data Bus Line	5	R/W	Read/Write
12	DB5	Data Bus Line	4	RS	Register select input
11	DB4	Data Bus Line	3	Vcontrast	Contrast (0V for max contrast)
10	DB3	Data Bus Line	2	**GND (0V)**	Power supply
9	DB2	Data Bus Line	1	**Vcc +5V**	Power supply
8	DB1	Data Bus Line			
7	DB0	Data Bus Line			

Note1: Pin 1 is +5V and pin 2 is GND! This is different from most other HD44780 compatible models.
Note2: Pin 14 is marked with a "1" on the back of the PCB. Do not get confused by this.



4.5 Circuit Description

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

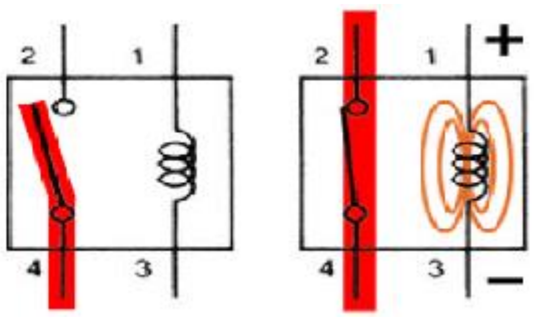
4.6 IR TRANSMITTER & RECEIVER:

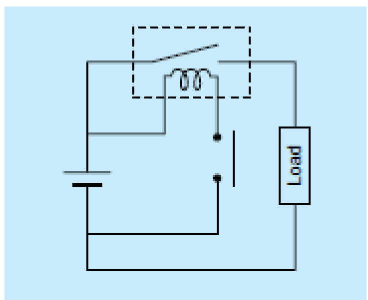
- IR stands infrared red spectrum, by using this region we can detect numbert of obstacles passing through a region.
- IR consists of two sections one is transmitter section and other is receiver section.

- Transmitter section:it consists of a LED which continuosly transmits a modulated signalof range(40 to 250khz).
- Receiver section: Receiver section consists of a photo diode it will continuosly track and receive the ir rays from LED
- Functionality: The IR transmitter will continuosly transmit IR rays and receiver (photo diode) will continuosly receive the rays whenever some obstacle passes through the IR pair the tansmission will be breaked by which we come to know that an intruder have been detected. In this way this communication will help us in safeguarding and many other applications.

4.7 Relays

A relay is an electrically operated switch. These are remote control electrical switches that are controlled by another switch, such as a horn switch or a computer as in a power train control module, devices in industries, home based applications. Relays allow a small current pin, 4-pin, 5-pin, and 6-pin, single switch or dual switches. Relays are used throughout the automobile. Relays which come in assorted sizes, ratings, and applications, are used as remote control switches. A typical vehicle can have 20 relays or more.





4.7.1 BASICS ON RELAY HANDLING

- To maintain initial performance, care should be taken to avoid dropping or hitting the relay.
- Under normal use, the relay is designed so that the case will not detach. To maintain initial performance, the case should not be removed. Relay characteristics cannot be guaranteed if the case is removed.
- Use of the relay in an atmosphere at standard temperature and humidity with minimal amounts of dust, SO 2, H 2 S, or organic gases is recommended.
- Please avoid the use of silicon-based resins near the relay, because doing so may result in contact failure. (This applies to plastic sealed type relays, too.)
- Care should be taken to observe correct coil polarity (+, –) for polarized relays.
- Proper usage requires that the rated voltage be impressed on the coil. Use rectangular waves for DC coils and sine waves for AC coils.
- Be sure the coil impressed voltage does not continuously exceed the maximum allowable voltage.
- Absolutely avoid using switching voltages and currents that exceed the designated values.
- The rated switching power and life are given only as guides. The physical phenomena at the contacts and contact life greatly vary depending on the type of load and the operating conditions. Therefore, be sure to carefully check the
- type of load and operating conditions before use.
- Do not exceed the usable ambient temperature values listed in the catalog.
- Use the flux-resistant type or sealed type if automatic soldering is to be used.
- Use alcohol based cleaning solvents when cleaning is to be performed using a
sealed type relay.
- Avoid ultrasonic cleaning of all types of relays.
- Avoid bending terminals, because it may cause malfunction.
- As a guide, use a Faston mounting pressure of 40 to 70N {4 to 7kgf}for relay with tab terminals.

A relay is used to isolate one electrical circuit from another. It allows a low current control circuit to make or break an electrically isolated high current circuit path. The basic relay consists of a coil and a set of contacts. The most common relay coil is a length of magnet wire wrapped around a metal core. When voltage is applied to the coil, current passes through the wire and creates a magnetic field. This magnetic field pulls the contacts together and holds them there until the current flow in the coil has stopped. The diagram below shows the parts of a simple relay.

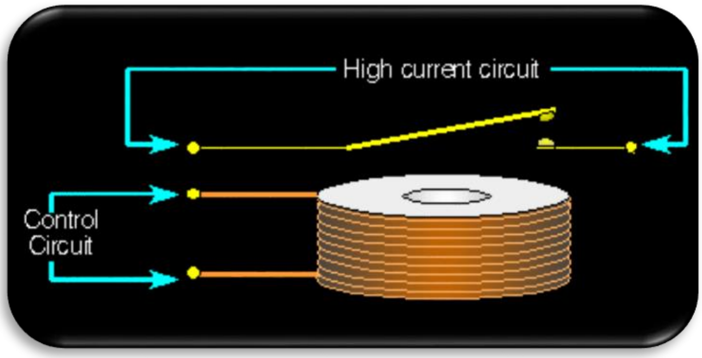


Fig.4.7: Relay

4.7.2 Operation:

When a current flows through the coil, the resulting magnetic field attracts an armature that is mechanically linked to a moving contact. The movement either makes or breaks a connection with a fixed contact. When the current is switched off, the armature is usually returned by a spring to its resting position shown in figure 6.6(b). Latching relays exist that require operation of a second coil to reset the contact position.

By analogy with the functions of the original electromagnetic device, a solid-state relay operates a thyristor or other solid-state switching device with a transformer or light-emitting diode to trigger it.

4.7.3 Pole and throw

Since relays are switches the terminology applied to switches is also applied to relays. A relay will switch one or more *poles*, each of whose contacts can be *thrown* by energizing the coil in one of three ways:

- Normally-open (**NO**) contacts connect the circuit when the relay is activated; the circuit is disconnected when the relay is inactive. It is also called a **Form A** contact or "make" contact.
- Normally-closed (**NC**) contacts disconnect the circuit when the relay is activated; the circuit is connected when the relay is inactive. It is also called a **Form B** contact or "break" contact.
- Change-over (**CO**), or double-throw (**DT**), contacts control two circuits: one normally-open contact and one normally-closed contact with a common terminal. It is also called a **Form C** contact or "transfer" contact ("break before make"). If this type of contact utilizes a "make before break" functionality, then it is called a **Form D** contact.

4.7.4 SPST

SPST relay stands for Single Pole Single Throw relay. Current will only flow through the contacts when the relay coil is energized.

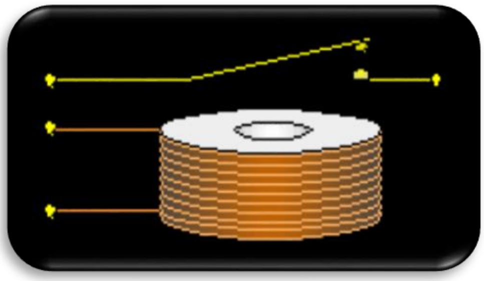


Fig.4.8: SPST Relay

4.7.5 SPDT Relay

SPDT Relay stands for Single Pole Double Throw relay. Current will flow between the movable contact and one fixed contact when the coil is De-energized and between the movable contact and the alternate fixed contact when the relay coil is energized. The most commonly used relay in car audio, the Bosch relay, is a SPDT relay.

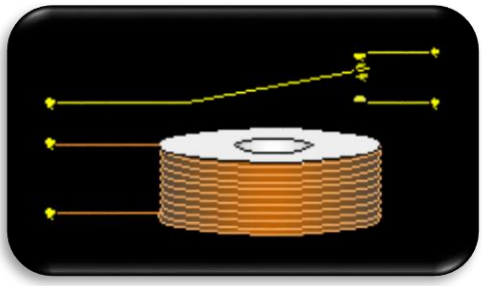


Fig4.9: SPDT Relay

4.7.6 DPST Relay

DPST relay stands for Double Pole Single Throw relay. When the relay coil is energized, two separate and electrically isolated sets of contacts are pulled down to make contact with their stationary counterparts. There is no complete circuit path when the relay is De-energized.

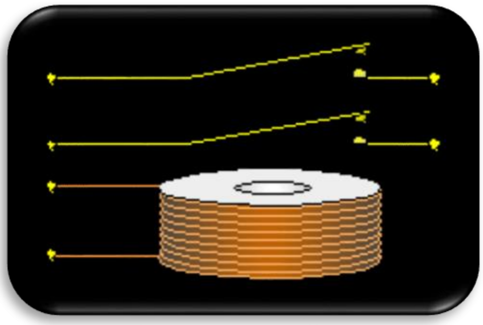


Fig.4.10: DPST Relay

4.7.7 DPDT Relay

DPDT relay stands for Double Pole Double Throw relay. It operates like the SPDT relay but has twice as many contacts. There are two completely isolated sets of contacts.

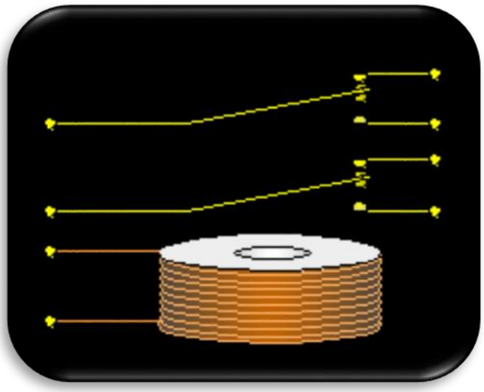


Fig.4.11: DPDT Relay

This is a 4 Pole Double Throw relay. It operates like the SPDT relay but it has 4 sets of isolated contacts.

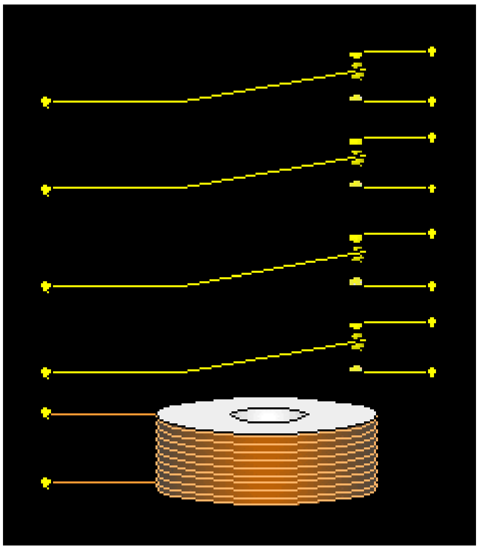


Fig.4.12: 4 Pole Double Throw relay

4.8 Types of relay:

- Latching Relay
- Reed Relay
- Mercury Wetted Relay
- Machine Tool Relay
- Solid State Relay (SSR)

4.8.1 Latching relay

Latching relay, dust cover removed, showing pawl and ratchet mechanism. The ratchet operates a cam, which raises and lowers the moving contact arm, seen edge-on just below it. The moving and fixed contacts are visible at the left side of the image.

A **latching relay** has two relaxed states (bi-stable). These are also called "impulse", "keep", or "stay" relays. When the current is switched off, the relay remains in its last state. This is achieved with a solenoid operating a ratchet and cam mechanism, or by having two opposing coils with an over-center spring or permanent magnet to hold the armature and contacts in position while the coil is relaxed, or with a remanent core. In the ratchet and cam example, the first pulse to the coil turns the relay on and the second pulse turns it off. In the two coil example, a pulse to one coil turns the relay on and a pulse to the opposite coil turns the relay off. This type of relay has the advantage that it consumes power only for an instant, while it is being switched, and it retains its last setting across a power outage. A remanent core latching relay requires a current pulse of opposite polarity and change state.

Fig.4.13: Latching

4.8.2 Reed relay

A reed relay has a set of contacts inside a vacuum or inert gas filled glass tube, which protects the contacts against atmospheric corrosion. The contacts are closed by a magnetic field generated when current passes through a coil around the glass tube. Reed relays are capable of faster switching speeds than larger types of relays, but have low switch current and voltage ratings.

4.8.3 Mercury-wetted relay

A mercury-wetted reed relay is a form of reed relay in which the contacts are wetted with mercury. Such relays are used to switch low-voltage signals (one volt or less) because of their low contact resistance, or for high-speed counting and timing applications where the mercury eliminates contact bounce. Mercury wetted relays are position-sensitive and must be mounted vertically to work properly. Because of the toxicity and expense of liquid mercury, these relays are rarely specified for new equipment. See also mercury switch.

4.8.4 Machine tool relay

A machine tool relay is a type standardized for industrial control of machine tools, transfer machines, and other sequential control. They are characterized by a large number of contacts (sometimes extendable in the field) which are easily converted from normally-open to normally-closed status, easily replaceable coils, and a form factor that allows compactly installing many relays in a control panel. Although such relays once were the backbone of automation in such industries as automobile assembly, the programmable logic controller (PLC) mostly displaced the machine tool relay from sequential control applications.

4.8.5 Solid-state relay

A solid state relay (SSR) is a solid state electronic component that provides a similar function to an electromechanical relay but does not have any moving components, increasing long-term reliability. With early SSR's, the tradeoff came from the fact that every transistor has a small voltage drop across it. This voltage drop limited the amount of current a given SSR could handle. As transistors improved, higher current SSR's, able to handle 100 to 1,200 Amperes, have become commercially available. Compared to electromagnetic relays, they may be falsely triggered by transients.

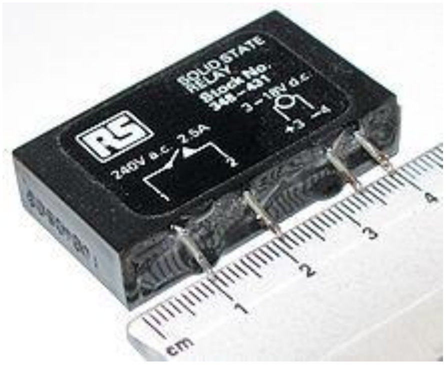


Fig.4.14: Solid relay, which has no moving parts

4.9 Specification

- Number and type of contacts – normally open, normally closed, (double-throw)
- Contact sequence – "Make before Break" or "Break before Make". For example, the old style telephone exchanges required Make-before-break so that the connection didn't get dropped while dialing the number.
- Rating of contacts – small relays switch a few amperes, large contactors are rated for up to 3000 amperes, alternating or direct current
- Voltage rating of contacts – typical control relays rated 300 VAC or 600 VAC, automotive types to 50 VDC, special high-voltage relays to about 15 000 V
- Coil voltage – machine-tool relays usually 24 VAC, 120 or 250 VAC, relays for switchgear may have 125 V or 250 VDC coils, "sensitive" relays operate on a few milli-amperes

4.10Applications:

Relays are used:

- To control a high-voltage circuit with a low-voltage signal, as in some types of modems.

- To control a high-current circuit with a low-current signal, as in the starter solenoid of an automobile.
- To detect and isolate faults on transmission and distribution lines by opening and closing circuit breakers (protection relays),
- To isolate the controlling circuit from the controlled circuit when the two are at different potentials, for example when controlling a mains-powered device from a low-voltage switch. The latter is often applied to control office lighting as the low voltage wires are easily installed in partitions, which may be often moved as needs change. They may also be controlled by room occupancy detectors in an effort to conserve energy,
- To perform logic functions. For example, the boolean AND function is realized by connecting relay contacts in series, the OR function by connecting contacts in parallel. Due to the failure modes of a relay compared with a semiconductor, they are widely used in safety critical logic, such as the control panels of radioactive waste handling machinery.
- As oscillators, also called vibrators. The coil is wired in series with the normally closed contacts. When a current is passed through the relay coil, the relay operates and opens the contacts that carry the supply current. This stops the current and causes the contacts to close again. The cycle repeats continuously, causing the relay to open and close rapidly. Vibrators are used to generate pulsed current.
- To generate sound. A vibrator, described above, creates a buzzing sound because of the rapid oscillation of the armature. This is the basis of the electric bell, which consists of a vibrator with a hammer attached to the armature so it can repeatedly strike a bell.
- To perform time delay functions. Relays can be used to act as an mechanical time delay device by controlling the release time by using the effect of residual magnetism by means of a inserting copper disk between the armature and moving blade assembly.

CHAPTER-5

HARDWARE AND SOFTWARE COMPONENTS

MICRO CONTROLLER UNIT

5.1 ATMEGA328

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 Analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The ATmega8 microcontroller contains 32 general purpose working registers. As shown in the below figure these registers are directly connected to ALU. Two registers can carry one single instruction consequently in one clock cycle.

Specifications:

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 Ma
DC Current on 3.3V Pin	50 Ma
Flash Memory	32 KB (0.5 KB is used for Boot loader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

Table.5.1: Atmega328 Specifications

5.1.1Arduino:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

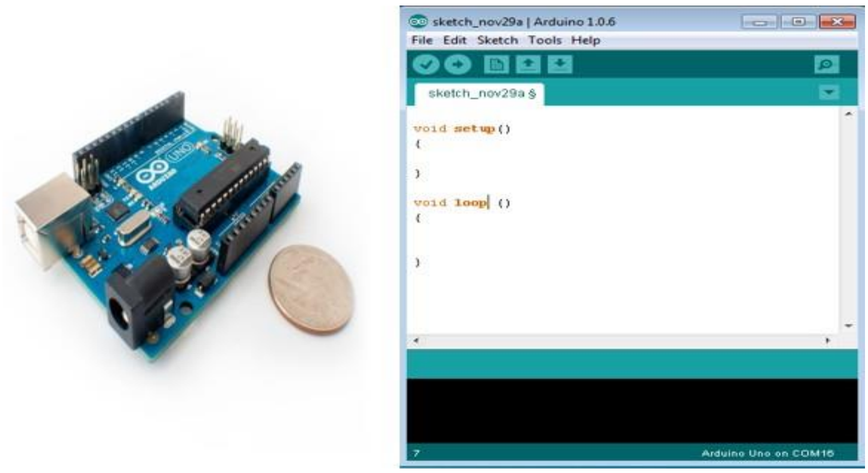


Fig.5.1: Arduino Uno

5.1.2 Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

ATMEGA328:

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header

Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compati

LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table.5.2: Arduino Boards Based On ATMEGA328 Microcontroller

ATMEGA32u4

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table.5.3: Arduino Boards Based On ATMEGA32u4 Microcontroller

ATMEGA2560

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16 U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table.5.4: Arduino Boards Based On ATMEGA2560 Microcontroller

AT91SAM3X8E

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table.5.5: Arduino Boards Based On AT91SAM3X8E Microcontroller

5.1.3 Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

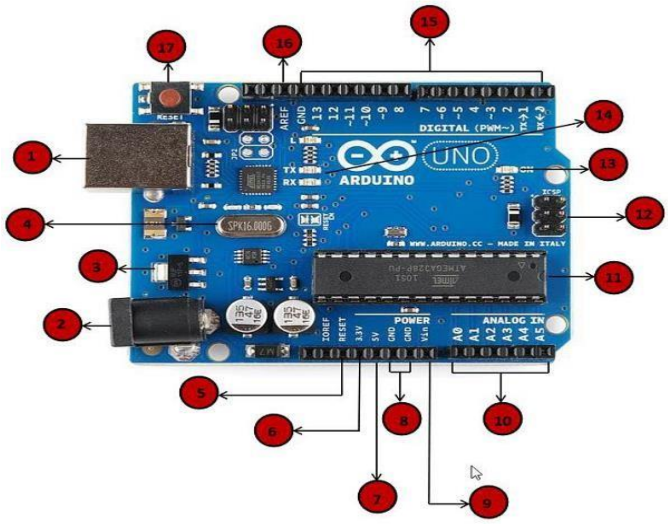


Fig.5.2: Arduino Uno Board Pins

Pin number and Pin Use

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC.

	For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Table.5.6: Pin Explanation of Arduino

5.1.4Arduino Family

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

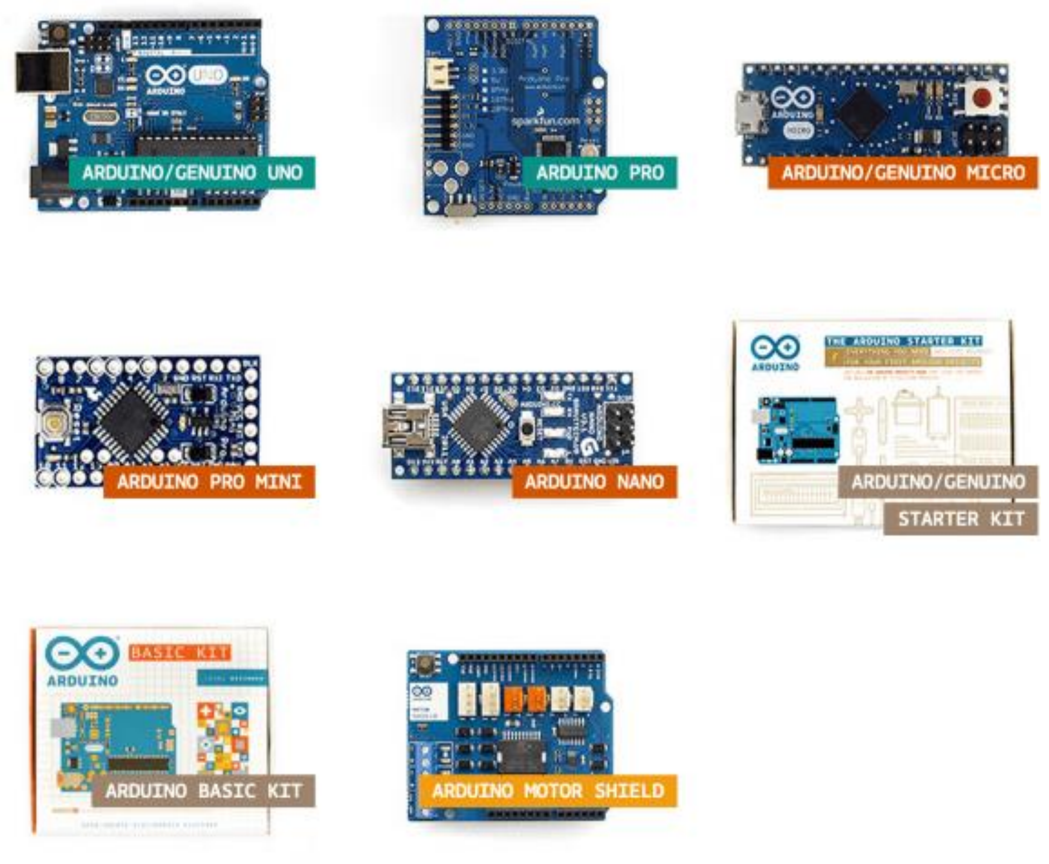


Fig.5.3: Arduino Family

5.1.5 Shields

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

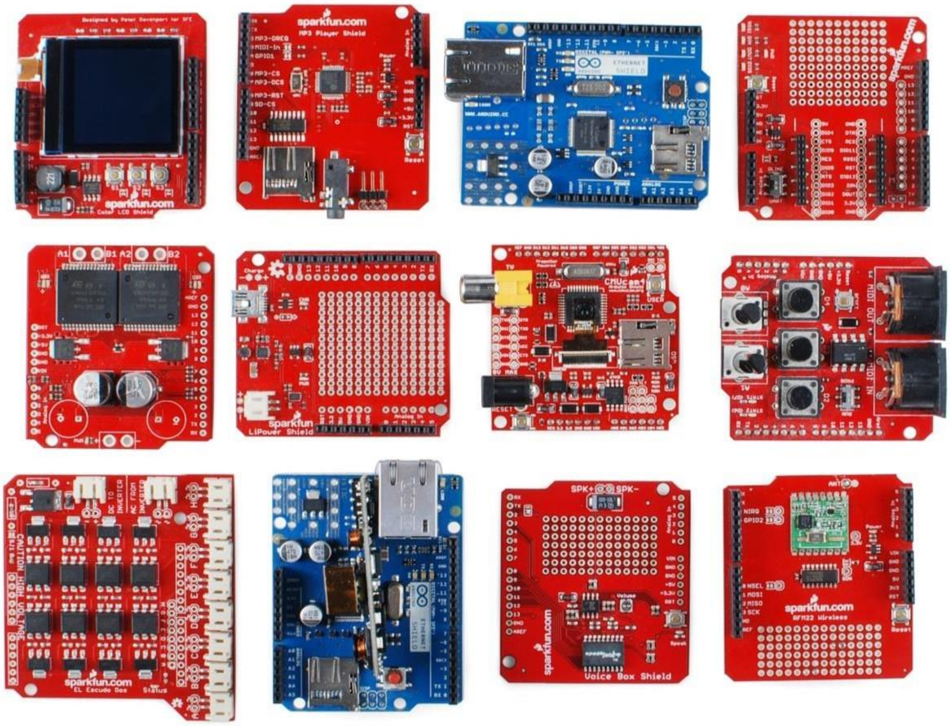


Fig.5.4: Arduino Shields

5.1.6 Pin Description Of Atmega328

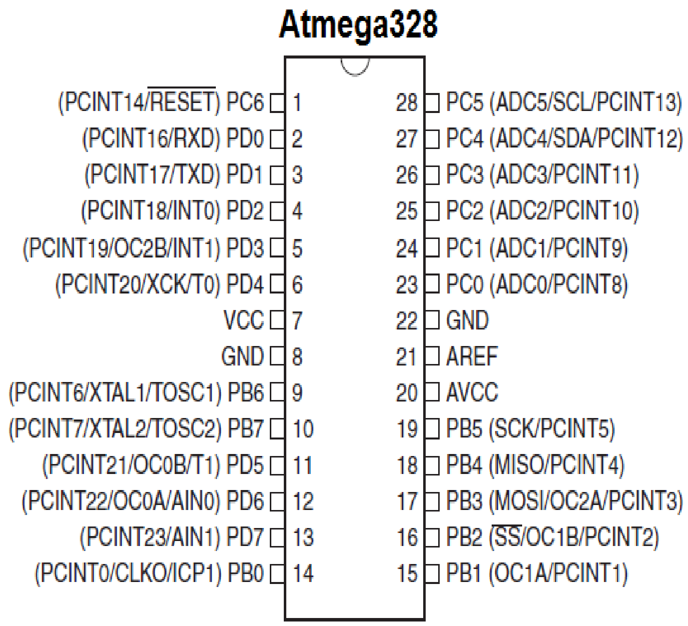


Fig.5.5: Pin Description Of ATMEGA328

5.1.7 Advantages Of Arduino

- It is cheap
- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
- For beginners, it is very simple to use.

5.1.8 Applications

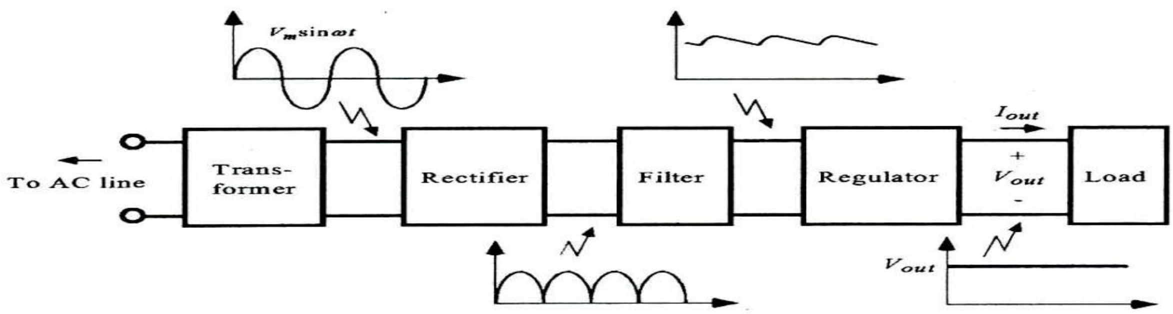
ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

5.2 POWER SUPPLY UNIT

5.2.1 Introduction:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices.A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as “Regulated D.C Power Supply”.

Block Diagram Of Power Supply:



Components of a typical linear power supply

Fig.5.6: (a) Block Diagram of Power Supply

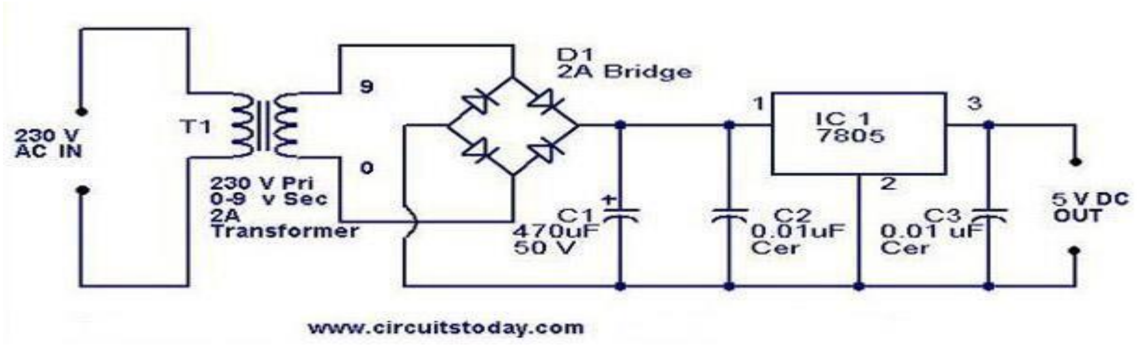


Fig.5.6: (b) Schematic Diagram of Power Supply

Description Of Power Supply:

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a different form of energy – such as solar, mechanical, or chemical - into electrical energy.

A power supply provides components with electric power. The term usually pertains to devices integrated. For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

5.2.2Transformer:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

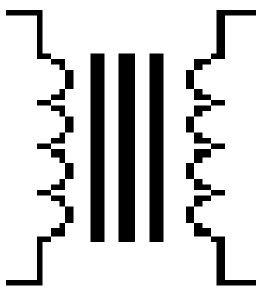


Fig.5.7: (a) Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a tranducer

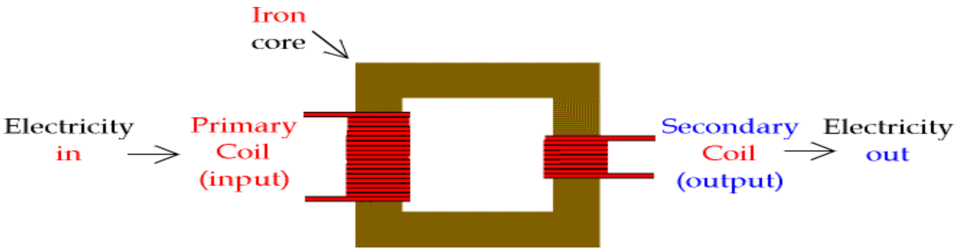


Fig.5.7: (b) Transformer

- **Basic Principle Of Transformer:**

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

From Faraday's Law

$$\frac{V_S}{V_P} = \frac{N_S}{N_P}$$

For ideal transformer

The voltage ratio is equal to the turns ratio, and power in equals power out.

From conservation of energy

$$P_P = V_P I_P = V_S I_S = P_S$$

- **Working Of Transformer:**

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below.There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

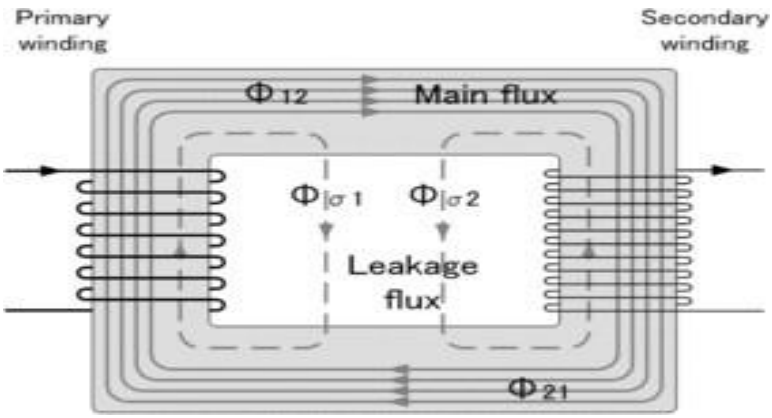


Fig.5.8: Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced EMF. The iron core is laminated to prevent 'eddy currents' flowing in the core.

These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1.They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2.Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

- **Classification Of Transformer:**

  - Step-Up Transformer
  - Step-Down Transformer

Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltageapplied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down

transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn's ratio of the two sets of windings determines the amount of voltage transformation.

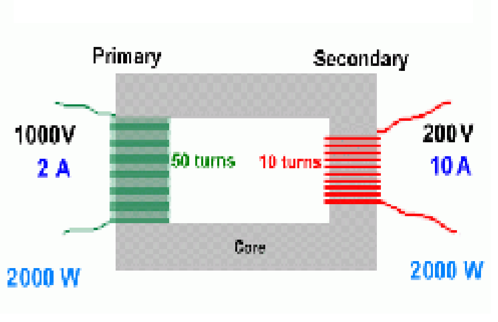


Fig.5.9: (a) Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated.Because of this it is possible to do some non-standard applications using standard transformer.single phase steps down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase steps up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.

- **Step-Up Transformer:**

A step up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply. A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

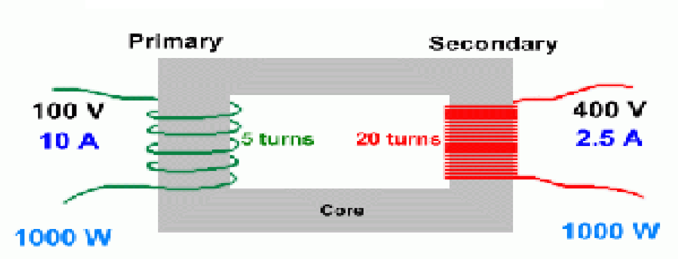


Fig.5.9: (b) Step-Up Transformer

- **Diodes:**

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valve.

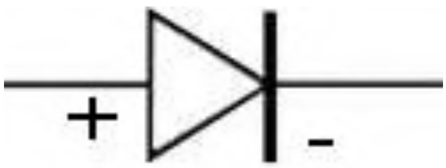


Fig.5.10: Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

- **Rectifier:**

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

- o **The Half-Wave Rectifier:**

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure.

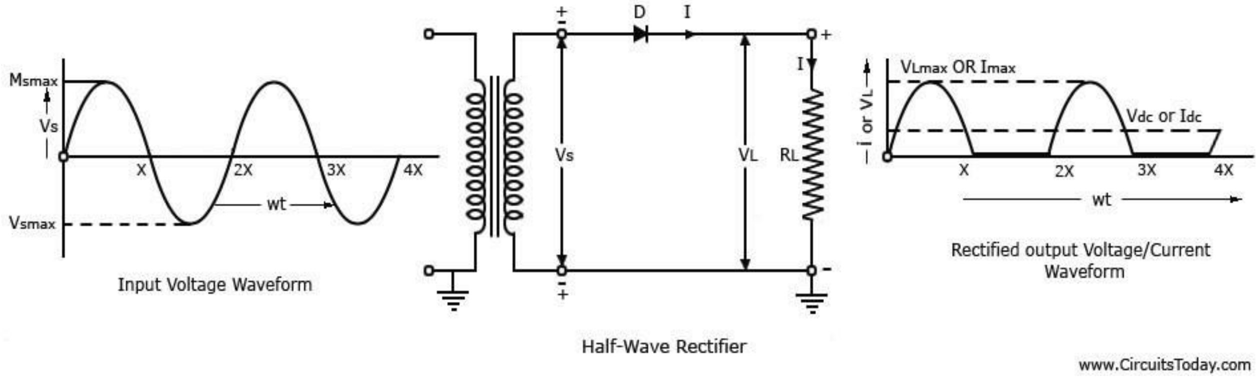


Fig.5.11: (a) Half Wave Rectifier

Figure 2 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

- o **The Full-Wave Rectifier:**

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC.

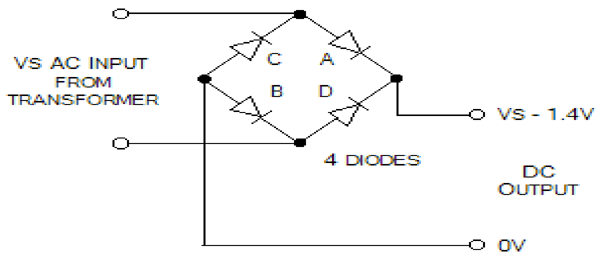


Fig.5.11: (b) Full Wave Rectifier

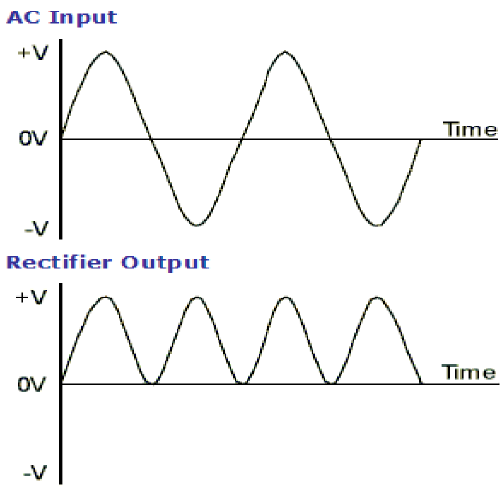


Fig.5.11: (c) Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased. While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

- **Capacitor Filter:**

The capacitor-input filter, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

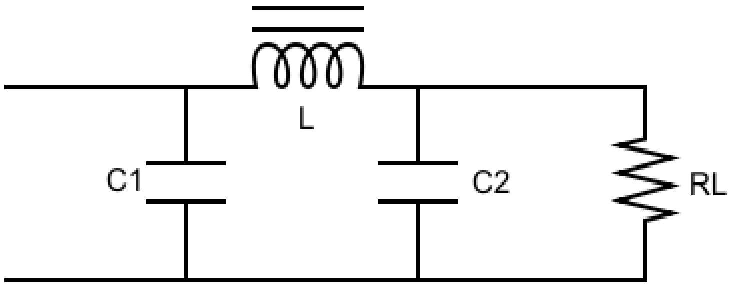


Fig.5.12: Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result, the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result, only the DC component appears across the load RL.

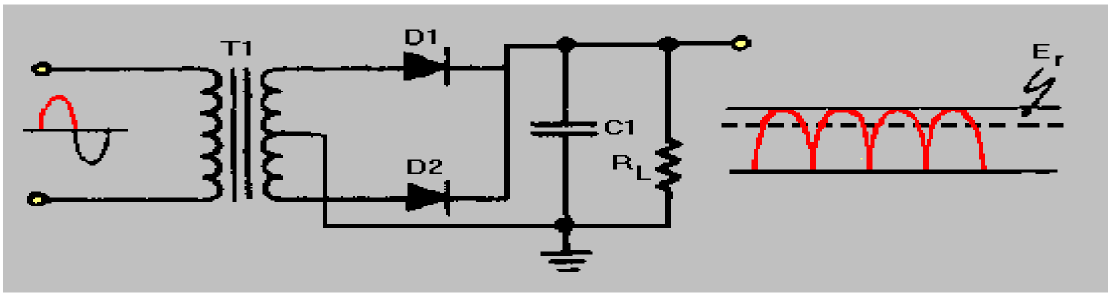


Fig.5.13: Centered Tapped Full-Wave Rectifier with a Capacitor Filter

Voltage Regulator:

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages.

There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:’78’ indicate the positive series and ‘xx’ indicates the voltage rating. Suppose 7805 produces the maximum 5V.’05’indicates the regulator output is 5V.

79xx:’78’ indicate the negative series and ‘xx’ indicates the voltage rating. Suppose 7905 produces the maximum -5V.’05’indicates the regulator output is -5V.

These regulators consist the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

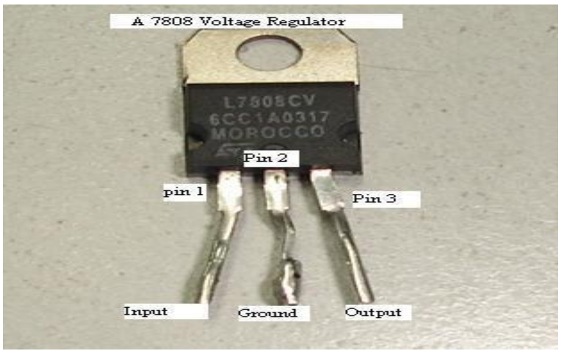


Fig.5.14: Regulator

5.3 SOFTWARE EXPLANATION:

- **Software Requirements**
  - Proteus simulation
  - Arduino software
  - Programming language

5.3.1Arduino software:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IOT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

5.3.2 Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux.

Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments.Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handy board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you
want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version
- Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money
- Getting Started with Arduino and Genuino products:-

Install the Arduino Software (IDE) on Windows PCs-

…This document explains how to install the Arduino Software (IDE) on Windows machines.

- Download the Arduino Software(IDE)
- Proceed with board specific instructions.

5.3.3How to Download the Arduino Software (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

- **Installation:**

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1

First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following imag.

e.

s

Fig.5.15: (a) Standard USB Cable

Fig.5.15: (b) USB cable A to Mini-B

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

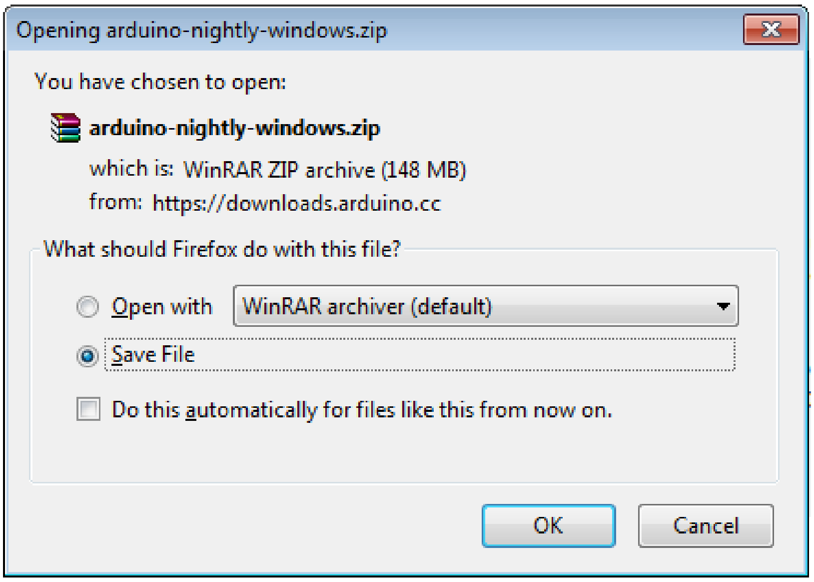


Fig.5.16: Process of Downloading the Arduino

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply.

If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe).

Double-click the icon to start the IDE.

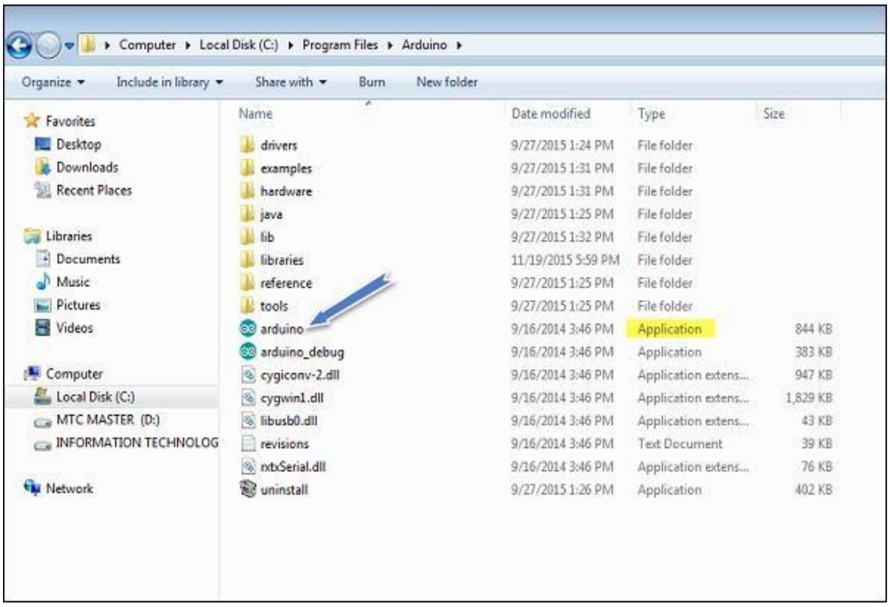


Fig.5.17: Launching Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.

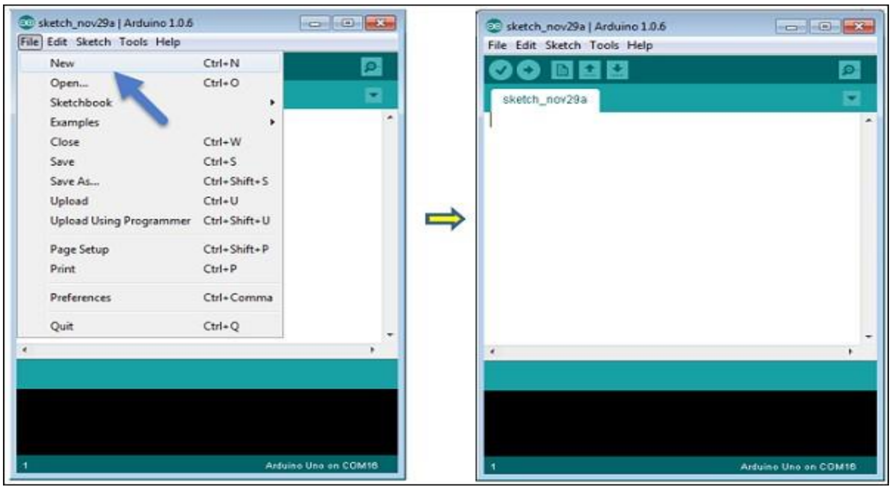


Fig.5.18: Opening Editor Window in IDE

To open an existing project example, select File → Example → Basics → Blink.

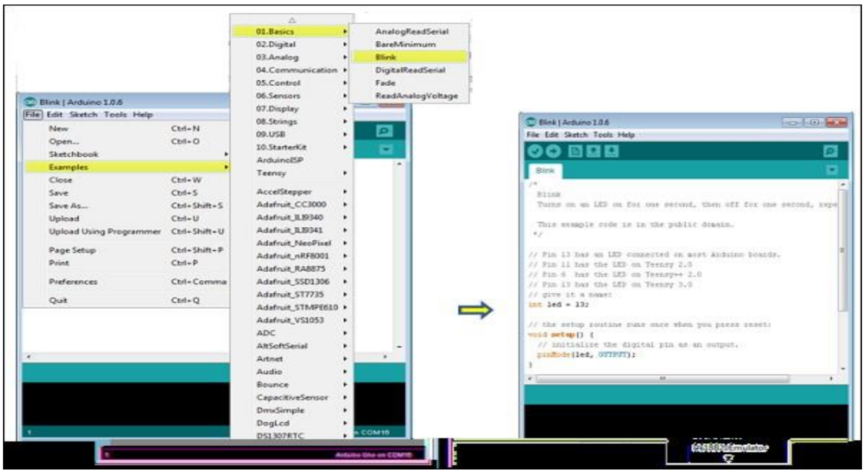


Fig.5.19: Opening Existing File in IDE

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

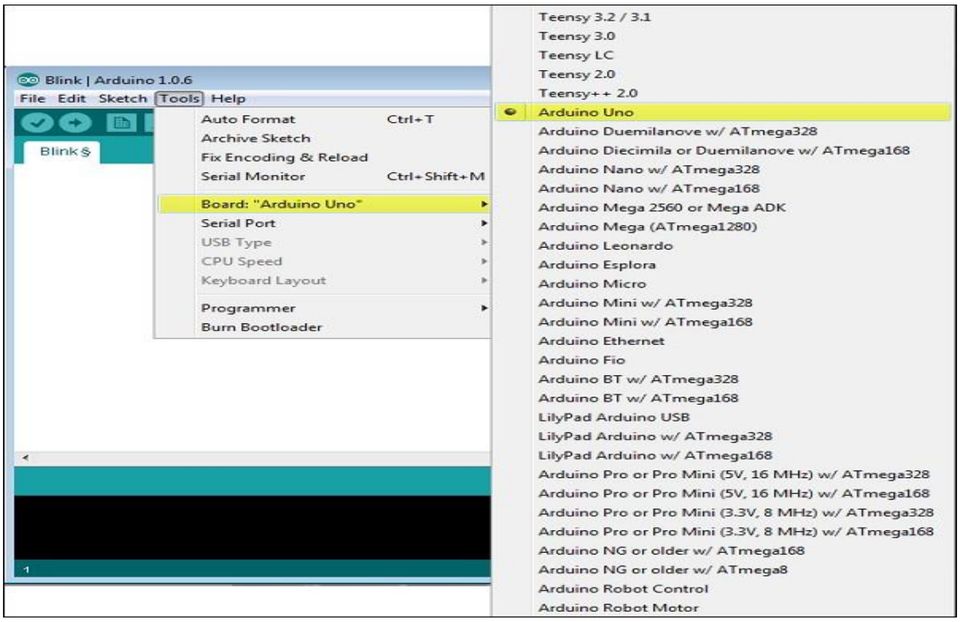


Fig.5.20: Selecting Arduino Board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports).

To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

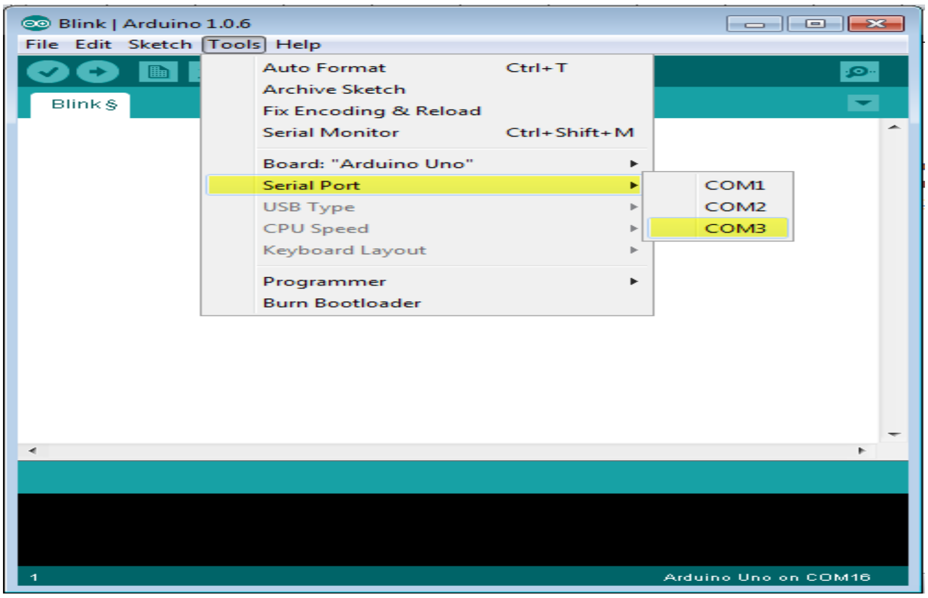


Fig.5.21: Selecting Serial Port

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

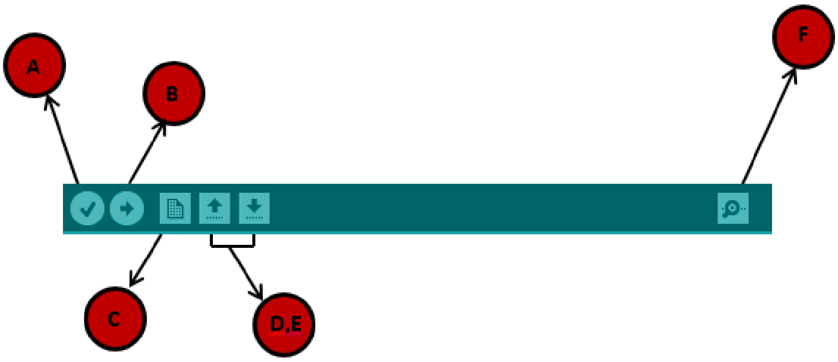


Fig.5.22: IDE Tool Bar

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

5.3.4 Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24 V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

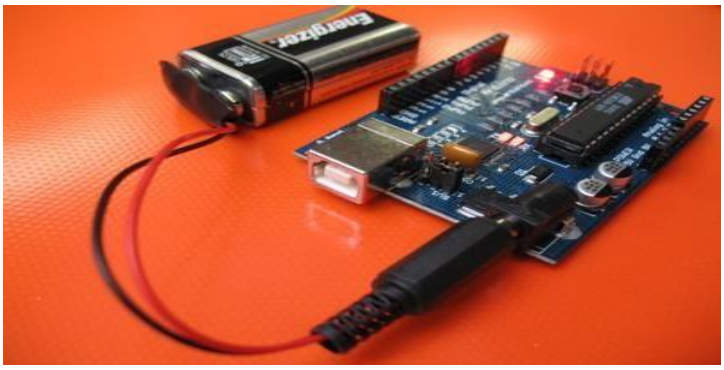


Fig.5.23: Arduino with Battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called “sketches”, but here we will just call them programs.

In the editing window that comes up, enter the following program show in Figure:4.66, paying attention to where semi-colons appear at the end of command lines.

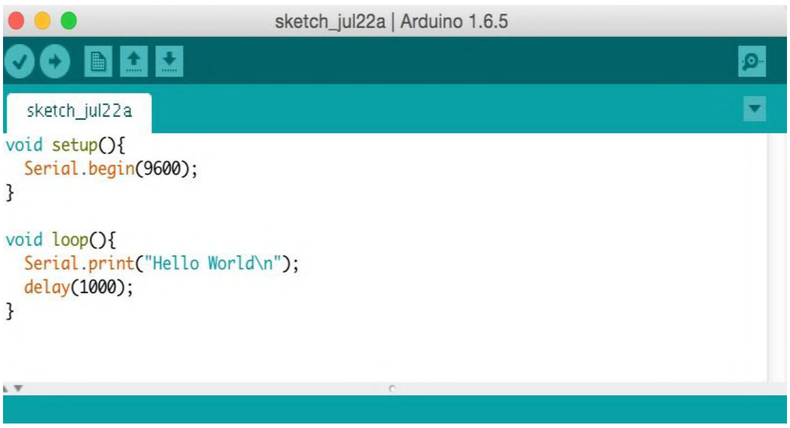


Fig.5.24: Writing the Program in IDE

5.4 MC Programming Language: Embedded C

This is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements. Embedded C is perhaps the most popular languages among Embedded Programmers for programming embedded Systems.There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

5.5 Proteus:

5.5.1 Proteus:

Proteus is a simulation and design software tool developed by Lab Centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

5.5.2 About Proteus

It is a software suite containing schematic, simulation as well as PCB designing.ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

5.5.3 Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components.

It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

5.5.4 Starting New Design

Step 1: Open ISIS software and select new design in File menu

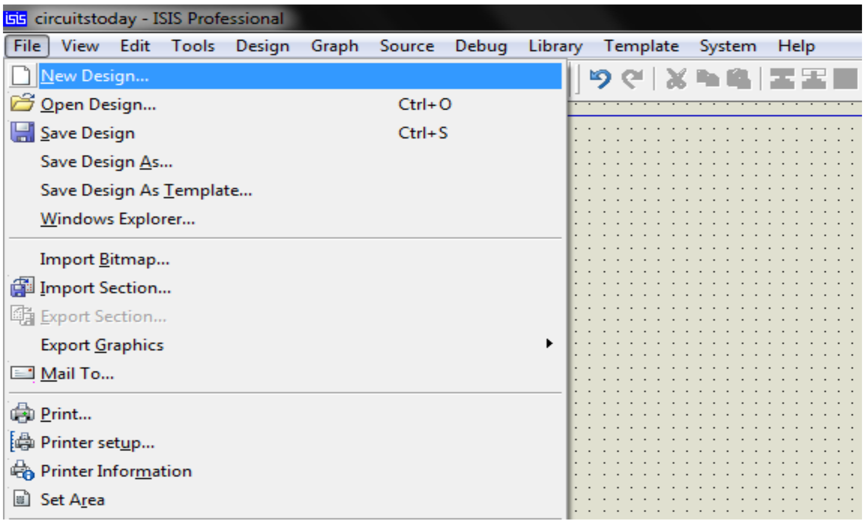


Fig.5.25: Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

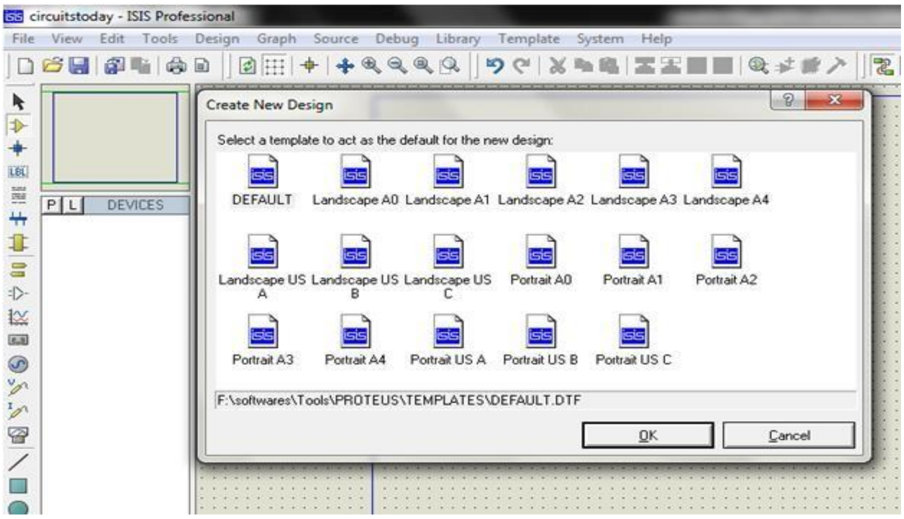


Fig.5.26: Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

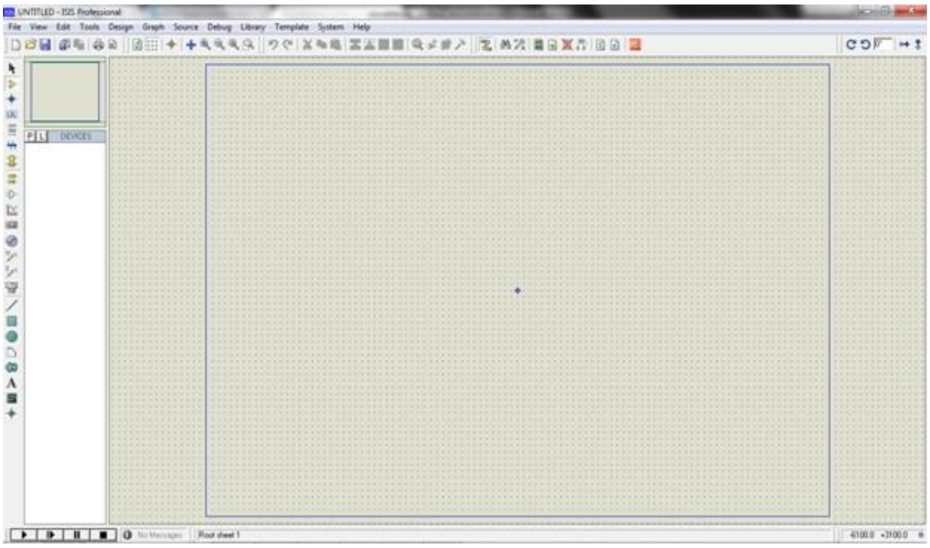


Fig.5.27: Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

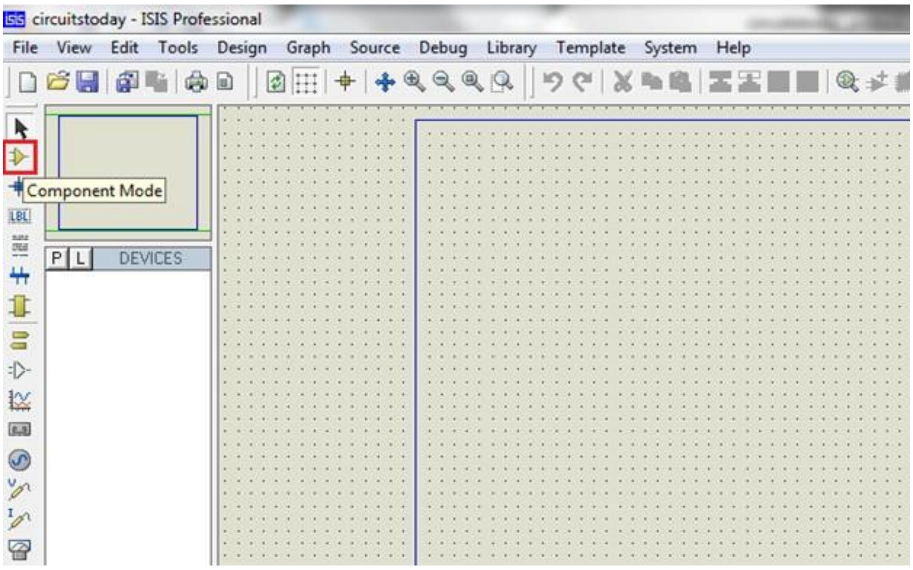


Fig.5.28: Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

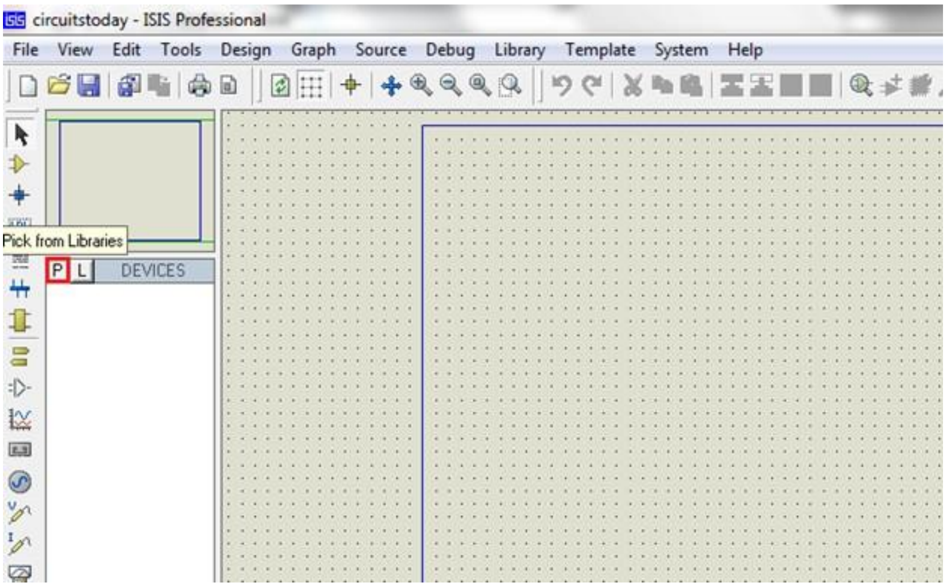


Fig.5.29: Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

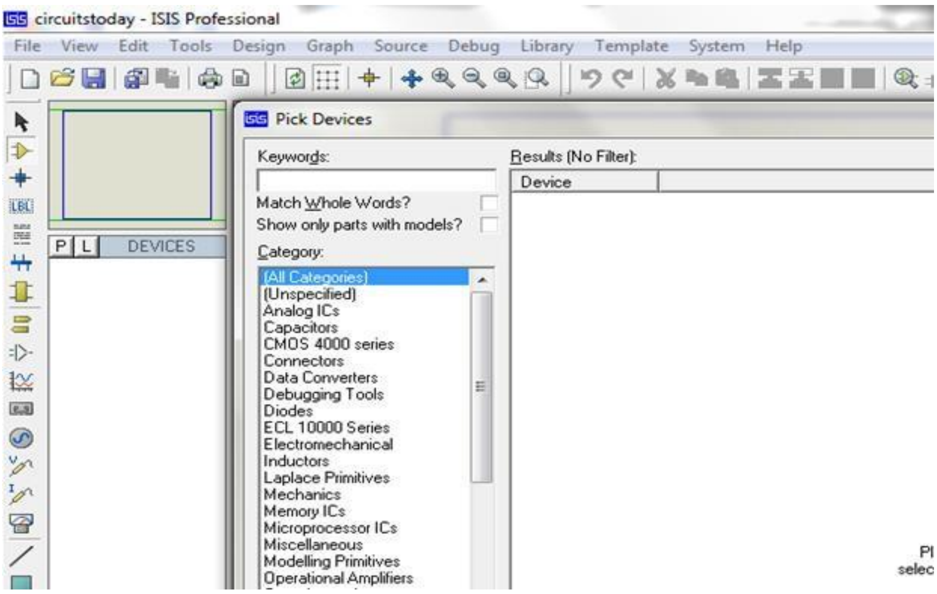


Fig.5.30: Keywords Textbox

Example shows selection of push button. Select the components accordingly.

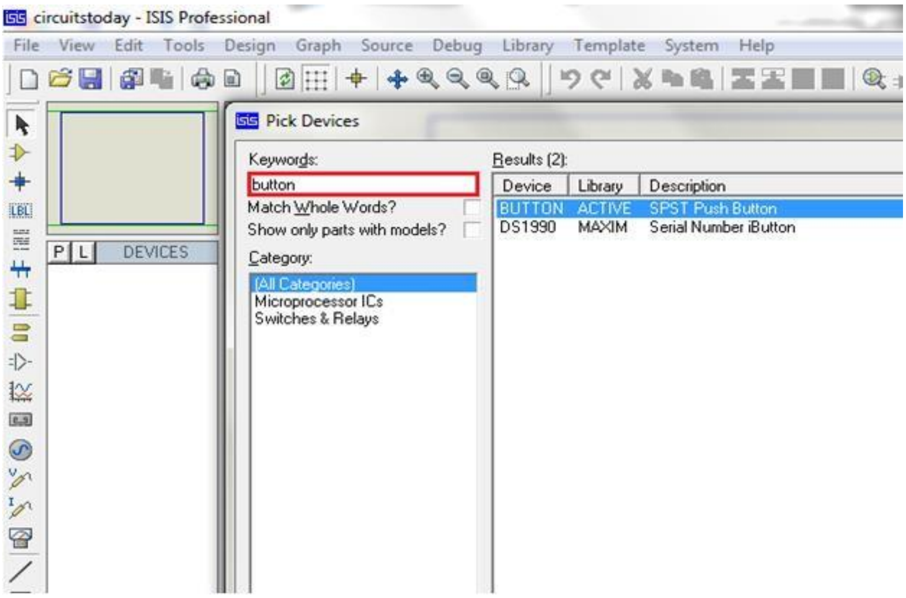


Fig.5.31: Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

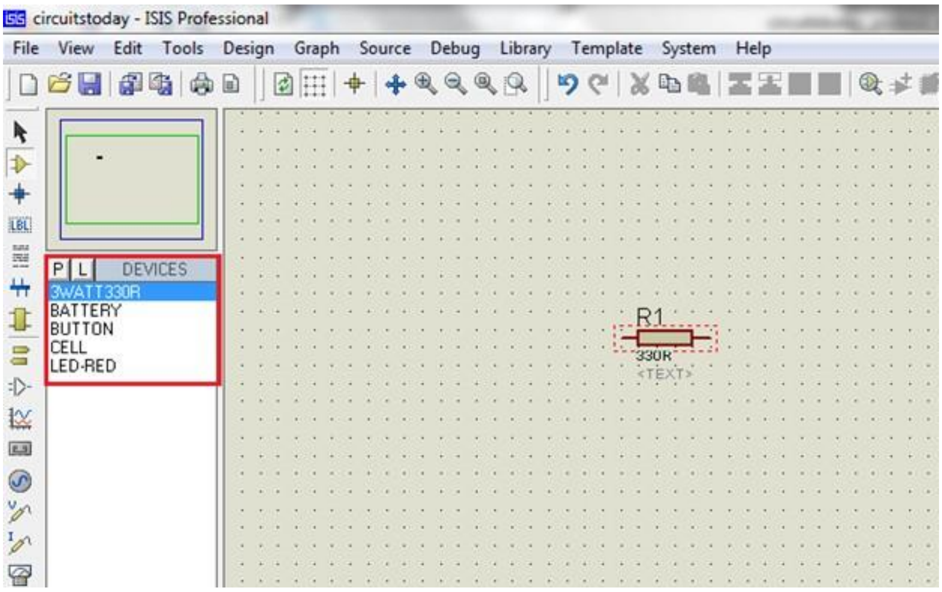


Fig.5.32: Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

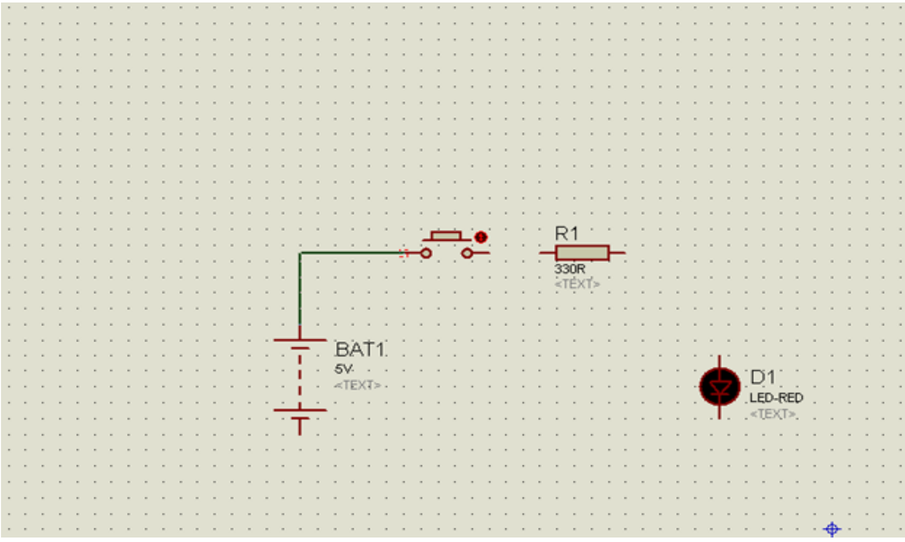


Fig.5.33: Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

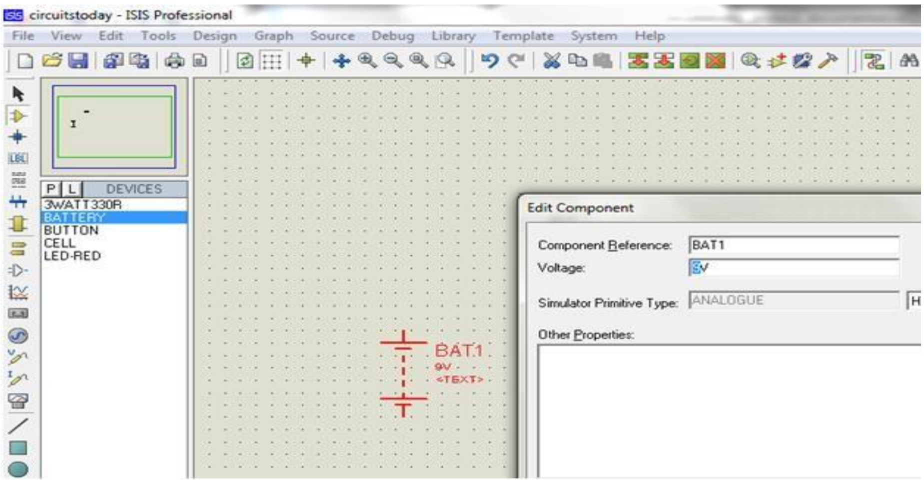


Fig.5.34: Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

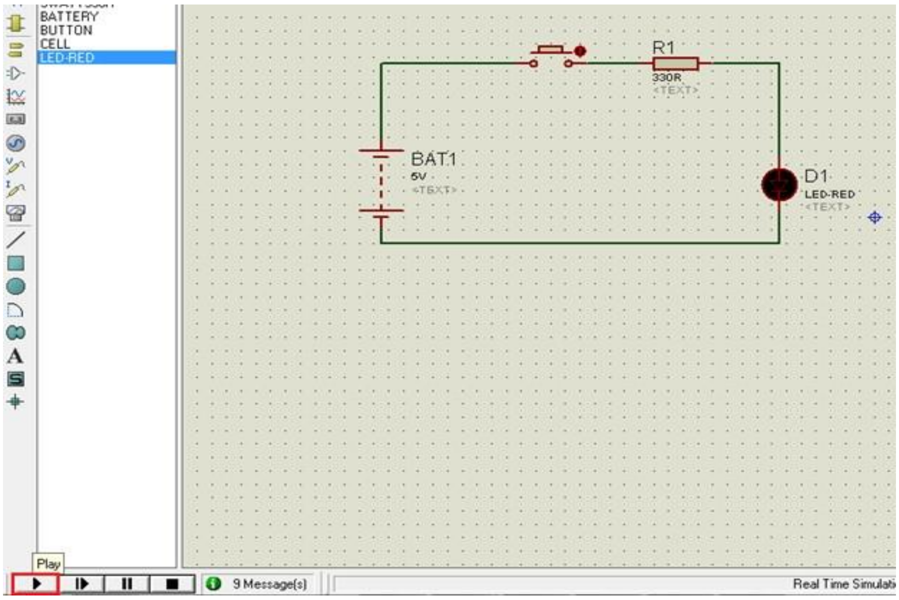


Fig.5.35: Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

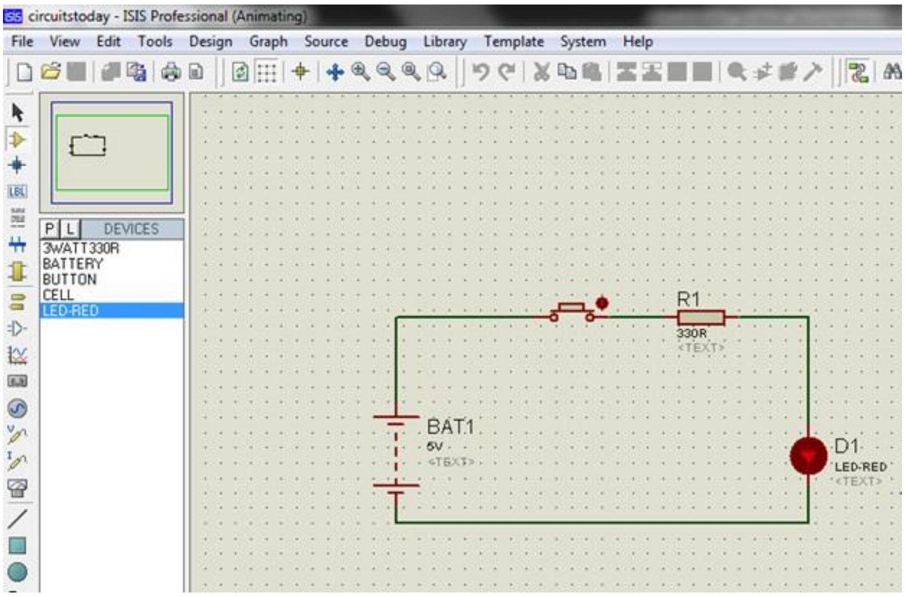


Fig.5.36: Simulation Animating

Simulation can be stepped, paused or stopped at any time.

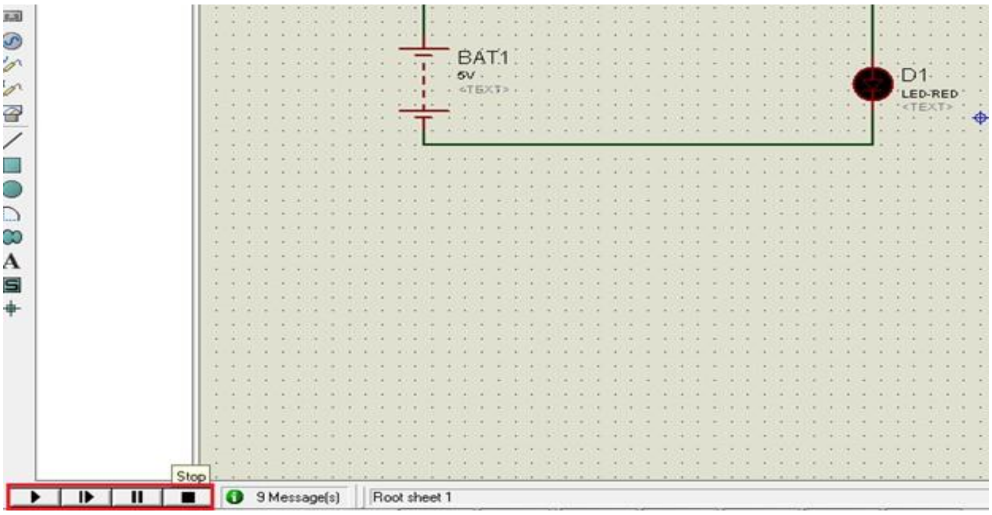


Fig.5.37: Simulation Step-Pause-Stop Buttons

5.5.5 Project Code:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);


int sense=A4,relay=6,buz=7;



 void setup() {
 Serial.begin(9600);
  pinMode(relay,OUTPUT);
  pinMode(buz,OUTPUT);;
  pinMode(sense,INPUT);



 digitalWrite(relay,LOW);
 digitalWrite(buz,HIGH);

  lcd.begin(16,2);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("MOBILE CHARGING ON COIN INSERTION");

  Serial.begin(9600);
```

```
    delay(3000);


  lcd.clear();lcd.setCursor(0,0);lcd.print("PLS INSERT UR COIN");
              delay(2000);
}

int count = 0,sig = 0;
void loop() {
     while(1)
     {
    // lcd.clear();
     lcd.setCursor(0,1);lcd.print("TIME:");lcd.print(count);delay(2000);
    if(digitalRead(sense)==LOW && sig == 0)
    {
      count =count + 60;
      sig =1;
      // lcd.clear();
       //lcd.setCursor(0,1);lcd.print("TIME:");lcd.print(count);
       digitalWrite(relay,HIGH);
    }
      if(sig == 1)
      {
    //lcd.clear();lcd.setCursor(0,1);lcd.print("TIME:");lcd.print(count);
       //lcd.clear();
     //  lcd.setCursor(0,0);lcd.print("sig");delay(2000);
       count--;
        if ( count <= 10)
       {
         //lcd.clear();

  lcd.setCursor(0,0);lcd.print("LOWBALANCE");digitalWrite(buz,LOW);delay(2000);digital
  Write(buz,HIGH);
       }

        if ( count == 0)
       {
         //lcd.clear();
         lcd.setCursor(0,0);lcd.print("NIL BALANCE");delay(2000);
         digitalWrite(relay,LOW);digitalWrite(buz,HIGH); sig = 0 ;
       }
      }

      //}

       /* if ( count <= 15)
       {
        // lcd.clear();
         lcd.setCursor(0,0);lcd.print("LOW BALANCE");delay(2000);
       }
```

```
 if ( count == 0)
       {
      //  lcd.clear();
        lcd.setCursor(0,0);lcd.print("NIL BALANCE");delay(2000);
        digitalWrite(relay,LOW);sig = 0 ;
      }*/
    // }


     }

}
```

CHAPTER-6

RESULTS & OUTPUTS

6.1 RESULT:

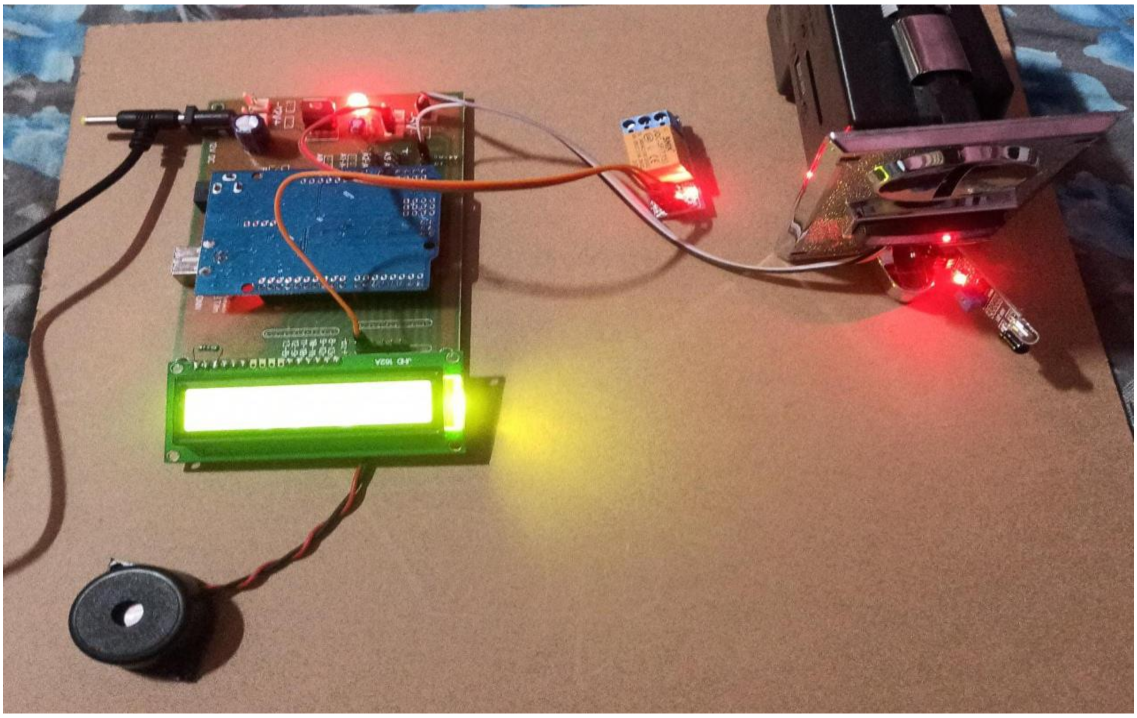


Fig.6.1: Final Output

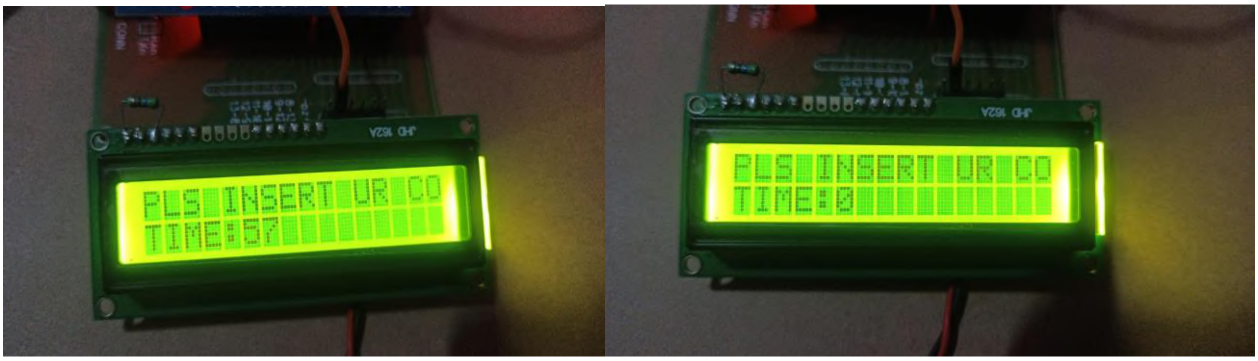


Fig.6.2: LCD Outputs

CHAPTER-7

APPLICATIONS AND ADVNTAGES

7.1 APPLICATIONS OF THE PROJECT:

- Useful to public for using coin to charge the mobile phone at any place.
- It can be used for different type of mobiles.
- It is used for emergency charging purpose.
- It can be installed railway stations, bus stops, villages and rural areas and public places.
- It can be installed in office and colleges for pay charging facility.

7.2 ADDVANTAGES OF THE PROJECT:

- Simple and hand efficient
- Less expensive
- Reduced man power
- Low power consumption
- Installation is easy
- It can be useful while travelling and when we don't have charger with us during travelling.
- Simple to operate

CHAPTER-8

CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSON:

The coin based mobile charger is used in public places for charging all types android cell phones just like charging it normally owing to the fact that it relayed the electrically through the coin based mobile charger nedded to bring the mobile phone back to life as a result it reduces certain risks and my save life. A novel method of charging mobile batteries of different manufactures using solar power has been designed and developed for rural and remote areas where the grid power is not available at any time at any palace

8.2 FUTURE SCOPE:

It can be installed on railway stations, bus stops and public places for pay charging facility.The system is very much useful foe emergency charging purposes or in rural areas.

The project can be used in the following areas

1. Railway station: This type of project is used in railway station for public palace.

2. Shop:-coin based project charger is install any shop and cam money

3. Rural areas:-This project is installing in rural areas where the power grid is not available at any time.

4. Public Place:-This project is very useful to people who are wing mobile phone wihout charging condition in public places

REFERENCES:

[1].M.S.Varadarajan “Coin Based Universal Mobile Battery Charger”. 2012 IOSR Journal of Engineering (IOSRJEN)

[2].K S.B.Sridevi, A. Sai Suneel. Nalini, “International Journal of Innovative Research in Science, Engineering and Technology”, ISSN: 2319-8753, ISO 3297: 2007 Vol. 3, Issue 2,PP.9603-9608, February 2014.

[3].Daniel A. Pritchard, Sun Tracking by Peak Power Positioning for Photovoltaic Concentrator Arrays, control systems magazine, 2011.

[4]. Robert Weissbach, Pennsylvania State University-Erie AC 2007-1213: A MICROCONTROLLER-BASED SOLAR PANEL TRACKING SYSTEM.

[5] K M Trautz P P Jenkins, R. J. Walters, D. Scheiman, R. Hoheisel, R. tatavarti, R. Chan, H. Miyamoto, J.G.

[6] S.Banu Pratap, R. Priyanka, G.Gunnu, Dr.Sujatha, Coin Based Cell Phone Charger, International Journal Of Reserch And Technology Issue3, Ma rch-2013

A

MAJOR PROJECT REPORT

on

BUILD, OWN VIRTUAL ASSITANT

Submitted by

Ms. B. Aasrita (17K81A04D2)

Mr. B. Abhinay Reddy (17K81A04D4)

Mr. M. Roshan Kumar (17K81A04F8)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

K. Naga Venkateshwara Rao

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500100

NBA &NAAC A+ Accredited

JUNE 2021

St. MARTIN'S ENGINEERING COLLEG
UGC AUTONOMOUS

Dhulapally, Secunderabad-500 100
NBA &NAAC A+ Accredited

CERTIFICATE

This is to certify that the major-project work entitled **"Build, Own Virtual Assitant"** is a bonafide work carried out by **B. Abhinay (17K81A04D4), B. Aasrita (17K81A04D2), M. Roshan Kumar (17K81A04F8),** in partial fulfilment of the requirements for the degree of **Bachelor of Technology in Electronics & Communication Engineering** by the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2020-2021.The results embodied in this report have not been submitted to any other University or Institution for the award of any degree.

INTERNAL GUIDE	HEAD OF DEPARTMENT
Mr. K. Naga Venkateshwara Rao	Dr. B. Hari Krishna,
Assistant Professor M.Tech (Ph.D),	Professor, Ph.D
Department of ECE	Department of ECE

INTERNAL EXAMINER **EXTERNAL EXAMINAR**

Place:

Date:

DECLARTION

We, the student of '**Bachelor of Technology in Department of Electronics and Communication Engineering**', session: 2017 – 2021, **St. Martin's Engineering College**, hereby declare that the project work entitled '**Build, Own Virtual assistant** is the outcome of our own Bonafede work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

B. ABHINAY REDDY (17K81A04D4)
M. ROSHAN KUMAR (17K81A04F8)
B. AASRITA (17K81A04D2)

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **B. ABHINAY REDDY** WITH ROLL NO.**17K81A04D4**, **B. AASRITA** WITH ROLL NO.**17K81A04D2**, **M. ROSHAN KUMAR** WITH ROLL NO.**17K81A04G5**, OF B. TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING** DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "ROAD MAP TO 6G: AI EMPOWERED WIRELESS NETWORK" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS &
PLACEMENTS LASYA
IT SOLUTIONS PVT
LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road,
Secunderabad 500014Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to our College Management for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr. P. Santosh Kumar Patra, our beloved Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B. Hari Krishna, Head & Professor,** the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our department **Project Coordinator, Dr. A. Anand Kumar,** Professor for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our Class Project Coordinator, **Mr. Nishakar, Department of ECE** and our project Internal guide, **Mr. K Naga Venkateshwara Rao, Assistant Professor, Department of ECE** for guidance and encouragement in carrying out this project work.

1. B. Abhinay Reddy (17K81A04D4)
2. B. Aasrita (17K81A04D2)
3. M. Roshan Kumar (17K81A04F8)

ABSTRACT

Adoption of social network sites and use of smart phones with number of sensors in them has digitized user's activities in real-time. Smart phone applications such as calendar, email, and notes contain lot of user information and may other provide a view into user's activities, while sensors such as GPS sensor can be used to passively find information about the user. In addition to this user and device data, these devices have access to the Internet that can be leveraged to build powerful applications. Personal assistant software (smart agent) can be used as an interface to the digital world to make the consumption of this information timely and efficient for the user's specific tasks. Goal of this project is to design personal assistant software that understands the semantics of the task, is able to decompose the task into multiple tasks withinthe context of the user and plan these tasks for the user. It is designed and developed using python (version 3.6) with the help of pycharms community interface.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.0 INTRODUCTION

In today's era almost all tasks are digitalized. We have Smartphone in hands, and it is nothing less than having world at your fingertips. These days we are not even using fingers. We just speak of the task and it is done. There exist systems where we can say Text Dad, "I'll be late today." And the text is sent. That is the task of a Virtual Assistant. It also supports specialized task such as booking a flight or finding cheapest book online from various ecommerce sites and then providing an interface to book an order are helping automate search, discovery and online order operations.

Virtual Assistants are software programs that help you ease your day to day tasks, such as showing weather report, creating reminders, making shopping lists etc. They can take commands via text (online chat bots) or by voice. Voice based intelligent assistants need an invoking word or wake word to activate the listener, followed by the command. For this project the wake word is Escobar (we can use any of the name we like too). We have so many virtual assistants, such as Apple's Siri, Amazon's Alexa and Microsoft's Cortana. For this project, wake word was chosen as Reddy.

This system is designed to be used efficiently on desktops. Personal assistant software improves user productivity by managing routine tasks of the user and by providing information from online sources to the user. Reddy is effortless to use. Call the wake word 'Reddy' followed by the command. And within seconds, it gets executed.

Voice searches have dominated over text search. Web searches conducted via mobile devices have only just overtaken those carried out using a computer and the analysts are already predicting that 50% of searches will be via voice by 2020.Virtual assistants are turning out to be smarter than ever. Allow your intelligent assistant to make email work for you. Detect intent, pick out important information, automate processes, and deliver personalized responses. This project was started on the premise that there is enough openly available data and information on the web that can be utilized to build a virtual assistant that has access to making intelligent decisions for routine user activities.

1.1 OBJECTIVE OF THE STUDY

The main objective of the project is to build our own virtual assistant with no cost forour daily tasks performed.

1.2 AIM OF THE STUDY

The primary aim for doing this project is to build a virtual assistant on our own in aeasy way and the interfacing is very fast and user friendly.

1.3 MATERIALS REQUIRED

- Personal Computer(PC) with OS(Operating System)
- Python Installer(version > 3.6)
- Pycharm Compiler
- Network(Ethernet or WIFI)

1.4 ORGANIZATIONS OF CHAPTERS

1.4.0 Introduction

Virtual Assistant is a device which is used to reduce the time of user for certainapplications and helps to do tasks.

1.4.1 Literature Survey

Speech recognition has a long history with several waves of major innovations. Speech recognition for dictation, search, and voice commands has become a standard feature on smartphones and wearable devices.

1.4.2 Project Design

- The goal of the project is to build our own virtual assistant.

- Using python language.
- Perform regular tasks by giving voice command.

1.4.3 Project Testing

Virtual Assistant works on giving command by the user. To perform any task, it requires a wake word. So, it will be in listening state if there is no wake word matching by the command given by user.

1.4.4 Conclusion and Future Enhancement

Nowadays all the virtual assistants are being integrated or manufactured at industrial standards and are given to the public. This project deals with making their own virtual assistant at mature level later it may be done even at teen level. In Future we can also build our own hardware of the virtual assistant rather than buying the fully designed product.

CHAPTER 2

LITERATURE SURVEY

2.0 LITERATURE REVIEW ON RESEARCH AREA

A computer primarily based approach for performing a command via a voice consumer interface on a subset of objects. The subset is selected from a fixed of items, each having an object type at least one taggable field is associated with the object type and has a corresponding value. The set of objects is saved in the laptop memory. An utterance is acquired from the person and consists of a command, an object type choice, a tag-gable field selection, and a price for the taggable discipline. Responsive to the utterance, at least one item is retrieved from the set of gadgets, the item of the sort selected through the user and having a price within the taggable area selection that matches the taggable field fee obtained from the user the command is done on the item. The object includes textual content that's converted to voice output [1]. They envisioned that someday computers will recognize natural language and count on what we need, whilst and where we need it, and proactively whole responsibilities on our behalf. However, speech recognition and machine getting to know have persevered to be refined, and based records served through packages and content providers have emerged. We agree with that as computer systems turn out to be smaller and greater ubiquitous [e.g., wearables and Internet of Things (IoT) [2]. The recognizer is designed to change a verbal articulation from a individual into an alternate method of data (e.g., text). A handheld individual colleague including a voice-recognizer and a characteristic dialect processor is disclosed. This snippet of data can be a plan for the day, data in the individual's logbook or data from the individual's address book, Such as a telephone number [3]. Speech recognition has a long history with several waves of major innovations.

Speech recognition for dictation, search, and voice commands has become a standard feature on smartphones and wearable devices. Design of a compact large vocabulary speech recognition system that can run efficiently on mobile devices, accurately and with low latency. [1] This is achieved by using a CTC based LSTM acoustic model which predicts context independent phones and is compressed to a tenth of its original size using a combination of SVD-based compression and quantization. Quantized deep neural networks (DNNs) and on-the-fly language model rescoring to achieve real-time performance on modern smartphones.

2.1 REVIEW ON RELATED LITERATURE

•[1]Veton Këpuska,"Next-Generation of Virtual Personal Assistants, Published in2018-uses of the multi-modal dialogue systems.

• [2]Sumit Kumar Sarda, "VPA: Virtual Personal Assistant" Published in 2017 -reduces the utilization of input devices

•[3]P. Milhorat,"Building the next generation of personal digital Assistants," Published in 2014 -adapted to a user's preferences are in distinct needs.

2.1 CONCLUSIONS ON REVIEWS

As the use and purpose of virtual assistant is easy and more in today's world. This project is to design a free cost virtual assistant.

CHAPTER 3

PROJECT DESIGN

3.0 OVERVIEW OF THE DESIGN

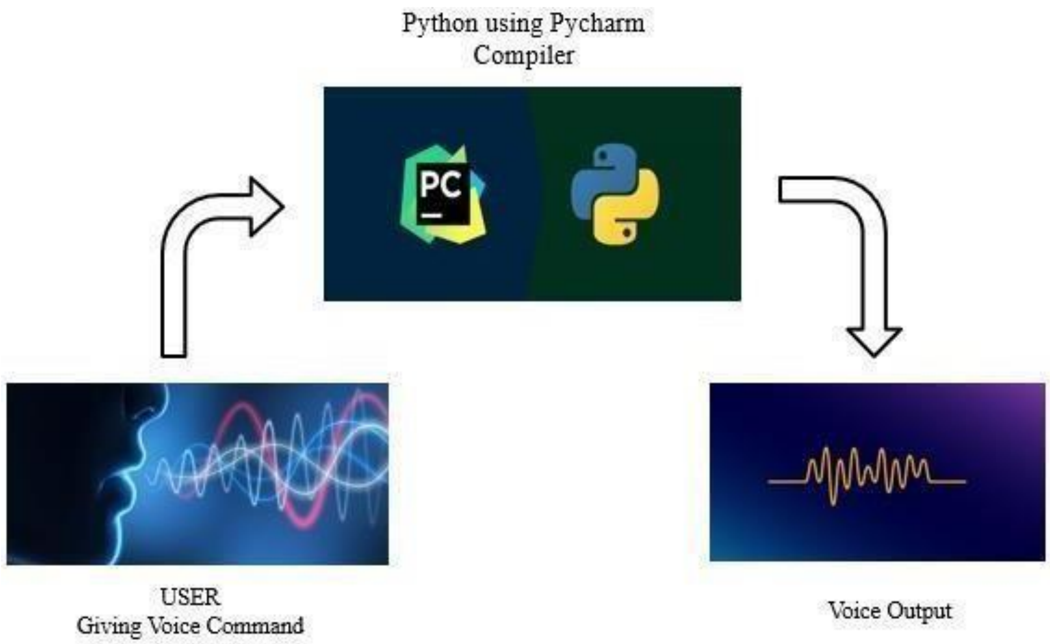


Figure 3. 1 Logic Diagram of Python code Working.

3.0.1 Virtual Assistant

A virtual assistant is an independent contractor who provides administrative services to clients while operating outside of the client's office. A virtual assistant typically operates from a home office but can access the necessary planning documents, such as shared calendars, remotely.

People employed as virtual assistants often have several years of experience as an administrative assistant or office manager. New opportunities are opening up for virtual assistants who are skilled in social media, content management, blog post writing, graphic design, and Internet marketing. As working from home has become more accepted for both workers and employers, the demand for skilled virtual assistants is expected to grow.

3.0.2 Key Points of Virtual Assistant

- A virtual assistant is a self-employed worker who specializes in offering administrative services to clients from a remote location, usually a home office.
- Typical tasks a virtual assistant might perform include scheduling appointments, making phone calls, making travel arrangements, and managing email accounts.
- Some virtual assistants specialize in offering graphic design, blog writing, bookkeeping, social media, and marketing services.
- For an employer, one advantage of hiring a virtual assistant is the flexibility tocontract for just the services they need.

3.0.3 How a Virtual Assistant Works

Virtual assistants have become more prominent as small businesses and startups rely on virtual offices to keep costs down and businesses of all sizes increase their use of the Internet for daily operations. Because a virtual assistant is an independent contractor, a business does not have to provide the same benefits or pay the same taxes that it would for a full-time employee.

Also, since the virtual assistant works offsite, there is no need for a desk or other workspace at the company's office. A virtual assistant is expected to pay for and provide their own computer equipment, commonly used software programs, and high-speed Internet service.

3.0.4 Virtual Assistant Duties

The specific duties of a virtual assistant vary according to the needs of the client and the terms of the contract. Some virtual assistants handle clerical and bookkeeping tasks, while others may post regular updates to social media or write articles for a blog. A well- rounded virtual assistant may also handle travel arrangements, appointment scheduling, data entry, and online file storage.

3.0.5 Virtual Assistant Qualifications

While there are no hard-and-fast educational requirements to become a virtual assistant, many clients will look for virtual assistants who have some higher-level education or specialized training. A few online companies and community colleges offer courses and certification for virtual assistant skills.

A virtual assistant should be tech-savvy, having a wide range of computer skills and a high level of proficiency with commonly used software and business programs. A virtual assistant who specializes in bookkeeping should be adept at basic accounting tasks, such as account reconciliations and double-entry bookkeeping.

3.0.6 Benefits of a Virtual Assistant

For the client, one advantage of hiring a virtual assistant is the flexibility to contract for just the services they need. Depending on the terms of the agreement, some virtual assistants may be paid by the task rather than by the hour. In contrast, employees in a traditional office setting usually must be paid for a fixed number of hours per day.

3.0.7 Coding Techniques

The project is based on the theories related to various aspects of software engineering principles and software development model; Java programming skills and Android tutorials, Database management and network communication technologies. The database and the web service in this project are put on the windows azure cloud; developers will never be required to run the web service and database locally. The cloud platform will handle the execution and maintenance. Hence, cloud computing is an important concept and theory guide the development.

3.0.7.0 Cloud Computing

Cloud computing refers to the delivery of computing and storage capacity as a service to a heterogeneous community of end-recipients. The name comes from the use of clouds as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts services with a user's data, software and computation over a network. It has considerable overlap with software as a service (SaaS). [5]

3.0.7.1 Software Engineering Principles

Extreme programming will direct the development process of the project, it focus on the development cycle of defining the requirement, corresponding design and test, integration and simplicity; during the development, there should always be working in pair programming, as well as doing the revision control, calculate the velocity and efficiency. Extreme programming (XP) is a software development methodology which is intended to i mprove software quality and responsiveness to changing customer requirements. As a type of agile software development, it advocates frequent "releases" in short development cycles (t imeboxing), which is intended to improve productivity and introduce checkpoints where new customer requirements can be adopted. [6]

3.0.7.2 Java Programming

Java API and reference, which is helpful in guide programming in eclipse and construction of the framework, and the completion of the functions. Java is a programming language originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to bytecode (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write INTELLIGENT VOICE ASSISTANT 12 once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java is currently one of the most popular programming languages in use, particularly for client-server web applications, with a reported 10 million users. [7] [8]

3.0.7.3 Network communication technologies

The communication in this program is based on the predefined protocol, the communication within the program is implemented in following the pre-defined protocol, the other main part of the communication is between the Android program in eclipse and the cloud platform, this will be done by working with URL, WSDL file. Figure-3.2 shows some knowledge of cloud platform, URL and WSDL. [15]

The WSDL describes services as collections of network endpoints, or ports. The WSDL specification provides an XML format for documents for this purpose. The abstract definitions of ports and messages are separated from their concrete use or instance, allowing the reuse of these definitions. A port is defined by associating a network address with a reusable binding, and a collection of ports defines a service. Messages are abstract descriptions of the data being exchanged, and port types are abstract collections of supported operations. The concrete protocol and data format specifications for a INTELLIGENT VOICE ASSISTANT 14 particular port type constitutes a reusable binding, where the operations and messages are then bound to a concrete network protocol and message format. In this way, WSDL describes the public interface to the Web service. [16]

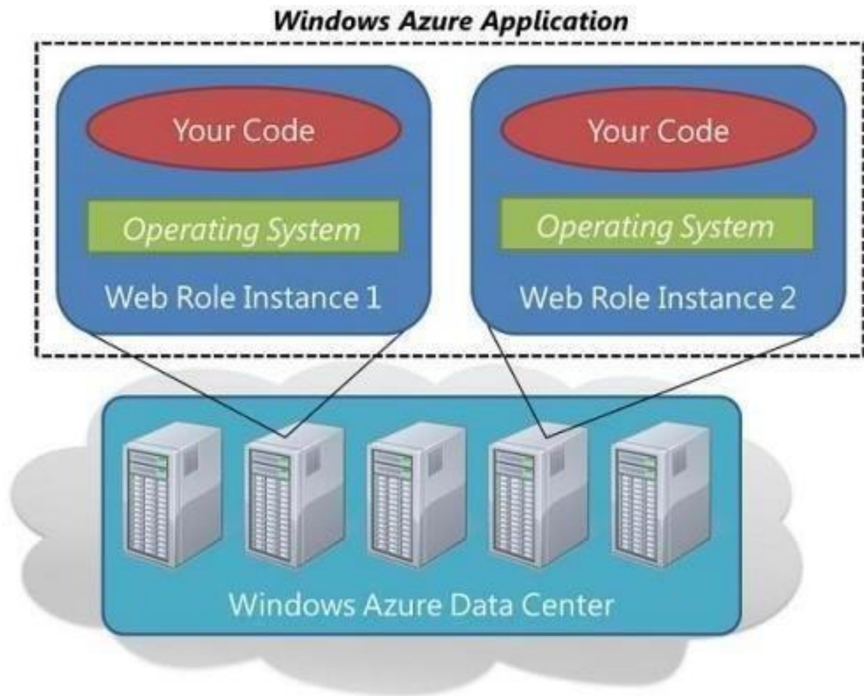


Figure 3. 2 Network Communication Technology

3.1 EQUIPMENT ANALYSIS

3.1.0 PYTHON

"Python is an interpreted, object-oriented, high-level programming language with dynamic semantics".[6] This language consist of mainly data structures which make it very easy for the data scientists to analyse the data very effectively. It does not only help in forecasting and analysis it also helps in connecting the two different languages.Two best features of this programming language is that it does not have any compilation step as compared to the other programming language in which compilation is done before the program is being executed and other one is the reuse of the code, it consist of modules and packages due to which we can use the previously written code any where in between the program whenever is required. There are multiple languages for example R., Java, SQL, Julia, Scala, MATLAB available in market which can be used to analyze and evaluate the data, but due to some outstanding features python is the most famous language used in the field of data science. Python is mostly used and easy among all other programming languages is due to the following reasons.

3.1.1.1 Data Structures In Python

Data structures are the way of storing the data so that we can easily perform different operations on the data whenever its required. When the data has been collected from the data source the data is available in different forms. So later it is easy for the data scientists to perform different operation on the data once it is sorted in to different data structures. Data structures are mainly classified in to two categories and then further their subcategories shown below.

- Primitive data structures
- Non-Primitive data structures

3.1.1 PYCHARM

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains (formerly known as IntelliJ).[5] It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda.

PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License,[7] and there is also Professional Edition with extra features – released under a proprietary license.

3.2 DEFINE THE MODULES

Following the functions of the data structures in python.

3.2.1 Primitive Data Structures

They are also called as basic data structures.This type of data structures contains simple values of the data.[7]

- Integers - All the whole numbers from negative infinity to positive infinity comes under integer data types. for example 4,9,-2,-6.

- Float - The decimal figure numbers or rational numbers comes under float data types. for example, 3.1,2.2,8.96.
- Strings - Collection of alphabets or characters are called strings. We enclose the string either in single or double quotes in python. for example ’hello’ and ”bread”.
- Boolean- These are the built in data types which take two values that are ’True’ and ’False’. True represents the 1 and False represents 0 in python.

3.2.2 Non-Primitive Data Structures

These are the derived type or reference variables data structures. They are called derived data structures because they derived from the basic data structures such as integer and float. Python has mainly five types of data structures.

- Array - Array is the collection of data types of same type. Arrays data structures are used mostly in the NumPy library of python. In the below example we have first imported the package array from NumPy library and defined the array as variable arr then divided the array by 7 and we have printed our array to get output.

```
In [3]: from numpy import array
        arr = array([28,35,42,56])
        arr = arr/7
        print(arr)

        [4. 5. 6. 8.]
```

Figure 3. 3 Array Example

- List – “A list is a value that contains multiple values in an ordered sequence”.[8]. Values in the list referred to list itself, that is the value can be stored in a variable or passed to a function. List are changeable and values in the list are enclosed inside a square bracket, we can perform

```
In [13]: List = ['orange', 'mango','kiwi','papaya']
         print(List)

         ['orange', 'mango', 'kiwi', 'papaya']
```

Figure 3.4 List Example

- Dictionary- These are nothing but a type of data structure which consist of key value pairs enclosed in the curly brackets. It is same as the any dictionary we use in day to day life in which we find the meaning of the particular words. So if I compare normal dictionary to this python dictionary data structure then the a word in a dictionary will be our key and its meaning will be the value of the dictionary. In the figure name, occupation and hobby are the keys and Suraj, data analyst and vlogging are the values assigned to the keys.

```
In [25]: dictionary = {'name':'Suraj', 'occupation':'data analyst','hobby':'vlogging'}
         dictionary['hobby']

Out[25]: 'vlogging'
```

Figure 3.5 Dictionary Example

- Tuple - A tuple is a list of non-changeable objects. The differences between tuples and lists are that the tuples cannot be changed, tuples use parentheses, whereas list uses square brackets.

```
In [16]: Tuple = ('orange', 'mango','kiwi','papaya')
         #If you can see the change values are same as list but enclosed in parentheses
         print(Tuple)

         ('orange', 'mango', 'kiwi', 'papaya')
```

Figure 3.6 Tuple Example

- Sets - Set are used for calculating mathematical operation such as union, intersection, and symmetric difference.

3.3 MODULE FUNCTIONALITY

3.3.1 Pyttsx3

It is a text-to-speech conversion library in Python. Unlike alternative libraries, itworks offline, and is compatible with both Python 2 and 3.

Installation:

pip install pyttsx3

If you recieve errors such as No module named win32com.client,

Usage:

importpyttsx3

engine = pyttsx3.init()

engine.say("I will speak this text")

engine.runAndWait()

3.3.2 PyWhatKit

PyWhatKit is a Python library with various helpful features. It is an easy to use librarywhich does not requires you to do some additional setup.

Installation:

This library can be installed by the pip command, open your command prompt and typein the following command…

pip install pywhatkit

Functions of this library:

First import the library using the command “import pywhatkit as kit” and then proceed to call the functions.

3.3.2.0 kit.sendwhatmsg()

This function can be used to send WhatsApp message at certain time. To any person with a phone number.

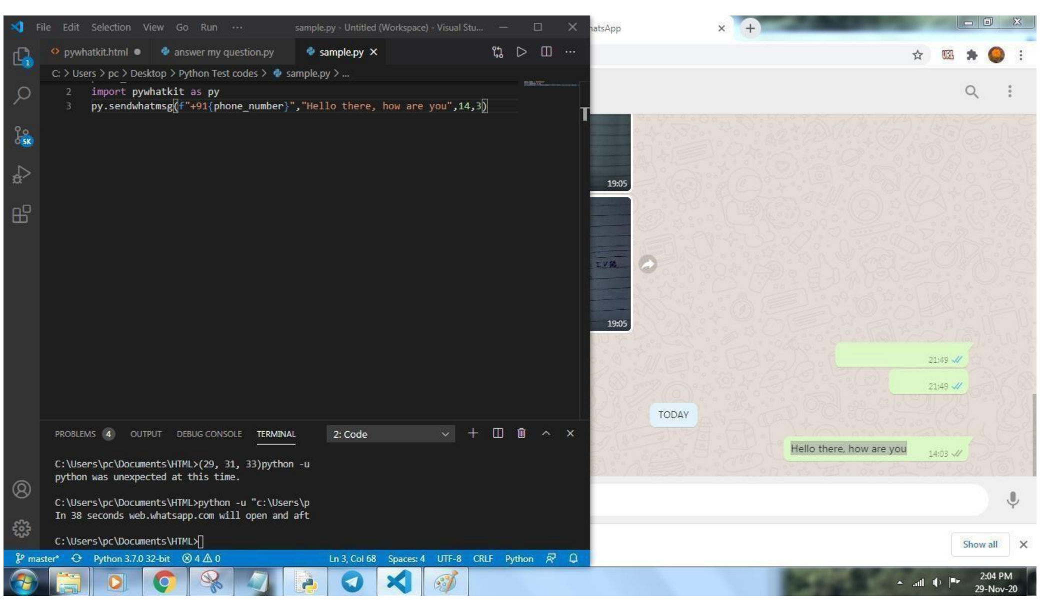


Figure 3.7 Sending WhatsApp message using pywhatkit library

The parameters are

phone_num (required) - Phone number of target with country code

message (required) - Message that you want to sendwhatmsg

time_hour (required) - Hours at which you want to send message in 24 hour format

time_min (required) - Minutes at which you want to send message

wait_time (optional, val=20) - Seconds after which the message will be

sent afteropening the web

print_waitTime (optional, val=True) - Will print the remaining time if set to true
Some common errors

CountryCodeException - Check if the phone number passed into the parameter has

country code

Message not getting delivered - Check internet speed and increase wait_time to

30 orabove.

CallTimeException - The web takes some time to load so some delay is required,

makesure the seconds left is greater than the wait_time

SyntaxError - Make sure the first two parameters are string and the rest are int

3.3.2.1 kit.playonyt()

This function can be used to search and play a particular video on YouTube byusing just the keyword, like "Shape of You song"

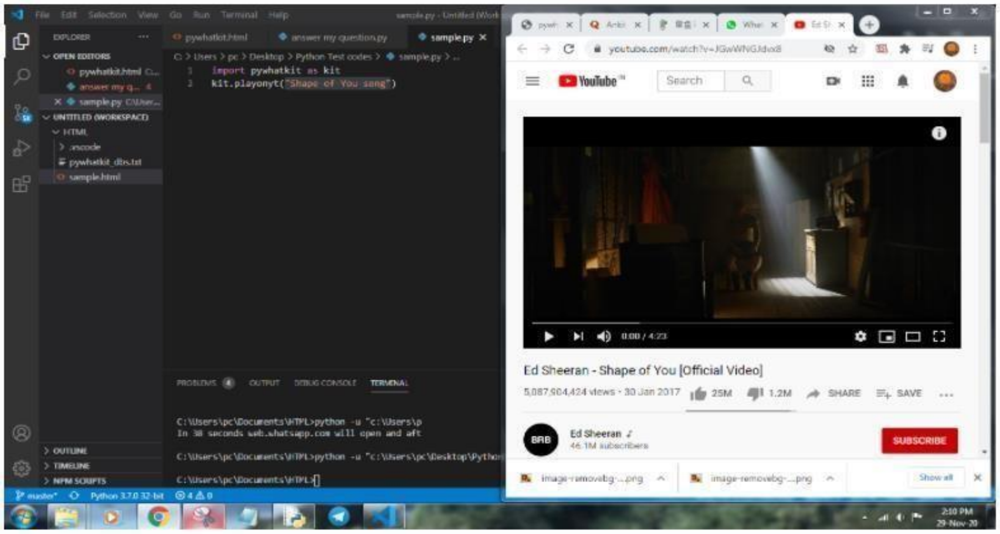


Figure 3.8 Video playing in youtube (Shape of you)

The parameters are topic (required) - Topic or title that is related to the video

Some common errors

Video not opening - Make sure the topic exists or you have provided proper spelling kit.search()

This function can be used to make a google search for any term

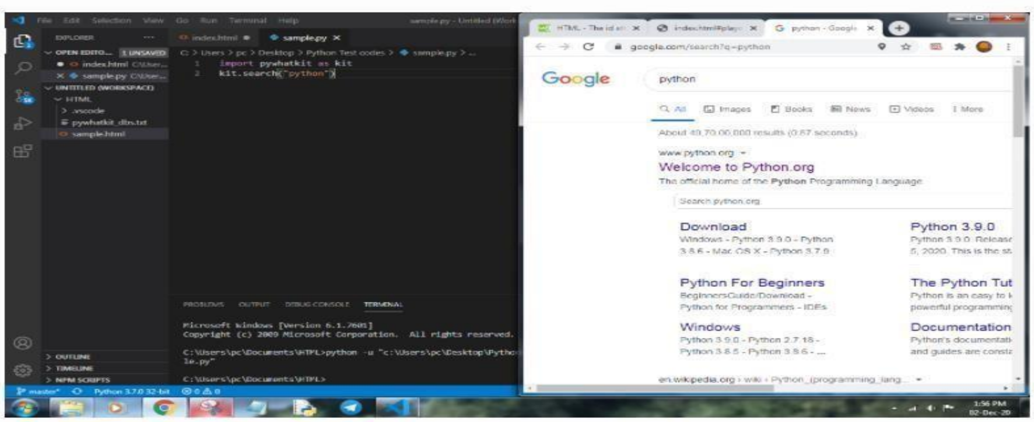


Figure 3.9 Web search

The parameters are
topic (required) - Topic or title that you want to search

3.3.2.2 kit.info()

This function can be used to fetch information about any topic.

The parameters are

topic (required) - Topic or title that you want to get information about lines (optional,val=3) - Number of lines that you want to print about it

Some common errors

Not returning paragraph - Make sure the topic exists and you are providing specifictitle.

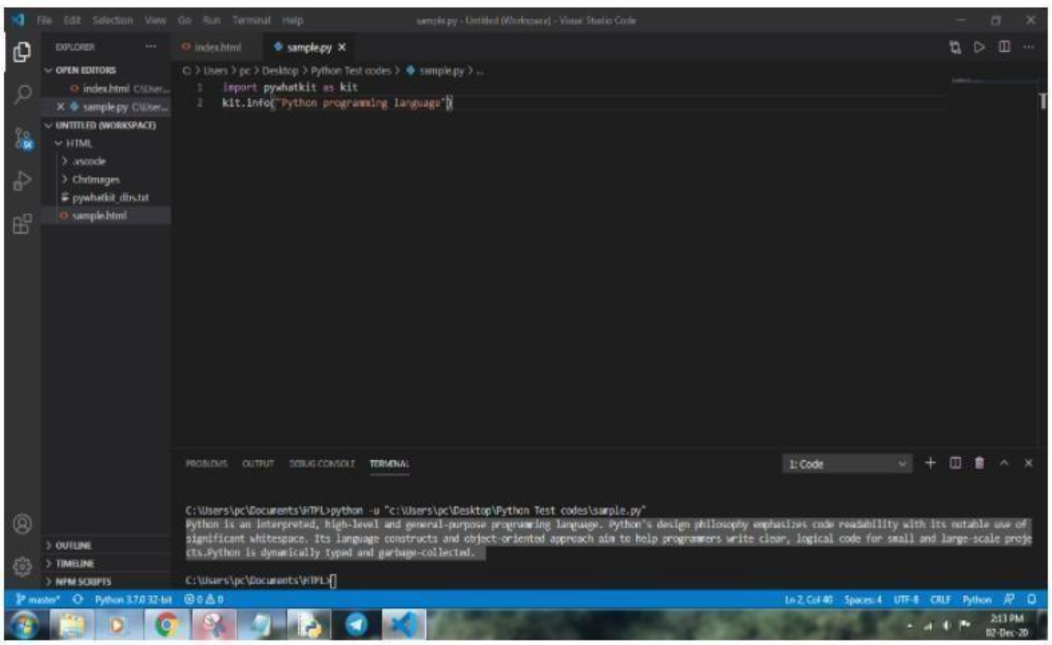


Figure 3.10 Information of Python programming language

3.3.3 PyJokes

One line jokes for programmers (jokes as a service).

Installation

Install the *pyjokes* module with.**Usage**

Once installed, simply call pyjoke from the command line or add it to your. bashrc file tosee a joke every time you open a terminal.

Use the -c flag to get jokes from a specific category.Options:

-c neutral [default] (neutral geek jokes)

-c chuck (Chuck Norris geek jokes)

-c all (all jokes)

-c twister (Tongue-twister)

You can also access the jokes in your own project by importing *pyjokes* and using thefunctions *get_joke* and *get_jokes*.

3.3.4 Wikipedia

Wikipedia is a Python library that makes it easy to access and parse data from Wikipedia.

Search Wikipedia, get article summaries, get data like links and images from a page, and more. Wikipedia wraps the MediaWiki API so you can focus on using Wikipedia data, not getting it.

Installation

To install Wikipedia, simply run:

$ pip install Wikipedia

Wikipedia is compatible with Python 2.6+ (2.7+ to run unittest discover) and Python 3.3+.

3.3.5 Tree Of Data Structure

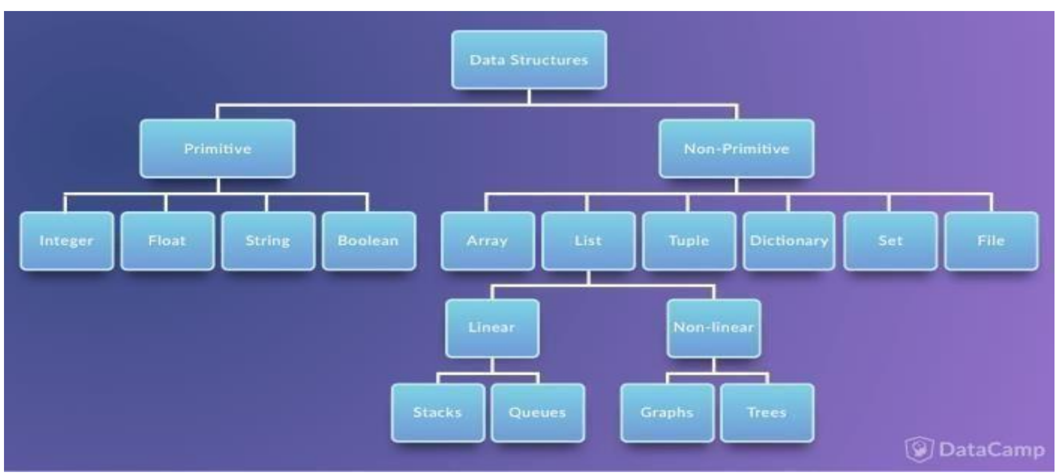


Figure 3.11 A data structure tree at glance

3.3.6 Operators

Operators are the symbols in python that are used to perform Arithmetic or logicaloperations. Following are the different types of operators in python.

3.3.6.1Arithmetic operators

Arithmetic operators carry out mathematical operations and they are mostly usedwith the numeric values.

Arithmetic operators		
Operator	Name	Example
+	Addition	A+B
-	Subtraction	A-B
*	Multiplication	A*B
/	Division	A/B
%	Modulus	A%B
**	Exponentiation	A**B
//	Quotient	A//B

Table 3. 1Arithmetic operators

Where A and B are the numeric values.

3.3.6.2 Assignment Operators

As the name decides this operators are used for assigning the values to thevariables.

ASSIGNMENT OPERATORS		
Operator	Example	may also be writ-ten
=	a = 6	a = 6
+=	a += 3	a = a + 3
-=	a -= 4	a = a - 4
*=	a *= 5	a = a * 5
/=	a /= 6	a = a / 6
%=	a %= 7	a = a % 7
//=	a //= 8	a = a // 8
**=	a **= 9	a = a ** 9
&=	a &= 1	a = a & 1

Table 3. 2 Assignment Operators

Here a is any value and number of operations are performed on this value.

3.3.6.3 Logical Operators

These operators are used to join conditional statements. Logical operators are theand, or and not operators.

Logical Operators		
Operator	Description	Example
and	if both statements are true it returns true	x <5 **and** x <10
or	if any of the two statement is true it returns true	x <4 **or** x <8
not	if the result is true it reverses the result and gives false	**not** (x <4 **and** x <8)

Table 3. 3Logical Operators

Here a is any value provided by us and on which multiple operations can be performed.

3.3.6.4 Comparison Operators

These operators are used to compare two different values.

Comparison operators		
Operator	Name	Example
==	Equal	a == b
!=	Not equal	a!=b
>	Greater than	a >b
<	less than	a <b
>=	Greater than equal to	a>= b
<=	less than equal to	a <=b

Table 3. 4Comparison operators

Here a and b are two different values and these values are compared.

3.3.6.5 Membership Operators

These operators are used to check membership of a particular value. It is used to check whether a specific value is present in the object or not.

Membership operators		
Operator	Description	Example
in	it returns a True if the value is present inside the object	a **in** b
not in	it returns a True if the value is not present inside the object	a **not in** b

Table 3. 5Membership operators

3.4 Condition Statements

3.4.1 If else statements

"The most common type of statement is the if statement. if statement consist of a block which is called as clause",[8] it is the block after if statement, it executed the statement if the condition is true. The statement is omitted if the condition is False. then the statement in the else part is printed.

If statement consist of following-

- If keyword itself
- Condition which may be True or False
- Colon
- If clause or a block of code

Below is the figure shows how If and else statements are used with description inside it.

```
In [4]: x = 32


        if x<40:
            print('x is less than 40')

        else:
            print('x is greater than 40')

        #In the above program we have if and else statements both
        #first line we have assigned value 32 to a variable x
        #second line contain keyword with condition x<40
        #If the condtion is true then the statement is printed
        #else the other condtion is executed in else statement

        x is less than 40
```

Figure 3.12 if else statement

3.4.2 elif statements

In this statement only one statement is executed, There are many cases in which there is only one possibility to execute. "The elif statement is an else if statement that always follows an if or another elif statement"[8]. The elif statement provides another condition that is checked only if any of the previous conditions were False. In code, an elif statement always consists of the following:. The only difference between if else and elif statement is that in elif statement we have the condition where as in else statement we do not have any condition.

elIf statement consist of following-

- elIf keyword itself
- Condition which may be True or False
- Colon
- elIf clause or a block of code

Below is the figure shows how elIf statement is used with description inside it.

```
In [9]: var = 't'

        if var == 'a':
            print('this is the vowel a')
        elif var == 'e':
            print('this is the vowel e')
        elif var == 'i':
            print('this is the vowel i')
        elif var == 'o':
            print('this is the vowel o')
        elif var == 'u':
            print('this is the vowel u')
        else:
            print('The value in variable var is constant')

        #In the above program we have if, else and elif statements.
        #first line we have assigned value t to a variable 'var'
        #second line contain keyword if with condition if var==a
        #If the condtion is true then the statement is printed
        #elif the other condtion is executed in elif statement in furhter lines
        #if all the conditions are not true then we come to else statement
        #and then statement in else block is printed on the output
```

```
The value in variable var is constant
```

Figure 3.13 elif example

3.5 Loops in Python

3.5.1 For loop

When do we use for loops?

for loops are traditionally used when you have a block of code which you want to repeat a fixed number of times. The Python for statement iterates over the members of a sequence in order, executing the block each time.[9]

Range statement –

This statement 'range()' is used with for loop statements where you can specify one value. For example, if you specify 10, the loop statement starts from 1 and ends with9, which is n-1. Also, you can specify the start and end values. The following examples demonstrate loop statements.

```
In [21]: for i in range(10):
             print(i)

0
1
2
3
4
5
6
7
8
9

In [ ]:
```

Figure 3.14 for example with range statement

3.5.2 While loop

While loops are used for repeating the section of code but not same as for loop, the while loop does not run n times, but until a defined condition is no longer met. If the condition is initially false, the loop body will not be executed at all.

The example is shown in figure 3.15 which says about the while loop process being performed.

```
In [30]: i=1 #we have assigned value one to a variable i
         while i<=5: #entering in while loop if condtion is true
             print(i)
             i=i+1
         print('done')
         i=2
         while i<=10:
             print('*'*i)
             i=i+1
         print('done')

1
2
3
4
5
done
**
***
****
*****
******
*******
********
*********
**********
done
```

Figure 3.15 While loop example

3.5.3 Module, Package and Functions

3.5.3.1 Module

Modules are Python files which has extension as .py. The name of the module will be the name of the file. A Python module can have a set of functions, classes or variables defined and implemented. Module has some python codes, this codes can define the classes, functions and variables. The reason behind using the module is that it organizes your python code by grouping the python code so that it is easier to use.

3.5.3.2 Package

A package consist of the collection of modules in which python codes are written with name init.py. It means that each python code inside of the python path, which contains a file named init.py, will be treated as a package by Python. Packages are used for organizing the module by using dotted names.

For Example - We have a package named simple package which consist of two modules a and b. We will import the module from package in following way.

from simple package import a, b

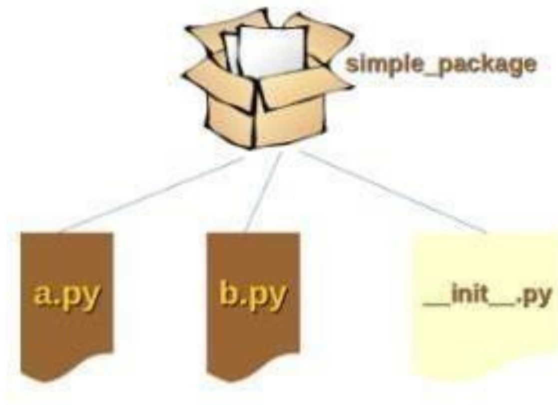


Figure 3.16 Packages Example [10]

3.5.3.3 Functions

A function is a python code which can be reused at any anytime in the whole python code. Function performs specific task whenever it is called during the program. With the help of function the program is divided in to multiple codes.

- **Built in functions** - The functions which are already in the python programming and have specific action to perform are called as built in functions. This function are immutable. Some examples of this functions are - chr() - used to get string print() - used to print an object in terminal min() - used to get minimum value in terminal
- **User defined functions** - This functions are user to defined functions and it starts with the key word 'def' as shown in the example below. We have defined the function names temperature and its task to be performed when called. Below is the example of it.

```
In [36]: def temperature():    #we have defined a function

             #this lines after the function are called function calls

                 print('Temperature is high in summer')
                 print('Temperature is low in winter')



         temperature()

         #when we called the function temperature
         #and everything inside the function is printed

         Temperature is high in summer
         Temperature is low in winter
```

Figure 3.17 Function example

3.5.4 Libraries in Python

Python library is vast. There are built in functions in the library which are written in C language. This library provide access to system functionality such as file input output and that is not accessible to Python programmers. This modules and library provide solution to the many problems in programming.

Following are some Python libraries.

- Matplotlib
- Pandas
- TensorFlow
- Numpy
- Keras
- PyTorch
- LightGBM
- Eli5
- SciPy

Libraries used for project are as follow-

- Pyttsx3
- Pywhatkit
- Datetime
- Wikipedia
- Pyjokes

3.5.5 Matplotlib

"Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy"[11]. Matlab provides an application that is used in graphical user interface tool kits. Another such libraby is pylab which is almost same as MATLAB.

It is a library for 2D graphics, it finds its application in web application servers, graphical user interface toolkit and shell. Below is the example of a basic plot in python.

```
In [10]: import matplotlib.pyplot as plt
         #we have imported matplotlib library first
         #we have imported the pyplot module from matplotlib library
         #we have give name 'plt' instead of using whole function name
         plt.plot([1,2,3],[1,3,4])
         #we used plot function to plot a graph
         #we have take simple list in plot function
         plt.xlabel('x label') #This function is used to name x axis
         plt.ylabel('y label') #This function is used to name y axis
         plt.title('basic plot') #This function used for title of grapho
         plt.show() #This function show the graph
```

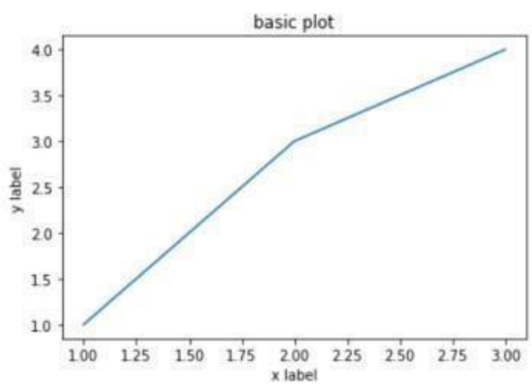


Figure 3.18 Matplotlib basic example

3.5.6 NumPy

NumPy is a library for the Python programming language, adding support for large, multidimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays”. The previous similar programming of NumPy is Numeric, and this language was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. [12] It is an open-source library and free of cost.

```
In [7]: import numpy as np  #we have imported numpy library

        data = np.arange(60)
        #arange is the range of the array function
        #we called arange function from numpy libraby
        data.shape = (10,6)
        #there will be 10 rows and 6 columns in array
        #now we have given the number of rows and columns
        print(data) #print the data
        print(len(data)) #gives length of array
        print(data.ndim) #gives dimension of array

        [[ 0  1  2  3  4  5]
         [ 6  7  8  9 10 11]
         [12 13 14 15 16 17]
         [18 19 20 21 22 23]
         [24 25 26 27 28 29]
         [30 31 32 33 34 35]
         [36 37 38 39 40 41]
         [42 43 44 45 46 47]
         [48 49 50 51 52 53]
         [54 55 56 57 58 59]]
        10
        2
```

Figure 3.19 NumPy basic example

3.5.7 Commercially Available Python Compilers

There are many compilers for python code which are using in our daily life, some of them are IDLE, SPYDER, Microsoft Visual Studio, Pycharm etc. and online compilers too. For our project we are using Pycharm compiler which is developed by JetBrains.

3.5.7.1 Pycharm

PyCharm is the most popular IDE used for Python scripting language. This chapter will give you an introduction to PyCharm and explains its features.

PyCharm offers some of the best features to its users and developers in the following aspects

- Code completion and inspection
- Advanced debugging
- Support for web programming and frameworks such as Django and FlaskFeatures of PyCharm

Besides, a developer will find PyCharm comfortable to work with because of thefeatures mentioned below −

Code Completion

PyCharm enables smoother code completion whether it is for built in or for an external package.

SQLAlchemy as Debugger

You can set a breakpoint, pause in the debugger and can see the SQL representation of the user expression for SQL Language code.

Git Visualization in Editor

When coding in Python, queries are normal for a developer. You can check the last commit easily in PyCharm as it has the blue sections that can define the difference between the last commit and the current one.

Code Coverage in Editor

You can run .py files outside PyCharm Editor as well marking it as code coverage details elsewhere in the project tree, in the summary section etc.

Package Management

All the installed packages are displayed with proper visual representation. This includes list of installed packages and the ability to search and add new packages.

Local History

Local History is always keeping track of the changes in a way that complements like Git. Local history in PyCharm gives complete details of what is needed to rollback and what is to be added.

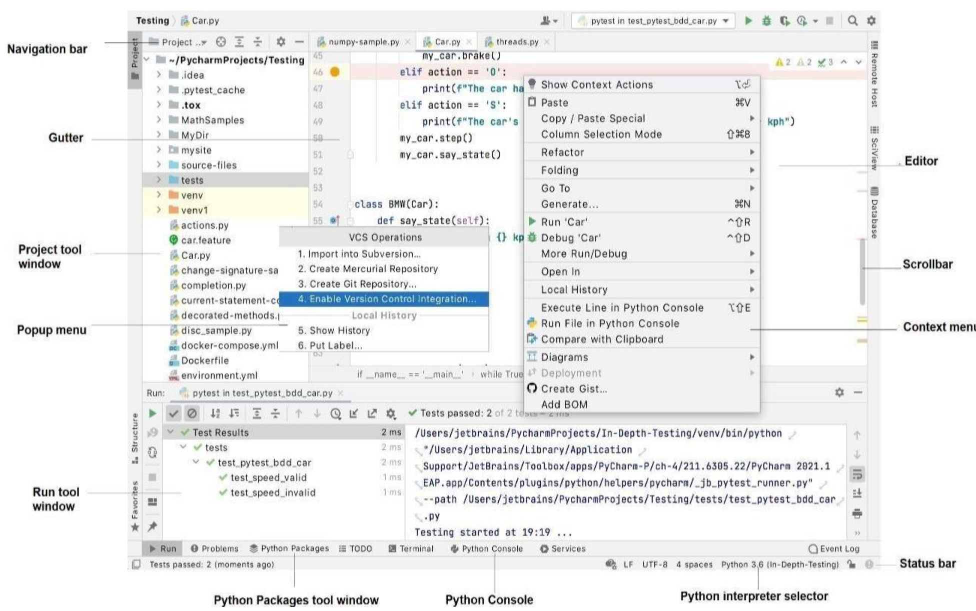
Refactoring

Refactoring is the process of renaming one or more files at a time and PyCharm includes various shortcuts for a smooth refactoring process.

3.5.7.1.1User Interface of PyCharm Editor

The user interface of PyCharm editor is shown as given below. Observe that the editor includes various features to create a new project or import from an existing project.

When you open a project in PyCharm, the default user interface looks as follows:



Depending on the set of plugins, PyCharm edition , and configuration settings, your IDE maylook and behave differently.

3.5.7.1.2 Editor

Focus: Escape

Use the editor to read, write, and explore your source code.

3.5.7.1.3 Action indicators and action list

Open action list: Alt+Enter

Different icons that appear in the left editor gutter are action indicators that help you notice available quick fixes and other actions.

Quick fixes and other commands relevant for the current caret position are available in the action list that you can open with Alt+Enter or by cliking an action indicator.

3.5.7.1.4 Navigation bar

Focus: Alt+Home

Show/hide: View | Appearance | Navigation Bar

The navigation bar at the top is a quick alternative to the Project tool window Workspace tool window where you can navigate the structure of your project and open files for editing.

Note: If VCS integration is enabled, items in the navigation bar are highlighted accordingto VCS file status colors.

Use the buttons to the right of the navigation bar to build run and debug your application, and perform basic version control operations (if the version control integration is configured). It also contains buttons to Run Anything (press Ctrl twice) and Search Everywhere (press Shift twice).

Note: The main toolbar with buttons for opening and saving files, undo and redo actions is hidden by default. To show it, select View | Appearance | Toolbar.

3.5.7.1.4 Status bar

Show/hide: **View | Appearance | Status Bar**

The left part of the status bar at the bottom of the main window shows the most recent event messages and descriptions of actions when you hover over them with the mouse pointer. Click a message in the status bar to open it in the Event Log. Right-click the message in the status bar and select Copy to paste the message text when you are

searching for a solution to a problem or need to add it to a support ticket or to the PyCharm issue tracker.

Use the quick access button to switch between tool windows and hide the tool window bars.

The status bar also shows the progress of background tasks. You can click to show the Background Tasks manager.

The right part of the status bar contains widgets that indicate the overall project and IDE status and provide access to various settings. Depending on the set of plugins and configuration settings, the set of widgets can change. Right-click the status bar to select the widgets that you want to show or hide.

View | Tool Windows

Tool windows provide functionality that supplements editing code.

For example, the Project tool window shows you the structure of your project, and the Run tool window displays the output of your application when you run it.

By default, tool windows are docked to the sides and bottom of the main window. You can arrange them as necessary, undock, resize, hide, and so on. Right-click the title of the tool window or click in the title for its arrangement options.

You can assign shortcuts to quickly access the tool windows that you frequently use. Some of them have shortcuts by default. For example, to open the Project tool window, press Alt+1, and to open the Terminal tool window, press Alt+F12. To jump from the editor to the last active tool window, press F12.

3.5.7.1.5 Context menu:

You can right-click various elements of the interface to see the actions available in the current context. For example, right-click a file in the Project tool window for actions

related to that file, or right-click in the editor to see actions that apply to the current codefragment.

Most of these actions can also be performed from the main menu at the top of the screenor the main window. Actions with shortcuts show the shortcut next to the action name.

3.5.7.1.6 Popup menus

Popup menus provide quick access for actions related to the current context. Here aresome useful popup menus and their shortcuts:

- Alt+Insert opens the **Generate** popup for generating boilerplate code based onthe context.
- Ctrl+Alt+Shift+T opens the **Refactor This** popup with a list of contextuallyavailable refactorings.
- Alt+Insert in the Project tool window opens the **New** popup for adding new filesand directories to your project.
- Alt+` opens the **VCS Operations** popup with contextually available actions foryour version control system.

You can create custom popup menus using quick lists of actions that you often use.

3.5.7.1.7 Main window

The main window contains all the information for a single PyCharm project. You can open multiple projects in multiple windows. By default, the window header contains the name of the project and the name of the currently open file. If there are multiple modules, it will also show the name of the relevant module.

Show full paths in the header

In the Settings/Preferences dialog Ctrl+Alt+S, open Appearance & Behavior |Appearance and select the Always show full paths in window header checkbox.

This will show the path to the project and to the current file.

CHAPTER 4

PROJECT IMPLEMENTATION

4.0 IMPLANTATION STAGES

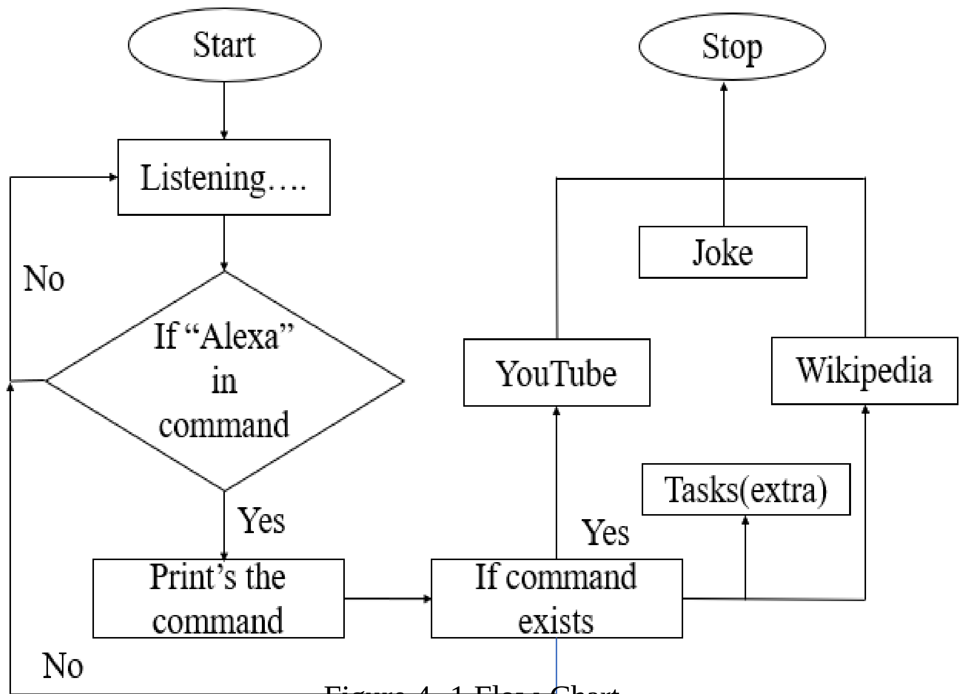


Figure 4. 1 Flow Chart

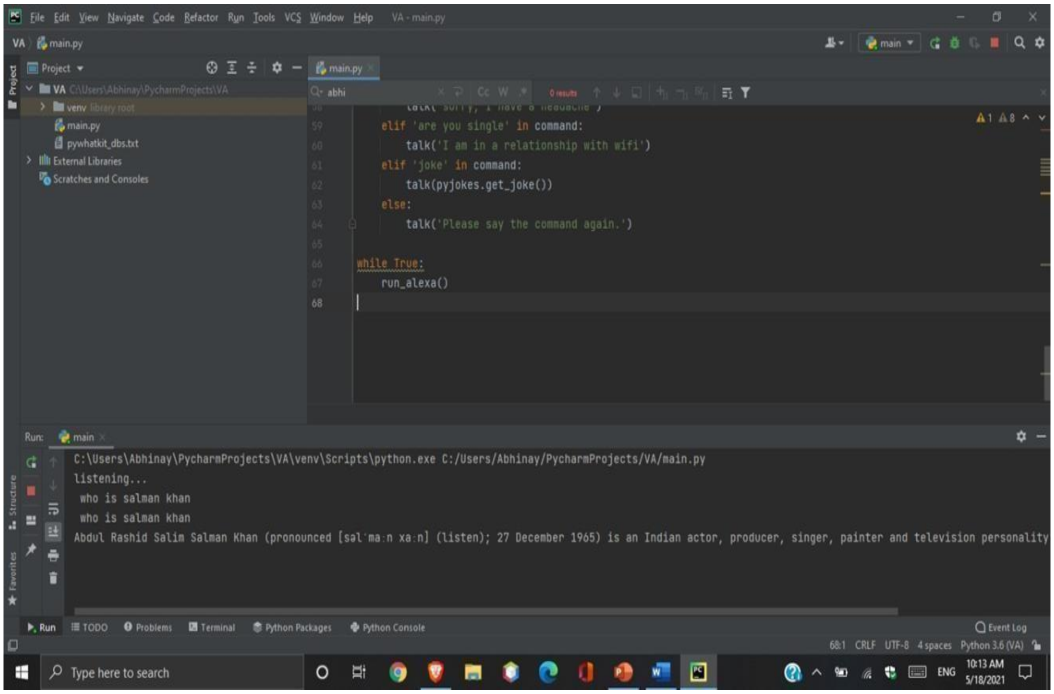
Stage 1: Switch on the PC/laptop.

Stage 2: Open pycharm community software and create new project for our virtualassistant software.

Stage 3: Write the required code for our VA using python language, which is easy to use.

Stage 4: Before running the code open terminal of software and import required librariesand download.

Stage 5: Now run the code and start talking to the assistant which will help us to do tasks.



4.1 RESULTS

Figure 4. 2 Data about a person

Here, the command given by the user is "Who is Salman Khan" with the wake word "Escobar". This is the request given by the user. The output by virtual assistant is in the form of Voice and Text out as seen in the screen.

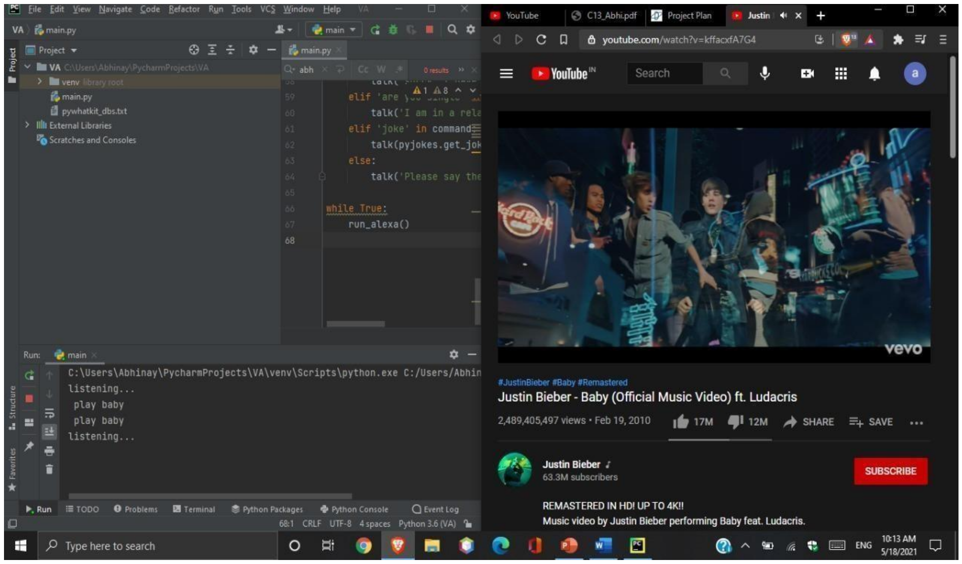


Figure 4. 3 Playing Song

Here, the command given by the user is “Play BABY” with the wake word “Escobar”. This is the request given by the user. The output by virtual assistant is in the form of Voice output by opening YouTube as seen in the screen. BABY(Song) is a trending song in internet past 10 years.

Similarly, there are many other applications like saying a JOKE, Time etc.

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

Personal assistant software improves user productivity by managing routine tasks of the user and by providing information from online sources to the user. As discussed earlier, technologies such as web services, sharing of data, linked data, shared ontologies, knowledge databases, and mobile devices are proving to be enablers for tools such as personal assistant software.

Nowadays all the virtual assistants are being integrated or manufactured at industrial standards and are given to the public. This paper deals with making their own virtual assistant at mature level later it may be done even at teen level.

PUBLICATION

- Publication : ICSMEC-21
- Paper ID : ICSMEC21-0015
- Paper Title : BUILD, OWN VIRTUAL ASSISTANT

Dear Author(s),

Warm Greetings from St.Martins Engineering College.

With heartiest congratulations we are pleased to inform you that based on the recommendations of the expert reviewers, your paper has been accepted for journal publication and oral presentation in the online mega International Conference on "Smart Modernistic in Electronics and Communication" (ICSMEC-21).

Paper ID:ICSMEC21-0015
Paper Title: BUILD, OWN VIRTUAL ASSISTANT

Paper Accepted.

Please Send the final camera ready paper Paper After Registration

REFERENCES

[1] Veton Këpuska, Gamal Bohouta, “Next-Generation of Virtual Personal Assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home) “, Published in 2018, ISBN- 99978-1-6/18 IEEE.

[2] Sumit kumar Sarda, Yash Shah, Monika Das, Nikita Saibewar, Shivprasad Patil,”VPA: Virtual Personal Assistant” Published in 2017 International Journal of Computer Applications (0975 – 8887) Volume 165 – No 1.

[3] P. Milhorat, S. Schlögl, G. Chollet, J. Boudy, A. Esposito and G. Pelosi, "Building the next generation of personal digital Assistants, "Published in 2014 1st International Conference on Advanced Technologies for Signal and Image Processing, ISBN 978 -1- 4799- 4888-8 IEEE.

[4] Gong, L.: San Francisco, CA (US) United States US 2003.01671.67A1 (12) Patent Application Publication c (10) Pub. No.: US 2003/0167167 A1 Gong (43) Pub. Date: 4 September 2003 for Intelligent Virtual Assistant

[5] Sarikaya, R.: The technology behind personal digital assistants. IEEE Signal Process.

Mag. 34, 67–81 (2017). https://doi.org/10.1109/msp.2016.2617341

[6] Tsiao, J.C.-S., Tong, P.P., Chao, D.Y.: Natural-Language Voice-Activated Personal Assistant, United States Patent (10), Patent No.: US 7,216,080 B2 (45), 8 May 2007

[7] Chung, H., Iorga, M., Voas, J., Lee, S.: Alexa, can I trust you? In: 2017 IEEEComputer Security (2017)

[8] Cowan, B.R.: What can i help you with?: infrequent users’ experiences of intelligent personal assistants. In: 2015 IEEE 10th International Conference on Industrial and Information Systems, ICIIS 2015, Sri Lanka (2015)

[9] Weeratunga, A.M., Jayawardana, S.A.U., Hasindu, P.M.A.K, Prashan, W.P.M., Thelijjagoda, S.: Project Nethra - an intelligent assistant for the visually disabled to interact with internet services.

BIBILOGRAPHY

http://en.wikipedia.org/wiki/Android-(operating-system)," n.d. http://en.wikipedia.org/wiki/Android_(operating-system). "Android System Architecture Diagram - Http://developer.android.com/images/system-architecture.jpg," n.d.http://developer.android.com/images/system-architecture.jpg.

"CALO Website - http://www.ai.sri.com/project/CALO," n.d. http://www.ai.sri.com/project/CALO.

"CALO Wiki Page - http://en.wikipedia.org/wiki/CALO," n.d. http://en.wikipedia.org/wiki/CALO.

"DBPedia Knowledge Base http://www.wiwiss.fuberlin.de/en/institute/pwo/bizer/research/publications/Mendes-JakobBizer-DBpedia-LREC2012.pdf," n.d. http://www.wiwiss.fuberlin.de/en/institute/pwo/bizer/research/publications/Mendes-akobBizer-DBpedia-LREC2012.pdf.

"DBPedia Usecases - http://wiki.dbpedia.org/UseCases," n.d.

"ESD.34 System Architecture Course at MIT - Principles of Architecture Assignment,"n.d.

"GoodRelations Ontology UML Diagram Http://www.heppnetz.de/ontologies/goodrelations/vl#uml," n.d. http://www.heppnetz.de/ontologies/goodrelations/vl#uml.

"Google Glass Website - Http://www.google.com/glass," n.d. http://www.google.com/glass. "Google Webservices Documentation

Https://developers.google.com/maps/documentation/webservices/,"
n.d.https://developers.google.com/maps/documentation/webservices/

"How SIRI Works - Interview with Tom Gruber, CTO of SIRI:
Http://www.novaspivack.com/technology/how-hisiri-works-interviewwith-
tom-

gruber-cto-of-siri," n.d
http://www.novaspivack.com/technology/how-hisiri-works-
interviewwith-tom- gruber-cto-of-siri.
"Introduction to Wordnet - an Online
Lexical Database
Http://wordnetcode.princeton.edu/5papers.pdf,"
n.d.
http://wordnetcode.princeton.edu/Spapers.pdf.

"Knowledge Databases - Http://www.mkbergman.com/913/metamodeling-
indomain-ontologies/," n.d.

http://www.mkbergman.com/913/metamodelingin-domain-
ontologies/. "Limitations in SPARQL 1.1 -
http://www.w3.org/2009/Talks/0615-qbe/," n.d.
http://www.w3.org/2009/Talks/0615-qbe/.
"Linked Open Data Cloud -
Http://richard.cyganiak.de/2007/10/lod/loddatasets_2011-09-
19_colored.png," n.d.
http://richard.cyganiak.de/2007/10/lod/lod-datasets_2011-09-
19_colored.html.

100

"OWL2 Primer - http://www.w3.org/TR/2012/REC-owl2-
primer20121211/SemanticWeb Presentation," n.d.

http://www.w3.org/TR/2012/REC-owl2-primer-

SIRI Demo - Http://vimeo.com/5424527," n.d. http://vimeo.com/5424527. "SIRI Patent Information - Http://www.patentlyapple.com/patentlyapple/2012/01/apple-introduces-us- to-siri-the-killer-patent.html," n.d. http://www.patentlyapple.com/patently-apple/2012/01/apple-introducesus-to-siri-the-killer-patent.html. "

SIRI Patent: US20120016678A1

http://www.google.com/patents?id=ISECAgAAEBAJ&printsec=abstract&zo o m=4#v=onepage&q&f=false," n.d.

http://www.google.com/patents?id=ISECAgAAEBAJ&printsec=abstract&zo o m=4#v=onepage&q&f=false.

SPARQL Cheatsheet –

http://www.slideshare.net/LeeFeigenbaum/sparql-cheatsheet," n.d. http://www.slideshare.net/LeeFeigenbaum/sparql-cheat-sheet.

"SPARQL Tutorial

Http://www.slideshare.net/ldodds/sparql-tutorial," n.d. http://www.slideshare.net/ldodds/sparql-tutorial.

"SPARQL W3C Resources - http://www.w3.org/2009/Talks/0615-qbe/," n.d.

http://www.w3.org/2009/Talks/0615-qbe/.

"Why Use Logic? - Presentation at Http://www.cyc.org," n.d. "Wordnet Reference Manual - http://wordnet.princeton.edu/man/wngloss.7WN.html," n.d. http://wordnet.princeton.edu/man/wngloss.7WN.html. "Wordnet: Design, Content andLimitations by Christian Fellbaum

http://dydan.rutgers.edu/Workshops/Semantics/slides/fellbaum.pdf," n.d. http://dydan.rutgers.edu/Workshops/Semantics/slides/fellbaum.pdf. 101 "Yelp

Webservices Documentation -
Http://www.yelp.com/developers/documentation," n.d.
http://www.yelp.com/developers/documentation.

APPENDICES

```
import speech_recognition as srimport pyttsx3
import pywhatkitimport datetime import wikipediaimport pyjokes


listener = sr.Recognizer()engine = pyttsx3.init()

voices = engine.getProperty('voices') engine.setProperty('voice', voices[1].id)

def talk(text):

engine.say(text)

engine.runAndWait()

def take_command():try:
    with sr.Microphone() as source:print('listening...')
    voice = listener.listen(source)
    command = listener.recognize_google(voice)command = command.lower()
    if escobar in command:
        command = command.replace(escobar, '')print(command)
    except:
    pass
return command




def run_escobar():
command =
        take_command()
        print(command)
if 'play' in command:
song =
```

```
        command.replace('pla
        y', '')talk('playing ' +
        song)
        pywhatkit.playonyt(so
        ng)
elif 'time' in command:
time =
        datetime.datetime.now().strftime('%I:%M
        %p')talk('Current time is ' + time)
elif 'who the heck is' in command:
person = command.replace('who the heck
        is', '')info =
        wikipedia.summary(person, 1)
        print(info)
talk(info)
elif 'date' in command:
        talk('sorry, I have a
        headache')
elif 'are you single' in command: talk('I am in a relationship with wifi')
elif 'joke' in command: talk(pyjokes.get_joke())

else:
    talk('Please say the command again.')
    while True:
        run_escobar
```

A

Project Report

On

BROTH CULTURE ACIDITY CONTROL SYSTEM ON NFT-BASED HYDROPONIC PLANTS

Submitted by

1) K. Jagadeesh (17K81A04F2)

2) Nikhil Sharma (17K81A04G0)

3) V. Hareesh (17K81A04H6)

In partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

UNDER THE GUIDANCE OF

Ms.D.Kirtana, M.Tech.

Assistant Professor

DEPARTMENT OF Electronics and Communication Engineering

ST.MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

ST.MARTIN'S ENGINEERING COLLEGE

An UGC Autonomous Institute

Dhulapally, Secunderabad – 500 100

NBA & NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the project entitled **Broth Culture Acidity Control System On NFT-Based Hydroponic Plants**, is being submitted by **K.Jagadeesh (17K81A04F2), Nikhil Sharma(17K81A04G0), V.Hareesh (17K81A04H6)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **Electronics and Communication Engineering** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Internal Guide	Head of the Department
Ms.D. Kirtana	Dr.B. Hari Krishna
Department of ECE	Department of ECE
Internal Examiner	External Examiner

Place:

Date:

TUESDAY, 15 JUNE 202

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **HAREESH** WITH ROLL NO.**17K81A04H6, JAGADISH** WITH ROLL NO.**17K81A04F2**, **NIKHIL SHARMA** WITH ROLL NO.**17K81A04G0,** OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**BROTH CULTURE ACIDITY CONTROL SYSTEM ON NFT- BASED HYDROPONIC PLANTS**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT

DIRECTOR

TRAININGS & PLACEMENTS LASYA IT
SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad
500014 Email : contact@lasyainfotech.com, ov@lasyainfotech.com

Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of Bachelor of Technology in Department of 'Electronics and Communication Engineering ', session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled as "Broth Culture Acidity Control On NFT –based Hydroponic Plants" , is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

K.Jagadeesh	**Nikhil Sharma**	**V.Hareesh**
(17K81A04F2)	**(17K81A04G0)**	**(17K81A04H6)**

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna**, **Head & Professor**, the Dept. of **Electronics and Communication Engineering**, for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in our memory for years to come.

We would like to express our gratitude to **our dept. project coordinator Dr.A.Anand**, for his involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time to come.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri, Professor in IT and Dr.R.Santosh Kumar, Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work.

1. K.Jagadeesh
2. Nikhil Sharma
3. V.Hareesh

ABSTRACT

Nutrient Film Technique (NFT) hydroponic cultivation could be an alternative to overcome the increasing food need. Nutritional needs supply of hydroponic plant are one of parameters that need to be considered. NFT hydroponic nutrient controlling system aims to facilitate farmers to maintain the amount of nutrition according to plant needs. This system uses linear regression method to control the amount of nutrients. The accuracy results obtained from controlling system experiment are 87.84%.

)

Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

1.1 Overview of the project:

There is an increasing need to re-circulate and reuse nutrient solutions in order to reduce environmental and economic costs. However, managing the nutrient solution is one of biggest challenges in hydroponics. Many research scientists dump out nutrient solutions and refill at weekly intervals. Some authors have recommended automated measurement and control of individual nutrients in solution as essential to nutrient control. However, two decades of research in hydroponics has shown us that dumping and replacing solution is unnecessary. Monitoring ions in solution at frequent intervals is extremely expensive and not always necessary; in fact the rapid absorption and consequent depletion of some nutrients often causes people to add toxic amounts of nutrients to the solution.

Managing nutrients by mass balance. During the past 18 years, we have managed nutrients in closed hydroponic systems according to the principle of "mass balance," which means that the mass of nutrients is either in solution or in the plants. We add nutrients to the solution depending on what we want the plant to take up. Plants quickly remove their daily ration of some nutrients while other nutrients accumulate in the solution. This means that the concentrations of nitrogen, phosphorous, and potassium can be at low levels in the solution (0.1 mM or a few ppm) because these nutrients are in the plant, where we want them. Maintaining a high concentration of nutrients in the solution can result in excessive uptake that can lead to nutrient imbalances. For example, the water removed from solution through transpiration must be replaced and it is necessary to have about 0.5 mM phosphorous in the refill solution. If the refill solution was added once each day, the phosphorous would be absorbed by the plant in a few hours and the solution phosphorous concentration would be close to zero. This does not indicate a deficiency; rather it indicates a healthy plant with rapid nutrient uptake. If phosphorous was maintained at 0.5 mM in the recirculating solution, the phosphorous concentration in the plant could increase to 1% of the dry mass, which is 3 times higher than the optimum in most plants. This high phosphorous level can induce iron and zinc deficiency (Chaney and Coulombe, 1982). Feeding plants in this way is like the daily feeding of a pet dog, some dogs would be seriously overweight if their food bowls were kept continuously full.

1.2 Objectives of the Study:

High yielding and high grade of crops are essential in modern day agriculture, this can only be achieved by smart farming technology which is used for making farms more intelligent in sensing its controlling parameters. Manual monitoring is in practice which is a very trivial task because the plants may die out if there is no proper care is taken. The architecture of this hydroponic system which is fully automatic that can be integrated into the agricultural curriculum while introducing business skills. The automatic monitoring and control of the environmental events such as light intensity, pH, electrical conductivity, water temperature, and relative humidity is carried out by lodging sensors and actuators onto the system. The maintenance and automated monitoring are done by the intervention of the IoT that are used to transfer and retrieve data to the internet (mass storage) and a mobile app is used to communicate the current status of the hydroponic system to the user through the use of internet to their mobile phones. This futuristic system can use high data analytics and prolonged data gathering to improve the accuracy of reckoning.

1.3 Scope of the Study:

Hydroponic is a method where the crops are grown in the absence of soil the nutrients that are acquired from the soil are given to them artificially. The term Hydroponics was acquired from the Greek words ‘hydro’ means water and ‘ponos’ means labour. This soil less culture of originating crops often involves their roots to be immersed in the nutrient solution along with some gravels or perlite medium. The maximum yield is achieved by the supply of sufficient quantity of nutrients and optimum microclimatic conditions are the main goal of hydroponics. Since soil is excluded from production process there will not be any problem related to soil borne diseases, pests and weeds. By the exclusion of these problems, there will not be any usage of harmful plant protection chemicals, so that there is a fresh and healthy yield of crops by the hydroponic method. The set-up of hydroponic only demands limited space and limited quantity of water as they recirculate and reuse the water. This eliminates the problems that are caused by soil. This limited space requirement also favours hydroponic as it can be accommodated in terraces, balconies and courtyards. So, there is a high probability of growing crops in urban areas, where cultivable land is limited. Hydroponics does not cause any adverse effect on the quality of fruits and flowers produced by it.

1.4 Methodology:

The seed of the desired crop is used and placed in the crop bed in phase-2 manually. A domestic power source is used to power the system. The user can select the crop that is to be planted from his mobile application which is connected through IoT. After the crop selection is done, the water is pumped from the main tank to the automatic nutrient mixing tank, the water pump stops pumping once the water level is reached. Here the water is mixed with the nutrients inappropriate proportions according to selected crop. After the completion of this process the user is notified through the mobile application. Users can also see level of nutrients present in water through his mobile application. The nutrient rich water is then flowed through pH tank with the help of solenoid valve. The user is notified once the pH tank is filled with the nutrient rich water. The pH tank has a pH sensor which monitors the pH of the water. If the pH of water is undesirable, then the system sends an alert to the user that pH is not in correct proportion and asks the user to enable the solenoid valve S2 with 'YES' or 'NO'. If the answer is 'YES' than the solenoid valve S2 opens and the water is poured out to reuse tank. If the pH of water is in correct proportion then the water is oxygenated by the oxygen pump. Once the water is completely oxygenated the user gets notified that pH is OK and water is oxygenated and ready to flow to the plants. Water pump P2 is used to pump water to the plants as well as from the plants by this way the water is circulated. The temperature and humidity of the environment is measured by temperature sensor and the readings are shown in the mobile application. Normally a camera is used to monitor the growth of the plants and also looks for any infection on the plants, these processes are updated to the user through the mobile application. Once the plants are ready to harvest, system sends the notification, your plant is ready to harvest. If the camera spots any infection in plants due to insects or disease, this turns the system into red alert by sending the red alert message to the user via application. Here digital image processing is used to point out the infection on the plants, the user is notified once the infection is detected. The system compares this image with the data in the cloud. If the symptoms of the image taken with the existing data are similar, then the system identifies the name of the disease or insect that has infected the plant and sends the complete details about the infection with its cure to the user via application.

CHAPTER-2

LITERATURE SURVEY

2.1 Key concept:

Hydroponics or soilless cultivation has been widely used in different countries because of its feasibility and environmental safety. This technique can be considered as the best alternative in areas where serious soil and water problems like soil born pests and diseases, soil and water salinity, chemical residues in soil and water, shortage of water etc. exist. In hydroponics, plants are grown by directly supplying optimum amount of nutrients in water. Composition of nutrient solution, electrical conductivity, pH and oxygen concentration have direct influence on the yield and quality of crops grown under hydroponics. If any of these factors are non-optimal, crops expresses stress symptoms.

Comparison of Land, Water and Energy requirement of Lettuce grown using Hydroponic cultivation vs conventional cultivation

Hydroponics is more efficient compared to traditional land based agricultural methods. In the case of lettuce, hydroponics offered 11 ± 1.7 times higher yields per area and saved up to 95% of water (20 ± 3.8 L/kg/y of hydroponics vs. 250 ± 25 L/kg/y of traditional land based)– mainly lost through evaporation, if not absorbed. The downside is that it required a tremendous amount of energy (90,000 ± 11,000 kJ/kg/y) compared to the 1100 ± 75 kJ/kg/y of traditional land based.

The literature work regarding the project was started in the initial weeks of 2021, few fundamental publishing and the writings helped to understand about the subject and it's evolution over time and even it's limitations. The present subject deals with, how to make the existing system more efficient by employing Embedded System technologies.

1. The writing work By Stephanie Mohammed , in the year 2018, titled as 'Tomorrow's Agriculture "NFT Hydroponics"-- Grow Within Your Budget'. This brilliant piece of work, highlights this culture of growing as one of the most efficient agricultural practices for a sustainable market using particularly NFT which opened doors for further research in the field.
2. D. Yolanda, H. Hindersah, F. Hadiatna and M. A. Triawan, "Implementation of Real-Time Fuzzy Logic Control for NFT-Based Hydroponic System on Internet of Things Environment", *ICSET*,2016, ***in desta's research, fuzzy logic was used** *
3. R. E. Saputra, B. Irawan and Y. E. Nugraha, "System design and implementation automation system of expert system on hydroponics nutrients control using forward chaining method",2018, * **As per the research an expert system is used with a forward chaining method that takes into account and draws conclusions from input parameters on the condition of nutrient solution** *

4. M. I. Alipio, A. E. M. Dela Cruz, J. D. A. Doria and R. M. S. Fruto, "A smart hydroponics farming system using exact inference in Bayesian network", *2017* , ***In Melchizedek's research an exact inference in Bayesian Network used to developed a smart hydroponics system***
5. C. Ramos, L. Nobrega, K. Baras and L. Gomes, "Experimental NFT hydroponics system with lower energy consumption", 2019, ***This technique put the root of a plant in a shallow, water that is rich with nutrients needed by the plant***
6. A. Nursyahid, T. Aprilian, T. A. Setyawan, Helmy, A. S. Nugroho and D. Susilo, "Automatic sprinkler system for water efficiency based on lora network", *2019,* ***Besides, with advancements in IoT, automatic controlling and monitoring the plants will become easier to perform***

.

CHAPTER-3

PROJECT DESIGN

3.1 Introduction to Embedded System:

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious. All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.

The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes.

The applications software on such processors is sometimes referred to as firmware.

The simplest devices consist of a single microprocessor (often called a "chip”), which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine.

As the embedded system is the combination of both software and hardware

3.2 Block diagram of embedded system:

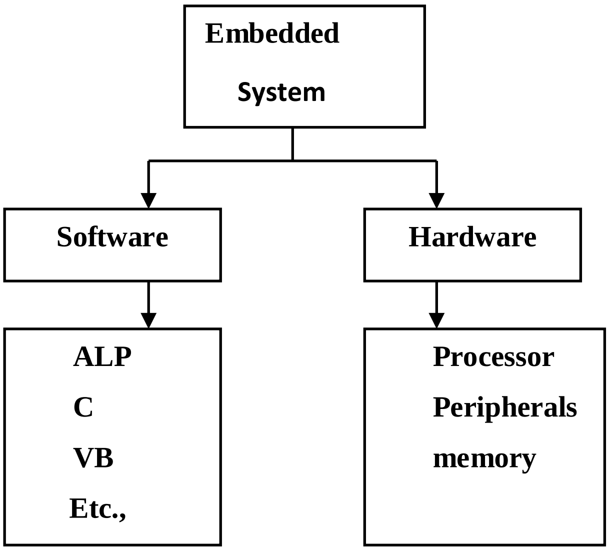


Fig 1: Block diagram of embedded system

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory

3.3 Applications of embedded systems:

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

3.4 Micro Processor (µp):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Processors are classified into four types like:

- Micro Processor (µp)
- Micro controller (µc)
- Digital Signal Processo1r (DSP)
- Application Specific Integrated Circuits (ASIC)

3.5 Three Basic Elements of a Microprocessor:

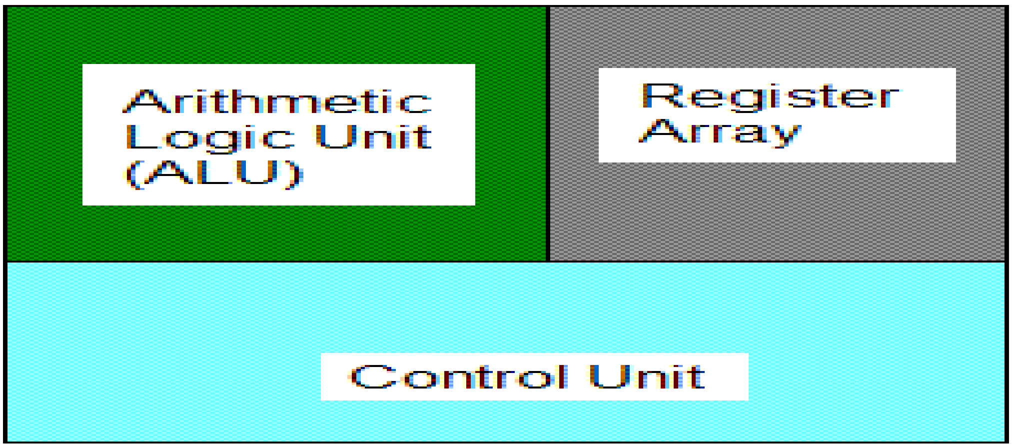


Fig 2:Three basic elements of a microprocessor

3.6 Harvard Architecture:

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electromechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

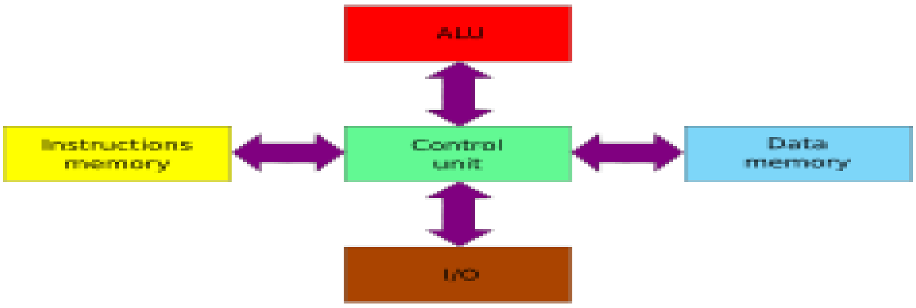


Fig 3: Harvard Architecture

3.6.1 Uses of the Harvard architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

➢ Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behaviour must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.

➢ Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

3.6.2 Von-Neumann Architecture

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. They cannot be performed at the same time. The von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A stored-program digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

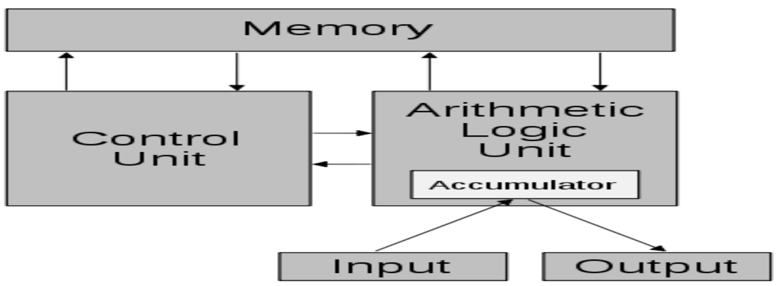


Fig 4: Schematic of the Von-Neumann Architecture

3.7 ATMEGA328

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The ATmega8 microcontroller contains 32 general purpose working registers. As shown in the below figure these registers are directly connected to ALU. Two registers can carry one single instruction consequently in one clock cycle.

Specifications:

Table 1: Atmega328 specifications

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 Ma
DC Current on 3.3V Pin	50 Ma
Flash Memory	32 KB (0.5 KB is used for Boot loader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

3.7.1 Arduino:

Arduino is a prototype platform (open source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

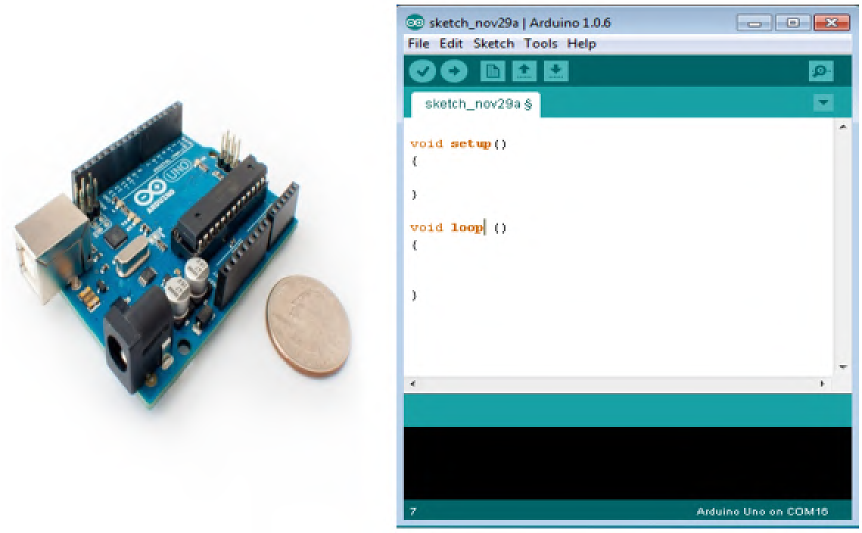


Fig 5: Arduino Uno

3.8 Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Table 2: Arduino boards based on ATMEGA328 microcontroller

Board Name	**Operating Volt**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header

Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compati
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3: Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 4: Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header

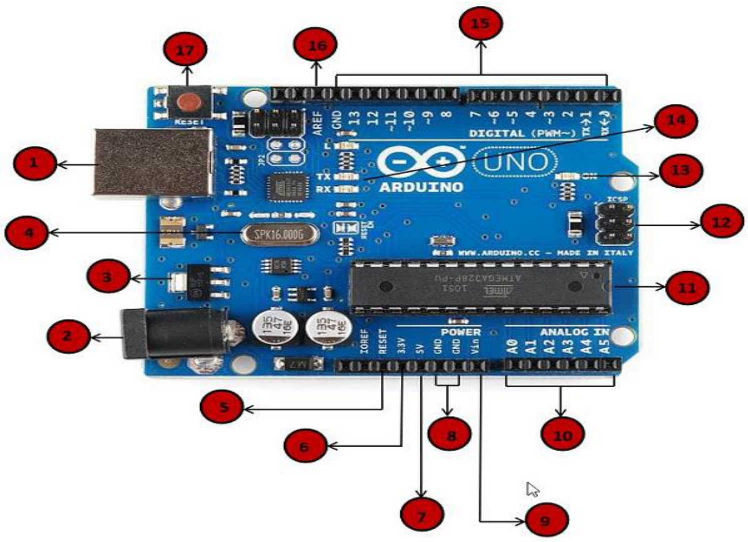
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 5: Arduino boards based on AT91SAM3X8E microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

3.8.1 Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.



1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage

	(between 0 and 5 Volts) as the upper limit for the analog input pins.

3.9 Arduino Family

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project , check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

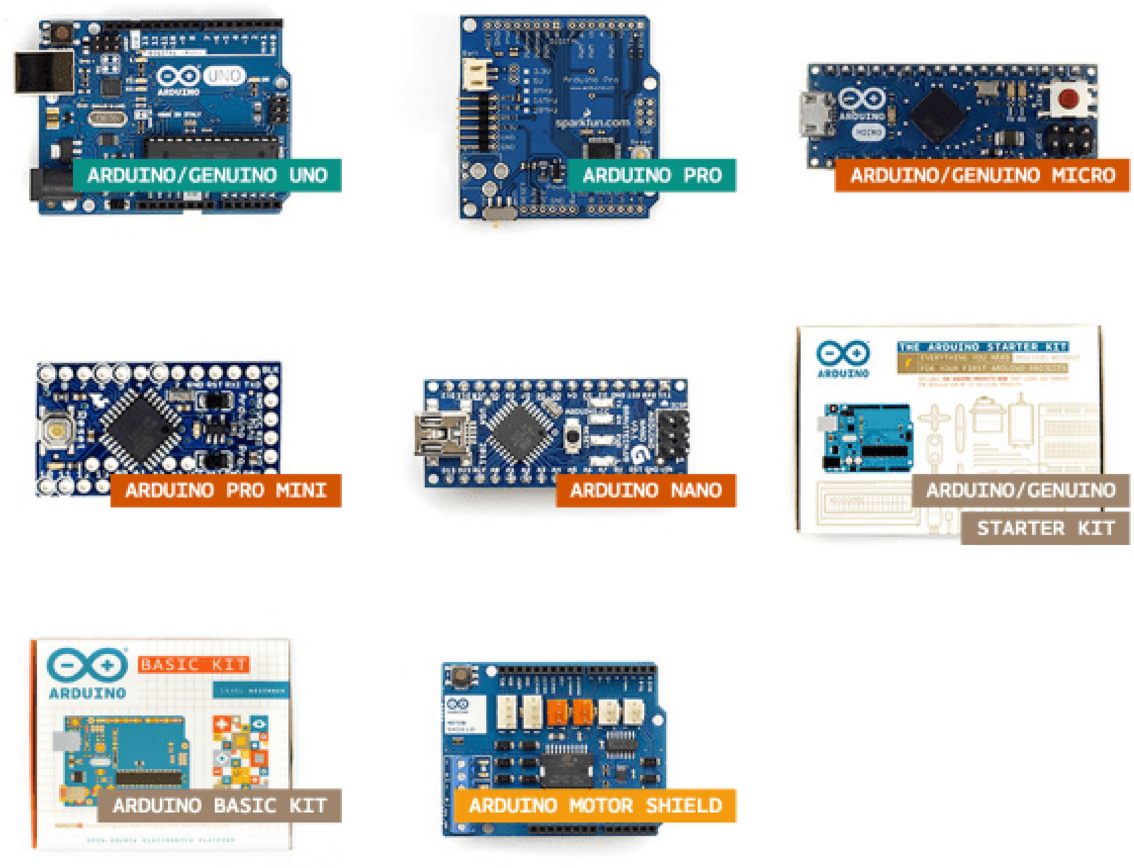


Fig 6: Arduino Family

3.10 Shields

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

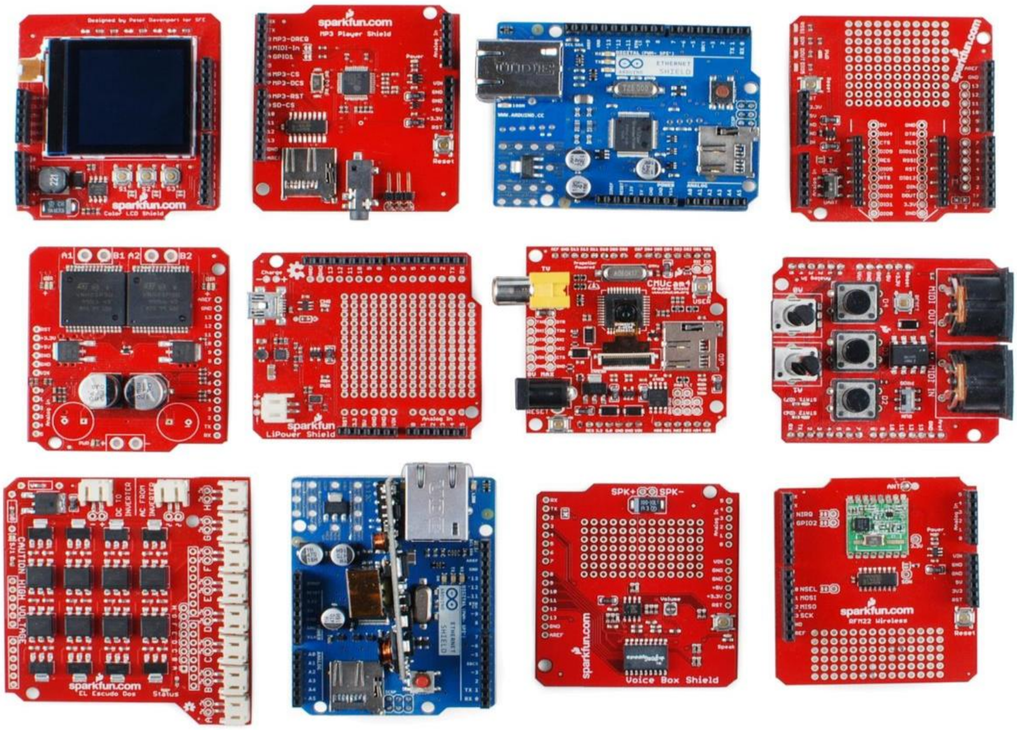


Fig 7: Arduino Shields

3.11 Pin description of ATMEGA328

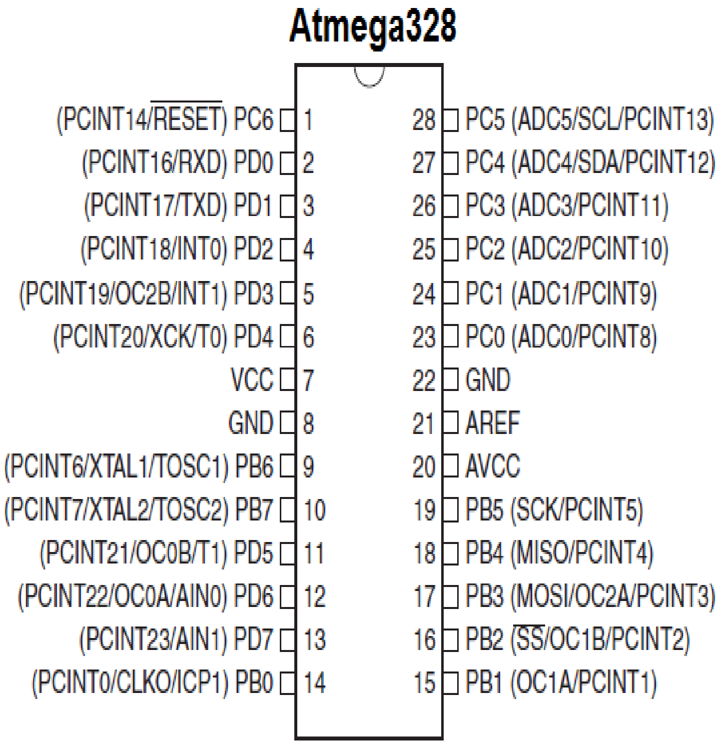


Fig 8: Pin description of ATMEGA328

3.12 Advantages of arduino

- It is cheap
- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
- For beginners, it is very simple to use.

3.13 Applications

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

3.14 Power Supply Unit

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as "Regulated D.C Power Supply".

3.14.1 Block Diagram Of Power Supply:

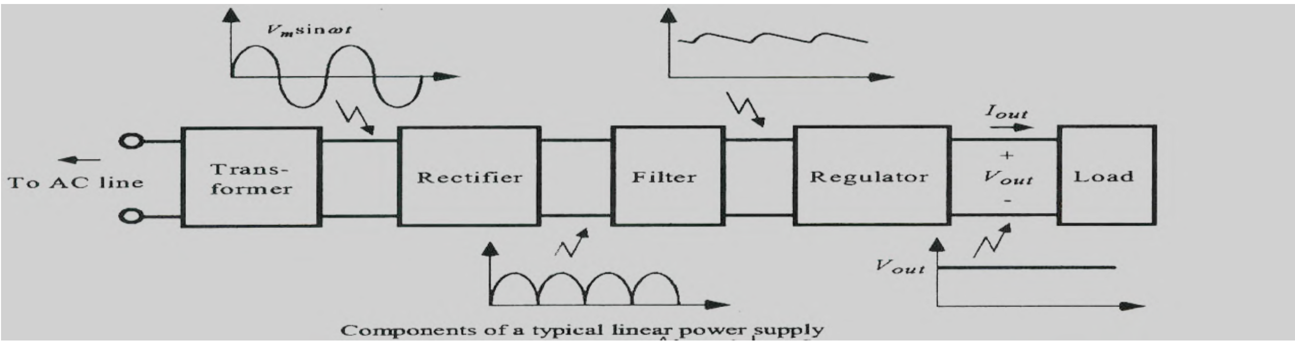


Fig 9: Block Diagram of Power Supply

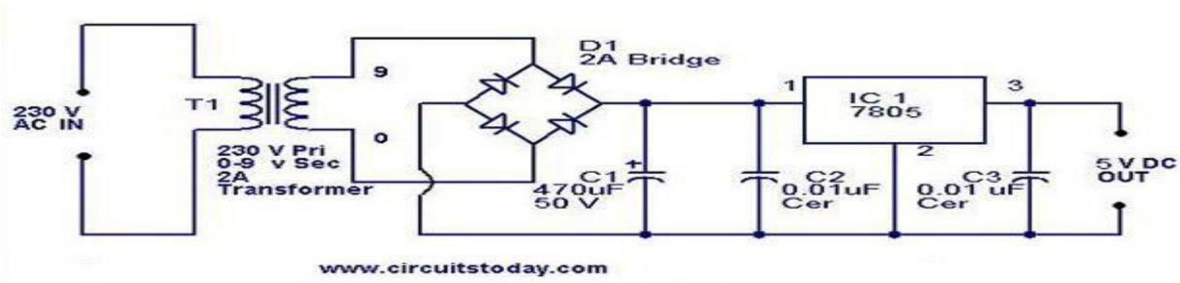


Fig 10: Schematic Diagram of Power Supply

3.14.2 Description Of Power Supply:

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a a different form of energy – such as solar, mechanical, or chemical - into electrical energy.

A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered. For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

3.15 Transformer:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

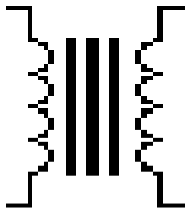


Fig 11: Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

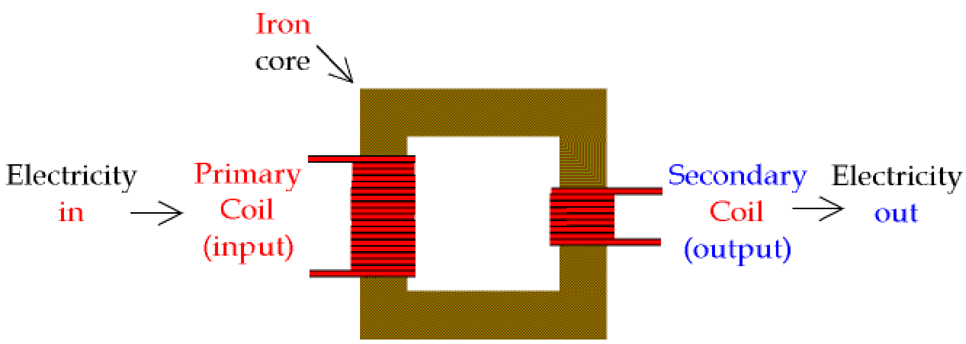


Fig 12: Transformer

3.15.1 Basic Principle of Transformer:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

From Faraday's Law

$$\frac{V_S}{V_P} = \frac{N_S}{N_P}$$

For ideal transformer

The voltage ratio is equal to the turns ratio, and power in equals power out.

From conservation of energy

$$P_P = V_P I_P = V_S I_S = P_S$$

3.15.2 Working of Transformer:

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

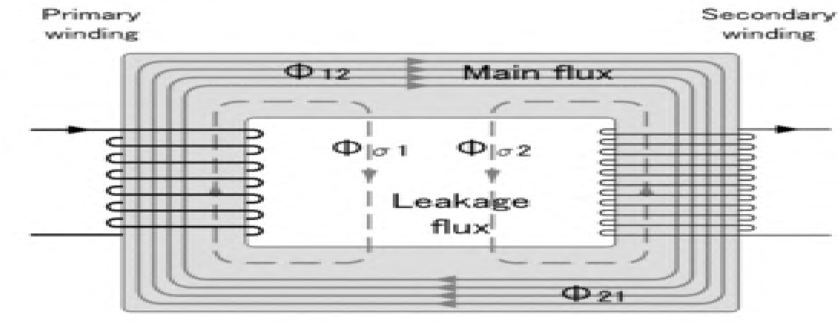


Fig 13: Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f. The iron core is laminated to prevent 'eddy currents' flowing in the core. These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

3.15.3 Classification Of Transformer:

- Step-Up Transformer
- Step-Down Transformer

Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn’s ratio of the two sets of windings determines the amount of voltage transformation.

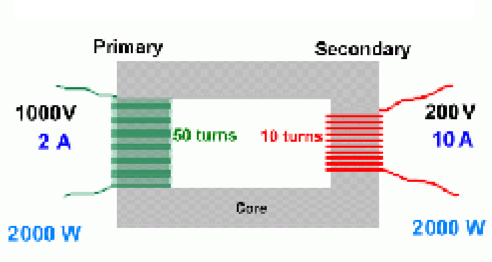


Fig 14: Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single phase steps down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase steps up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

Step-Up Transformer:

A step up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply. A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

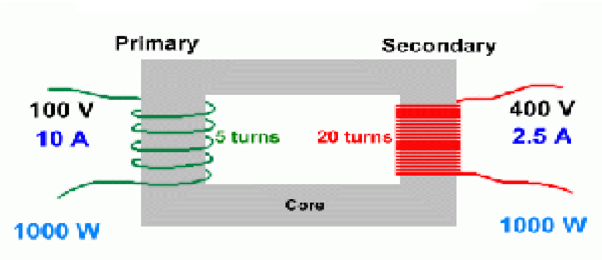


Fig 15: Step-Up Transformer

3.16 Diodes:

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

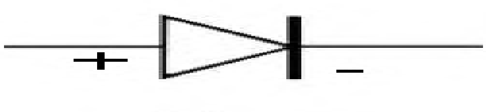


Fig 16: Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

3.17 Rectifier:

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

3.17.1 The Half-Wave Rectifier:

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure.

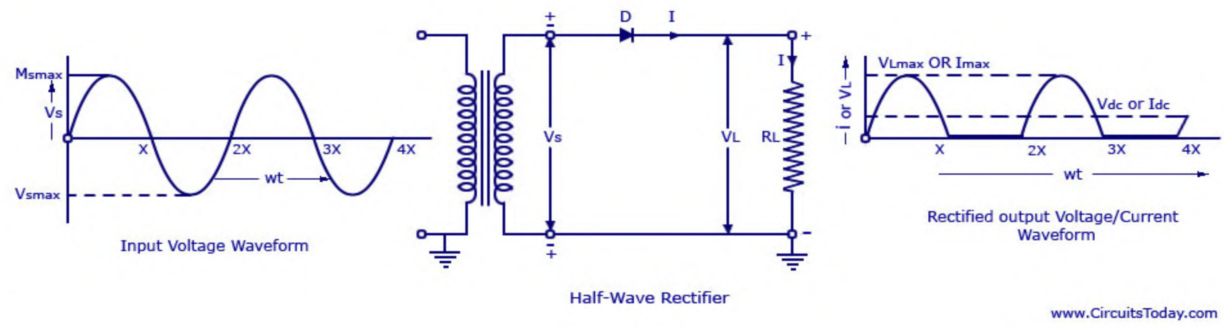


Fig 17): Half Wave Rectifier

Figure 2 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

3.17.2 The Full-Wave Rectifier:

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC.

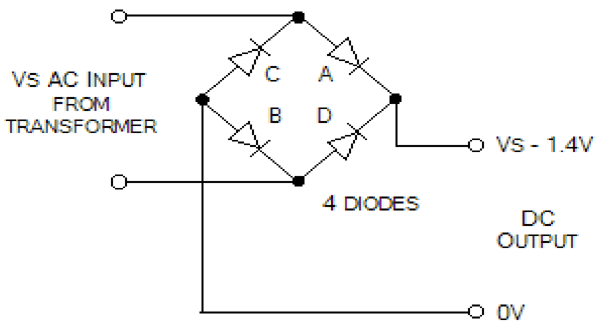


Fig 18: Full-Wave Rectifier

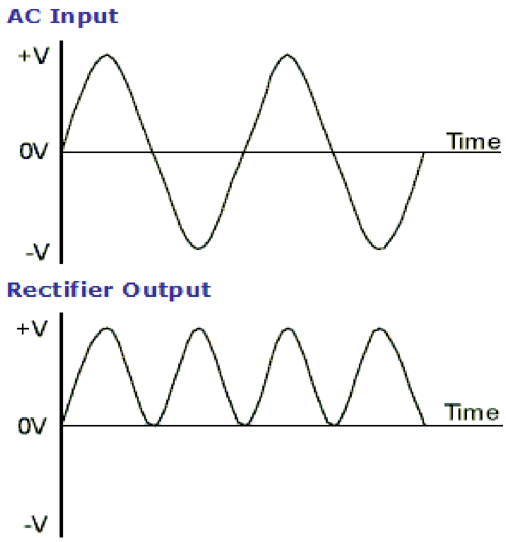


Fig 19): Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased. While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

3.18 Capacitor filter:

The capacitor-input filter, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

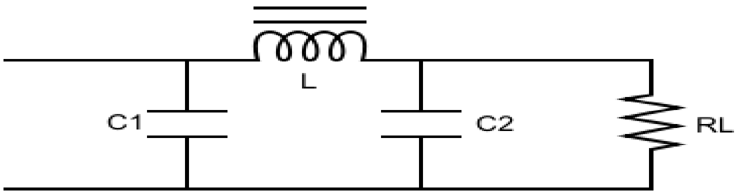


Fig 20 : Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result, the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result, only the DC component appears across the load RL.

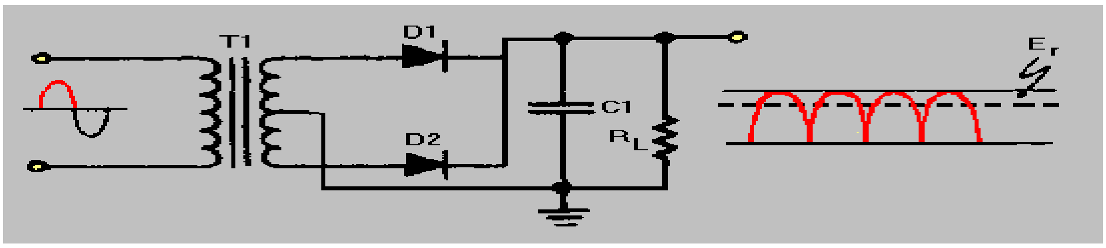


Fig 21: Centered Tapped Full-Wave Rectifier with a Capacitor Filter

3.19 Voltage regulator:

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx'indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx'indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consist the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

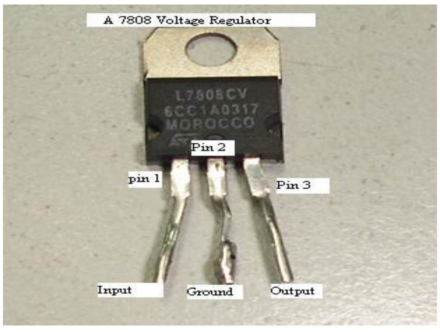


Fig 22: Regulator

CHAPTER-4

PROJECT IMPLEMENTATION

4.1 Hardware Requirement:

- Arduino Uno
- LCD
- Relays
- AC Pump
- DC Pumps
- Ultrasonic Sensor
- TDS Control
- pH Sensor
- WIFI Module

4.2 Block Diagram Of Projet:

Block Diagram:

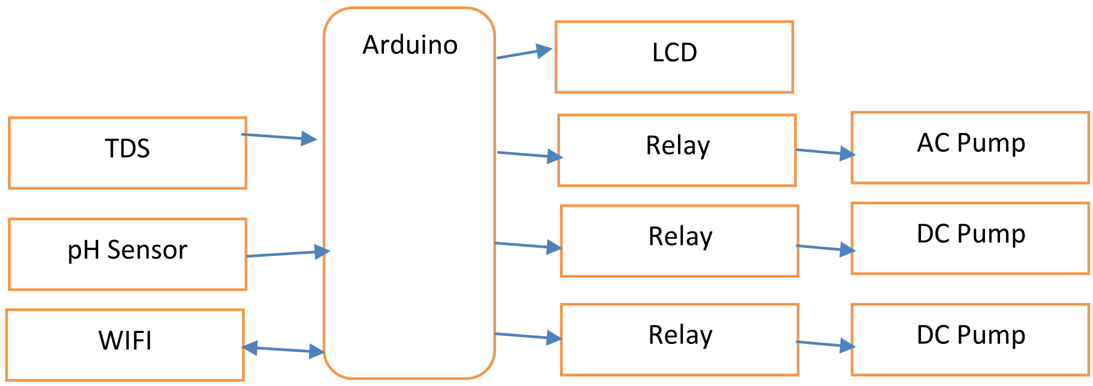


Fig 23: Block diagram of the project

4.3 ESP8266 WIFI Module:

The **ESP8266 Wi-Fi Module** is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your **Wi-Fi** network. The **ESP8266** is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

4.3.1 Character configuration:

- 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2),
- general-purpose input/output (16GPIO),
- Inter-Integrated Circuit (I²C) serial communication protocol,
- Analog-to-digital conversion (10-bitADC)
- Serial Peripheral Interface (SPI) serial communication protocol,
- I²S (Inter-IC Sound) interfaces with DMA(Direct Memory Access) (sharing pins with GPIO),
- UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2),and
- Pulse-width modulation (PWM).

4.3.2 Applications:

- Smart Home Appliances
- Improved Home safety and security
- Home air quality and water quality monitoring
- Natural Language-based voice assistants

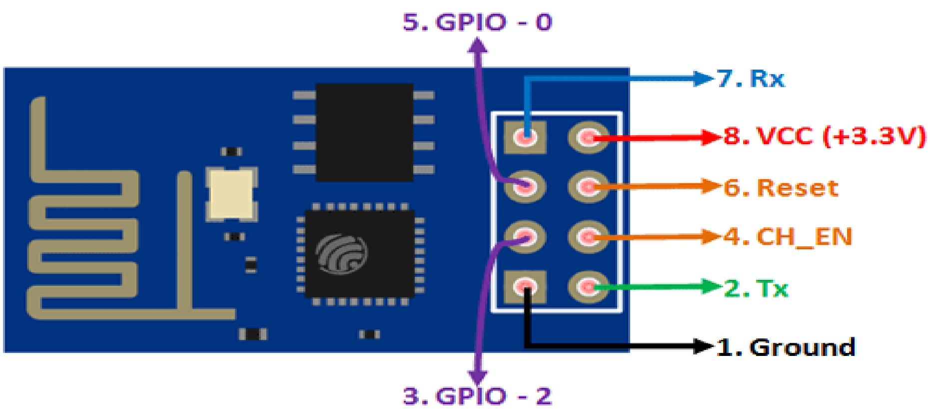
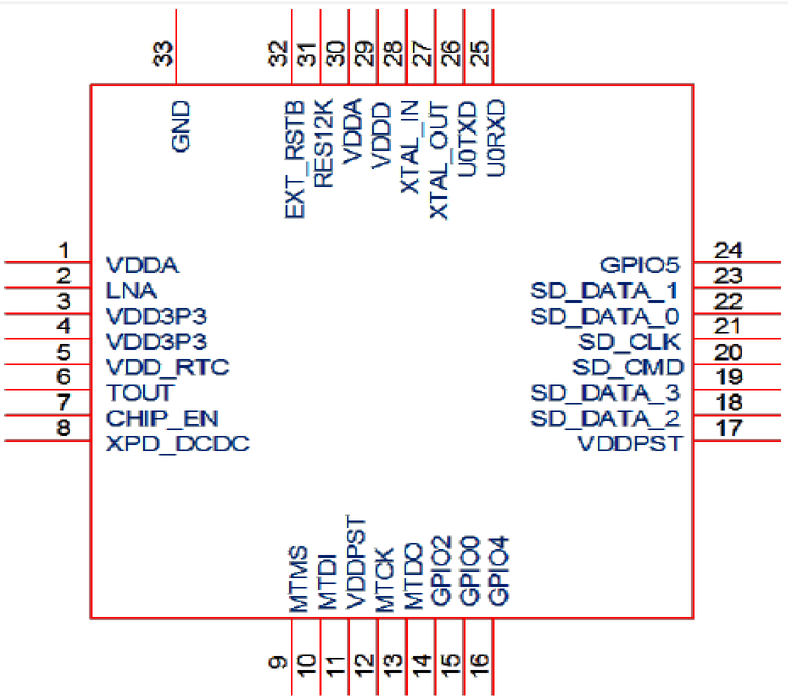


Fig 24: ESP8266 wifi module pin out

4.3.3 Structure and configuration:

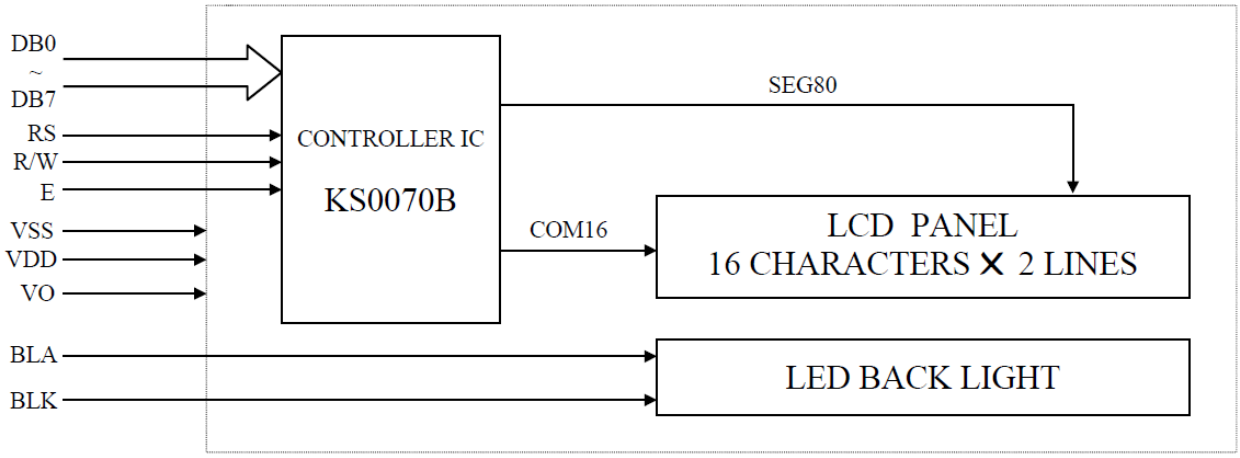


4.4 16 * 2 Alphanumeric LCD:

4.4.1 Description

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

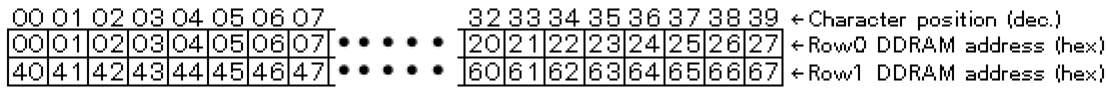
4.4.2 Block Diagram



4.4.3 Display Data RAM

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.



DDRAM Address for 2 Line LCD

4.4.4 Character Generator ROM

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 4 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. Userdefined character patterns are also available by mask-programmed ROM.

4.4.5 Busy Flag

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

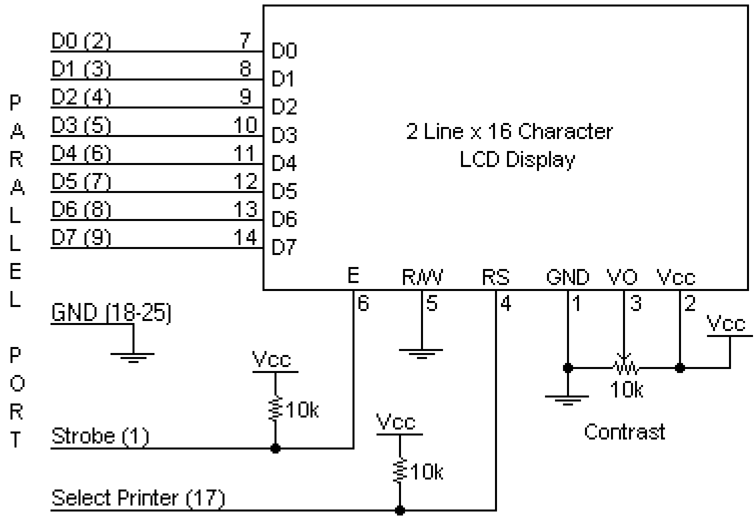
4.4.6 Instruction Register(IR) and Data Register(DR)

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD.
Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

4.4.7 16 x 2 Alphanumeric LCD Module Features

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

4.4.8 Schematic



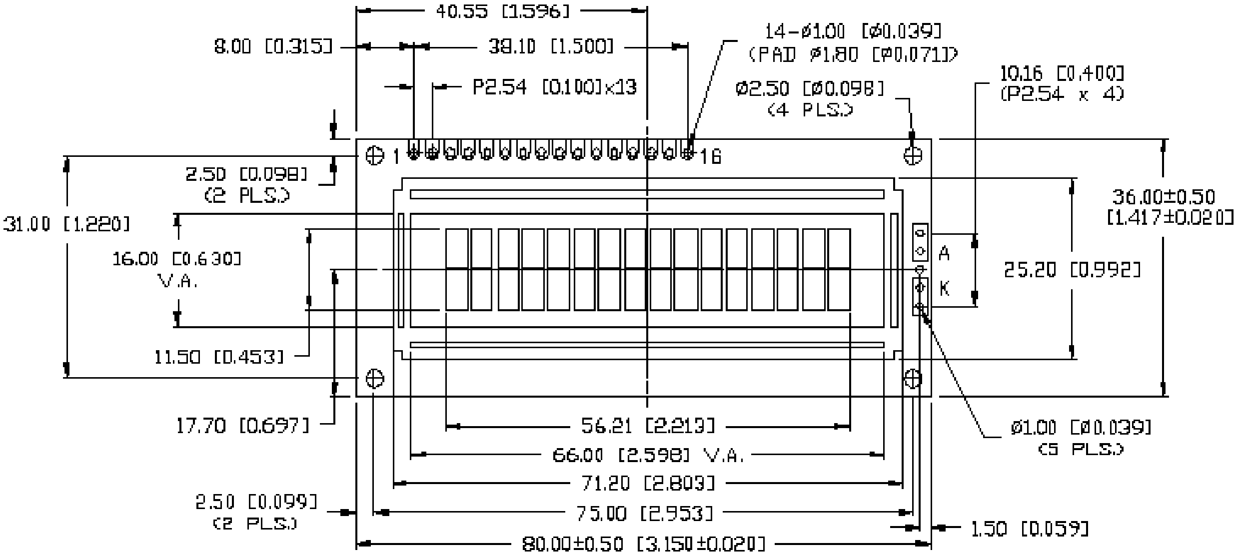
4.4.9 Specifications

Connector Pin Assignment:

Pin	*Symbol*	*Function*	*Pin*	*Symbol*	*Function*
14	DB7	Data Bus Line	6	E	Enable
13	DB6	Data Bus Line	5	R/W	Read/Write
12	DB5	Data Bus Line	4	RS	Register select input
11	DB4	Data Bus Line	3	Vcontrast	Contrast (0V for max contrast)
10	DB3	Data Bus Line	2	**GND (0V)**	Power supply
9	DB2	Data Bus Line	1	**Vcc +5V**	Power supply
8	DB1	Data Bus Line			
7	DB0	Data Bus Line			

Note1: Pin 1 is +5V and pin 2 is GND! This is different from most other HD44780 compatible models.

Note2: Pin 14 is marked with a “1” on the back of the PCB. Do not get confused by this.



Circuit Description

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

4.5 Relay

A relay is used to isolate one electrical circuit from another. It allows a low current control circuit to make or break an electrically isolated high current circuit path. The basic relay consists of a coil and a set of contacts. The most common relay coil is a length of magnet wire wrapped around a metal core. When voltage is applied to the coil, current passes through the wire and creates a magnetic field. This magnetic field pulls the contacts together and holds them there until the current flow in the coil has stopped. The diagram below shows the parts of a simple relay.

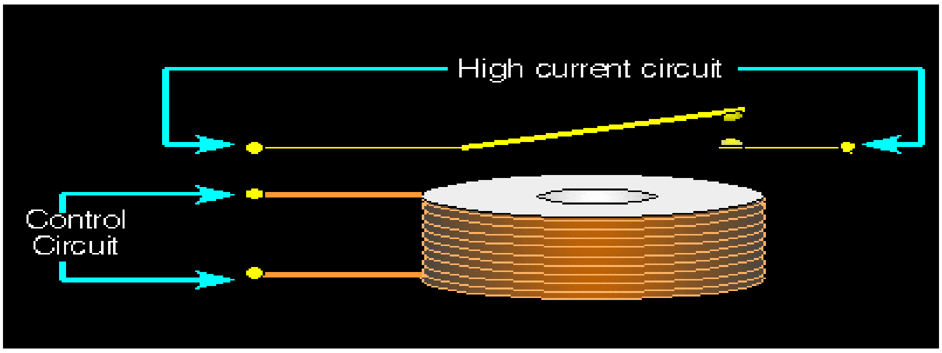


Figure:5.6 Relay

Operation:

When a current flows through the coil, the resulting magnetic field attracts an armature that is mechanically linked to a moving contact. The movement either makes or breaks a connection with a fixed contact. When the current is switched off, the armature is usually returned by a spring to its resting position shown in figure 6.6(b). Latching relays exist that require operation of a second coil to reset the contact position.

By analogy with the functions of the original electromagnetic device, a solid-state relay operates a thyristor or other solid-state switching device with a transformer or light-emitting diode to trigger it.

4.5.1 Pole and throw

SPST

SPST relay stands for Single Pole Single Throw relay. Current will only flow through the contacts when the relay coil is energized.

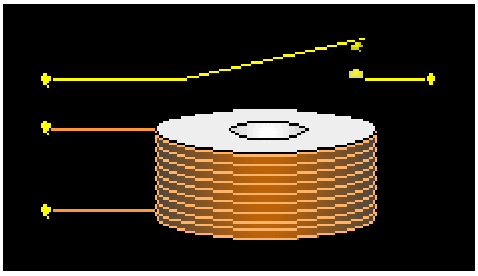


Fig 25: SPST Relay

SPDT Relay

SPDT Relay stands for Single Pole Double Throw relay. Current will flow between the movable contact and one fixed contact when the coil is De-energized and between the movable contact and the alternate fixed contact when the relay coil is energized. The most commonly used relay in car audio, the Bosch relay, is a SPDT relay.

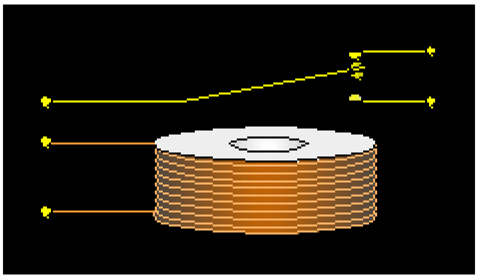


Fig 26: SPDT Relay

DPST Relay

DPST relay stands for Double Pole Single Throw relay. When the relay coil is energized, two separate and electrically isolated sets of contacts are pulled down to make contact with their stationary counterparts. There is no complete circuit path when the relay is De-energized.

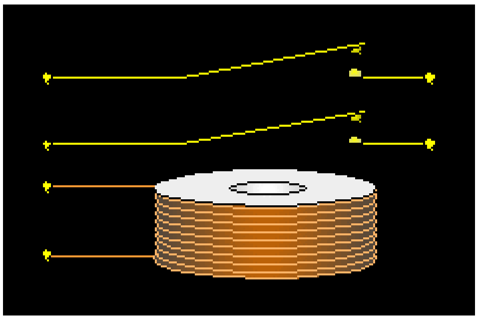


Fig 27: DPST Relay

DPDT Relay

DPDT relay stands for Double Pole Double Throw relay. It operates like the SPDT relay but has twice as many contacts. There are two completely isolated sets of contacts.

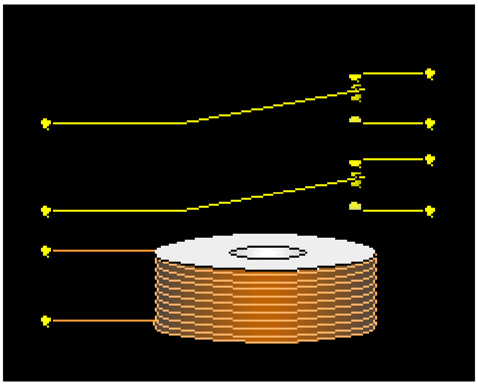


Fig 28: DPDT Relay

This is a 4 Pole Double Throw relay. It operates like the SPDT relay but it has 4 sets of isolated contacts.

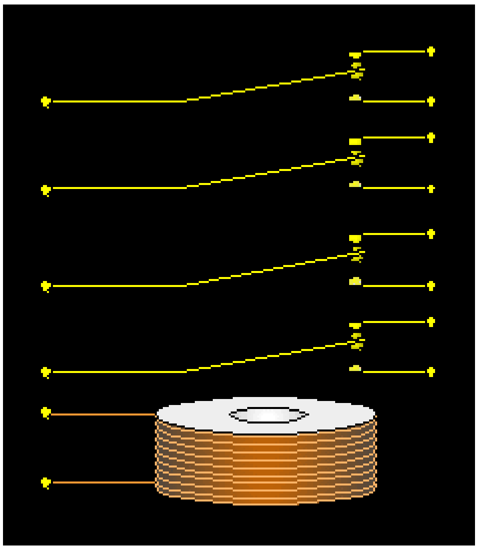


Fig 29: Pole Double Throw relay

4.5.2 Types of relay:

1. Latching Relay
2. Reed Relay
3. Mercury Wetted Relay
4. Machine Tool Relay
5. Solid State Relay (SSR)

Latching relay

Latching relay, dust cover removed, showing pawl and ratchet mechanism. The ratchet operates a cam, which raises and lowers the moving contact arm, seen edge-on just below it. The moving and fixed contacts are visible at the left side of the image.

A **latching relay** has two relaxed states (bi-stable). These are also called "impulse", "keep", or "stay" relays. When the current is switched off, the relay remains in its last state. This is achieved with a solenoid operating a ratchet and cam mechanism, or by having two opposing coils with an over-center spring or permanent magnet to hold the armature and contacts in position while the coil is relaxed, or with a remanent core. In the ratchet and cam example, the first pulse to the coil turns the relay on and the second pulse turns it off. In the two coil example, a pulse to one coil turns the relay on and a pulse to the opposite coil turns the relay off. This type of relay has the advantage that it consumes power only for an instant, while it is being switched, and it retains its last setting across a power outage. A remanent core latching relay requires a current pulse of opposite polarity to make it change state.

Fig 30: Latching relay

Reed relay

A reed relay has a set of contacts inside a vacuum or inert gas filled glass tube, which protects the contacts against atmospheric corrosion. The contacts are closed by a magnetic field generated when current passes through a coil around the glass tube. Reed relays are capable of faster switching speeds than larger types of relays, but have low switch current and voltage ratings.

Mercury-wetted relay

A mercury-wetted reed relay is a form of reed relay in which the contacts are wetted with mercury. Such relays are used to switch low-voltage signals (one volt or less) because of their low contact resistance, or for high-speed counting and timing applications where the mercury eliminates contact bounce. Mercury wetted relays are position-sensitive and must be mounted vertically to work properly. Because of the toxicity and expense of liquid mercury, these relays are rarely specified for new equipment. See also mercury switch.

Machine tool relay

A machine tool relay is a type standardized for industrial control of machine tools, transfer machines, and other sequential control. They are characterized by a large number of contacts (sometimes extendable in the field) which are easily converted from normally-open to normally-closed status, easily replaceable coils, and a form factor that allows compactly installing many relays in a control panel. Although such relays once were the backbone of automation in such industries as automobile assembly, the programmable logic controller (PLC) mostly displaced the machine tool relay from sequential control applications.

Solid-state relay

A solid state relay (SSR) is a solid state electronic component that provides a similar function to an electromechanical relay but does not have any moving components, increasing long-term reliability. With early SSR's, the tradeoff came from the fact that every transistor has a small voltage drop across it. This voltage drop limited the amount of current a given SSR could handle. As transistors improved, higher current SSR's, able to handle 100 to 1,200 Amperes, have become commercially available. Compared to electromagnetic relays, they may be falsely triggered by transients.

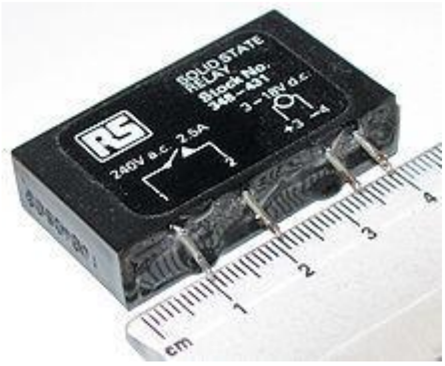


Fig 31: Solid relay, which has no moving parts

4.5.3 Specification

- Number and type of contacts – normally open, normally closed, (double-throw)
- Contact sequence – "Make before Break" or "Break before Make". For example, the old style telephone exchanges required Make-before-break so that the connection didn't get dropped while dialing the number.
- Rating of contacts – small relays switch a few amperes, large contactors are rated for up to 3000 amperes, alternating or direct current
- Voltage rating of contacts – typical control relays rated 300 VAC or 600 VAC, automotive types to 50 VDC, special high-voltage relays to about 15 000 V
- Coil voltage – machine-tool relays usually 24 VAC, 120 or 250 VAC, relays for switchgear may have 125 V or 250 VDC coils, "sensitive" relays operate on a few milli-amperes

4.5.4 Applications:

Relays are used

- To control a high-voltage circuit with a low-voltage signal, as in some types of modems,
- To control a high-current circuit with a low-current signal, as in the starter solenoid of an automobile,
- To detect and isolate faults on transmission and distribution lines by opening and closing circuit breakers (protection relays),
- To isolate the controlling circuit from the controlled circuit when the two are at different potentials, for example when controlling a mains-powered device from a low-voltage switch. The latter is often applied to control office lighting as the low voltage wires are easily installed in partitions, which may be often moved as needs change. They may also be controlled by room occupancy detectors in an effort to conserve energy,
- To perform logic functions. For example, the boolean AND function is realized by connecting relay contacts in series, the OR function by connecting contacts in parallel. Due to the failure modes of a relay compared with a semiconductor, they are widely used in safety critical logic, such as the control panels of radioactive waste handling machinery.
- As oscillators, also called vibrators. The coil is wired in series with the normally closed contacts. When a current is passed through the relay coil, the relay operates and opens the contacts that carry the supply current. This stops the current and causes the contacts to close again. The cycle repeats continuously, causing the relay to open and close rapidly. Vibrators are used to generate pulsed current.
- To generate sound. A vibrator, described above, creates a buzzing sound because of the rapid oscillation of the armature. This is the basis of the electric bell, which consists of a vibrator with a hammer attached to the armature so it can repeatedly strike a bell.
- To perform time delay functions. Relays can be used to act as an mechanical time delay device by controlling the release time by using the effect of residual magnetism

by means of a inserting copper disk between the armature and moving blade assembly.

4.6 Software Requirements

- Proteus simulation
- MC Programming Language: Embedded C
- Arduino software

4.6.1 Arduino Software:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals

- has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers c an make their own version
- Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money
- Getting Started with Arduino and Genuino products:-

Install the Arduino Software (IDE) on Windows PCs-

…This document explains how to install the Arduino Software (IDE) on Windows machines.

- ✓ Download the Arduino Software(IDE)
- ✓ Proceed with board specific instructions.

How to Download the Arduino Software (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

Installation:

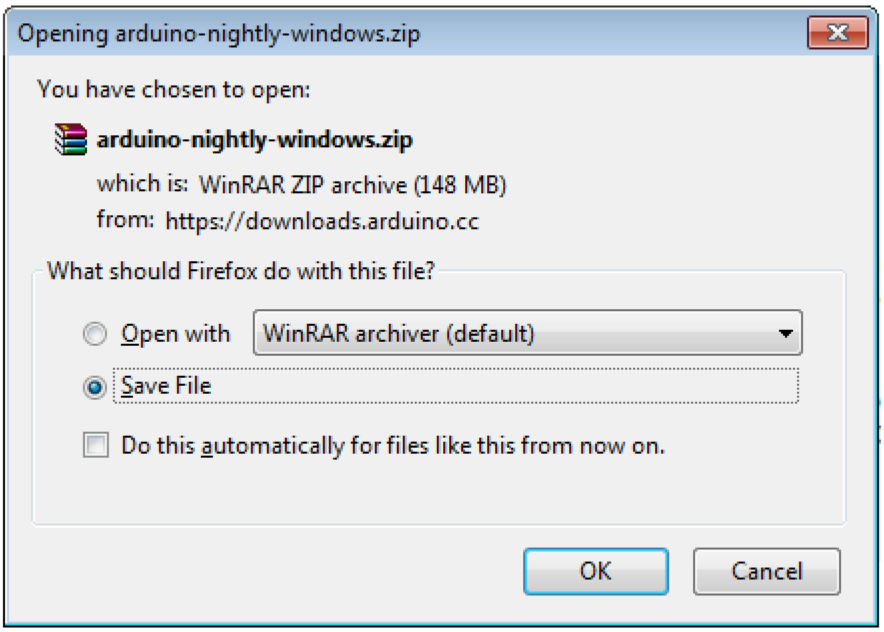
In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



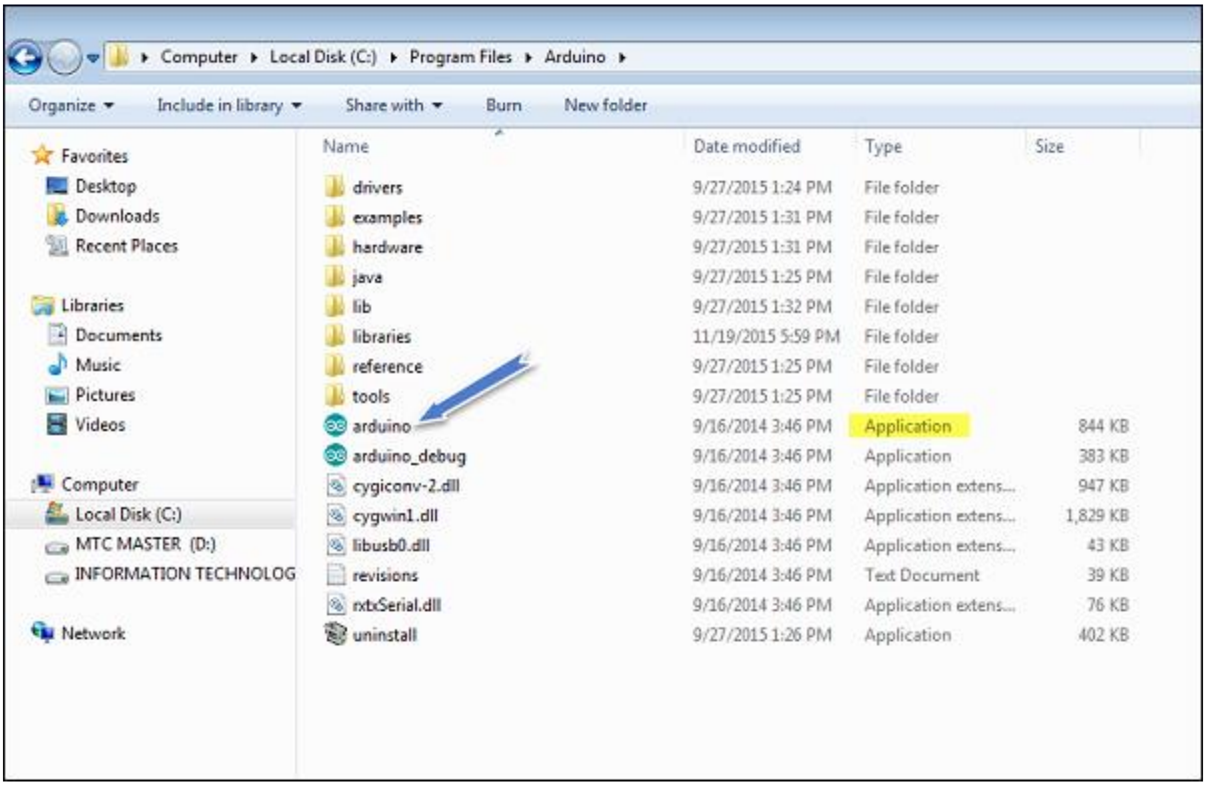
Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 – Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

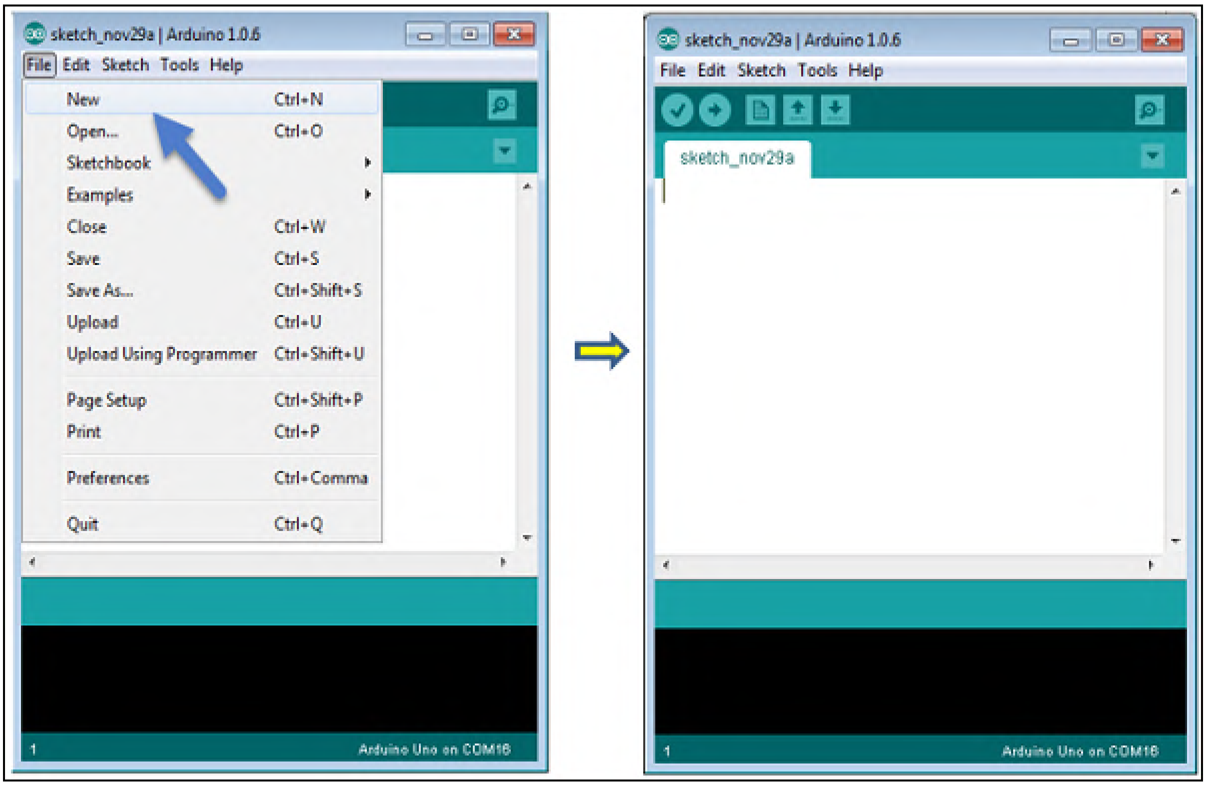


Step 5 – Open your first project.

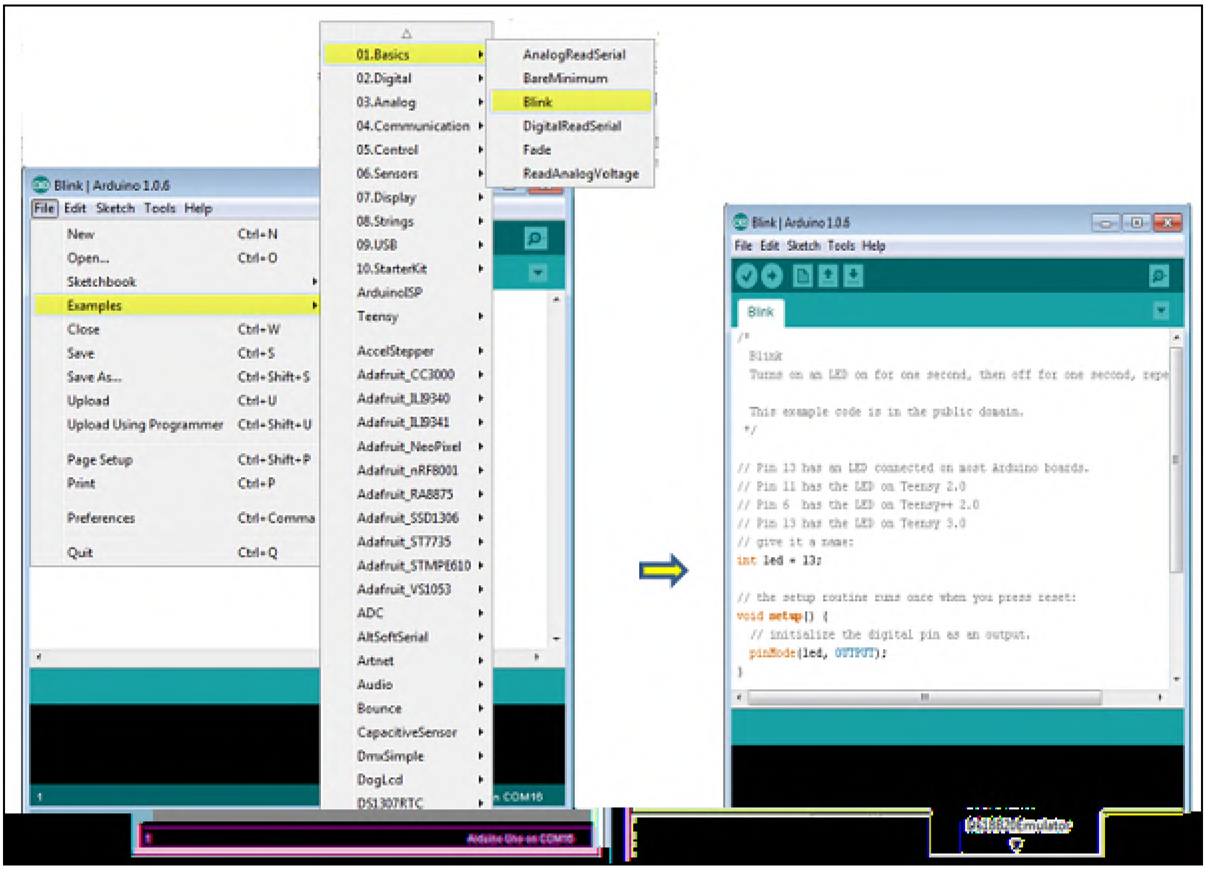
Once the software starts, you have two options –

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.



To open an existing project example, select File → Example → Basics → Blink.

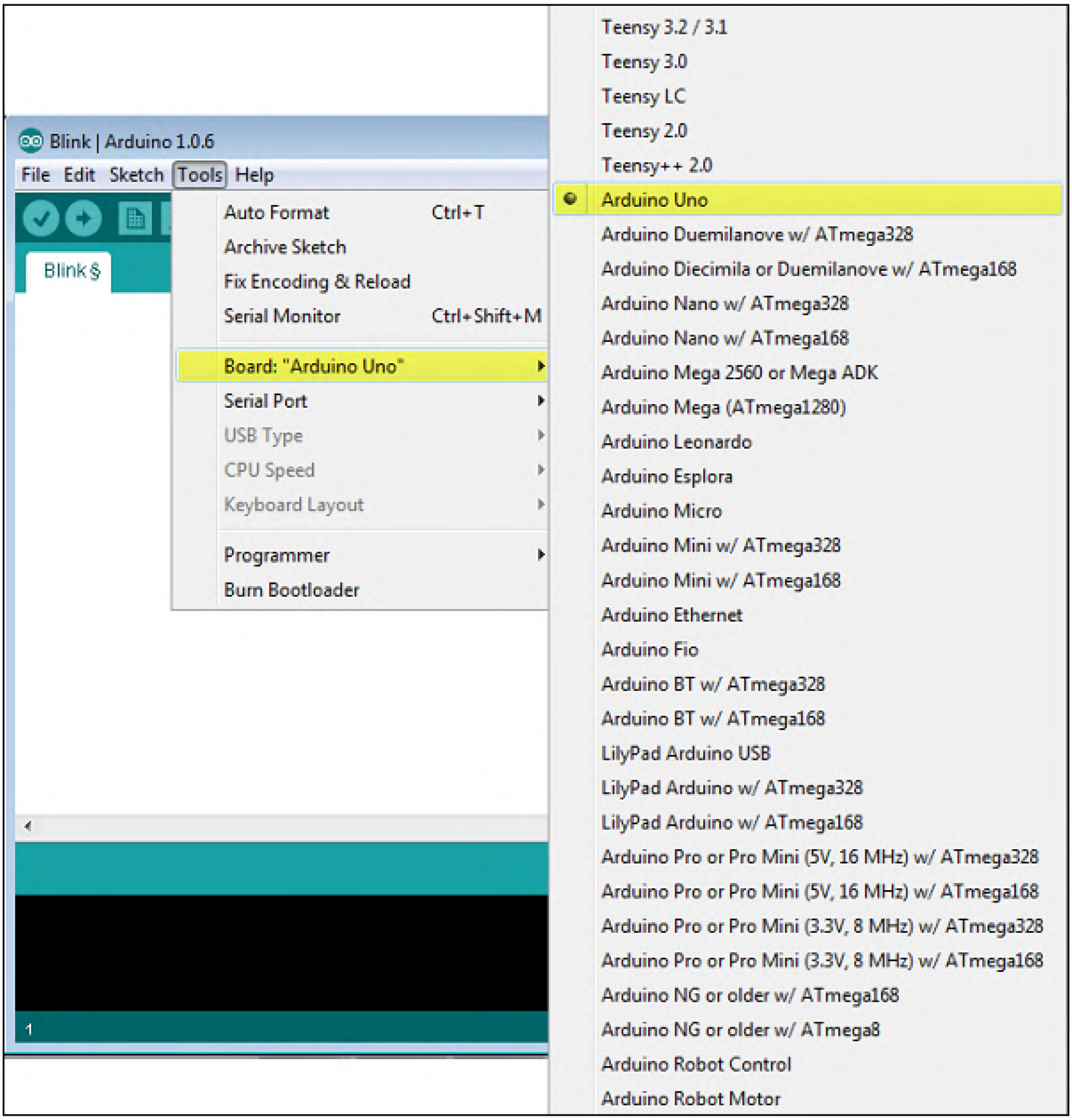


Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

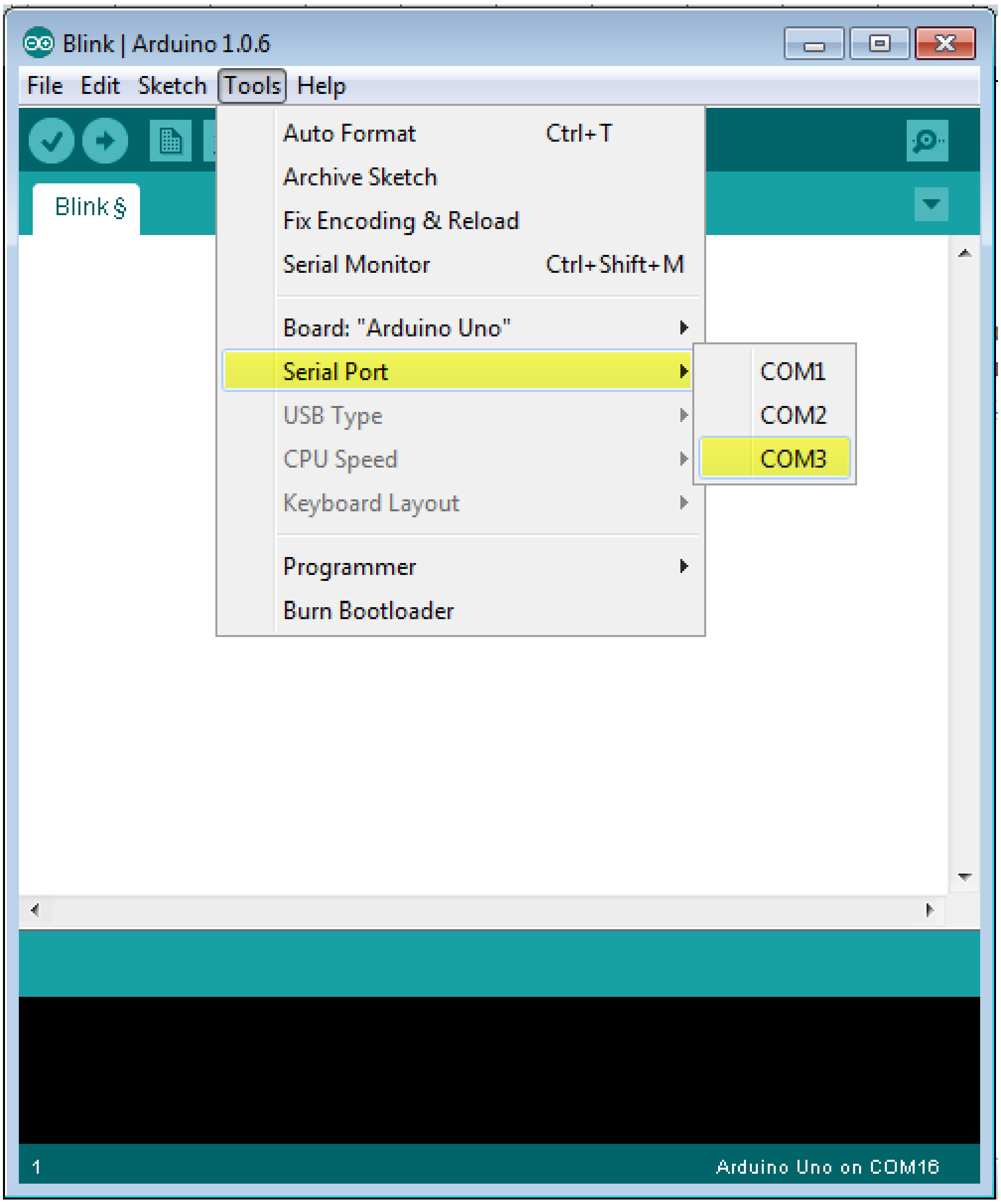
Go to Tools → Board and select your board.



Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

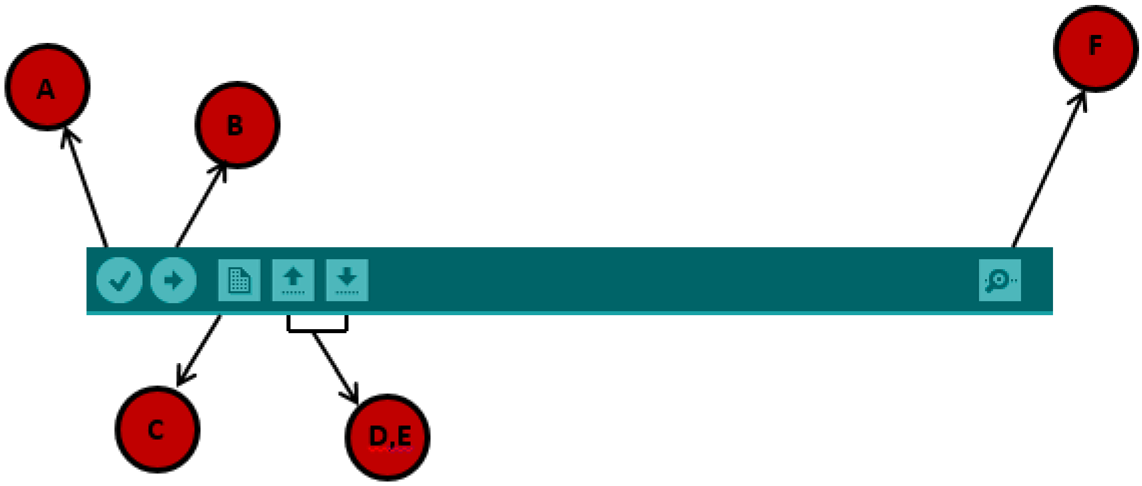
Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.



Step 8 – Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.



A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24 V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

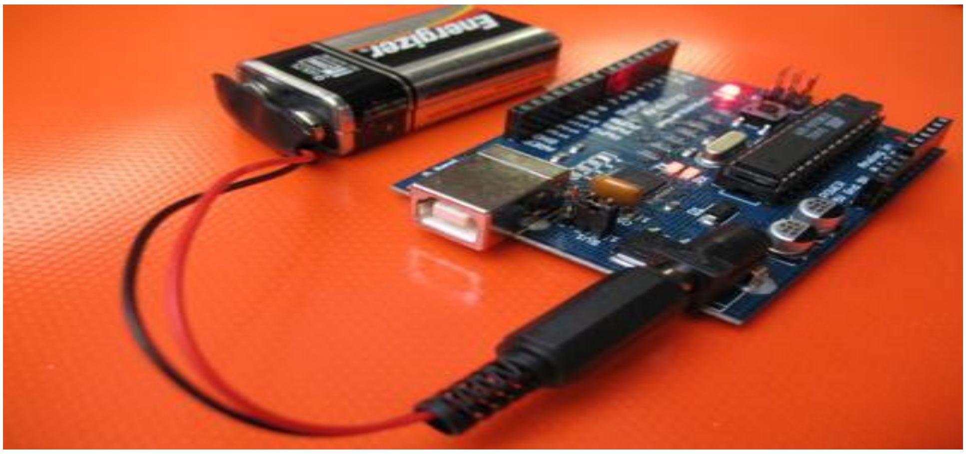


Fig 32: Arduino with battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called "sketches", but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

4.7 MC Programming Language: Embedded C

This is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements. Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

4.8 Proteus:

Proteus:

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.

The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design

Step 1: Open ISIS software and select New design in File menu

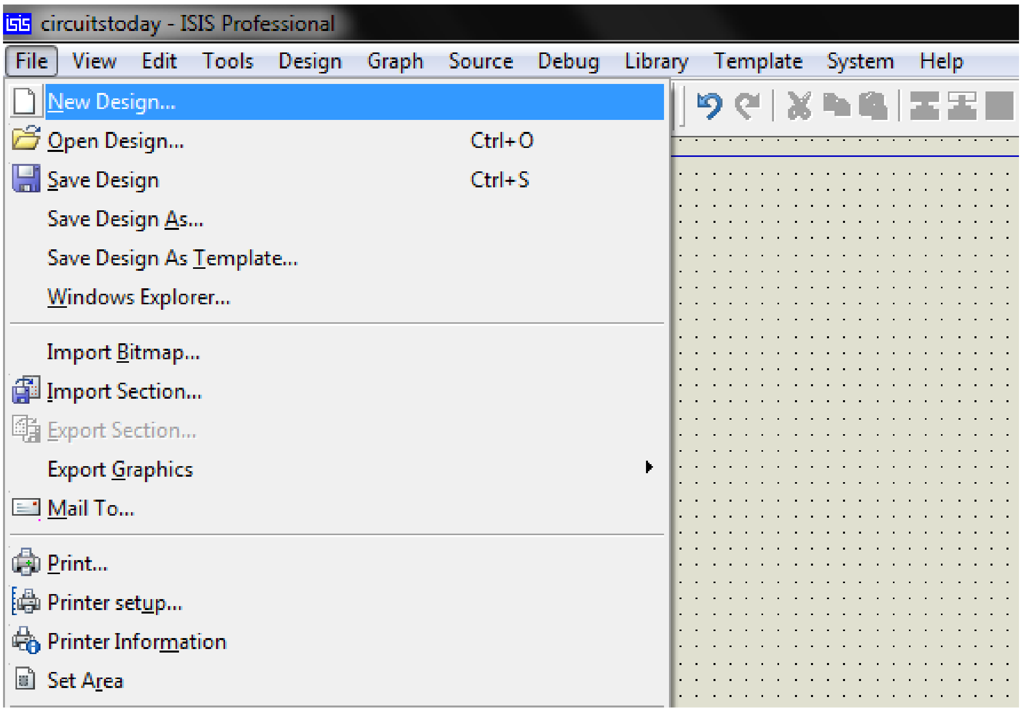


Fig 33: Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

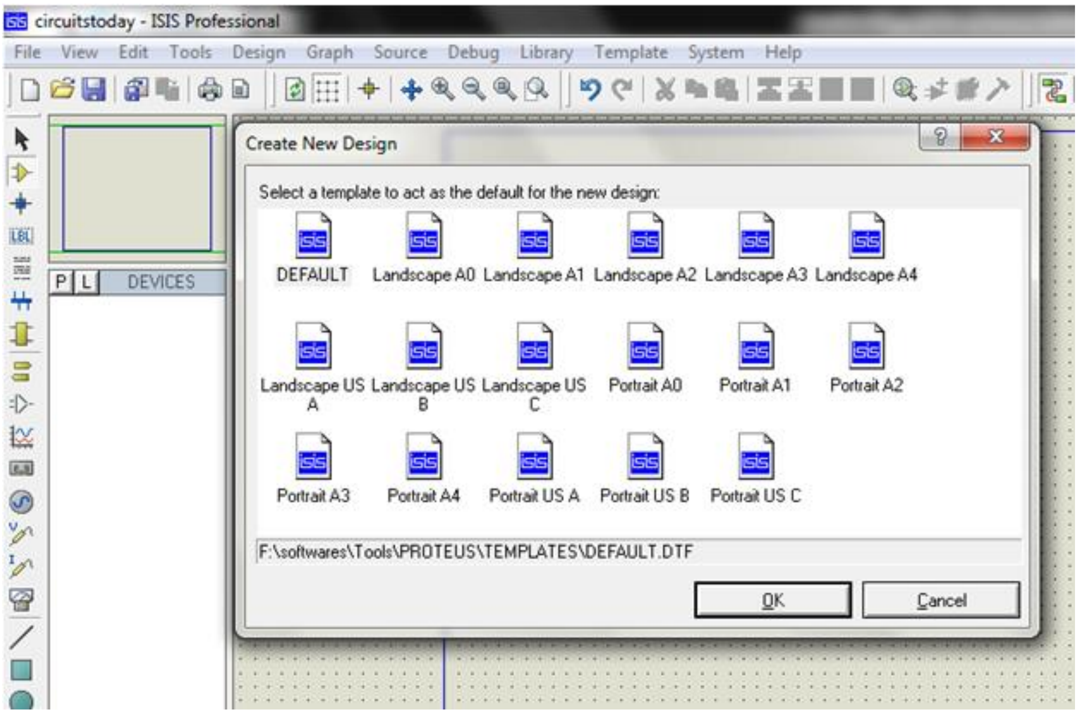


Fig 34: Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

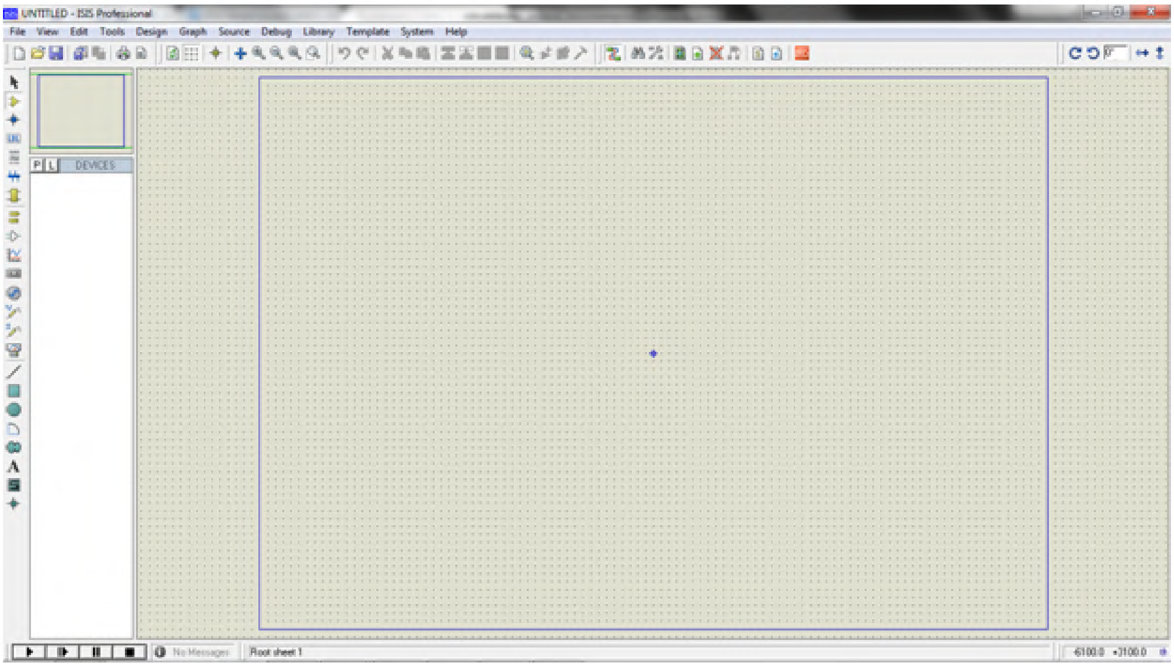


Fig 35: Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

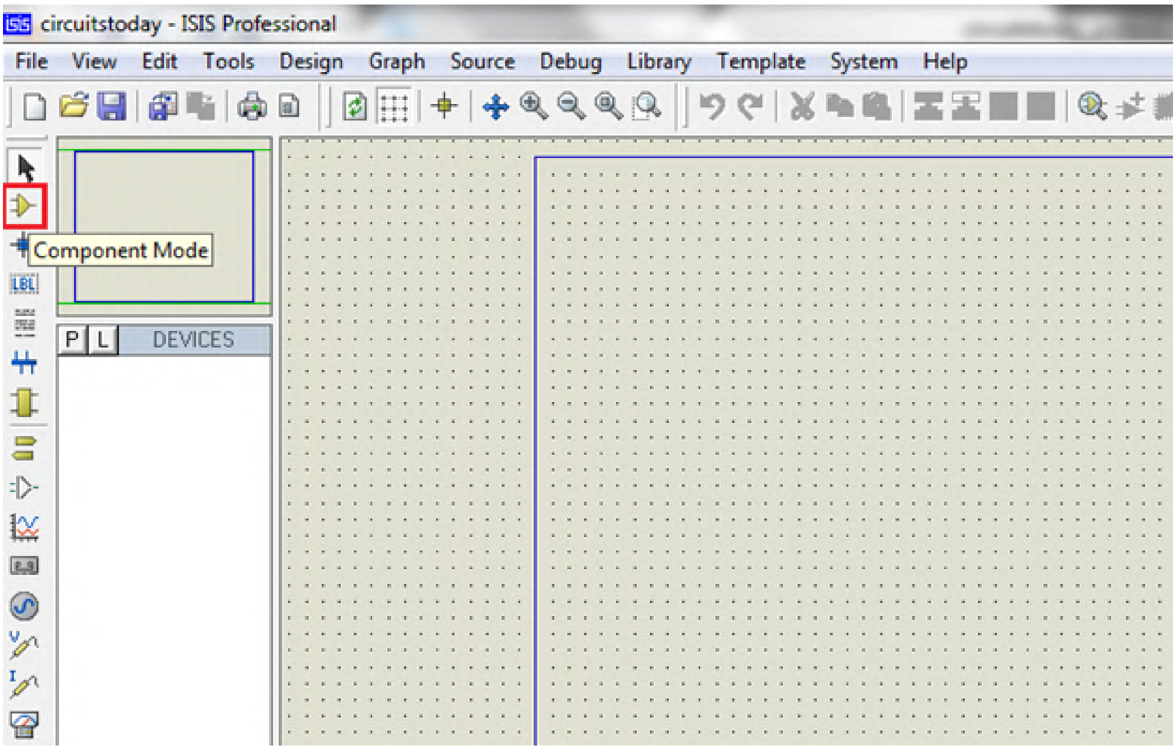


Fig 36: Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

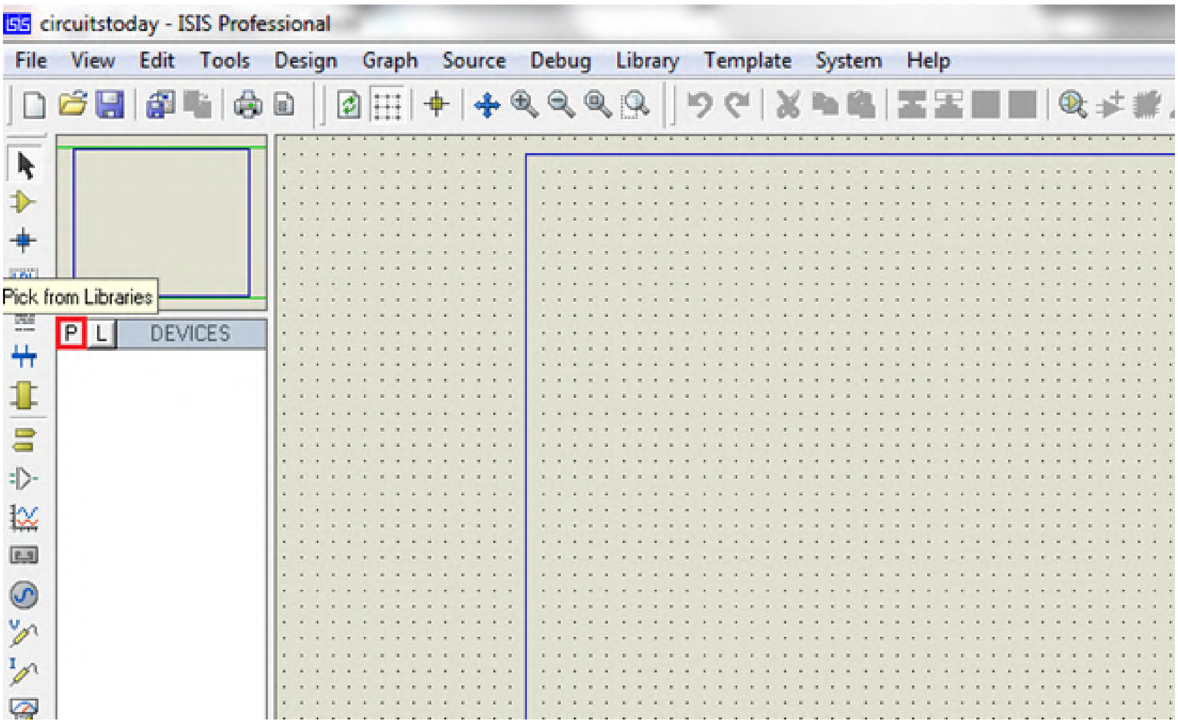


Fig 37: Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

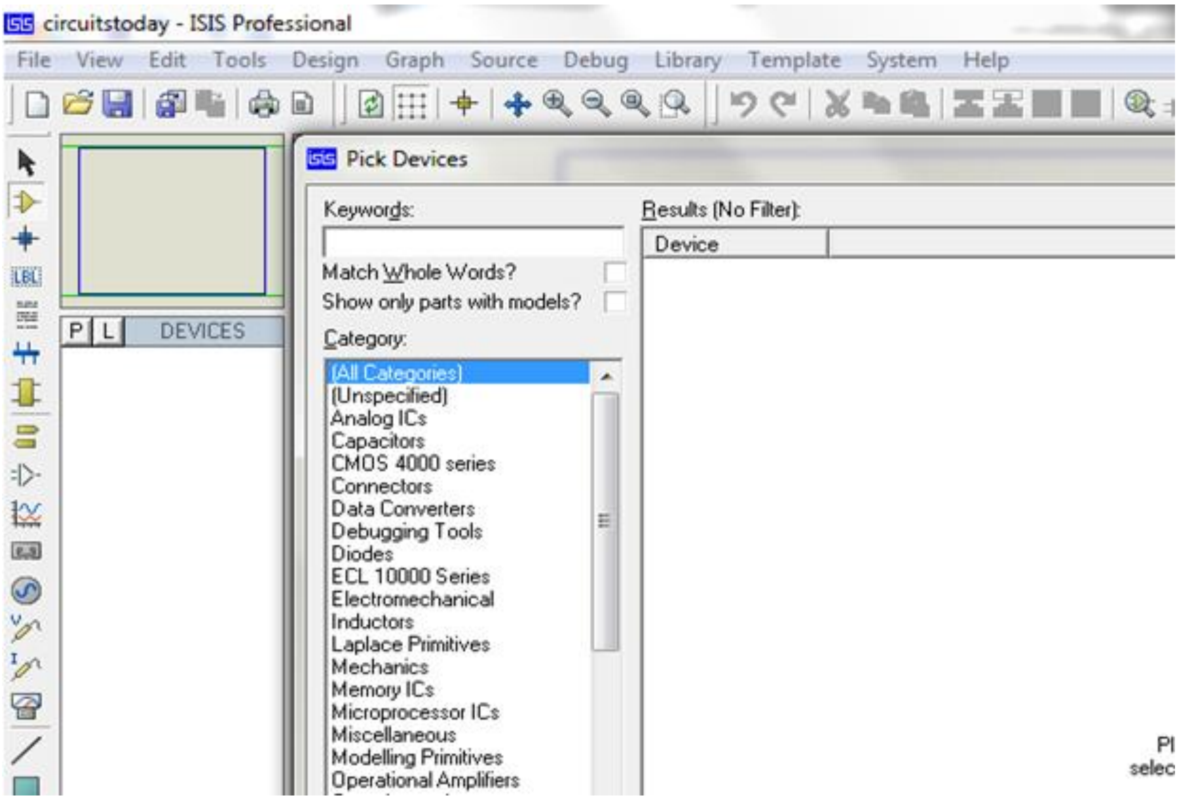


Fig 38: Keywords Textbox

Example shows selection of push button. Select the components accordingly.

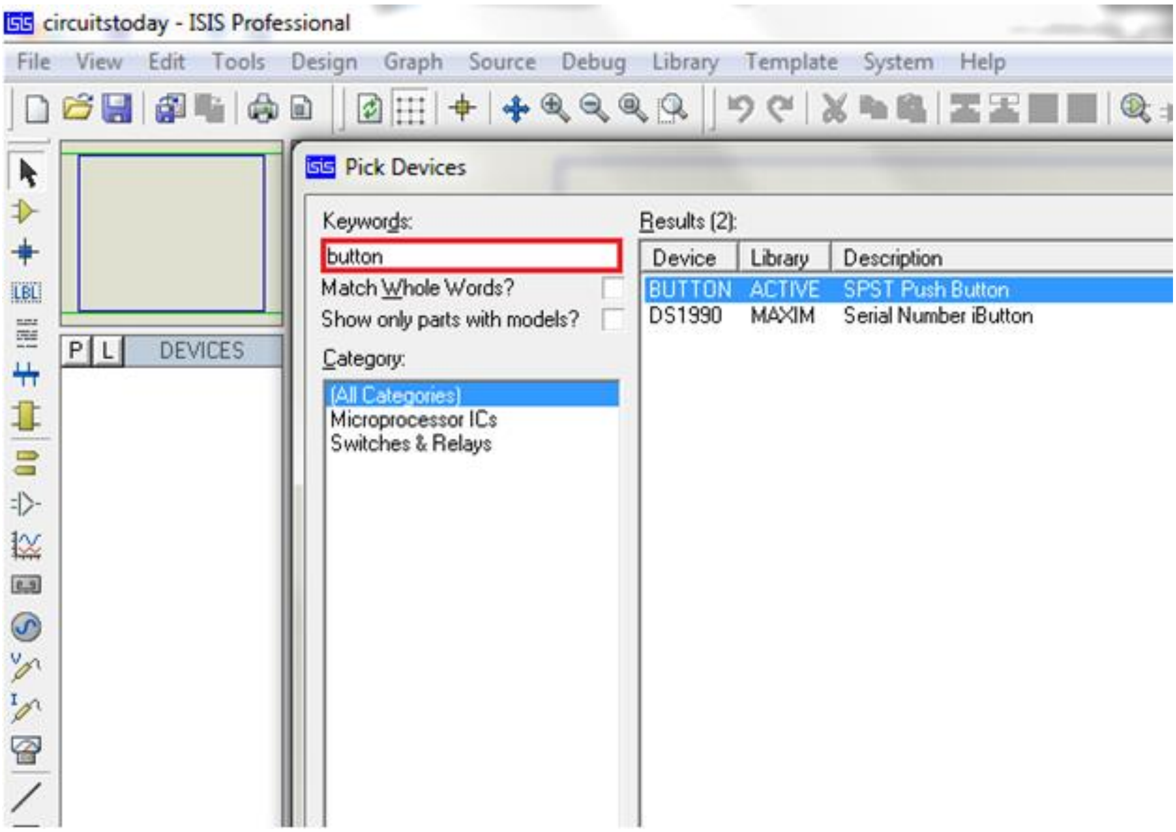


Fig 39: Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

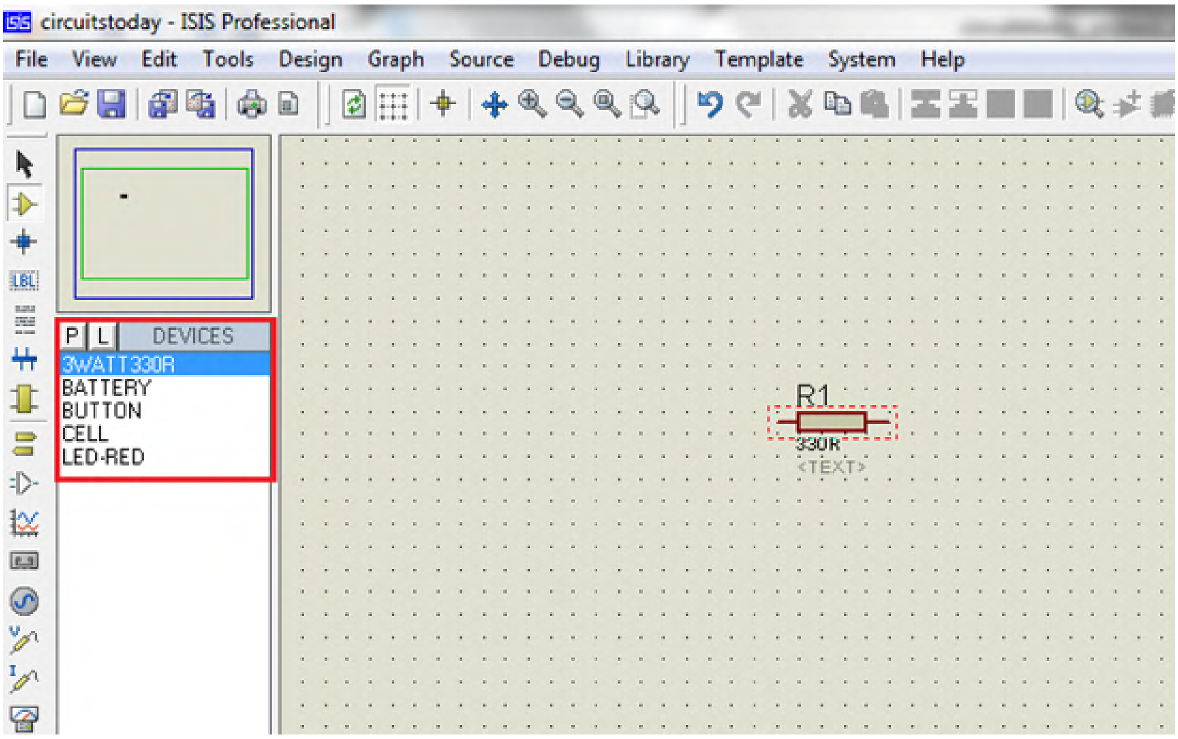


Fig 40: Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

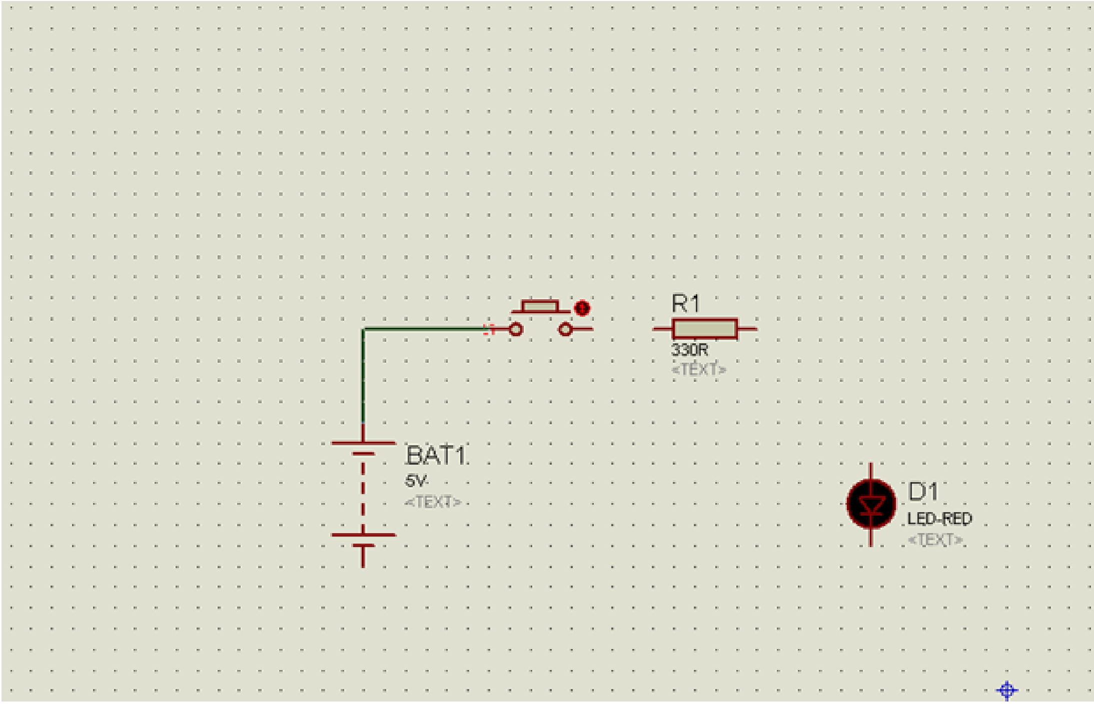


Fig 41: Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

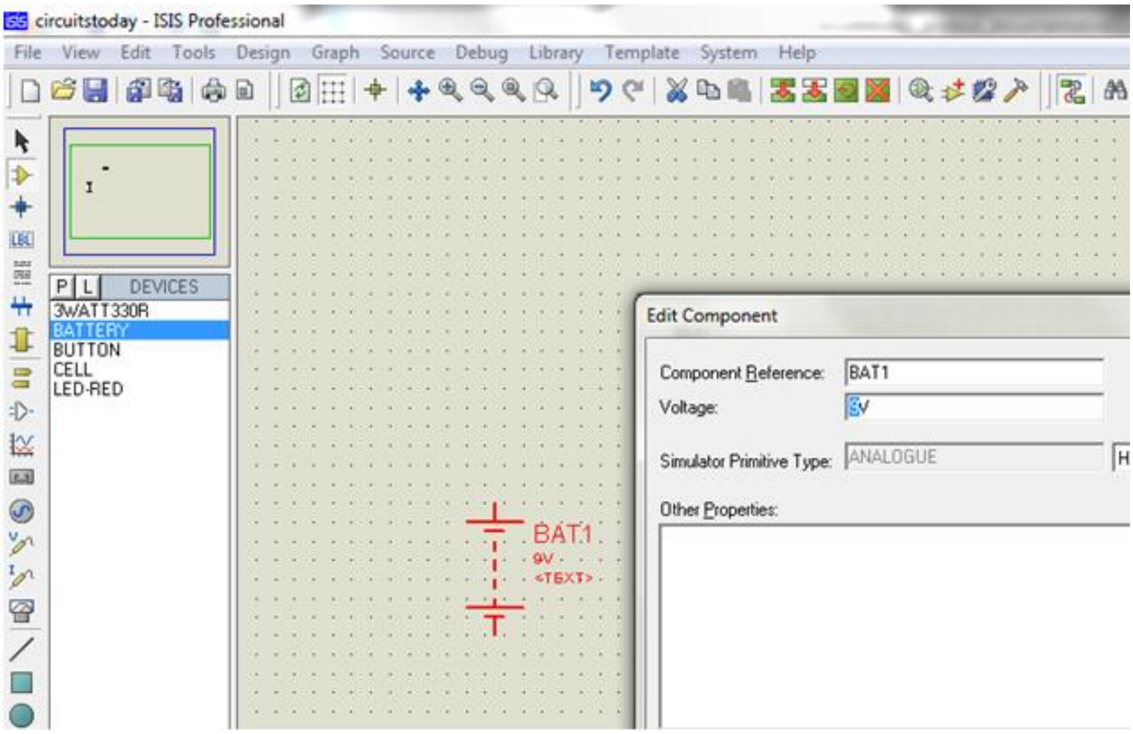


Fig 42: Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

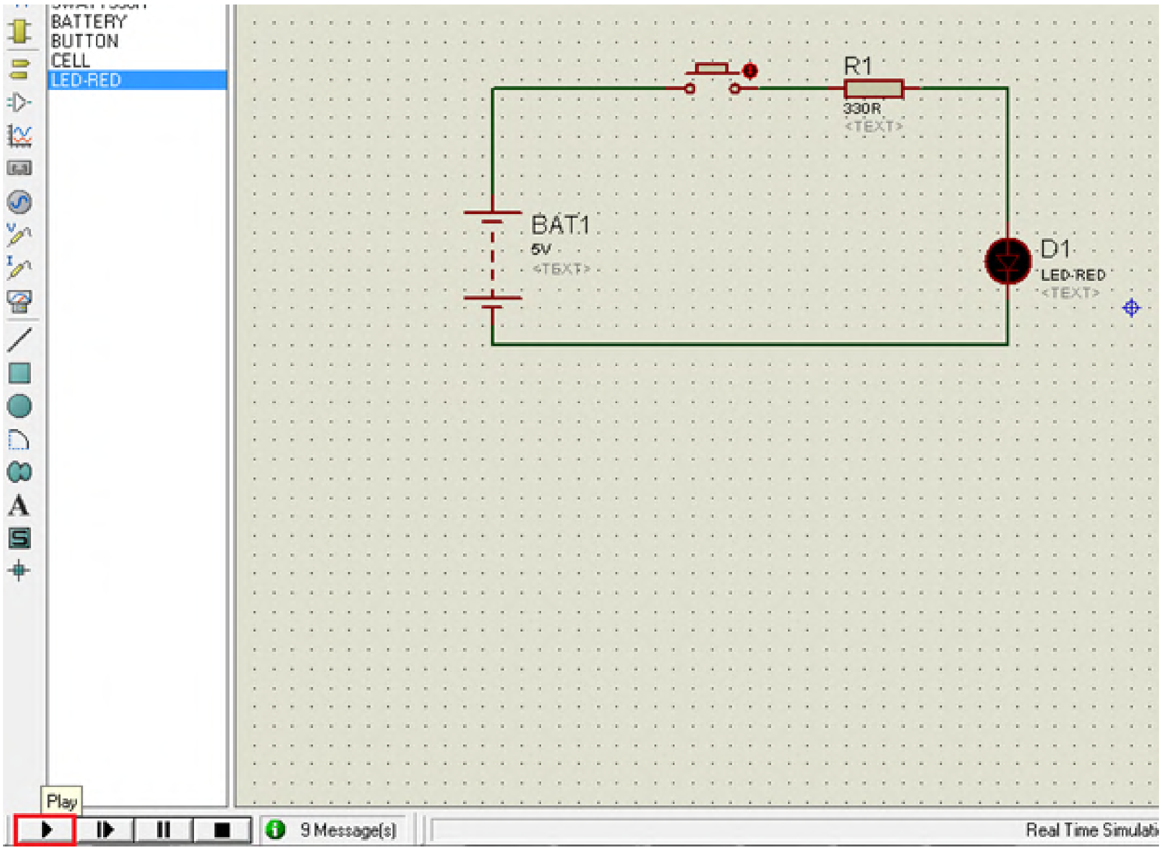


Fig 43: Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

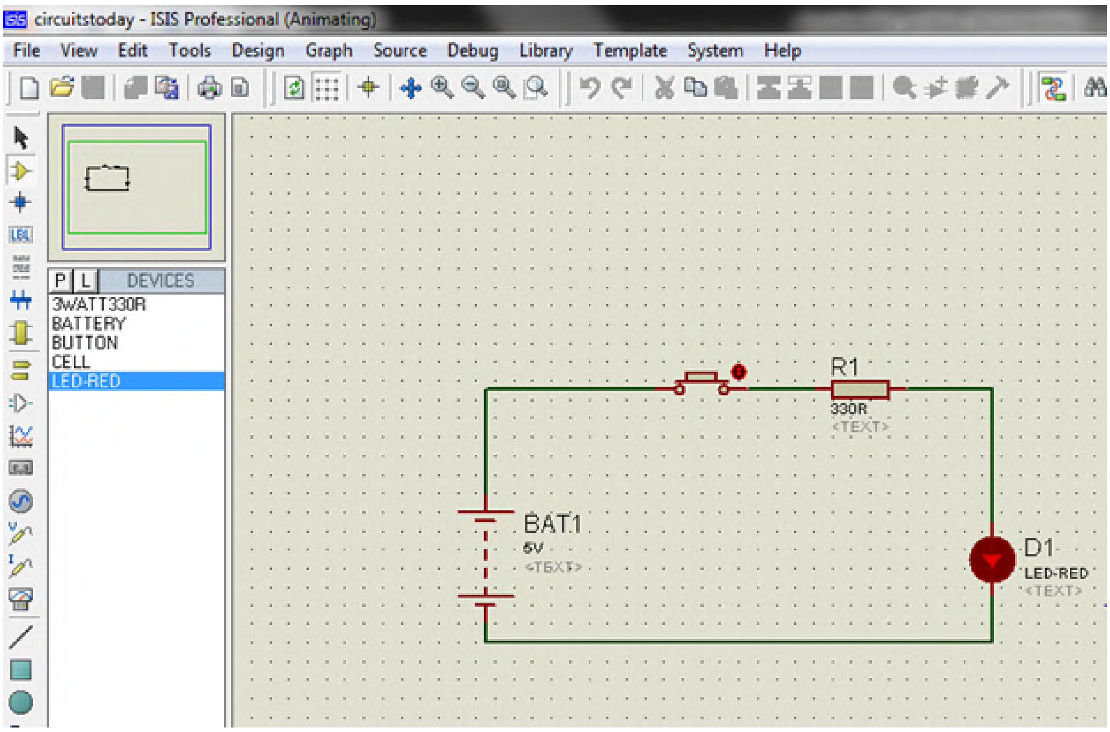


Fig 44: Simulation Animating

Simulation can be stepped, paused or stopped at any time.

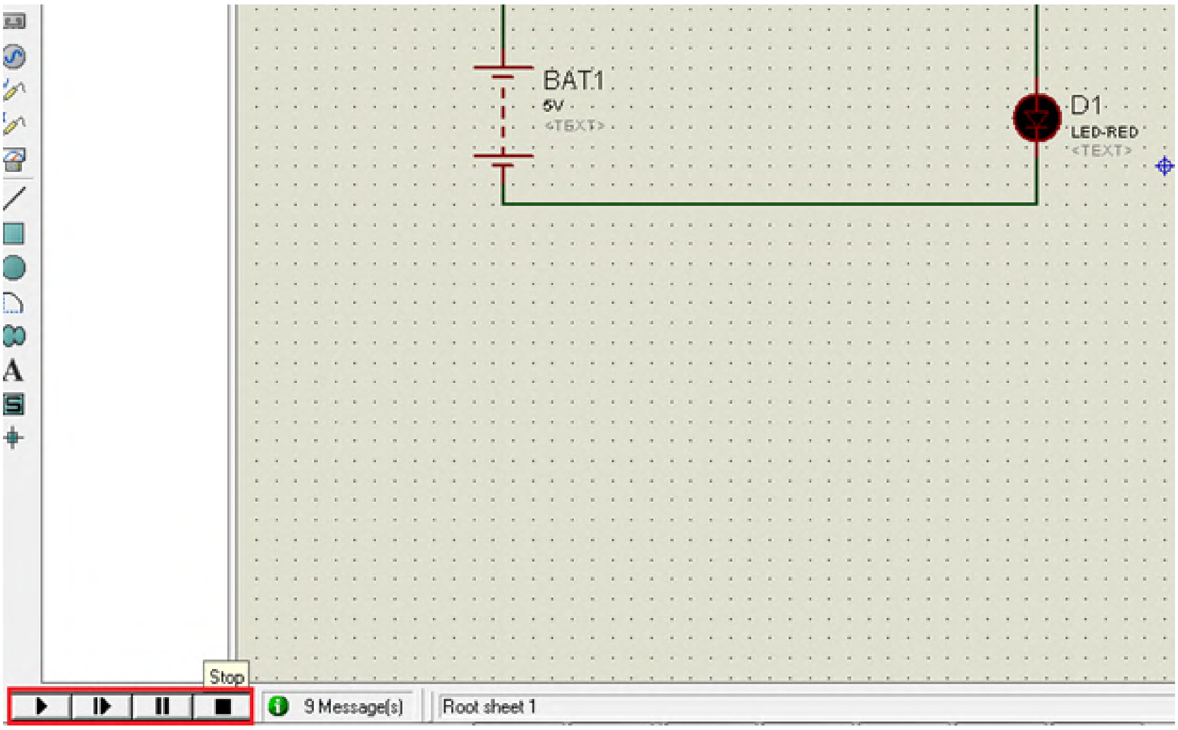


Fig 45: Simulation Step-Pause-Stop Buttons

4.9 Telnet:

Telnet is a client-server protocol, based on a reliable connection-oriented transport. Typically, this protocol is used to establish a connection to Transmission Control Protocol (TCP) port number 23, where a Telnet server application (telnetd) is listening. Telnet, however, predates TCP/IP and was originally run over Network Control Program (NCP) protocols.

Even though Telnet was an ad hoc protocol with no official definition until March 5, 1973,[3] the name actually referred to *Teletype Over Network Protocol* as the RFC 206 (NIC 7176) on Telnet makes the connection clear:[4]

The TELNET protocol is based upon the notion of a virtual teletype, employing a 7-bit ASCII character set. The primary function of a User TELNET, then, is to provide the means by which its users can 'hit' all the keys on that virtual teletype.[5]

Essentially, it used an 8-bit channel to exchange 7-bit ASCII data. Any byte with the high bit set was a special Telnet character. On March 5, 1973, a Telnet protocol standard was defined at UCLA[6] with the publication of two NIC documents: Telnet Protocol Specification, NIC 15372, and Telnet Option Specifications, NIC 15373.

Many extensions were made for Telnet because of its negotiable options protocol architecture. Some of these extensions have been adopted as Internet standards, IETF documents STD 27 through STD 32. Some extensions have been widely implemented and others are proposed standards on the IETF standards track (see below) Telnet is best understood in the context of a user with a simple terminal using the local Telnet program (known as the client program) to run a logon session on a remote computer where the user's communications needs are handled by a Telnet server program.

When Telnet was initially developed in 1969, most users of networked computers were in the computer departments of academic institutions, or at large private and government research facilities. In this environment, security was not nearly as much a concern as it became after the bandwidth explosion of the 1990s. The rise in the number of people with access to the Internet, and by extension the number of people attempting to hack other people's servers, made encrypted alternatives necessary.

Experts in computer security, such as SANS Institute, recommend that the use of Telnet for remote logins should be discontinued under all normal circumstances, for the following reasons:

- Telnet, by default, does not encrypt any data sent over the connection (including passwords), and so it is often feasible to eavesdrop on the communications and use the password later for malicious purposes; anybody who has access to a router, switch, hub or gateway located on the network between the two hosts where Telnet is being used can intercept the packets passing by and obtain login, password and whatever else is typed with a packet analyzer.
- Most implementations of Telnet have no authentication that would ensure communication is carried out between the two desired hosts and not intercepted in the middle.
- Several vulnerabilities have been discovered over the years in commonly used Telnet daemons.

These security-related shortcomings have seen the usage of the Telnet protocol drop rapidly,[7] especially on the public Internet, in favour of the Secure Shell (SSH) protocol, first released in 1995. SSH has practically replaced Telnet, and the older protocol is used these days only in rare cases to access decades old legacy equipment that does not support more modern protocols.[8] SSH provides much of the functionality of telnet, with the addition of strong encryption to prevent sensitive data such as passwords from being intercepted, and public key authentication, to ensure that the remote computer is actually who it claims to be. As has happened with other early Internet protocols, extensions to the Telnet protocol provide Transport Layer Security (TLS) security and Simple Authentication and Security Layer (SASL) authentication that address the above concerns. However, most Telnet implementations do not support these extensions; and there has been relatively little interest in implementing these as SSH is adequate for most purposes.

It is of note that there are a large number of industrial and scientific devices which have only Telnet available as a communication option. Some are built with only a standard RS-232 port and use a serial server hardware appliance to provide the translation between the TCP/Telnet data and the RS-232 serial data. In such cases, SSH is not an option unless the interface appliance can be configured for SSH.

Telnet is still used by hobbyists, especially among Amateur radio operators.
The Winlink protocol supports packet radio via a Telnet connection.

4.9.1 Telnet 5250

IBM 5250 or 3270 workstation emulation is supported via custom telnet clients, TN5250/TN3270, and IBM servers. Clients and servers designed to pass IBM 5250 data streams over Telnet generally do support SSL encryption, as SSH does not include 5250 emulation. Under IBM i (also known as OS/400), port 992 is the default port for secured telnet.

CHAPTER-5

RESULT & CONCLUSION

5.1 Application of the Project:

- Food production Industry
- Hydroponic Farming machinery
- Agricultural Industry
- and many technologies in the emerging time

5.2 Advantages of the Project:

- Maximizes space utilization
- Conserves water
- Produces high yield per area
- And the nest part is "needs no soil"

5.3 Result:

Fig 46: The Prototype ready Image

Fig 47: LCD output 1

Fig 48: LCD output 2

5.4 Conclusion:

We are trying here too develop a hydroponic farm that help in managing the time in planting of crop using wireless sensor network which will be deployed at various points and senses various environmental parameters, gather data to the center node by using wireless protocol through IoT. Still it has been observed that to monitor the environmental factors is not the total solution to increase the yield of cropping, hence automation in agriculture can only avoid the other factors and problems that reduces the productivity. We are expecting that our proposed system can effectively control the PH value or concentration and electrical conductivity in nutrition liquid. Plant performance may be optimized by controlling the climate and lighting. There will be lesser requirements in cost and also easy to maintain which controls the important factors such as light, water level temperature and humidity throughout the year is needed.

5.5 Future Scope:

One can implement a few more sensors and connect it to the satellite as a global feature of this system,

1) Adding more sensor to monitor other environmental parameters such as CO2,Pressure and Oxygen Sensor
2) In aircraft, navigation and military there is a great scope of this real-time system.

3) It can also be implemented in hospitals or medical institutes for the research & study in “Effect of Weather on Health and Diseases”, hence to provide better precaution alerts.
4) To make the system solar powered.
5) To install Portable, lightweight and sturdy systems; compatible for use in urban environment.

REFERENCES

[1] Trejo Téllez LI, Gómez-Merino FC. Editors. Nutrient Solutions for Hydroponic Systems, Hydroponics – A Standard Methodology for Plant Biological Researches. Rijeka: Croatia. 2012.
[2] Steiner AA. A Universal Method for Preparing Nutrient Solutions of a Certain Desired Composition.
Plant and Soil. 1961; 15(2): 134-154.
[3] Nemali KS, Van Iersel MW. Light Intensity and Fertilizer Concentration: I. Estimating Optimal
Fertilizer Concentration from Water-Use Efficiency of Wax Begonia. Hort Science. 2004; 39(6): 1287-
1292.
[4] Sonneveld C, Voogt W. Plant Nutrition of Greenhouse Crops. New York: Springer.2009.
[5] Samarakoon UC, Weerasinghe PA, Weerakkody AP. Effect of Electrical Conductivity [EC] of the
Nutrient Solution on Nutrient Uptake, Growth and Yield of Leaf Lettuce (Lactuca sativa L.) in
Stationary Culture. Tropical Agricultural Research. 2006; 18(1): 13-21.
[6] Gruda N. Do Soilles Culture Systems have an Influence on Product Quality of Vegetables? Journal of
Applied Botany and Food Quality. 2009; 82(2): 141-147.
[7] Delya, Buti. Design of Ebb and Flow Automatic Hydroponic System for Chilli Pepper Cultivation.
Jurnal Teknik Pertanian. 2015; 3(3).
[8] Suprijadi, Nuraini N, Yusuf M. Sistem kontrol nutrisi hidroponik dengan menggunakan logika fuzzy.
Jurnal Otomasi, Kontrol, dan Instrumentasi. 2009; 1(1):31-35.
[9] Domingues DS, Camara CAP Nixdorf SL, Takahashi HW. Automated system developed to control pH
and concentration of nutrient solution evaluated in hydroponic lettuce production. Computers and
Electronics in Agriculture. 2012; 84: 53–61
[10] Ming D, Bing-jie W, Hailing S, Yong-qi T. Study of an Improved Fuzzy Direct Torque Control of
Induction Motor. TELKOMNIKA. 2013; 11(4): 691-698.
[11] Prayitno A, Indrawati V, Utomo G. Trajectory Tracking of AR. Drone Quadrotor Using Fuzzy Logic
Controller. TELKOMNIKA. 2014; 12(4): 819-828.
[12] Prathisthaya, ME, Liyantono, Solahudin M. Rancang Bangun Sistem Kendali Proportional Integral-
Deritative pada Pengendalian Konsentrasi Larutan Nutrisi Hidroponik. Proceedings of the National
Seminar FTIP UNPAD – PERTETA – HIPI 2014. Jatinangor. 2014: 376-384
[13] Weerakkody WAP, Wakui K, Nukaya A. Plant nutrient uptake in recirculation culture of tomato under
growth stage based electrical conductivity adjustment. Journal of National Science of Sri Lanka.
2011; 39(2):139-147.
[14] Dermawati. Substitusi hara mineral organik terhadap inorganik untuk produksi tanaman pakchoy

(Brassica rapa L.) secara hidroponik. Thesis. Bogor: Postgraduate IPB. 2006.
[15] Kusumadewi S, Purnomo H. Aplikasi Logika Fuzzy untuk Pendukung Keputusan. Yogyakarta: Graha Ilmu. 2004.
[16] Siegenthaler J. Modern Hydronic Heating: For Residential and Light Commercial Buildings. New York: Cengage Learning. 2011:335.
[17] Prastya, Andika. Analisis Kelayakan dan Optimasi Usaha Budidaya Bayam Merah dan Kangkung Hidroponik Dengan Sistem NFT (Nutrient Film Technique) di PT. Joy Farm, Depok. Undergraduate thesis. Bogor: Institut Pertanian Bogor. 2009.
[18] International Journal of Advanced Research in Computer and Communication Engineering ISO 3297:2007 Certified Vol. 5, Issue 9, September 2016
[19] International Journal of Engineering Trends and Technology (IJETT) – Volume 32 Number 2- February 2016
[20] International Journal of Engineering Science and Computing, May 2017
[21] Sagar J. S. T. , M. S. Balamurugan and J. A. Vivek, "A wireless framework for automotive monitoring systems," in Indian Journal of Science and Technology, Vol 8(19), IPL0146, August 2015
[22] https://www.openhacks.com/uploadsproductos/rain_sensor_module.pdf
[23]http://designinformaticslab.github.io/productdesign_tutorial/2017/01/24/soilmoisture_sensor.html
[24] content://com.sec.android.app.sbrowser/readinglist/0721190655.mhtml
[25] https://www.hackster.io/techmirtz/using-16x2-lcd-with-arduino-d89028
[26] https://www.arduino.cc/en/Guide/ArduinoUnoWiF

A

Project report on

6G WIRELESS COMMUNICATIONS : VISION AND POTENTIAL TECHNIQUES

Submitted By

RAMYA BAKI (17K81A04C9)

B.MAHITHA REDDY (17K81A04D9)

P.KRISHNA VAMSI (17K81A04G4)

in partial fulfilment for the award of degree

of

BACHELOR OF TECHNOLOGY

IN

Electronics & Communication Engineering

Under the Guidance of

Mrs.G.Laxmi Priyanka

M.Tech. Assistant Professor

St. MARTIN'S ENGINEERING COLLEGE

(Autonomous Institute)

Dhulapally, Secunderabad-500 100.

JUNE 2021

St.MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)
An UGC Autonomous Institute
Dhulapally, Secunderabad-500 100
NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

BONAFIDE CERTIFICATE

Certified that this project report titled "6G WIRELESS COMMUNICATIONS: VISION AND POTENTIAL TECHNIQUES", is a bonafide work of **1. Ms. Ramya Baki (17K81A04C9), 2. Ms. B. Mahitha Reddy (17K81A04D9), 3. Mr. P. Krishna Vamshi (17K81A04G4)** who carried out the work under my supervision, for the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electronics And Communication Engineering. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion.

INTERNAL GUIDE

Mrs. G. Laxmi Priyanka
Assistant Professor
Department of ECE

HEAD OF THE DEPARTMENT

Dr.B. Hari Krishna
Professor
Department of ECE

Internal Examiner

External Examiner

Place: Secunderabad
Date:

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **RAMYA BAKI** WITH ROLL NO.**17K81A04C9, B. MAHITHA REDDY** WITH ROLL NO.**17K81A04D9**, **P.KRISHNA VAMSHI** WITH ROLL NO.**17K81A04G4,** OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**6G WIRELESS COMMUNICATIONS : VISION AND POTENTIAL TECHNIQUES**" AT OUR DEVELOPMENT CENTER,KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. ★ Kompally-500014. ★

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS LASYA IT
SOLUTIONS PVT LTD.

LasyaITSolutionsPvtLtd,BehindCinePlanet,Kompally,MedchalRoad,Secunderabad500014
Email :contact@lasyainfotech.com,ov@lasyainfotech.com
Website :www.lasyainfotech.com| contact: 7330666881/82/83/84/86

DECLARATION

We declare that this project report titled "6G WIRELESS COMMUNICATIONS: VISION AND POTENTIAL TECHNIQUES" submitted in partial fulfillment of the degree of B. Tech in Electronics And Communication Engineering record of original work carried out by us under the guidance and supervision of Mrs. G. Laxmi Priyanka, and has not formed the name basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have be encited.

Ramya Baki	**B. Mahitha Reddy**	**P.Krishna Vamsi**
(17K81A04C9)	**(17K81A04D9)**	**(17K81A04G4)**

// ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

We extended our deep sense of gratitude to Principal, **Dr. P. SANTOSH KUMAR PATRA**, St. Martin's Engineering College, Dhulapally, for permitting us to undertake this project.

We are also thankful to **Dr.B.Hari Krishna** Head of the Department, Electronics And Communications, St. Martin's Engineering College, Dhulapally, for his support and guidance throughout our project and as well as our project coordinator **Dr.A.Anand**, Professor in Electronics And Communication Engineering, for his valuable support.

We would like to express our sincere gratitude and indebtedness to our project guide **Mrs.G.Laxmi Priyanka**, Assistant Professor, department of Electronics and Communication and our class project coordinator **Mr.K.Nishakar**, Assosiate professor, department of Electronics and Communication, St. Martin's Engineering College, Dhulapally, for their support and guidance throughout our project.

Finally, we express thanks to our college level project coordinators **Dr.Malayadri, professor in IT and Dr.R.Santhosh Kumar,Associate professor in CSE** for helping and advising us to successfully complete this project. Furthermore, we would like to thank our family and friends for their moral support and encouragement.

We express thanks to all those who have helped us in successfully completing the project.

.

1. Ramya Baki
2. B. Mahitha Reddy
3. P. Krishna Vamshi

ABSTRACT

The demand for wireless connectivity has grown exponentially over the last few decades. With the fast development of smart terminals and emerging new applications (e.g., real-time and interactive services), wireless data traffic has drastically increased, and current cellular networks (even the forthcoming 5G) cannot completely match the quickly rising technical requirements. To meet the coming challenges, the sixth generation (6G) mobile network is expected to cast the high technical standard of new spectrum and energy-efficient transmission techniques. In this Project, we sketch the potential requirements and present an overview of the latest research on the promising techniques evolving to 6G, which have recently attracted considerable attention. Moreover, we outline a number of key technical challenges as well as the potential solutions associated with 6G, including physical-layer transmission techniques, network designs, security approaches, and test bed developments.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1.1 Introduction to project:

With the maturity and forthcoming commercialization of the fifth generation (5G), the expectation and development of 6G mobile network have attracted a great deal of attention. In the past two years, some countries have released relevant research plans concerning the development of 6G. For example, in September 2017, the European Union launched a three-year research project on the basic 6G technologies. The main task is to study the next generation forward error correction coding, advanced channel coding, and channel modulation technologies for wireless terabit networks (https://futurecomresearch.eu). At the end of 2017, China began to study the 6G mobile communication system to meet the inconstant and rich demands of the Internet of Things (IoT) in the future, such as medical imaging, augmented reality, and sensing (www.china.org.cn). In April 2018, the Academy of Finland announced an eight-year research program, "6Genesis," to conceptualize 6G through a joint effort of the University of Oulu and Nokia. More recently, the U.K. government has invested in some potential techniques (e.g., €15 million in quantum technology studies) for 6G and beyond (https://www.standard.co.uk/ tech/quantum-technologies), some universities in the United States have lunched research on terahertz-based 6G wireless networks, and South Korea Telecom (SKT) has started 6G research based on the cell-free and non-terrestrial network techniques. In [1], based on the regularity of market entry of past commercial wireless communication systems and the expectation for 6G, the authors forecasted that 6G will start its commercialization in 10 years.

1.2 Project Overview:

The sixth generation mobile and wireless communication network can integrate the satellite communication networks and 5G to provide global coverage. The satellite communication network may consist of telecommunication satellite networks, earth imaging satellite networks and navigation satellite networks. The goal of 6G is to integrate these kinds of satellite networks to provide network position identifier, multimedia and internet connectivity, and weather information services to the mobile users. The four countries which have these satellite systems are: The Global Position

System(GPS) by USA, the Galileo by Europe, the COMPASS by China and the GLONASS by Russia. If 6G integrates with 5G with these satellite networks, it would have four different standards. So handoff and roaming will be a big issue in 6G.The radio over fiber system is already in existence, but with the advent of 6G technology, mankind will be more close to any extraterrestrial civilization in the universe.

1.3 Project Objective:

The objective of our project is

- Multiband Ultrafast speed transmission.
- Super flexible integrated network.
- Multi-mode multi domain joint transmission.
- Intelligent transmission.

CHAPTER 2
LITERATURE SURVEY

2.1 Literature Survey:

Estimated that the international standardization bodies will sort out the standards for

year 2035. While the rollout of 5G is still underway, the researchers across the tentative timeline for the implementation of 5G, B5G, and 6G standards by2020 (IMT-2020 Standard) in 2015 for the 5G network standards. At the same time, standardization of 6G (ITU-R IMT-2030) by the end of the year 2030, whereas 3GPP

workgroup for exploring the system technologies for B5G/6G systems in July communication technologies .

The vision of 5G technologies is extended for the 6G networks by speculating the visionary technologies for next-generation wireless systems in [5]. Different networking scenarios are presented in [15]. The authors in [12] and [13] give a predictive technical framework for industries in future generations of communication systems mainly focusing on the specifications of future generations of the communication system. Cell-less architecture, decentralized networking, and resource allocation, and three- dimensional radio connectivity including the vertical direction are expected in next- generation communication systems. The evolution of wireless systems from 1G to 6G is outlined in [14]. The authors in [15] presented the role of intelligent surfaces in the architecture of 6G networks.

The authors in [16,17] presents the expected technologies, possible applications of 6G. The articles present the system-level perspective of the 6G scenario with use cases, vision, and technologies. The authors in [25] analyze the application of blockchain for the security and privacy measures in upcoming 6G networks. The potential role of optical communication in 5G/B5G and 6G communication networks is described in [5]. The article [26] presents the feasibility of the application of mmWave communication in satellite communication as an enabler of 6G networks. The article [27] gives an analysis of potential applications of device-to-device communication in 6G. The authors in [9] elaborate on the multiple challenges in integrating artificial intelligence (AI) and its potential role in future communication networks.

The authors in [31, 32], have focused on the vision for the next generation of wireless communication systems. Blockchain and AI are the potential technologies for the next generation communication systems. Blockchain can be used for efficient resource sharing and AI can be implemented for the robust, self-organizing, self-healing, and self-optimizing wireless network [33].

By using millimeter-wave (mmWave) and terahertz (THz) frequency bands, massive bandwidth, and highly directive antennas will be available to the 6G mobile devices to enable new applications and seamless coverage [33]. Federal Communications Commission (FCC) has commercialized these frequency bands in 2019 [34]. Ultra-high-precise positioning will become available with 6G due to high-end imaging and direction-finding sensors, just like human eyes and ears. 6G mobile phones could be equipped with capable robots and intelligent algorithms [5].

The latency of the network in 6G will be minimized by using super-fast and high computational power processors both at the network and end devices. The mobile phone of the future network will be intelligent enough to sense the environment and give the precautionary and preventive measures. For example, these mobile phones will be capable to detect the air pollution level, toxic food materials, and explosive materials around us. These phones will replace the wallet, hard cash, and wristwatches by providing digital currencies, and smart watches, respectively. Similarly, smart goggles will replace glasses and smart phones. It is anticipated that 6G cellphones, coupled with the incredibly high directive and beam-steering antennas, would be capable enough to see through the walls by reconstructing the images by receiving the signals from multiple levels of density of the environment in the vicinity [9]. This feature would be useful for extracting minerals and elements from rocks, exploring underground natural reserves, and detecting arms. Apart from this, 6G mobile phones will have tremendous features of providing position, location, and range with very high accuracy. This will be helpful for maritime and underwater communication and positioning.

Self-driving cars, which are already being developed in the initial phase these days, would make human life safer and more comfortable [5]. Holographic technologies and VR/augmented reality (AR) will break the barrier of distances. The digital revolution has transformed the way we play, talk, or work. In the recent era of the digital revolution, 5G has become the center of attention for everyone. Soon the mobile devices in our pockets will get the wireless speed approaching the fiber optic transmission speed, bringing 3D imagery, television, online games, and many other applications that we never imagined into our tablets or mobiles.

Special attention is paid to the improvement of the traffic prediction in [35]. Following the 6G vision and service requirements, some use case scenarios for the 6G, such as autonomous vehicles, smart cities, flying networks, holographic, telemedicine, and Tactile Internet, are discussed in [36]. Moreover, the reliability of the future wireless network is forecasted to be at the same or higher level as that of today's wired communication networks.

Some potential key enabling technologies encompassing block chain-aided decentralization of the network and machine learning (ML)-based intelligent communication system for the 6G are discussed in [37]. A comparative analysis between the key performance indicators (KPIs) for 5G and 6G is carried out in [31]. Practical applications including holography, ML, VR, Internet-of-Things, visible light communication (VLC), automated driving is discussed in [38].

2.2 Existing system:

The existing systems are 1G, 2G, 3G, 4G, 5G. The 1G network was all about voice. 2G network was all about voice and texting. 3G network was all about voice, texting, data. 4G network was everything in 3G but faster version and the 5G is even faster, has better battery life and very low latency than 4G.

2.3 Proposed system:

In general, the 6G mobile network is expected to provide ultrafast speed, greater capacity, and ultra-low latency for supporting the possibility of new applications, such as fine medicine, intelligence disaster prediction, and surreal virtual reality (VR). Based on the former evolution rule of mobile networks, early 6G networks will be mainly based on the existing 5G architecture, inheriting the benefits achieved in 5G (e.g., the increased authorized frequency bands and the optimized de-centralized network architecture) and prodigiously changing the way we work and play. Around 2030, our society will likely become data-driven, enabled by nearly instantaneous, unlimited wireless connectivity [1]. As a result, 6G is expected to advance the wireless technologies we are familiar with today and achieve considerably enhanced system performance. As a vision for the future, in terms of speed, 6G will probably utilize higher frequency spectrum than previous generations in order to improve the data rate expected to be 100 to 1000 times faster than that of 5G [2]. To be specific, 6G networks will allow hundred gigabits per second to terabit-per-second links by making use of multi-band high-spread spectrum; for example, the combination use of 1–3 GHz band, millimeter-wave (mmWave) band (30–300 GHz), and terahertz band (0.06–10 THz) [3]. On the other hand, in terms of capacity, compared to 5G, 6G will be able to flexibly and efficiently connect upper trillion-level objects rather than the current billion-level mobile devices. As a result, the 6G network becomes extremely dense, and its capacity may be 10 to 1000 times higher than that of 5G systems and networks. Furthermore, in terms of latency, from 2G to 5G, the evolution of mobile communication networks is centered on service people, and hence latency depends on human reaction times, such as the auditory reaction time (~100 ms), the visual reaction time (~10 ms), and the perceptual response time (~1ms).

For the application of tactile Internet, 5G technology will allow for a latency time of 1 ms; however, this is too long for Industrial IoT and some other latency-sensitive applications. For example, a minimal latency time is essential for decreasing collision rates and improving the safety in autonomous vehicles. For this purpose, 6G aims for an undectable or even nonexistent latency, since it can enhance the application of autonomous vehicles, augmented reality, and medical imaging. Indeed, with the emergence of more new unmanned and autonomous applications, the latency time no longer solely depends on human reaction times. While the preliminary sketch of 6Gis

being drawn up, efforts on configuring the potential techniques to match the aforementioned appealing vision remain in a nascent stage. It is worth noting that in [1], the authors first provided a general survey of different wireless generations and then highlighted an initial sketch of 6G based on the requirements of future users. Compared to [1], this article summarizes the potential requirements and the latest research on promising techniques toward the evolution to 6G. Another important purpose of this article is to provide the scientific community with an overview of the most challenging aspects in the focused context of 6G mobile networks and to give helpful suggestions for overcoming these challenges.

Laptops, mobile data traffic has observed an exponential growth during the past few years, and this growth is expected in next few years as well. With the increase in the number of mobile users, not only the mobile traffic has increased but every user wants higher data rate with more accuracy and reliability. This considerable amount of mobile data traffic is challenging to manage with current technologies.

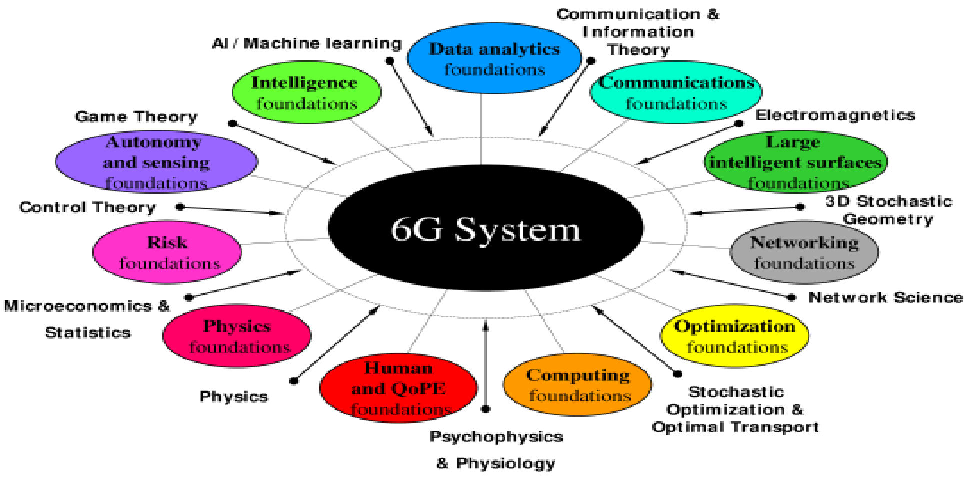


Fig 2.1:Vision of 6G wireless systems.

Future generation network called fifth generation (5G) must accommodate this huge traffic and address the current limitation of data rates reliability, and efficiency. Currently, many technologies are trying to answer all these problems in mobile communication, but none have found an optimal solution. On the top of that, these technologies must take complexity, energy efficiency, reliability into account while designing the new system. Recently, a technology called massive MIMO has been proposed, which uses hundreds or even thousands of antennas at the base station, and it can serve tens of user simultaneously [1]. These thousands of antennas focus the transmission and reception of signal onto smaller region and help the system to achieve high diversity and multiplexing gains to improve reliability and increase data rate [2]. In massive MIMO systems, a user sends pilot towards the base station and based on these pilot signals base station estimates the channel between it and the user. The base station should have knowledge of channel during both uplink and downlink as massive MIMO is a technology that is dependent upon spatial multiplexing [3]. There are several advantages of using these systems such as high spectral efficiency, antenna array gain, high reliability, robustness to internal jamming and interference and energy efficiency. Along with these advantages, Massive MIMO comes up with certain challenges as well, and one of the major challenges that massive MIMO is facing is high computational complexity and poor bit error rate (BER) performance during received signal detection at the base station which due to the higher number of antennas at the base station and more number of users. In Massive MIMO, all the signal transmitted by the user terminals superimpose at the base station and thus interfere with each other. There are several algorithms or methods for Massive MIMO detection and non- linear detectors like sphere detector, and successive interference cancellation detector are computationally very complex. Therefore, these methods are not recommended. Linear detectors are computationally less complex than non-linear detector, but the performance is much degraded. All the conventional detection methods like Maximum Likelihood detection (ML), Minimum mean square error detection (MMSE), Zero-forcing detection (ZF) are not very efficient in terms of performance and complexity. During detection, these linear methods include inversion of high dimensional matrices which drastically increases the complexity of the system and this complexity increases exponentially with more number of antennas. [4]. In this paper, we present a low complex and efficient algorithm for detection of Massive MIMO systems which is based upon modified Approximate Message Passing(AMP)algorithm.

Wireless data traffic had been dramatically increasing over the past few years. Nonetheless, the existing techniques are not satisfying with the users' needs in terms of the emergence of applications for daily routines (e.g., proximity aware services). Therefore, there is a wave of popular interest to seek for new paradigms to deal with this problem. In the coming fifth generation (5G) cellular networks, emerging technologies will lead to both disruptive architectural and component design changes. For instance, in 5G wireless communication systems, diverse researchers study different aspects of millimeter wave transmission, which are plentiful because spectrums have become scarce at microwave frequencies. Massive multiple-input multiple-output, which could increase the system throughput is proposed to utilize a very high number of antennas. We know that 2G–3G–4G cellular networks were built under the design premise of having complete control from the infrastructure side. However, this assumption should be dropped in the 5G systems. The base-station-centric architecture of cellular systems may change, and intelligence at the device side, within different layers of protocol stack, should be exploited, for example, by allowing device-to device (D2D)connectivity.

Device-to-device communication defined as a direct communication between two mobile users without traversing the base station or the core network is considered to be a promising technique, which also offloads the increasing data traffic into user equipments (UEs). In a traditional cellular network, it is implicitly implied that two parties willing to establish the same call will not be in close proximity to each other.

Therefore, all communications must go through the base station. However, in the age of data, mobile users in today's cellular networks are potentially in range for direct communicationsusinghighdatarateservices.

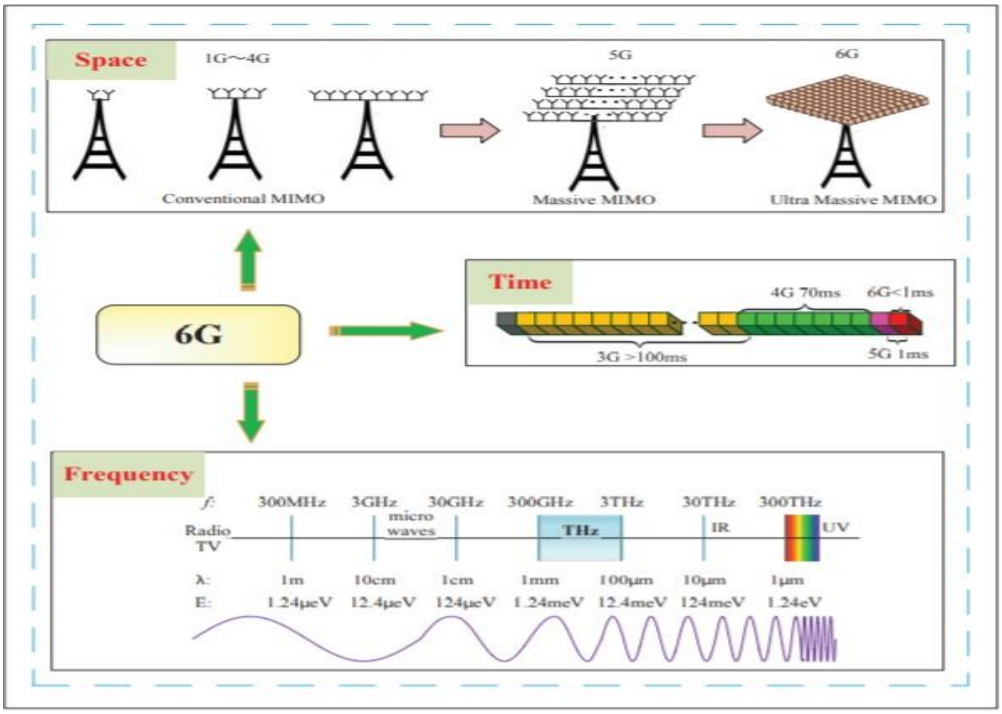


Fig 2.2 : A sketch of 6G based on the time-frequency-space resource utilization.

Thus,D2Dcommunication,whichcan decrease latency and increase resource utilization had been proposed as a means of taking advantage of the physical proximity of communicating devices. Figure 1 shows a simple example of D2D communication. The majority of the literatures in D2D communication proposed to use the cellular spectrum for both D2D and cellular communication. Most of these previous studies have focused on issues such as resource allocation and interference mitigation. Although, few existing studies have investigated the D2D access procedure. Here we review the literatures related to device discovery and access procedure. In TR 22.803, the D2D discovery is categorized into several types, which are summarized in. In addition, the D2D discovery procedure and long-term evolution (LTE)-based design are also discussed in. Yanget al. proposed a distributed peer discovery protocol for LTE-A networks. In, they provided an overview of the new agreements in third generation partners hip project LTE radio access networks related to evaluation methodology and channel modelling for D2D discovery and communications. Honget al. proposed a D2D discovery and link setup procedure and analyzed its performance in terms of energy consumption and delay by utilizing the measurement results of real

LTE smart phones. However, all of the existing works lack the overall performance analysis based on the Markov process model. In this work, we will provide the system model based on the Markov process and present the performance analysis. Moreover, we give our proposal on the Vienna Matlab platform, which is a system level Matlab simulator developed by Vienna University of Technology and obtain the simulation results.

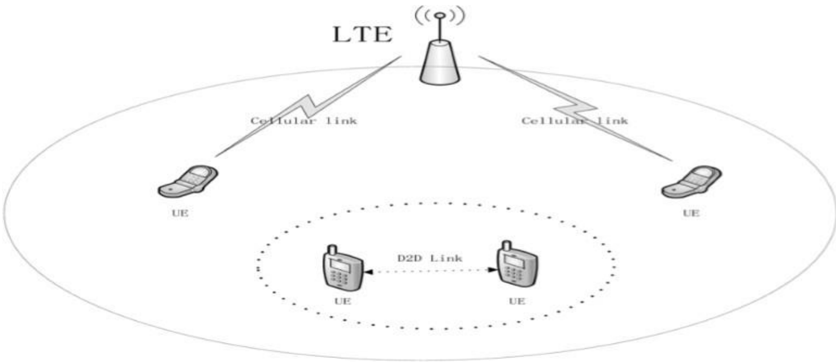


Fig 2.3 : Device-to-device communication in a cellular network.

CHAPTER 3
SOFTWARE INTRODUCTION

3.1 Introduction to MATLAB:

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.

- Typical uses include.
- Math and computation.
- Algorithm development.
- Data acquisition.
- Modelling, simulation, and prototyping.
- Data analysis, exploration, and visualization.
- Scientific and engineering graphics.
- Application development, including graphical user interface building.

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN. The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

3.2 The MATLAB system:

The MATLAB system consists of five main parts

3.2.1 Development Environment:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

3.2.2 The MATLAB Mathematical Function Library:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

3.2.3 The MATLAB Language:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

3.2.4 Graphics:

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully

customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

3.2.5 The MATLAB Application Program Interface(API):

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files. Various toolboxes are there in MATLAB for computing recognition techniques, but we are using IMAGE PROCESSING toolbox.

3.3GRAPHICAL USER INTERFACE(GUI):

MATLAB’s Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions)files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function’s GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M function is executed.

- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed. To launch GUIDE from the MATLAB command window, type guide filename Where filename is the name of an existing FIG-file on the current path. If filename is omitted, GUIDE opens a new (i.e., blank)window.

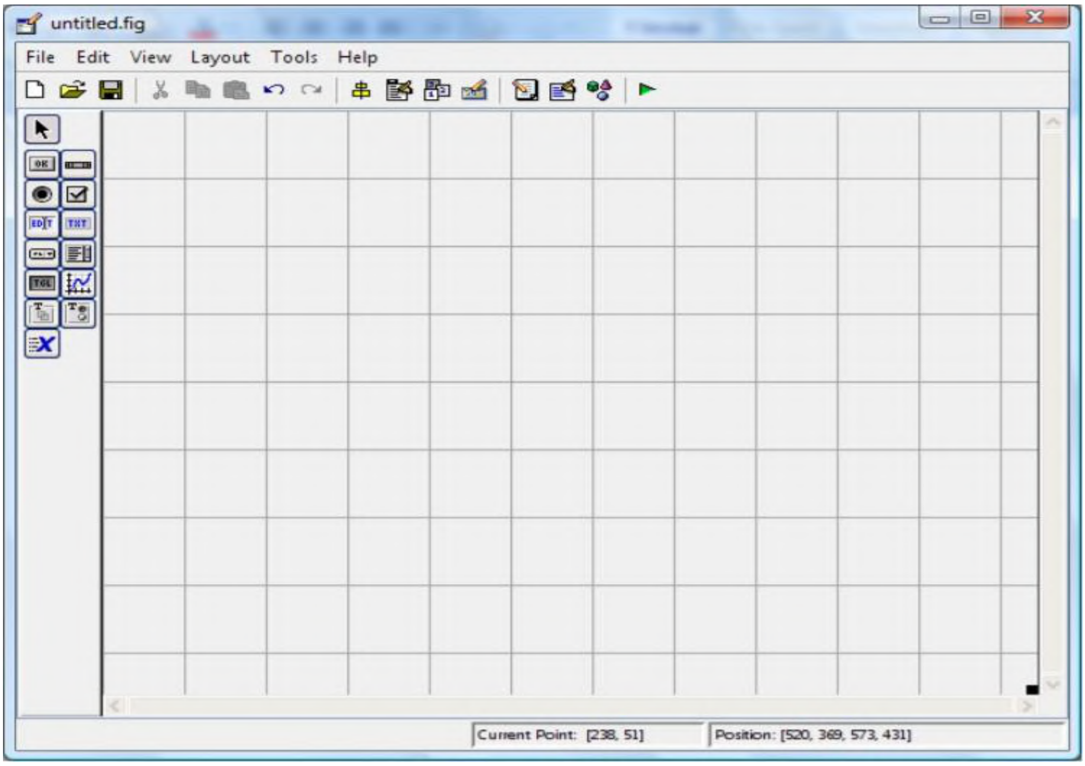


Fig 3.1 :Matlab procedure.

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed. GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

3.4 Getting Started:

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos. At the heart of MATLAB is a new language you must learn before you can fully exploit its power. YoucanlearnthebasicsofMATLABquickly,andmasterycomesshortlyafter.You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

3.5 Introduction:

Describes the components of the MATLAB system.7

3.5.1 Development Environment:

Introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

3.5.2 Manipulating Matrices:

Introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

3.5.3 Graphics:

Introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

3.5.4 Programming with MATLAB:

Describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

3.6 DEVELOPMENT ENVIRONMENT :

3.6.1 Introduction:

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

3.6.2 Starting and Quitting MATLAB :

- **Starting MATLAB:**

On a Microsoft Windows platform, to start MATLAB, double-click the

MATLAB shortcut icon on your Windows desktop. On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop. You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

- **Quitting MATLAB:**

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

3.6.3 MATLAB Desktop:

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB. The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

3.6.4 Desktop Tools:

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools.

The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger

- Command Window
- Command History
- Launch Pad
- Help Browser

- **Command Window:**

Use the Command Window to enter variables and run functions and M-files.

- **Command History:**

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

- **Running External Programs:**

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacsmagik.m invokes an editor called emacs for a file named magik.m.

When you quit the external program, the operating system returns control to MATLAB.

- **Launch Pad:**

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

- **Help Browser:**

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents. To open the Help browser, click the help button in the toolbar, or type help browser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

- **Help Navigator:**

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

- **Display Pane:**

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

- **Browse to other pages:**

Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

- **Bookmark pages:**

Click the Add to Favorites button in the toolbar.

- **Print pages:**

Click the print button in the toolbar.

- **Find a term in the page:**

Type a term in the Find in page field in the toolbar and click Go. Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

- **Current Directory Browser:**

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

- **Search Path:**

To determine how to execute functions you call, MATLAB uses a search path to find Mfiles and other MATLAB-related files, which are organized in directories on your file system.

Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and Math Works toolboxes are included in the search path.

3.6.5 Workspace Browser:

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and who's. To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function. The workspace is not maintained after you end the MATLAB session.

To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

3.6.6 Array Editor:

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

3.6.7 Editor/Debugger:

Use the Editor/Debugger to create and debug M-files, which are programs you write to run MATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

3.6.8 Creating GUIs with GUIDE:

MATLAB implements GUIs as Figure windows containing various styles of unicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of the set asks are simplified by GUIDE, MATLAB's graphical user interface development environment.

3.7 GUI Development Environment:

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks – the functions that execute when users activate components in the GUI.

3.7.1 The Implementation of a GUI:

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIGURE-file - contains a complete description of the GUI Figure and all of its children (uicontrols and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the callbacks, which are defined as subfunctions. This M-file is referred to as theapplication M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIGURE-file.

The following diagram illustrates the parts of a GUI implementation.

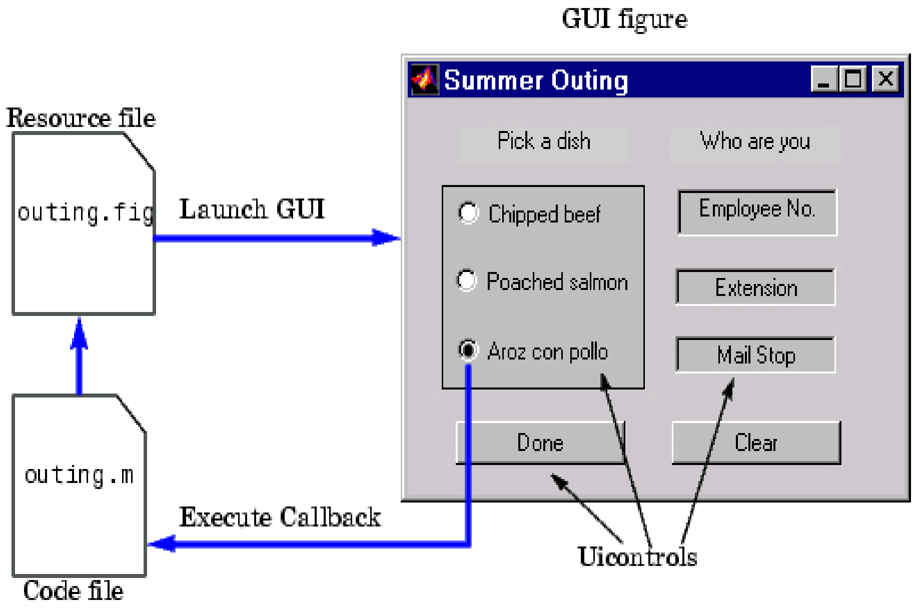


Fig 3.2 Graphical user blocks.

3.7.2 Features of the GUIDE-Generated Application M-Fil:

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Con Figuring Application Options for information on these features). The M- file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIGURE-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIGURE-file generated by the Layout Editor.

3.7.3 Beginning the Implementation Process:

To begin implementing your GUI, proceed to the following sections: Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIGURE-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques used in the application M-file.

Application Examples - a collection of examples that illustrate techniques which are useful for implementing GUIs.

3.7.4 Command-Line Accessibility:

When MATLAB creates a graph, the Figures and axes are included in the list of children of their respective parents and their handles are available through commands such as find object, set, and get. If you issue another plotting command, the output is directed to the current Figure and axes.

GUIs are also created in Figure windows. Generally, you do not want GUI Figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI Figures, resulting in the graph appearing the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

3.7.5 User Interface Control:

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB unicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons

- Frames
- Radio Buttons
- Listboxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

- **Push Buttons**

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its non depressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

- **Programming the Callback**

When the user clicks on the pushbutton, its callback executes. Push buttons do not return a value or maintain a state.

- **Toggle Buttons**

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the non depressed state and again executes its callback.

- **Programming the Callback**

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)

button_state = get(h,'Value');

if button_state == get(h,'Max')

 % toggle button is pressed elseif button_state == get(h,'Min')

 % toggle button is not pressed end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) =rand(16,128);
a(:,:,2) =rand(16,128);
a(:,:,3) =rand(16,128);
set(h,'CData',a)
```

- **Radio Buttons**

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected. Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group.

MATLAB sets the Value propertyto1onthe radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

Functionmutual_exclude(off) set(off,'Value',0)

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual exclude being made from the first radio button's callback in a group of four radio buttons.

function varargout = radiobutton1_Callback(h,eventdata,handles,varargin) off =

[handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];

mutual_exclude(off)

% Continue with callback

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

- **Checkboxes**

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode(e.g., display a toolbar or generate callback function prototypes). The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked. Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

functioncheckbox1_Callback(h,eventdata,handles,varargin) if (get(h,'Value') == get(h,'Max'))

% then checkbox is checked-take approriate action else

% checkboxis not checked-take approriate action end

- **Edit Text**

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback. function

edittext1_Callback(h,eventdata, handles,varargin)

user_string = get(h,'string');

% proceed with callback...

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (is nan) and displays an error dialog(errordlg).

functionedittext1_Callback(h,eventdata,handles,varargin)user_entry

=str2double(get(h,'string'));

if isnan(user_entry)

errordlg('You must enter a numeric value','BadInput','modal') end

% proceed with callback... Triggering Callback Execution

On UNIX systems, clicking on the menu bar of the Figureure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editabletextboxhasfocus,clickingonthemenubardoesnotcausetheeditabletext callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

- **Static Text**

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it.

- **Frames**

Frames are boxes that enclose regions of a Figureure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

- **List Boxes**

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The Listbox Top property defines which string in the list displays as the topmost item when the list box is not large enough to display all list entries. Listbox Top is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer Single or Multiple Selection.

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection. Selection Type

Listboxes differentiate between single and double clicks on an item and set the Figures Selection Type property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This

means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the listbox Value property (and possibly the Figures Selection Type property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string (") and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

- **Popup Menus**

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

- **Programming the Popup Menu:**

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

function varargout = popupmenu1_Callback(h,eventdata,handles,varargin) val =

get(h,'Value');

switch valcase 1

% The user selected the first item case 2

% The user selected the second item

% etc.

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

function varargout =popupmenu1_Callback(h,eventdata,handles,varargin) val =get(h,'Value');

string_list = get(h,'String');

selected_string = string_list{val}; % convert from cell array to string% etc.

- **Enabling or Disabling Controls**

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is grayed out
inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback

routines to execute: First the Figures Window Button DownFcn callback executes. Then the control's Button DownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

- **Axes:**

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

- **Axes Callbacks:**

Axes are not uicontrol objects, but can be programmed to execute a callback whenusersclickamousebuttonintheaxes.UsetheaxesButtonDownFcnproperty to define the callback.

3.7.6 Plotting to Axes in GUIs:

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example, axes(handles.axes1) makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axes. See GUI with Multiple Axes for and example that uses two axes.

3.7.7 Figure:

Figure are the windows that contain the GUI you design with the Layout Editor. See the description of Figure properties for information on what Figure characteristics you can control.

CHAPTER 4

ARCHITECTURE OF 6G NETWORK

4.1 Architecture of 6GNetwork:

Next-generation wireless networks will consist of massive number of connected devices and with the base stations (BSs)/access points (APs) leading to mMTC. Multiple BSs/APs may serve one or more devices simultaneously to form a coordinated multi-point (CoMP) transmission [39]. The huge amount of data produced by massive devices will require very high-performance processing units and robust backhauling links. The central processing units may utilize ML and AI algorithms and the backhauling links may utilize optical fiber and or photonic communications. Remote user, in 6G communication systems, can use several relays or transmitters for a remote user to transmit, and the user's SINR may be improved by using the technique of diversity as in virtual MIMOsystems.

By intelligent networking, all the end devices would be aware of the location and features of BSs/APs in their vicinity, and all of the BSs/APs would be aware of the locations, features, and QoS requirements of devices in their vicinity. Robust interference management/optimization techniques can be applied to maximize the efficiency of the wireless network. Central processing units will be fast enough to manage and switch the resources (bandwidth, time, power) among multiple end-users, and data processing will be conducted at the base-band processing units (BPUs). The air interface is the main component that causes a major improvement in the wireless generations. Orthogonal frequency division multiplexing (OFDM) played a major role in the development of 4G, as code division multiple access (CDMA) was the key player in 3G. Similarly, the development of the new air interface will be an essential component of 6G system architecture.

AI and ML is another crucial component of the 6G system architecture. AI and ML will play an important role in the self-organization, self-healing, self-configuration of 6G wireless systems. Spectrum congestion has also pushed the 6G to adopt a new spectrum for communication. Therefore, this new spectrum will also be an active component in the 6G system architecture. Since 6G will accommodate a wide range of communication devices ranging from IoTs to live HD video transmission, 6G will need to be in line with all previous technologies. Therefore, a flexible and multi-radio access technologies (RAT) system architecture will be an essential component in the 6G network.

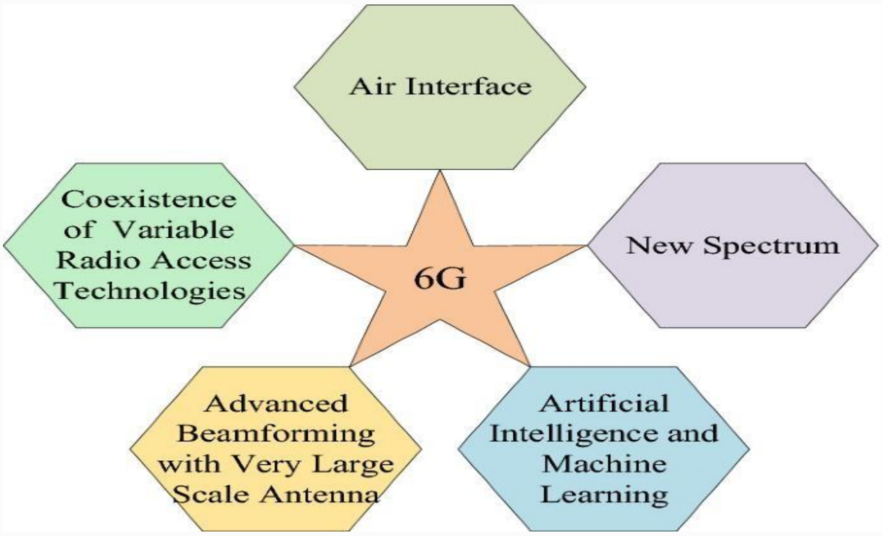


Fig 4.1 : Architecture of 6G network.

4.1.1 Air interface:

Since 6G will concentrate on the current terahertz frequency range with extremely wide bandwidths available, it will bring up new obstacles to interact efficiently at these frequencies. Getting a secure transmission infrastructure that has an adequate range and isn't power-hungry will be the answer here. The availability of incredibly wide bandwidths would change the emphasis from spectrally optimized solutions to improved coverage solutions. In these new frequency spectrums, the tradeoff between spectrum performance, power efficiency, and coverage will play a key role in developing devices. This will lead to the design of a modern air interface where more consideration can be paid to single-carrier systems. The OFDM scheme would be revisited for lower frequency ranges where spectral efficiency will be important as it does not use the energy effectively because of the cyclic prefix, which is just the duplication of information and does not hold any additional information. Furthermore, a high peak-to-average power ratio (PAPR) makes the power amplifiers complex and expensive.

Many researchers have proposed the non – orthogonal multi places (NOMA)as a promising new scheme for the B5G/6G mobile networks [40,41,42]. In NOMA, all of the users are allowed to access the complete resource(frequency band) simultaneously. Some researchers have suggested the rate-splitting multiple access (RSMA) as a new access technology for 6G communication systems [43,44,45]. Both NOMA and RSMA rely on the successive interference cancellations (SIC) to decode the information for the user. RSMA uses the SICs to decode the common message firstly and then decode the private message. Both schemes need to be matured enough before practical deployment. A new AI-based software-defined air interface is presented in [46], where the authors proposed an intelligent air interface switching system for user QoS enhancement.

3GPP release 15, reveals the specifications for the 5G-New Radios (NR), in which multiple waveform configurations and two sets of frequencies are defined. By adopting the variable numerologies (symbol duration, sub-carrier spacing, and pilot spacing), we can give the transmitter leverage to self-organize and self-configure according to the channel conditions and service required. This is often useful on different measurements. For instance, by reducing the symbol length, low latency can be achieved, and increasing the spacing of the sub-carrier can be helpful in reducing the phase noise in mmWave and sub-mmWave. In high mobility situations, optimizing the sub-carrier width can also be helpful for Doppler shift compensation.

4.1.2 New spectrum:

mmWave is already a candidate for 5G, but it is not exploited to its full potential as the beam forming algorithms are not mature enough. It requires improvements in the networks when personal BSs and satellite connectivity can get merged into cellular communication. In the previous generations, the spectrum is divided for multiple services, for instance, television (TV) services, military communications, and cellular communication [47].

Therefore, the idea of using an unlicensed spectrum is proposed, i.e., to use the mmWave, THz band, and visible light spectrum, simultaneously [48,49,50,51,52]. These bands are never used for any communication. The problem with the higher frequency band, though, is that the signal is attenuated very rapidly about the distance travelled. For example, a 3G or 4G BS can have a coverage of about several miles whereas a 5G or 6G BS coverage may limit to only a few hundreds of meters. To resolve this issue in mmWave and THz communications, the idea of using massive multiple inputs and multiple outputs (MIMO) and beam forming emerged, which is described in the next subsections.

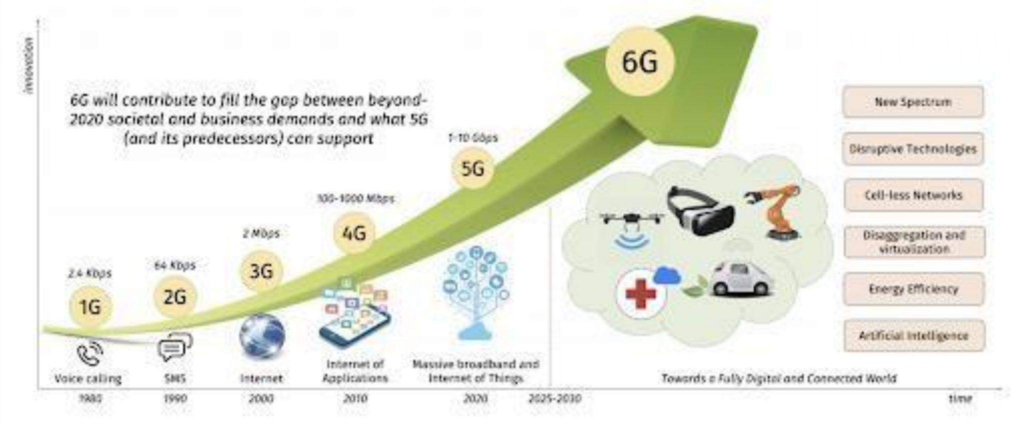


Fig 4.2 :mmWave network dimensions.

4.1.3 Artificial intelligence/machine learning:

By offering pervasive, secure, and close proximity-instant wireless networking for humans and computers, 6G wireless communication networks would be the core of society’s digital transition. A broad variety of emerging developments, such as self-driving cars and voice assistants, have been made possible by recent advancements in ML research. B5G/6G wireless networks have increased complexity, requiring smarter methods for handling any losses and handling network features, detecting anomalies, and understanding KPI trends. This can be done by introducing solutions for ML and SDN. In order to preserve a certain level of KPI, ML/AI will boost the decision-making process. The operation and implementation of RAN for 6G needs a new strategy. Incorporating AI in wireless algorithms (e.g., for channel estimation, for channel state information (CSI) feedback, and decoding, etc.) may bring a change in the direction of these algorithms [53]. Application of ML, DL [54], and AI algorithms to the communication network, we can instantly manage the resources as per the user requirements. The probability of choosing the best solution is improved in this way and the network can maintain its optimum state.

4.1.4 Advanced beam forming with very large scale antenna (VLSA):

The idea of beam forming is to steer the beam to only the desired direction or user. Since energy is not spread in all directions, the transmission range is thus improved by concentrating the beam in one direction.

4.1.5 Intelligent reflecting surfaces(IRSs):

Intelligent Reflecting Surfaces (IRSs) can be the potential area for beam forming in 6G [55].IRSs are composed of thin electromagnetic materials, which can reflect/configure the incoming electromagnetic rays in an intelligent way by configuring the phase of reflected rays by a software [55]. Indeed, IRSs use at a large number of low-power and low-cost passive elements to reflect the incident signals with configurable phase shifts without the requirements of additional power, encoding, decoding, modulation, demodulation requirements. IRSs are installed on the important points and locations such as high-rise buildings, advertising panels, vehicles (cars, airplanes, unmanned aerial vehicles (UAVs)), and even the clothes of the pedestrians. The main advantage of the IRS is that it can enhance the signal-to-interference-plus-noise-ratio (SINR) with no change in the infrastructure or the hardware of the communication network. Also, there is no need for extra power required for the installation.

IRS can reduce the hardware complexity at the receiver and the transmitter by reducing the number of antennae installed at them, thereby, reducing the radio frequency (RF) chains at the transmitter and the receiver. IRS can replace the conventional relays system due to its advantages in terms of power, spectral efficiency, and reduced hardware complexity [56]. IRS can be used in the deep-fade and non-line-of-sight (NLOS) communication environment. The principle by which SINR is enhanced at the receiver is optimally controlling the phases of the incident ray at multiple elements of the IRS, to produce useful beam forming at the receiver [56]. Degradation factors such as noise and interference have no impact on the IRS. All these features of the IRS make it a promising technique for the B5G/6G communication systems.

4.1.6 Orbital angular momentum (OAM)-aided MIMO:

A new dimensional property of the electromagnetic waves (EW) was discovered in the 1990s termed as the orbital angular momentum (OAM). This discovery promised the transmission of multiple data streams over the same spatial channel. An EM wave carrying the OAM has the phase rotation factor of $\exp(-jl\phi)\exp(-jl\phi)$, where l is OAM state number represented in integer and $\phi\phi$ is transverse azimuth angle [57,58,59]. The main advantage of OAM over other beam forming techniques is that OAM can have an unlimited number of orthogonal modes, which allows the EW to multiplex multiple data streams over the same spatial channel, thereby, enhancing the spectral efficiency and transmission capacity. OAM support a high number of user in mode division multiple access (MDMA) scheme without utilizing extra resources (i.e., frequency, time, and power). The flexibility of OAM to be used in narrowband and wideband frequency hopping scheme makes it an attractive scheme for low probability of interception (LPI) applications. OAM-based MIMO systems have advantages over the conventional MIMO systems in terms of capacity and long-distance line-of-sight (LoS) coverage [60]. Therefore, OAM has great potential for applications in 6G wireless networks.

4.1.7 Coexistence of variable radio access technologies:

6G can lead to a ubiquitous networking infrastructure where users would not only be left with the option of selecting the best communication network. Each node in this network would, however, be intelligent enough to sense the conditions of the channel and the specifications of QoS at any other node. For example, the use case and the availability of network will decide the network as cellular, wireless LAN, Bluetooth, and ultra-wideband (UWB), etc. 6G communication standard should, therefore, be designed in such a way that it will converge all of the wireless technologies. Communication with Wi-Fi, Bluetooth, UWB, VLC, UAVs, biosensors, and satellite communications can all integrate into 6G and should fall under one standard such that all of them can connect with each other. The Wi-Fi operating at 2.4 GHz has already entered deeply into IoTs as most of the appliances are now connected through this network [61,62,63].

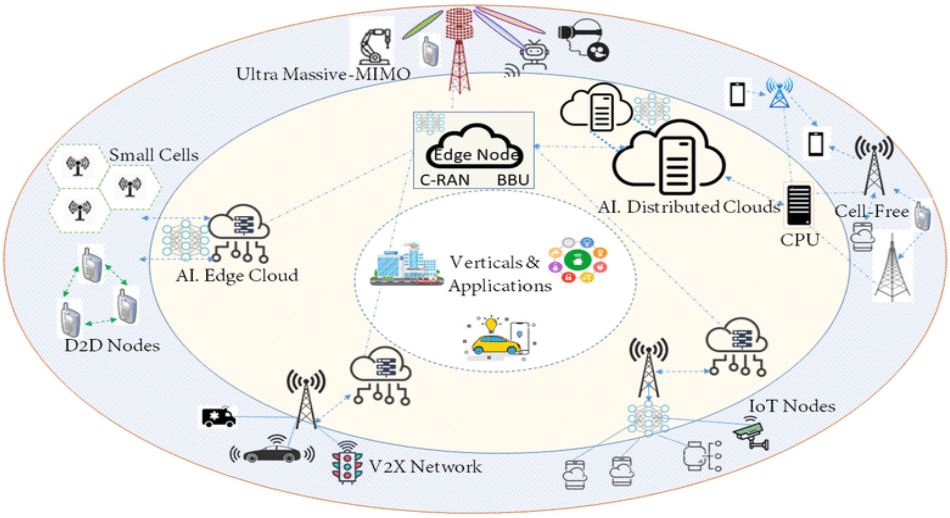


Fig 4.3 : Network level of 6G architecture.

By merging all these technologies, 6G would be able to utilize the massive infrastructure deployed by previous technologies, which otherwise can cost 6G a huge revenue. The features in the previous technologies, such as network densification, high spectral efficiency, high throughput, low latency, high reliability, and massive connectivity should be converged in 6G. 6G technology should also keep the trend of offering new services by applying the new technologies, such as AI/ML, VLC, quantum communications (QC), and quantum machine learning (QML). These services may include but are not limited to smart cars, smart homes, smart wearable, and 3D mapping [64].

4G introduced mobile Internet, mobile TV, and HD videos. AR/VR, ultra-HD (UHD) videos, wearable devices, vehicle-to-infrastructure (V2X), smart city, telemedicine, and IoTs concepts are introduced in 5G. 6G is projected to have applications such as space tourism, Tactile Internet, fully automated cars, holographic verticals, deep-sea sight, digital sensing, and Internet-of-bio-Nano-things (IoBNT). Figure 4b shows that how KPIs are changing with the evolution of wireless generations from 1G to 6G. All Internet protocol (IP) and the ultra-broadband concept is introduced in 4G. The concepts of cloudification, softwarization, slicing, virtualization, and wireless worldwide web (WWW) are introduced in 5G. Integration of intelligence with cloudification, softwarization, slicing, and virtualization will be introduced in 6G communication systems. The initial stage of the wireless communication system is the development of the advanced mobile phone system (AMPS). Global systems for mobile (GSM) and general packet radio systems (GPRS) family is developed in 2G wireless systems. Code-division multiple access (CDMA) family shifted the wireless systems from 2G to the 3G. OFDM with the integration of turbo codes and MIMO systems is the key technology for 4G communication systems. 5G communication systems brought some new technologies such as cloud/fog/edge computing, massive MIMO, SDN, mmWave and sub mmWave (NR) along with low-density parity-check (LDPC) and polar codes. ML, AI, blockchain, THz communication, orbital angular momentum multiplexing (OAM Mux), spatial Modulation (SM)-MIMO and intelligent re-configurable reflecting surfaces are the new technological domains in 6G.

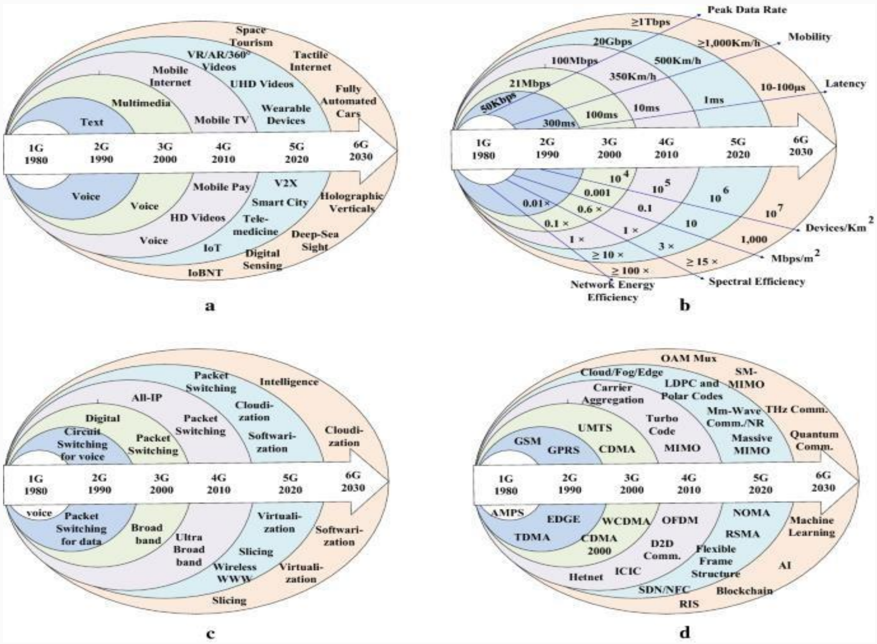


Fig 4.4 : Evolution of wireless communication, with timeline, from 1G to 6Gbased on (a) Applications (b) KPIs (c) Network characteristics and (d)Technological development.

CHAPTER 5

NETWORK DIMENSIONS AND POTENTIAL TECHNIQUES

Network intelligence will be an essential component of 6G networks and the network will take actions dynamically according to the environmental conditions. The idea of clouds, fog, and edge computing is applied for fast access to services. The features of self-optimization, self-organization, self-reconfiguration will be achieved through softwarization, virtualization, and slicing. The detailed discussion on each network dimension is given as follows.

5.1 Intelligence:

Researchers believe that AI will play a defining role in the IoTs and IoBNTs driven world [65]. The potential shift from 5G to the 6G will be to determine an efficient way to transmit data. The ideal system will be the one that is free from human intervention at all [66].

5.2 Cloudification/fog/edge:

Thousands of sensors are installed in the industries and hundreds of the sensors are installed in homes. It is very difficult to connect all these sensors with wires [67], and, all these devices can produce a large amount of data. Also, these devices are smart and intelligent, capable of making smart decisions and less processing power. Therefore, we need to offload the data from cloud to edge and device end. To reduce the processing delay, we need to shift the process near to end devices in terms of cloud/fog. We need to place the workload closer to the edge for a better quality of service.

5.3 Softwarization:

Main driving force behind the development of B5G and 6G networks is to provide services such as self-organization, configurability, programmability, flexibility, and heterogeneous use-cases. It is difficult to install the hardware equipment which provides all of the mentioned functionalities. By realizing the functionalitiesby underlying networks, softwarization and virtualization have emerged as the two most demanding paradigms for B5G/6G networks [68].

Softwarization is the term used for the set of interfaces and protocols which can allow the network to be configured in software by decoupling the control and user plane. The user plan usually consists of a set of distributed and stateless routing tables at which packet switching is performed at a very high speed. These tables are updated by the centralized control plan which maintains the end-to-end routing information for multiple services. Data and control management operations are exchanged between the service consumer and the SDN provider [36]. SDN provider ultimately forwards the required service to the service consumer. These services are controlled by the service consumer by taking acting on these virtual resources.

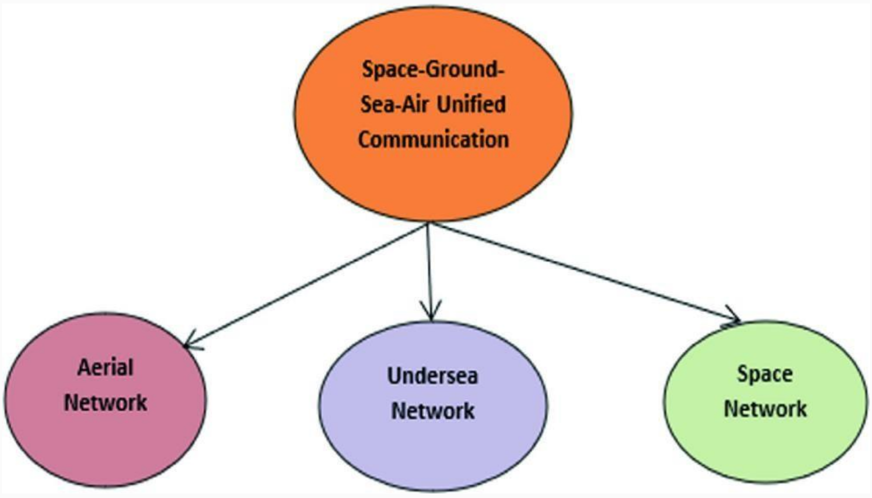


Fig 5.1 : Green 6G network era.

5.4 Virtualization:

Network function virtualization enables the software functions to be performed in the virtual machines and allows the access of common shared physical resources such as storage, networking, and computations. Containers are used to instantiate multiple functions within the same virtual machine. Dynamically varying network demand such as offered services and network traffic can be handled by dynamically instantiating the virtual machines. The services, which can be virtualized, include but are not limited to load management, mobility management, baseband processing unit services, evolved packet core (EPC) functions. Network function virtualization(NFV), unlike traditional mobile networks, provides the leverage to route the packet of each service between virtual network functions (VNFs) [69]. Further, the routing services are provided with very few overheads. Also, routing and traffic flow are smoothly maintained without any interruption even if a new VNF is added or removed according to the traffic demands.

5.5 Slicing:

One of the key network abilities that will allow us to build a flexible network on top of the common physical infrastructure is network slicing. As 5G continues to take shape, network slicing will become the fundamental technology to enable a wide range of use cases. Taking a single piece of network infrastructure and being able to cost-effectively deliver multiple logical networks over the same common physical infrastructure [69]. The slices can be allocated to some specific use-cases such that we can have a slice for IoTs, slices can be allocated to a class of service, slices can be allocated to a class of customers, slices can be allocated to some specific mobile network operators, slices can also be allocated to network types such as wireless vs wired or consumer vs businesses.

In network slicing, the biggest difficulty is configuring new slices, since it affects all the network components. However, with the need to create customized services and deliver a service with a very specific requirement, we can create slices such as slice for automotive, healthcare, utility.

It is possible to communicate with the devices with very low data-rates such as biosensor and IoTs, and at the same time, it will enable high data rate communication such as HD video transmission in smart cities. Communication will be possible in a fast-moving bullet train, airplane. It also shows that all of the networks will be merged all together. Further, the buildings and surfaces in smart cities can be equipped with the IRS that could enhance the coverage and quality of service (QoS) of each communicating device. For the maritime communication scenarios, the robust underwater data links will enable the communication between ships, submarines, and sensors at the deep sea level [70, 71]. Besides, innovative technologies such as

AR/VR, haptics, and ML will further reduce the effect of physical distances around the globe.

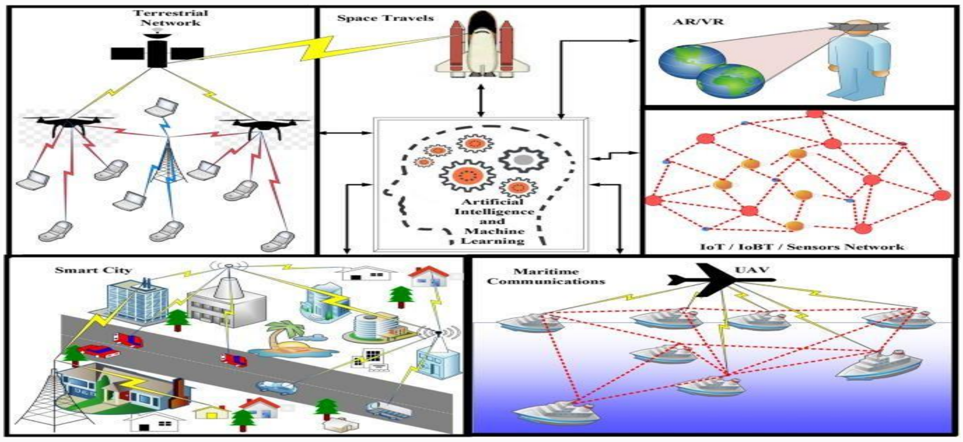


Fig5.2 : Depiction of space-air-ground-sea based integrated next-generation communication system.

The 6G core network is shown to have upgraded to the basic 5G core network based on intelligence, high computational power, and high capacity. By integrating BSs/APs, satellites, and UAVs, the access network is upgraded similarly. There is a vertical hand-off in 6G in addition to the horizontal as in that of 5G. Besides, fog computing and MEC are an integral component of the 6G network infrastructure, that reduces latency and bandwidth utilization for regularly needed services by a massive number of devices on the user plan.

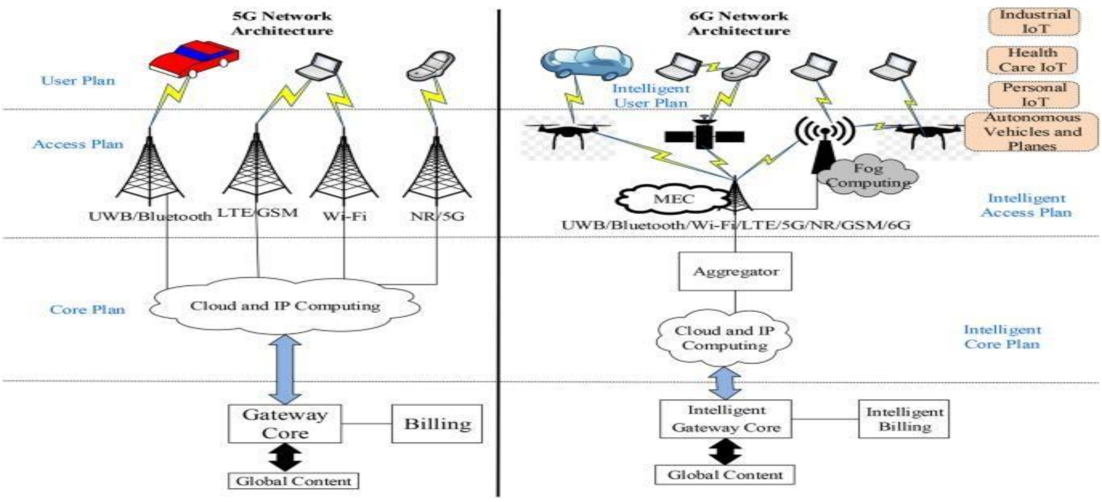


Fig 5.3 : A comparative analysis between 5G and 6G network architecture.

5.6 Device to device communication:

Device-to-Device (D2D) communication in cellular networks is defined as direct communication between two mobile users without traversing the Base Station (BS) or core network. D2D communication is generally non-transparent to the cellular network and it can occur on the cellular frequencies (i.e., inband) or un licensed spectrum (i.e.,outband).

In a traditional cellular network, all communications must go through the BS even if communicating parties are in range for proximity-based D2D communication. Communication through BS suits conventional low data rate mobile services such as voice call and text messaging in which users are seldom close enough for direct communication. However, mobile users in today's cellular networks use high data rate services (e.g., video sharing, gaming, proximity-aware social networking) in which they could potentially be in range for direct communications (i.e., D2D). Hence, D2D communications in such scenarios can greatly increase the spectral efficiency of the network. The advantages of D2D communications go beyond spectral efficiency; they can potentially improve throughput, energy efficiency, delay, and fairness.

Table I. Device-to-device strategy list.

Parameters	Length/bit	Value	
Communication patterns	2	00	Orthogonal
		01	Reused
		10	Cellular
Power indicator	1	0	High
		1	Low
Spectrum	8	0	Whole spectrum
		[7:4]	Starting
		[3:0]	Ending
Scan time	8	Unit: ms	

Fig 5.4 : Device to device strategy list.

5.7 Data delivery in non-cooperative D2Dcommunication:

Existing data delivery protocols in D2D communications mainly assume that mobile nodes willingly participate in data delivery, share their resources with each other, and follow the rules of underlying networking protocols. Nevertheless, rational nodes in real-world scenarios have strategic interactions and may act selfishly for various reasons (such as resource limitations, the lack of interest in data, or social preferences).

For example, if a node has limited battery resources or the cost of the network bandwidth delivered by mobile network operators is high, it would not willingly relay data for others until appropriate incentives are provided. Meanwhile, malicious nodes may attack the network in different ways to disturb the normal operation of the data transmission process. An adversary, for example, may drop received messages but produce forged routing metrics or false information with the aim of attracting more messages or decreasing its detection probability. This issue becomes more challenging when colluding attackers boost their metrics to deceive the attack detection systems. Dealing with non-cooperative mobile nodes is very challenging because of the distributed network model and intermittent access of nodes to central authorities.

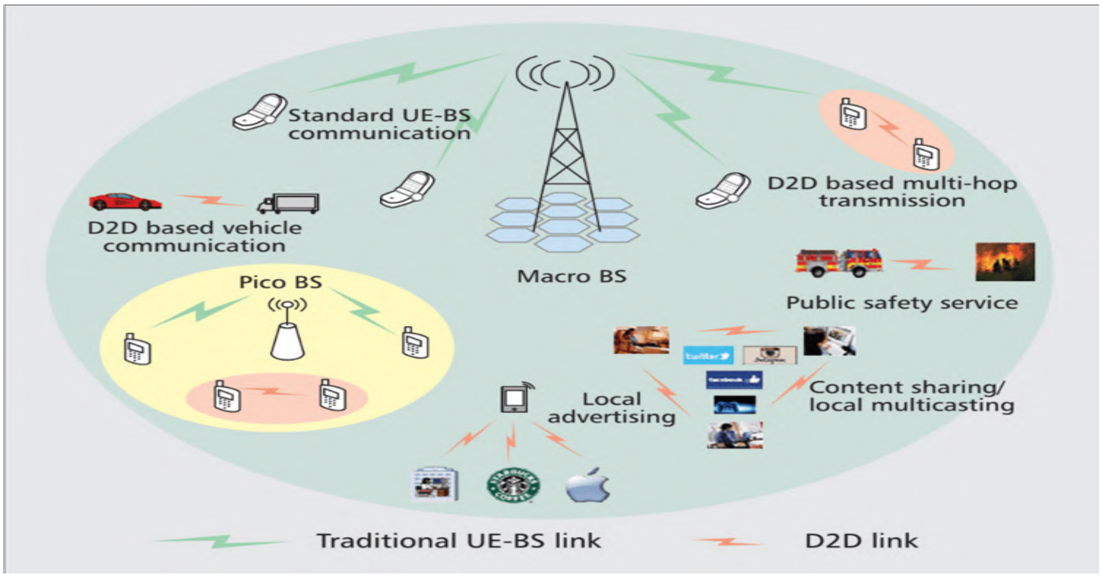


Fig 5.5 : Device to device communication.

5.8 D2Dapplications:

D2D Communications is used for

- Local Services: In local service, user data is directly transmitted directly between the terminals and doesn’t involves network side, e.g. social media apps, which are based on proximity service.

- Emergency communications: In case of natural disasters like hurricanes, earthquakes etc., traditional communication network may not work due to the damage caused. Ad-hoc network can be established via D2D which could be used for such communication in such situations.

- IoT Enhancement: By combining D2D with Internet of things (IoT) , a truly interconnected wireless network will be created. Example of D2D-based IoT enhancement is vehicle-to-vehicle (V2V) communication in the Internet of Vehicles (IoV). When running at high speeds, a vehicle can warn nearby vehicles in D2D mode before it changes lanes or slows down.

5.9 Potential Features:

In order to provide satisfying services for Industry 4.0, personalized health services, virtual presence, and other challenging anticipated applications in the future, 6G needs to further enhance its scalability, flexibility, and efficiency by embracing novel techniques. Like the emergence of many new technologies when the wireless world moves toward 5G, the new requirements of 6G will influence the main technology trends in its evolution process. The success of 6G will have to leverage breakthroughs in novel technological concepts.

A vide range of recent research findings related to 6G design, including multi-band ultrafast-speed transmission techniques, super-flexible integrated network designs, multi-mode multi-domain joint transmission, as well as machine learning and big-data-assisted intelligent approaches.

5.9.1 Multi-Band Ultrafast-Speed Transmission:

For our bandwidth-hungry society, 2G, 3G, and 4G have used frequencies that reach approximately up to 6 GHz, while 5G systems exploit the range of less than 6 GHz as efficiently as possible by combining 24–100 GHz. Recently, developers are realizing that the current frequency bands may not be enough to serve the growing demands; for example, an uncompressed ultra high-definition video may reach 24 Gb/s, and some 3D videos may reach to 100 Gb/s [3]. As a result, in 6G, we will jump above 100 GHz, and the new radio will consider not only the traditional sub-5 GHz band but also validate little-explored frequency sources such as mmWave and terahertz bands to overcome the spectrum scarcity and provide wide bandwidth from hundreds of megahertz to several gigahertz and even to terahertz. In recent years, a flurry of research activities have been reported concerning the use of multiple high-frequency bands for ultrafast-speed transmissions, which are recommended as promising solutions for 6G. Specifically, a consortium of DARPA, IBM, and Intel has focused on research into using 140 GHz, 220 GHz, and 340 GHz frequencies (https://spectrum.ieee.org). In early 2014, Akyildiz et al. provided an in-depth view of terahertz band in the range of 0.1–10 THz for supporting terabit-per-second high-speed communications [4]. A further advance of terahertz band communication is that it is enabled to simultaneously support macro-scale and micro-scale applications, such as terabit wireless local area networks (WLAN) and nano sensor networks. Teams at New York University are already working on terahertz research and quantum devices, with the goal that transmit rates in 6G are expected to be 1000 times faster than those in 5G. Researchers from the Fraunhofer Institute have cooperated with researchers from Heinrich Hertz Institute and other partners from industry to build a network connection in the terahertz frequency range that is sufficiently stable to allow for wireless data transmission at speeds of up to 400 Gb/s (https://www.networkworld.com). In [3], several terahertz-based medium access protocols for mobile heterogeneous networks were investigated, where conventional mWave, mmWave, and terahertz bands could coexist. Moreover, another trend in 6G may be dynamically using different frequencies, especially these unlicensed bands, in order to enjoy the benefits of the sharing architecture. For example, as predicted in [1], optical indoor communication (visible-light frequency band) is a promising high rate technique to replace conventional WLAN, since people spend most of their time indoors (more than 80 percent). Furthermore, we can also combine the fiber optic technology with the wireless-type transmissions for further improvement. The coexistence of multiple high-frequency bands and the dynamic utilization of different frequencies can be realized by advanced software defined radio (SDR) and software defined networking (SDN) techniques. Moreover, the emerging block chain technique may be an appealing solution to facilitate dynamic spectrum sharing in the future (https://venturebeat.com/). With these aforementioned novel techniques, it is expected that 6G networks will be easier to upgrade based on the existing 4G and future 5G equipment.

5.9.2 Super Flexible Integrated Network:

6G systems will also need to serve a wide range of applications in diverse scenarios, which have been defined in 5G. Moreover, with the rise of smart homes, buildings, cities, and society, 6G will meet the increased demands for human-to-machine and machine-to-machine communications, especially with the development of robotic and autonomous drone systems. For the purpose of the Internet of Everything (IoE) in the 6G systems will also need to serve a wide range of applications in diverse scenarios, which have been defined in 5G. Moreover, with the rise of smart homes, buildings, cities, and society, 6G will meet the increased demands for human-to-machine and machine-to-machine communications, especially with the development of robotic and autonomous drone systems. For the purpose of the Internet of Everything (IoE) in the 6G systems will also need to serve a wide range of applications in diverse scenarios, which have been defined in 5G. Moreover, with the rise of smart homes, buildings, cities, and society, 6G will meet the increased demands for human-to-machine and machine-to-machine communications, especially with the development of robotic and autonomous drone systems. For the purpose of the Internet of Everything (IoE) in the 6G systems will also need to serve a wide range of applications in diverse scenarios, which have been defined in 5G. Moreover, with the rise of smart homes, buildings, cities, and society, 6G will meet the increased demands for human-to-machine and machine-to-machine communications, especially with the development of robotic and autonomous drone systems. For the purpose of the Internet of Everything (IoE) in the 6G systems will also need to serve a wide range of applications in diverse scenarios, which have been defined in 5G. Moreover, with the rise of smart homes, buildings, cities, and society, 6G will meet the increased demands for human-to-machine and machine-to-machine communications, especially with the development of robotic and autonomous drone systems. For the purpose of the Internet of Everything (IoE) in the 6G systems will also need to serve a wide range of applications in diverse scenarios, which have been defined in 5G. Moreover, with the rise of smart homes, buildings, cities, and society, 6G will meet the increased demands for human-to-machine and machine-to-machine communications, especially with the development of robotic and autonomous drone systems. For the purpose of the Internet of Everything (IoE) in the 6G systems will also need to serve a wide range of applications in diverse scenarios, which have been defined in 5G. Moreover, with the rise of smart homes, buildings, cities, and society, 6G will meet the increased demands for human-to-machine and machine-to-machine communications, especially with the development of robotic and autonomous drone systems. For the purpose of the Internet of Everything (IoE) in the future, 6G will be an ultra-dense network with super flexibility, which can deftly integrate different techniques to simultaneously satisfy various different service requirements. Moreover, 6G networks will be decentralized and designed based on the integration of terrestrial communications, aerial networks, and satellite systems in order to accommodate emerging and urgently needed services (e.g., disaster prediction), and to realize global coverage and stringent seamless access, even for ocean and mountain areas. Recently, many efforts have been devoted to designing super flexible integrated networks. Specifically, in [3] the authors proposed a scalable heterogeneous network architecture for supporting THz band communications, where a novel medium access control (MAC) protocol facilitating multi-band transmission was proposed. In [5], a potential architecture of the space-terrestrial integrated network (STIN) that integrates the space network, Internet, and mobile wireless network was introduced. In general, STIN is a large-scale heterogeneous network with high flexibility, complex structure, and dynamic topology. By jointly exploiting the advantages of satellite systems, air segment networks, and ground segment systems, this multidimensional network will bring lots of benefits for future 6G wireless communications. In particular, as shown in [6], with the increasing number and types of aerial vehicles, such as balloon, airship, and unmanned aerial vehicle (UAV), the flying base station (FBS) assisted dynamic networks can be built to improve the conventional static structure. For example, in recent years, many different FBS projects have sprung up, such as the Google Loon project, the Facebook Aquila project, the ABSOLUTE project, and the Matternet project[6].

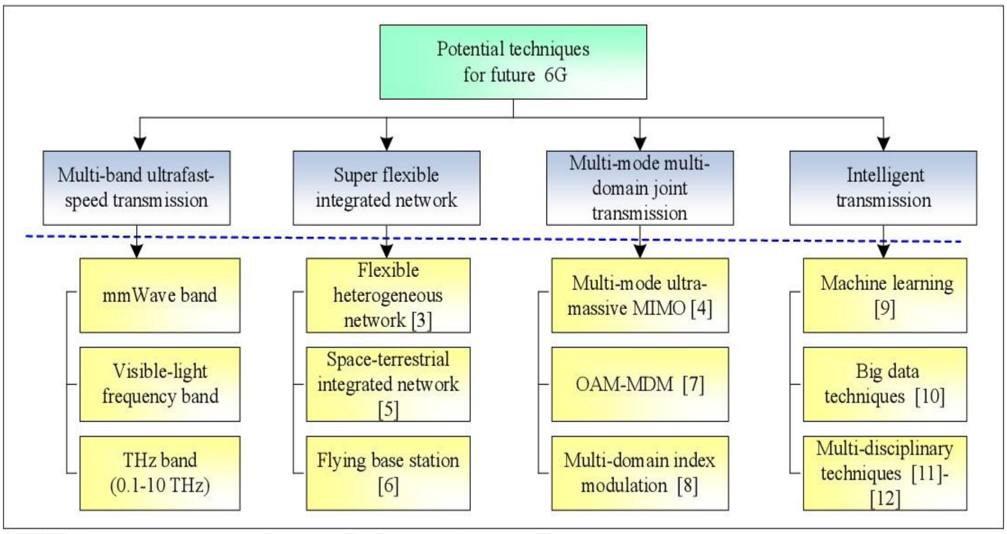


Fig 5.6 :Potential techniques of 6G.

5.9.3 Multi-Mode Multi-Domain Joint Transmission:

One of the most challenging tasks in 6G is to conceive suitable physical-layer transmission techniques to support the newly used spectral bands and enable new applications, including ultrahigh-speed indoor wireless services. For example, when THz band frequencies are utilized, how to deal with the high spreading loss and molecular absorption is a vital issue. For this research objective, many universities and research centers have begun to study the next generation forward error correction coding, advanced channel coding, and channel modulation technologies for multi-band ultrafast-speed wireless communications [1, 4]. In [4], a distance-adaptive physical layer design was proposed for mmWave and THz band communications, where each GHz (even THz) ultra-wide band was divided into narrower but still broadband sub-windows for allowing parallel multiple wide band transmissions.

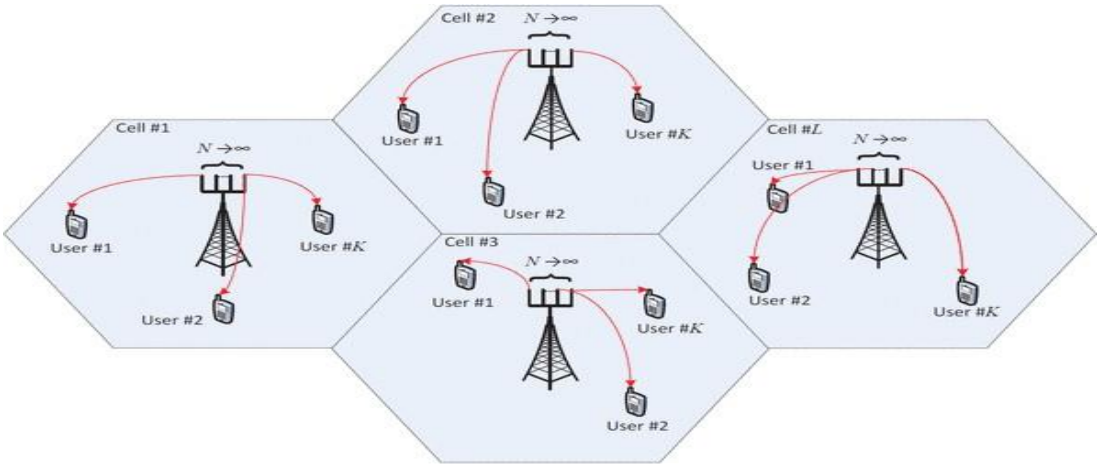


Fig 5.7 : Massive MIMO.

In each band, to add efficiency, novel multiple antenna techniques can be adapted, such as the PM-MI - MO technique. Specifically, as shown in [4], a novel class of PM-MIMO, namely UM-MIMO, is a promising solution for increasing the communication distance and improving the attainable capacity of the THz-band networks. Moreover, for further enhancing the PM-MIMO design, the multi-mode multiple antenna techniques, such as beam forming (BF) and spatial multiplexing (SMX), can be dynamically combined and adapted. BF can effectively decrease the effects of high attenuation at mmWave and THz bands, while SMX is able to increase the capacity per user. The above benefits of BF and SMX can be simultaneously obtained by adopting their combination in these parallel broad bands. Furthermore, to achieve the diversity gain inherited in MIMO for some specific applications, space diversity techniques, such as some classic space-time block codes (STBCs), can also be integrated. Besides the multi-mode techniques, multi-do-main joint transmission techniques are also promising for future 6G transmission. For example, orbital angular momentum-based mode-division multiplexing (OAM-MDM) is an emerging low-complexity but high-spectral-efficiency physical layer solution for short-distance line-of-sight wireless communications, which is capable of employing all available DoFs to convey the information over wireless links, including phase, polarization state, and other spatial DoFs [7]. In contrast, for innovatively exploiting the DoFs of multiple antennas (space-domain) and multiple carriers (frequency domain), in [8] a multi-do - main index modulation (MD-IM) technique was proposed, which relies on the generalized on/ off keying principle applied to any of the available signal resource domains to modulate the information bits on to the indices of the transmit resources, including the indices of subcarriers, transmit/receive antennas, code types, dispersion matrices, signal powers, precoding matrices, and so on. In general, OAM-MDM and MDIM create completely new dimensions for data transmission. The main benefit of OAM-MDM and MD-IM techniques is that they can be flexibly configured to satisfy different performance requirements for supporting a wide variety of applications, which is in accord with the basic requirement of 6G network design. As a further advance, the above-mentioned multi-mode and multi-domain techniques can be roundly and penetratively combined for more efficient and flexible designs.

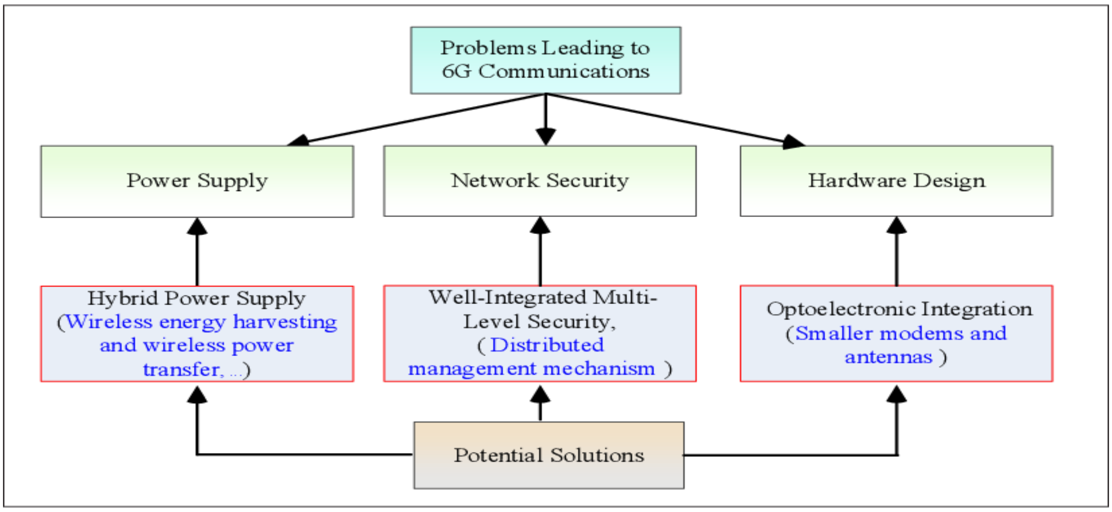


FIGURE 3. Some potential problems for 6G development and their promising solutions.

Fig 5.8 : Potential problems for 6G development.

5.9.4 Machine Learning and Big Data Assisted Intelligent Transmission:

Another trend predicted for 6G is intelligent networks and technologies to enable a fully immersive experience for users. For this research objective, 6G needs to be innovated by using and combining technologies from other fields. In recent years, breakthroughs have been made in the fields of artificial intelligence and machine learning technologies, such as deep learning neural network (DNN) algorithms [9]. In machine learning, the optimal solution (e.g., the optimal transmit mode) is capable of being obtained by classification or neural network learning instead of tedious calculation, where the classifiers and DNNs can be trained by offline datasets. These machine-learning-based methods are the best candidates to improve the design and optimization of wireless communication systems. Specifically, the key issues behind synchronization, channel estimation, equalization, MIMO signal detection, iterative decoding, and multi-user detection in wireless communication systems are similar to the theoretical basis of machine learning. 6G will be a big data-driven network, since it is expected to support extremely high data rates with heterogeneity in applications, devices, and networks. As a result, besides the machine learning techniques, some emerging big data techniques can be employed for further improving the 6G network design. For example, the latency will be improved through the use of machine learning and big data to determine the best way to transmit data from the user to the base station, since an intelligent 6G network will be capable of providing predictive analysis. Motivated by the above-mentioned appealing characteristics of machine learning and big data techniques, a flurry of research activities on intelligent wireless communication system designs have been sparked. Specifically, machine learning was considered as a key approaching [1] for realizing 6G fro mauser perspective. In [9] the authors summarized the machine learning techniques for massive MIMO optimization, heterogeneous network design, and device-to-device communications. In [10], a novel mobile network architecture enabling big data analytics was proposed for facilitating physical layer optimizations. It is worth noting in [9–11] that machine learning and big data techniques will not only deliver compelling system performance but also profoundly change the design and configuration of the future 6G networks (e. g., the physical-layer processing and MAC protocol). Furthermore, machine learning and big data analytics are not independent and unrelated in future 6G network design.

As noted in [11], by jointly utilizing these techniques, the mobile networks will become more promising in terms of self-adaptive, self-aware, and predictive ability. Indeed, intelligent 6G network design requires knowledge and methods from multi-disciplinary aspects. To be specific, methods in optimization theory, data mining, computer science, and even life science will be involved. Recently, based on the development of brain-machine interface techniques, mind-controlled machines are gradually being realized [12], which have led some scientists to speculate that mind-to-mind communications may be possible in 6G and beyond (https://www.iflscience.com/). Moreover, a 6G network may also include other communication types, such as molecular communications [13] for future micro- or nanoscale medical applications. In these long-term, forward-looking, and service-oriented visions, the biology and chemistry fields are involved in future communication theory.

5.10 Beam Management :

Even though 3GPP would not preclude the use of Sub 6 GHz deployment of 5G(NR), at least based on the current status it seems that most of the deployment would be in very high frequency (millimeter wave) and this high frequency deployment would be one of the most important characteristics of 5G (NR).

5.10.1 Why we need Beam ?

Mostly by Nature of the wave (by Physics), when we use low and mid range of frequency, we can transmit a signal in all direction (as in (A)) or relatively wide angles (as in (B)). However, when we use very high frequency, we would not have much choice except using a huge antenna array. As a result of using this kind of huge antenna array, the resulting radiation would be a beam as in (C). Refer to Why Massive MIMO page for the details of this background.

5.10.2 Why Beam Management / Beam Control?

I don't think transmitting signal in Beam in high frequency deployment would be the matter of choice. It is a kind of 'MUST' implementation. In case of low / mid frequency region without using massive antenna array (as in (A) / (B)), a single transmission would cover a lot of UEs simultaneously. However, when the radiation become beam-shaped as (C), it is very difficult to cover multiple UEs in single transmission unless those multiple UEs are located in very close proximity. To handle this problem, we need a very sophisticated idea of managing/controlling the beam to cover the multiple devices scattered in all directions and the management/control mechanism should be different depending on the situations. All of these collection of idea would fall into the title of "Beam Management" in the specification.

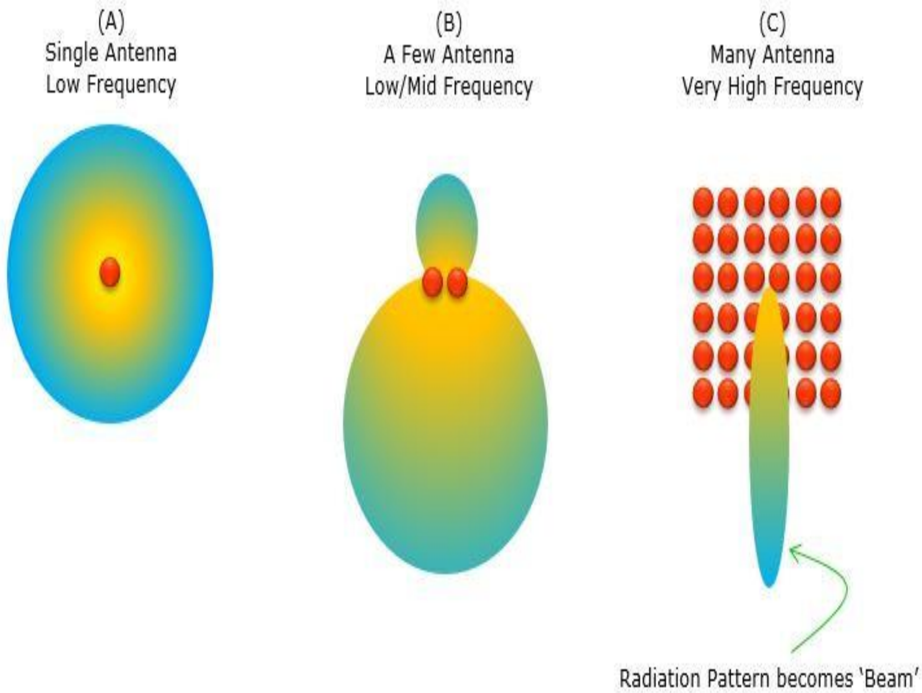


Fig 5.9 : 6G Beam radiation pattern.

5.10.3Beam Management/Control when a transmitter has no information on the location of the receiver

Now let's look into a more specific cases where the Beam Management/Control become crucial. As an example, let's think of following case. There is a Base Station with Massive MIMO operating at the very high frequency. There is a UE around the Base Station and you are just about to turning on the UE. Once the UE is turned on, it would start Synchronization process. For this step, the Base Station would transmit the special signal called Synchronization Signal and the signal should be able to reach to every UEs around the base station. However, here comes a serious problem with the base station sending signal in Beam. It is the fact that the signal beam can point to a very narrow area and it cannot cover a very wide area at the same time. Simply put, now you have the following question.

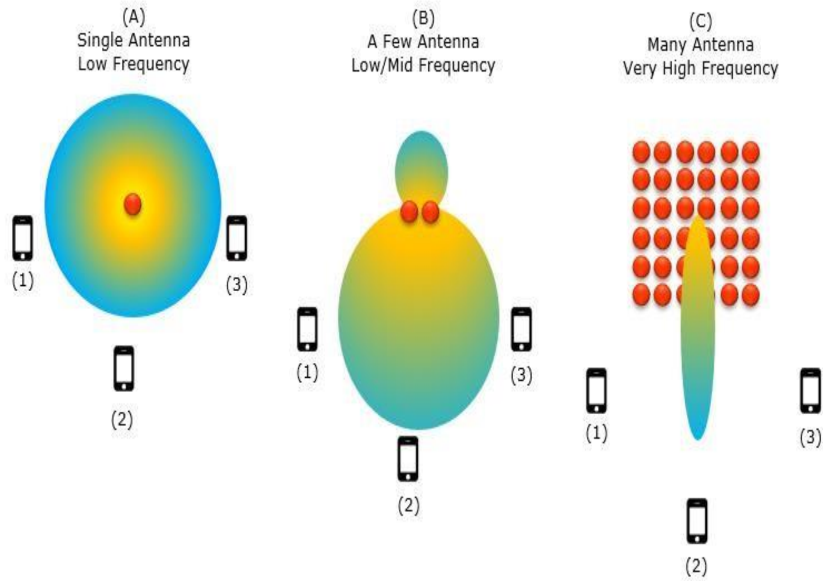


Fig 5.10 : Antenna range of 6G.

Fig 5.11 : Signal sharing.

What would be the answer for this ? If everything works as you draw in power point, you may draw a solution as follows. You may want to transmit a lot of beams in all direction simultaneously. Looks good ? Looks like a flower :).

Would the solution above be feasible, reasonable and effective ? The simple answer is NO (I would not explain why. You may easily guess why).

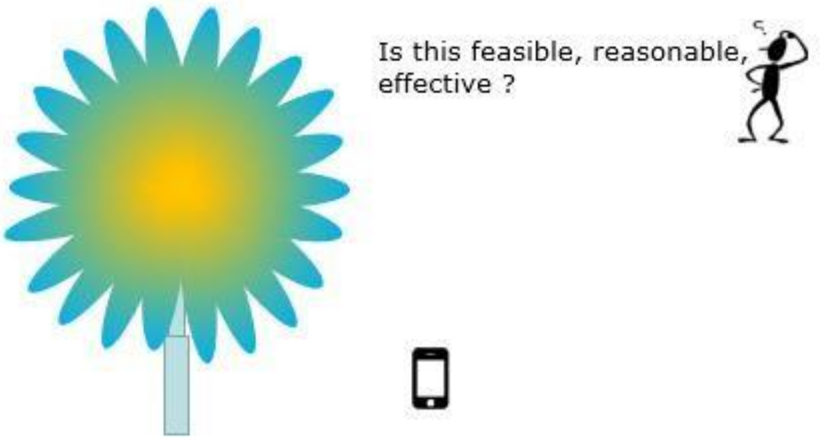


Fig 5.12 : Signal sharing Q/A.

Then what can be another idea (possible solution) for the problem ? There can be multiple ideas and proposals, but the most popular proposal as of now seems to be that the base station transmit the beam to a specific direction at a specific time and then change the direction a little bit in a next time frame and so on until it can scan all the area it should cover.

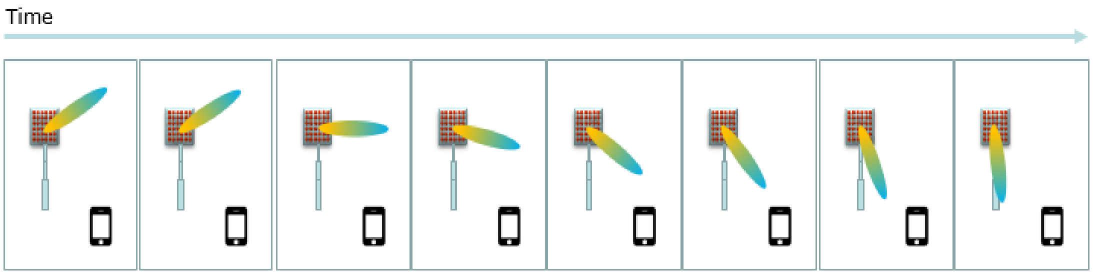


Fig 5.13 : 6G NR initial access procedure.

Then, the next question would be "how to reflect / implement this concept in the radio frame design ?". I would not go too much detail on this until this is explicitely determined in 3GPP TS (Technical Specification) document, but you can get the general ideas on various options /proposals from TDocs listed in Reference Section.

// Now that 3GPP Technical Specification on Beam Management has been released and I could write on this mechanism based on the formal specification. You may jump to NR Beam Management in a Nutshell section and read from there if you are interested in the formal specification.

5.10.4 Beam Management/Control when the connection is already established.

Now let's talk about more serious case of Beam Management. In terms of 3GPP TDocs, Beam Management handles mostly with this topic (Beam Management during the connected states) and the one mentioned in previous section are described as a part of the topic Cell Search / Initial Access.

Once UE gets into a connection states with a Network, at least one beam (or multiple beam) is properly in connection between UE and the network. Theoretically there can be so many different ways in which UE and Network beam is connected, but we can reduce it down to roughly four differences case as shown below.

In case 1, UE and Network is connected through a single TRP (Tx/Rx Point) and a single beam.

In case 2, UE and Network is connected through multiple TRP (Tx/Rx Point) and a single beam for each TRP.

In case 3, UE and Network is connected through a single TRP (Tx/Rx Point) and multiple beams.

In case 4, UE and Network is connected through multiple TRP (Tx/Rx Point) and multiple beams for each TRP.

You may think of many other cases and ask "How about this case ? How about that case ?". Whatever you think of and whatever you are asking, I think all of those would be valid thinking and valid question until 3GPP reach a explicit conclusion. So keep asking and try to find your own answers until you see the explicit 3GPP specification.

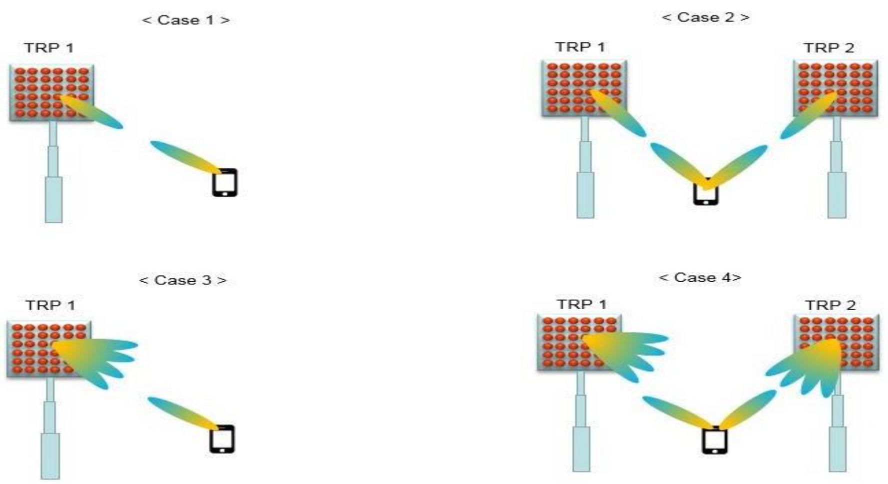


Fig 5.14 : Beam forcing.

Now the important and tough task is to maintain the connection. For this, I would not write much details until 3GPP specifies it in detail. Sorry for skipping too much with the excuse of 3GPP specification unavailability :). But I don't want to write many things now and rewrite too much after 3GPP Technical Specification come out. For now, my purpose is to give you very broad and general idea, and let you know what you may need to study in further details if you are really interested in technical details. For this purpose, I will list up most of TDocs about each topics in Reference section, so that you can get more detailed idea proposed by many companies / organizations in the industry.

The general idea of the beam management during the connected states would be

5.10.3 Network transmit a specific reference signal for beam management

5.10.4 UE detect the signal and perform some measurement and send feedback to Network

As you may notice, the general idea would be very similar to CSI report mechanism that are currently used in current LTE. However, a lot of details are yet to be determined. For example,

i) Baseband Signal (Symbol) generation formula

ii) Resource Allocation mapping (How to allocate these reference symbols to which specific resource element)

iii) How often UE need to perform these measurement

iv) How UE report the measurement result ? (via RRC messages ? or via MAC / PHY layer transactions ?)

NOTE : Now that the technical specification on all of the questions listed above has been released and I wrote a few separate pages regarding this topic. Refer to CSI RSsignal page and CSI Report page for further details based on 3GPP technical specification.

5.10.5 Where to point my beam?

We can think of this question in terms of two different cases : transmitting case and receiving case.

- **When transmitting?**

Now let's think of which direction a gNB or UE has to point its beam when they try to transmit the signal ?

The answer is simple. They (gNB or UE) has to transmit the signal in the direction that can reach the receiver with the best signal quality.

Then you would have another question. How can they(gNB or UE) figure out which direction is the one that can reach the receiver with the best signal quality ?

Now the answer would be a little bit trickier, but the big picture is as follows.

- When gNB is transmitting, gNB figure out this direction by evaluating the quality of a specific reference signal of multiple beam from UE. gNB evaluate the quality of the reference signal from each of the multiple beam and chose the best one. The reference signal from UE is called SRS.
- When UE is transmitting, UE figure out this direction by evaluating the quality of a specific reference signal of multiple beam from gNB. UE evaluate the quality of the reference signal from each of the multiple beam and chose the best one. The reference signal from gNB in this case can vary depending on situation. Sometimes it can be SSB and sometimes it can be CSI-RS. (NOTE : CSI-RS play many different roles in addition to beam management and very complex topic. Refer to CSI-RS signal generation and CSI report page for further details).

  NOTE : This kind of estimation of reference signal quality should be done sometime before they transmit signal.

3GPP TR 38.802 (V14.2.0)-6.1.6.1 describes on this situation as follows :

- TRP is able to determine a TRP Tx beam for the downlink transmission based on TRP's uplink measurement on TRP's one or more Rxbeams
- UE is able to determine a UE Tx beam for the uplink transmission based on UE's downlink measurement on UE's one or more Rxbeams

  NOTE : TRP is a transmission point of a gNB.

- ## When receiving?

Now let's think of which direction a gNB or UE has to point its beam when they try to receive signal ? (In this case, the word 'beam' may be a little misleading because the receiver does not form any real beam. So it would be better to change the phrase 'to point its beam' to 'to tune its receiver to a certain direction').

The answer is simple. They (gNB or UE) has to tune their receiver in the direction in which they can receive the signal from the transmitter with best quality.

Then you would have another question. How can they(gNB or UE) figure out which direction is the one in which they can receive the signal from the transmitter with best quality ?

Overall logic is as follows :

- When gNB receiving signal from UE, (before doing this) gNB is supposed to get the information of the best direction from UE in the form of CSI report.
- When UE receiving signal from gNB, (before doing this) UE is supposed to get the information of the best direction from gNB (gNB has detected the best direction based on the measurement of SRS signal quality of multiple beams from UE and indicates the UE of the best direction).

3GPP TR 38.802 (V14.2.0)-6.1.6.1 describes on this situation as follows :

- TRP is able to determine a TRP Rx beam for the uplink reception based on UE’s downlink measurement on TRP’s one or more Txbeams.
- UE is able to determine a UE Rx beam for the downlink reception based on TRP’s indication based on uplink measurement on UE’s one or more Tx beams.

CHAPTER 6

METHODOLOGY

6.1 System Model:

6.1.1 Block Diagram:

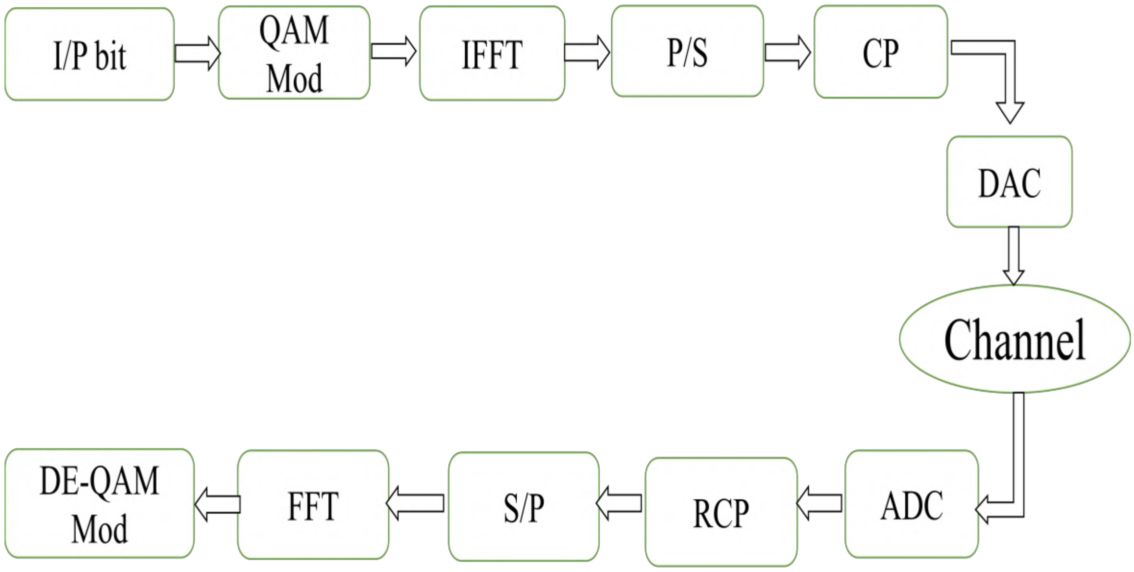


Fig 6.1:Block diagram.

- 6G is the sixth generation standard currently under development for wireless communications technologies supporting cellular data networks.
- 6G is not yet a functioning technology, and is instead in the early research phase. Mobile telecom companies are much too focused on 5G to deal with 6G in any significant way.
- Though researchers can't define how fast 6G will be yet, but estimates to have it around 100 times faster than 5G.

6.1.2 Flow Chart:

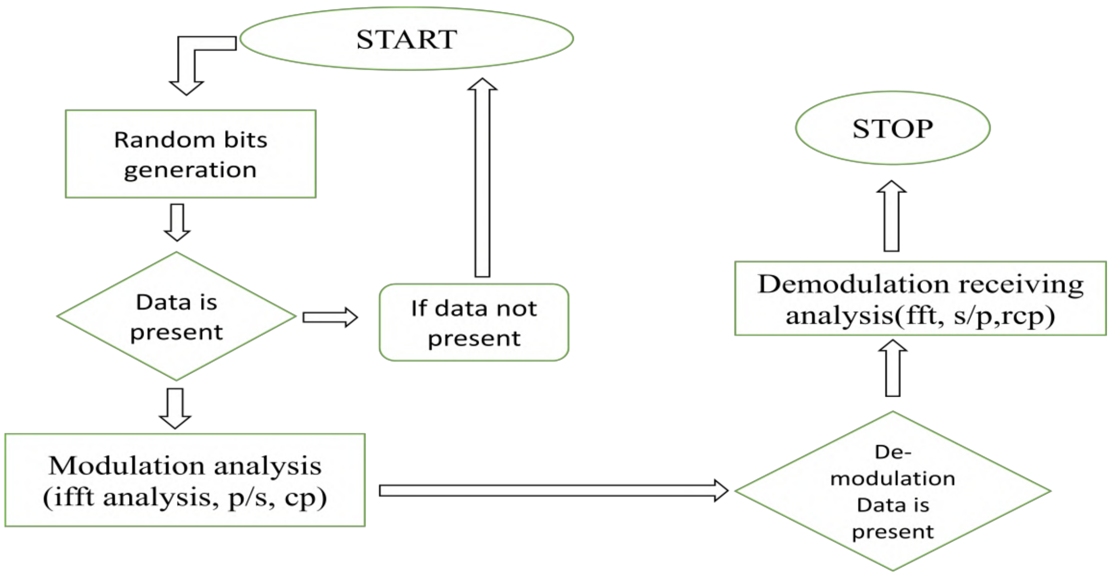


Fig 6.2:Flow chart.

From starting the process the digital signal in the form of input bits undergoes QAM modulation which undergoes changes in phase, amplitude and frequency. Then the output signal passes through IFFT where the bits in the data forms summation of all N sinusoids. IFFT block provides a simple way to modulate data onto N orthogonal subcarriers. The block of N output samples from IFFT make up a single OFDM signal and then it passes to cyclic prefix which enables the OFDM signal to operate reliably. The cyclic prefix acts as a buffer region to protect the OFDM signal from inter symbol interface. Then the signal is converted to analog and is sent into a channel to reduce the SNR and same process is repeated in reverse manner to require digital output for working.

CHAPTER 7
RESULT

7.1 Result:

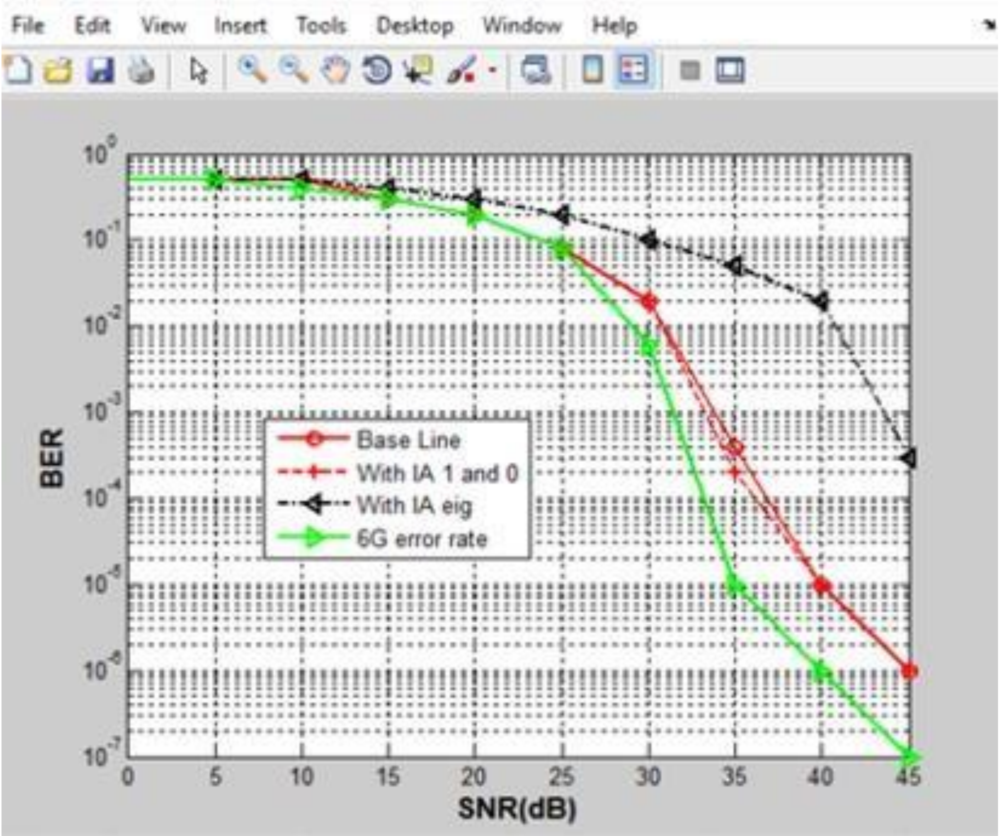


Fig 7.1 : Output 1 using 32:32 SNR algorithm.

Here in our project we obtained our output comparing the latancy rate and BER(Bit Error Rate) of 6G network and other networks.

Output 1 result shows the comparision of 6G and 5G networks BER using 32: 32 algorithm. In the above graph the green line indicates the 6G network and red line indicates the 5G network. The X-axis shows the SNR(Signal to Noise ratio) and Y- axis shows the BER. On comparing the 6G and 5G networks the green line i.e., 6G networks is showing the low BER around 10^-7 which is low compared to the 5G network. Low BER indicates the reduce of SNR and the data transmitted with low latency to the receiver.

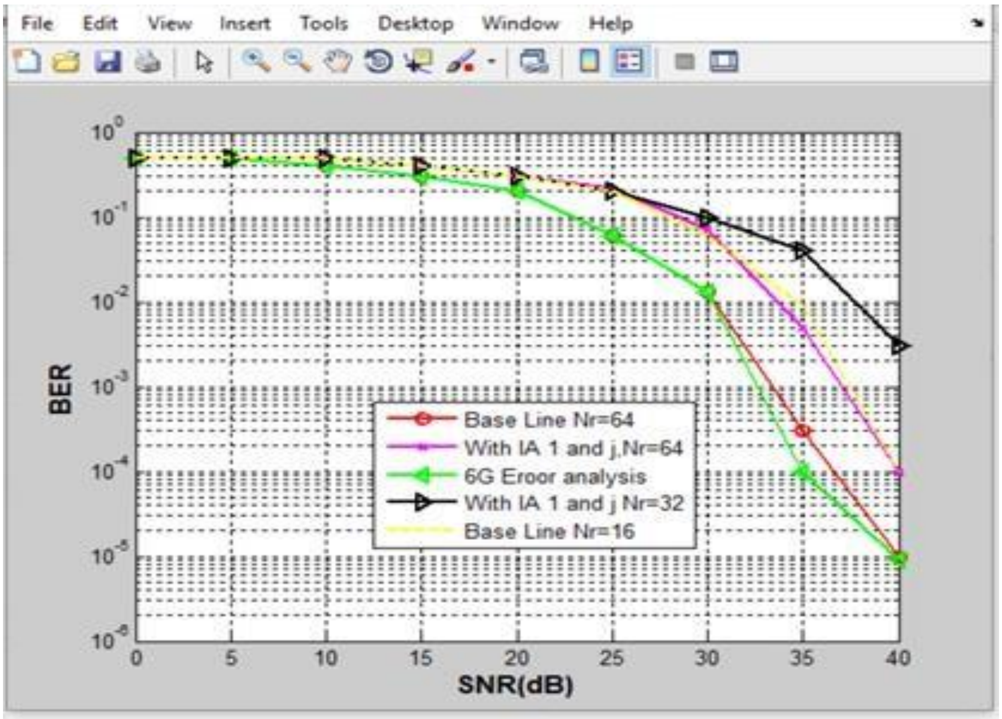


Fig 7.2: Output 2 using 64:64 SNR algorithm.

In this project, we put our proposal into the Vienna Matlab platform and receive the simulation results. Here, we assume that D2D communication reuses the downlink resource for transmission. Otherwise, the collision probability reduces as m increases. Therefore, we need to choose a suitable m that considers both the collision probability and the access probability. At this point, we assume that the arrival of the D2DUE's access request follows the Poisson process. This result will satisfy the need of a practical system performance. Through numerical simulation on the Vienna Matlab platform, we can obtain the average access delay under the premise of a differing number of preambles.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 Conclusion:

We have presented a vision for 6G mobile net - works, which can cater to the growing demands of IoE. We commence with a sketch of 6G from the viewpoint of time, frequency, and space resource utilization. Then we review some promising recent approaches that could move this vision closer to reality. Finally, we focus our attention on some challenges in 6G communication systems, which will hopefully serve as guidelines for their future development. We note that the major features of 6G networks are their flexibility and versatility, and the design of 6G networks is a truly multidisciplinary field of science. We expect that the new research on 6G will also impact the areas of medical imaging, semiconductors, spectroscopy, chemistry, and even biotechnology.

8.2 Future Scope:

In future 6G can be implemented for following:

- How to provide security.
- Virtualization of radio access interface.
- Mobility and localization.
- Rural Connectivity.

REFERENCES

1. Boccardi F, Heath RW, Lozano A, Marzetta TL, Popovski P. Five disruptive technology directions for 5G. IEEE Communications Magazine 2014; 52 (2):74–80.

2. Hu RQ, Qian Y. Resource Management for Heterogeneous Networks in LTE Systems. Springer: New York,2014.

3. Hu RQ, Qian Y. Heterogeneous Cellular Networks. John Wiley and Sons, Ltd.: New Jersey,2014.

4. Hu RQ, Qian Y. An energy efficient and spectrum efficient wireless heterogeneous network framework for 5G systems. IEEE Communications 2014; 52(5):94–101.

5. Ge X, Cheng H, Guizani M, Han T. 5G wireless backhaul networks: challenges and research advances. IEEE Network Magazine 2014; 28(6):6–11.

6. Wei L, Hu RQ, Qian Y, Wu G. Key elements to enable millimeter wave communications for 5G wireless systems. IEEE Wireless Communications 2014; 21(6):136–143.

7. Wu D, Wang J, Cai Y, Guizani M. Millimeter-wave multimedia communications: challenges, methodology, and applications. IEEE Communications Magazine 2015; 53(1):232–238.

8. Asadi A, Wang Q, Mancuso V. A survey on deviceto-device communication in cellular networks. IEEE Communications Surveys & Tutorials 2014; 16 (4): 1801–1819.

9. Gao G, Sheng X, Tang J, Zhang W, Zou S, Guizani M. Joint mode selection, channel allocation and power assignment for green device-to-device communications. In IEEE ICC, Sydney, Australia, 2014;10–14.

10. Wei L, Hu RQ, Qian Y, Wu G. Enabling device-todevice communications underlaying cellular networks: challenges and research aspects. IEEE Communications 2014; 52(6):90–96.

11. Hyunkee M, Jemin L, Sungsoo P, Daesik H. Capacity enhancement using an interference limited area for device-to-device uplink underlaying cellular networks. IEEE Transactions on Wireless Communications 2011; 10(12):3995–4000.

12. Xu C, Song L, Han Z, Zhao Q, Wang X, Cheng X, Jiao B. Efficiency resource allocation for device-to-device underlay communication systems: a reverse iterative combinatorial auction based approach. IEEE Journal on Selected Areas in Communications 2013; 31(9):348–358.

13. Zulhasnine M, Changcheng H, Srinivasan A. Efficient resource allocation for device-to-device communication underlaying LTE network. In 2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications (WIMOB), Niagara Falls, ON, 2010;368–375.

14. Lin Z, Gao Z, Huang L, Chen CY, Chao HC. Hybrid architecture performance analysis for device-to-device communication in 5G cellular network. Mobile Networks and Applications 2015; 31(9):348–358.

15. Shaoyi X, Haiming W, Tao C, Qing H, Tao P. Effective interference cancellation scheme for device-todevice communication underlaying cellular networks. In 2010 IEEE 72nd Vehicular Technology Conference Fall (VTC 2010-Fall), Ottawa, ON, 2010; 1–5.

16. Tao P, Qianxi L, Haiming W, Shaoyi X, Wenbo W. Interference avoidance mechanisms in the hybrid cellular and device-to-device systems. In 2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications, Tokyo, 2009; 617–621.

17. 3GPP. Tr 22.803 v12.01.0, 2013.

18. Ying P, Qiubin G, Shaohui S, Zheng YX. Discovery of device-device proximity: physical layer design for D2D discovery. In 2013 IEEE/CIC International Conference on Communications in China - Workshops (CIC/ICCC), Xi'an, 2013;176–181.

19. Zhu-Jun Y, Jie-Cheng H, Chun-Ting C, Hung-Yun H, Chin-Wei H, Ping-Cheng Y, Hsu CCA. Peer discovery for device-to-device (D2D) communication in LTE-A networks. In 2013 IEEE Globecom Workshops (GC Wkshps), Atlanta, GA, 2013; 665–670.

20. Simsek M, Merwaday A, Correal N, Guvenc I. Device-to-device discovery based on 3GPP system level simulations. In 2013 IEEE Globecom Workshops (GC Wkshps), Atlanta, GA, 2013;555–560.

21. Hong J, Park S, Kim H, Choi S, Lee KB. Analysis of device-to-device discovery and link setup in LTE networks. In 2013 IEEE 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), London, United Kingdom, 2013; 2856–2860.

22. Huang L, Su Z, Gao Z, Lin Z, Hu T, Liwang M. Oop-based device-to-device communication simulator design of LTE network. In 2013 IEEE 24th International Symposium on Personal Indoor andMobile.

APPENDICES

```
clear all;
close all;
clc;
%% input Parameters
    ofdm.Nb      = 10^2;
    ofdm.Nt      = 2;
    ofdm.Nr      = 4;
    ofdm.K       = 128;
    ofdm.G       = 1/4;
    ofdm.Mod     = 4;
    ofdm.PSpace  = 1;
    chan.SNR_dB  = 15;
    chan.L       = 6;

% control parameters
    ofdm.ifDemodulateData = 1;
    ofdm.ifDisplayResults = 1;
    M=0.5;i=1;
    n=0.01;beta_k=0.2;
    a=0.02;r=10;

if nargin > 2
%         error('Only two set as inputs thresholding')
elseif nargin == 2
% updating the set parameters
 freq_coeff = fieldnames(ofdmIn);
for nS = 1:length(freq_coeff)
ofdm.(freq_coeff{nS}) = ofdmIn.(freq_coeff{nS});
end
freq_coeff = fieldnames(chanIn);
for nS = 1:length(freq_coeff)
chan.(freq_coeff{nS}) = chanIn.(freq_coeff{nS});
end
elseif nargin == 1
freq_coeff = fieldnames(ofdmIn);
for nS = 1:length(freq_coeff)
ofdm.(freq_coeff{nS}) = ofdmIn.(freq_coeff{nS});
end
end
% dependent parameters
    ofdm.PPos     = 1:(ofdm.PSpace+1):ofdm.K;
    ofdm.PL       = length(ofdm.PPos);
    ofdm.DPos     = setxor(1:ofdm.K,ofdm.PPos);
    ofdm.DL       = length(ofdm.DPos);
    ofdm.BER      = 0;
    chan.sigma    = sqrt(10^(-0.1*chan.SNR_dB));
     y_j=sum(a*n*i*n*i+n*i);        %%%%%%%%%1

%%% QAM modulation
        qam_process          = 0:ofdm.Mod-1;
```

```
qam_process          = qammod(qam_process,ofdm.Mod);
qam_process          = abs(qam_process).^2;
qam_process          = mean(qam_process);
ofdm.ModNorm = 1/sqrt(qam_process);

%% sixth generation(6G) mobile network is expected
generation methodology
% symbol generation
ofdm.d = randi(ofdm.Mod,ofdm.DL,ofdm.Nb,ofdm.Nt)-1;
    k_j=a*beta_k-1;     %%%%%%%3
%% data Modulation
    ofdm.dMod   = zeros(ofdm.K,ofdm.Nb,ofdm.Nt);
if ofdm.DL > 0
for nt = 1 : ofdm.Nt
  ofdm.dMod(ofdm.DPos,:,nt) =
    ofdm.ModNorm*qammod(ofdm.d(:,:,nt),ofdm.Mod);
end
end
for nt = 1 : ofdm.Nt
ofdm.dMod(ofdm.PPos,:,nt) = repmat(exp(-sqrt(-
1)*2*pi*(nt-1)*chan.L*(1:ofdm.PL).'/ofdm.PL),1,ofdm.Nb);
end
% checking the power of the transmit signal (it has to
be 1 after normalization)
ofdm.pow = var(ofdm.dMod(:))+abs(mean(ofdm.dMod(:)))^2;
%% IFFT operation

ofdm.ifft   = zeros(ofdm.K,ofdm.Nb,ofdm.Nt);
for nt = 1 : ofdm.Nt
ofdm.ifft(:,:,nt) =
     sqrt(ofdm.K)*ifft(ofdm.dMod(:,:,nt),ofdm.K);
end
%% parallel to serial communication
ofdm.ifftG = [ofdm.ifft(ofdm.K*(1-
            ofdm.G)+1:ofdm.K,:,:);ofdm.ifft];

%% Add CP
   chan.Coeff =
1/sqrt(2)*1/sqrt(chan.L)*(randn(ofdm.Nt,ofdm.Nr,chan.L,o
fdm.Nb)+sqrt(-1)*randn(ofdm.Nt,ofdm.Nr,chan.L,ofdm.Nb));
   h=i*a;          %%%%%%%%2
%% Channel  filter with DAC
if ofdm.K*ofdm.G < chan.L+1
            error('differentiate input parameters')
end
    ofdm.Y = zeros(ofdm.K*(1+ofdm.G),ofdm.Nb,ofdm.Nr);
for nb = 1 : ofdm.Nb
for nt=1:ofdm.Nt
for nr=1:ofdm.Nr
ofdm.Y(:,nb,nr) = ofdm.Y(:,nb,nr) +
filter(squeeze(chan.Coeff(nt,nr,:,nb)),1,ofdm.ifftG(:,nb
,nt));
```

```
end
end
end
%% ADC
ofdm.Y = ofdm.Y + chan.sigma*1/sqrt(2)*(
randn(ofdm.K*(1+ofdm.G),ofdm.Nb,ofdm.Nr)+...
     sqrt(-1)*randn(ofdm.K*(1+ofdm.G),ofdm.Nb,ofdm.Nr)
);
%% Cyclic prefix removal
ofdm.fftG= fdm.Y(ofdm.K*ofdm.G+1:ofdm.K*(1+ofdm.G),:,:);
    s=h*a*i*n;      %%%%%%%%5
%% FFT operation
    ofdm.fft  = zeros(ofdm.K,ofdm.Nb,ofdm.Nr);
for nr = 1 : ofdm.Nr
ofdm.fft(:,:,nr)  =
1/sqrt(ofdm.K)*fft(ofdm.fftG(:,:,nr),ofdm.K);
end
%%  serial to parallel communication
  rcvd_sort_sgnl = dftmtx(ofdm.K);
  rcvd_sort_sgnl = rcvd_sort_sgnl(:,1:chan.L);
  chan.CoeffEst = zeros(ofdm.Nt,ofdm.Nr,chan.L,ofdm.Nb);

for nb = 1 : ofdm.Nb
for nr = 1 : ofdm.Nr
 chan.A = zeros(ofdm.PL,chan.L*ofdm.Nt);
for nt = 1 : ofdm.Nt
chan.A(:,(1:chan.L)+(nt-1)*chan.L) =
diag(ofdm.dMod(ofdm.PPos,nb,nt))*rcvd_sort_sgnl(ofdm.PPo
s,:);
end
ChanEst = pinv(chan.A)*ofdm.fft(ofdm.PPos,nb,nr);
for nt = 1 : ofdm.Nt
chan.CoeffEst(nt,nr,:,nb) = ChanEst((1:chan.L)+(nt-
1)*chan.L);
end
end
end
chan.MSE_Simulation = var(chan.Coeff(:)-
chan.CoeffEst(:));
chan.MSE_Theory     = chan.sigma^2/ofdm.PL;
if ofdm.ifDisplayResults
        disp(['MSE differentiate :
',num2str(chan.MSE_Theory)])
        disp(['MSE differentiate :
',num2str(chan.MSE_Simulation)])
end
%% Demodulation
if ofdm.ifDemodulateData == 1 && ofdm.DL > 0
chan.CoeffEstFreq=zeros(ofdm.K,ofdm.Nt,ofdm.Nr,ofdm.Nb);
for nb = 1 : ofdm.Nb
for nr = 1 : ofdm.Nr
for nt = 1 : ofdm.Nt
```

```
chan.CoeffEstFreq(:,nt,nr,nb) =
rcvd_sort_sgnl*squeeze(chan.CoeffEst(nt,nr,:,nb));
end
end
end
% demodulation
 ofdm.dDemod = zeros(ofdm.DL,ofdm.Nb,ofdm.Nt);
for nb = 1 : ofdm.Nb
for dl = 1 : ofdm.DL
ofdm.dDemod(dl,nb,:) =
pinv(reshape(chan.CoeffEstFreq(ofdm.DPos(dl),:,:,nb),ofd
m.Nt,ofdm.Nr).')...

*squeeze(ofdm.fft(ofdm.DPos(dl),nb,:));
end
end
% QAM de-modulation
ofdm.dEst = zeros(ofdm.DL,ofdm.Nb,ofdm.Nt);
for nt = 1 : ofdm.Nt
 ofdm.dEst(:,:,nt) = qamdemod(1/ofdm.ModNorm *
ofdm.dDemod(:,:,nt),ofdm.Mod);
end
```

A

PROJECT REPORT

On

WOMEN SAFETY PATROLLING ROBOT USING IOT

Submitted BY

1) Ms. J. Sangeetha (17K81A04E8)

2) Ms. E. Nagalakshmi (17K81A04E3)

3) Mr. BK. Yash Kumar (17K81A04D1)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN
ELECTRONICS & COMMUNICATION ENGINEERING

Under the Guidance of

Mr. M. Suresh Kumar
M.Tech. (Ph.D), Assistant Professor

DEPARTMENT OF

ELECTRONICS & COMMUNICATION ENGINEERING

St. MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad - 500 100

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the project entitled WOMEN SAFETY PATROLLING ROBOT USING IOT, is being submitted by1.**MS.J.Sangeetha(17K81A04E8), 2.MS. E.Nagalakshmi(17K81A04E3), 3 Mr. BK.Yashkumar(17K81A04D1)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN** ELECTRONICS AND COMMUNICATION ENGINEERING is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Mr.M.Sureshkumar	Head of the Department
Department of ECE	Dr.B.HariKrishna
	Department of ECE

Internal Examiner External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **E.NAGALAKSHMI** WITH ROLL NO.**17K81A04E3, J.SANGEETHA** WITH ROLL NO.**17K81A04E8**, **YASH KUMAR** WITH ROLL NO.**17K81A04D1,** OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**WOMEN SAFETY PATROLLING ROBOT USING IOT**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014 Email : contact@lasyainfotech.com, ov@lasyainfotech.com

Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of Bachelor of Technology in Department of ‘ELECTRONICS AND COMMUNICATION ENGINEERING’, session: 2017 – 2021, St. Martin’s Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled ‘WOMEN SAFETY PATROLLING ROBOT USING IOT is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

J.Sangeetha(17K81A04E8)

E.Nagalakshmi(17K81A04E3)

BK.Yashkumar(17K81A04D1)

ACKNOWLEDGEMENT

While bringing out this project report to its final form, we came across a number of people whose contributions in various ways helped my field of research and they deserve special thanks. It is a pleasure to convey our gratitude to all of them.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna**, **Head & Professor**, the Dept. of Electronics and communication Engineering for his/her advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our Dept .project coordinator Dr.A.ANAND , Professor** for his/her involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide Mr.M.SureshKumar Assistant professor,Department of ECE and our class project coordinator Mr.M.Nishakar Assiatant professor,Department of ECE for guidance and encouragement in carrying out this project work

Finally, We are highly grateful and obliged to **our college level Project Coordinator Dr. Maalyadri, Professor in IT and Dr. Santosh Kumar, Assistant Professor in CSE,** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

1. J.SANGEETHA
2. E.NAGALAKSHMI
3. BK.YASH KUMAR

ABSTRACT

Nowadays Women Safety is the biggest concern in many parts of the world. There is still a fear in alone areas for women as well as men. So here we propose a security patrolling robot using Arduino. The robotic vehicle moves at particular path and is equipped with sound sensors and camera which continuously sends live streaming of robot premises to server. It uses a predefined line to follow its path while patrolling. It stops at particular points and moves to next points if sound is detected. It monitors each area to detect any problems. It has the ability to monitor sound in the premises. Robot hears any sound after area is quite and it starts moving towards the sound on its predefined path. Thus we put forward a fully autonomous security robot that operates tirelessly and patrols large areas on its own to secure the facility.

TABLE OF CONTENTS

LIST OF FIGURES

TABLE OF TABLES

CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION ABOUT EMBEDDED SYSTEMS:

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.
The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes.

The applications software on such processors is sometimes referred to as firmware.
The simplest devices consist of a single microprocessor (often called a "chip”), which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine.

As the embedded system is the combination of both software and hardware

1.1.1 Block Diagram of Embedded System:

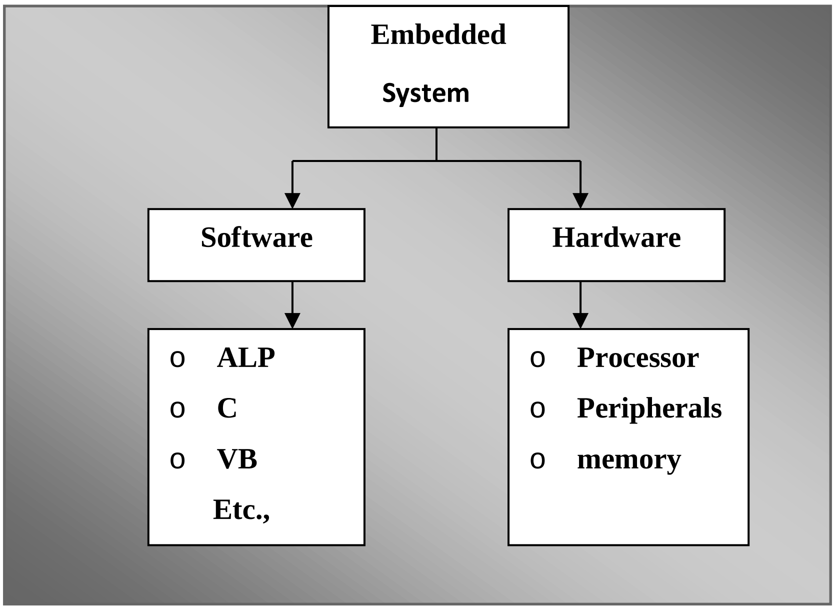


Fig.1.1: Block Diagram of Embedded System

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory

1.1.2 Applications of Embedded Systems:

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

1.2 MICRO PROCESSOR (µP):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Processors are classified into four types like:

- Micro Processor (µp)
- Micro controller (µc)
- Digital Signal Processo1r (DSP)
- Application Specific Integrated Circuits (ASIC)

1.2.1 Three Basic Elements of A Microprocessor:

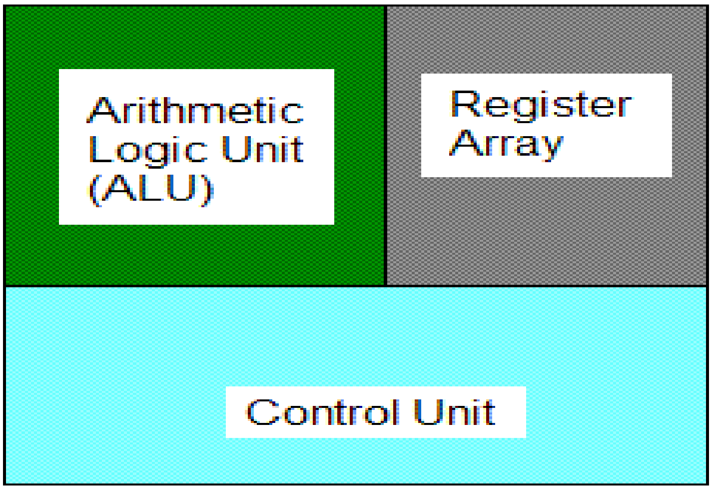


Fig.1.2: Three Basic Elements of a Microprocessor

1.3 HARVARD ARCHITECTURE:

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data.

The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

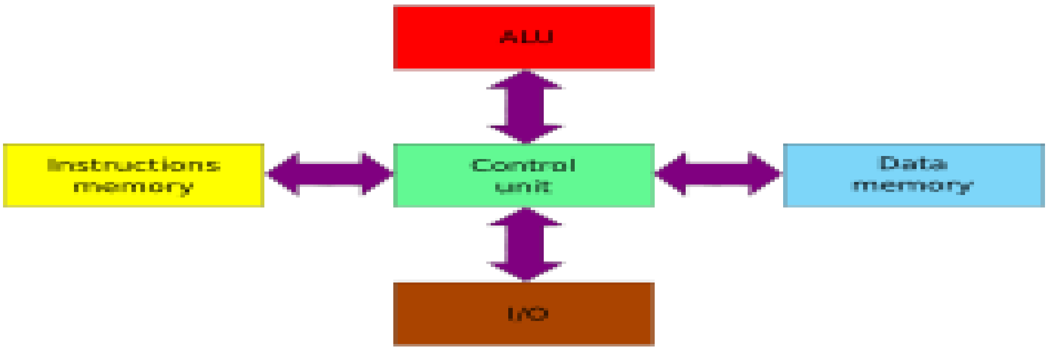


Fig.1.3: Harvard Architecture

1.3.1 Uses of The Harvard Architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

➢ Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behaviour must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.

- Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access.
- The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

1.4 VON-NEUMANN ARCHITECTURE

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. They cannot be performed at the same time. The von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A stored-program digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

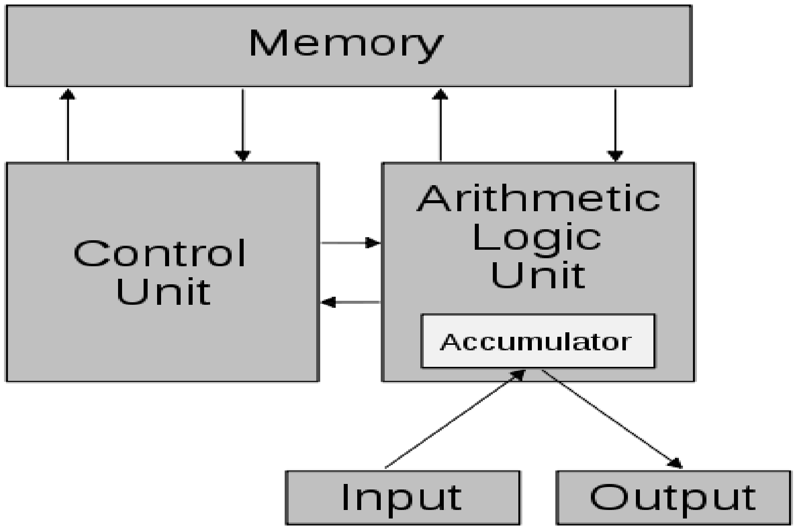


Fig.1.4: Schematic of the Von-Neumann Architecture

1.5 PROJECT DESCRIPTION

1.5.1 Objective:

The main objective of the project is to design and implement women safety patrolling robot using Arduino Uno. The system uses ESP32 camera-Night Vision and Sound Sensors mounted on robotic vehicle for securing women in any premises. The robotic vehicle moves at particular path and is equipped with ESP32camera and sound sensors. It uses a predefined line to follow its path while patrolling. It stops at particular points and moves to next points if sound is detected. The ESP32 Camera always set to ON to monitor premises even during night, works similar to CCTV cameras. Robot hears the sounds after area is quite and it starts moving towards the sound on its predefined path. Camera continuously sends the live video of surroundings to the to the IOT website. Here we use EMBEDDED SPOT website in which a person can provide Username and password and observes the live streaming of premises where the robot present.

Thus we put forward a fully autonomous women Safety robot that operates tirelessly and patrols large areas on its own.

1.5.2 Aim:

We are all aware of importance of women's safety, but we must realize that they should be properly protected. Women are not as physically strong as men, in an emergency situation a helping hand would be a relief for them. The best way to minimize your chances of becoming a victim of violent crime (robbery, sexual assault, rape, domestic violence) is to identify and call on resources to help them out of dangerous situations. Having these applications can reduce the risk and bring assistance when needed.

The main aim of this project to design a robot that is designed solely to serve the purpose of providing security and safety to women so that they never feel helpless while facing such social challenges

1.5.3 Components Used:

- Sound Sensor-2
- ESP-01 Wi-Fi Module
- LCD
- Power Supply
- L293D
- Motors
- ESP Camera
- Arduino
- Robot chassis

CHAPTER-2

LITERATURE SURVEY AND EXISTING METHODS

2.1 LITERATURE SURVEY

JGhanem Osman Elhaj Abdalla, finished a recognition framework with a covert authority robot with the raspberry Pi utilizing web protocol. It gives different thoughts concerning the observation of outskirts areas. Results from the camera are sent to the clients through the web server which can be posted on the website page simultaneously.

In 2013, Cheng Tang, Qunqun Xie, Guolai Jiang, Yongsheng Ou, make a street night subject to a planar reflection model. Till now, various structures are normal for daytime rehearses yet for night there is no such sort of research. This progress puts the any unused movement zone around evening. Here a planar reflection model is sensible to get the force transport of various pixels with an infrared camera. With that, a pixel-based arrangement is utilized to check the various pixels have a spot with the street or not.

In 2017, Kirk Mac Tavish , Michael Paton, and Timothy D. Barfoot , made night rider: visual odometry utilizing headlights.

2.2 EXISTING METHODS

In the existing system, the robot has to be controlled remotely where the person has to monitor the area using the remote control and more human resource is needed for this operation of the patrol robot. Some systems require a smartphone for controlling the night vision robot. Lacking few parts, autonomous operation requires a workforce to operate. Here In these project, we upgrade the robot from remotely controlled to autonomous and even used night vision camera to continuously monitor the robot surroundings through video which is sent to localhost network, therefore, human resource is reduced.

2.3 PROPOSED SYSTEM

In this proposed system, Arduino is installed with the night vision camera and sound sensors which help the system to go for the automation. Help to find the human or any problem detected using the sound sensor and according to the sound produced, it automatically goes to that area and it continuously send the live streaming of the robot premises to IOT server over the internet, In this project we use 2 sound sensors for detecting the sound from left and right, here we use the “HELP” command to be the sound to be detected whenever sound sensor-1 detects the sound the output goes low and moves to left side and similarly with right sound sensor, we use night vision camera’s so that, a person can monitor the premises of robot at any time ,therefore patrolling can be done even in night.

By simply entering the IP address of ESP32-CAM in any IoT web server any person can monitor the place and take actions accordingly to help the Women.

CHAPTER-3

BLOCK DIAGRAM

3.1 BLOCK DIAGRAM

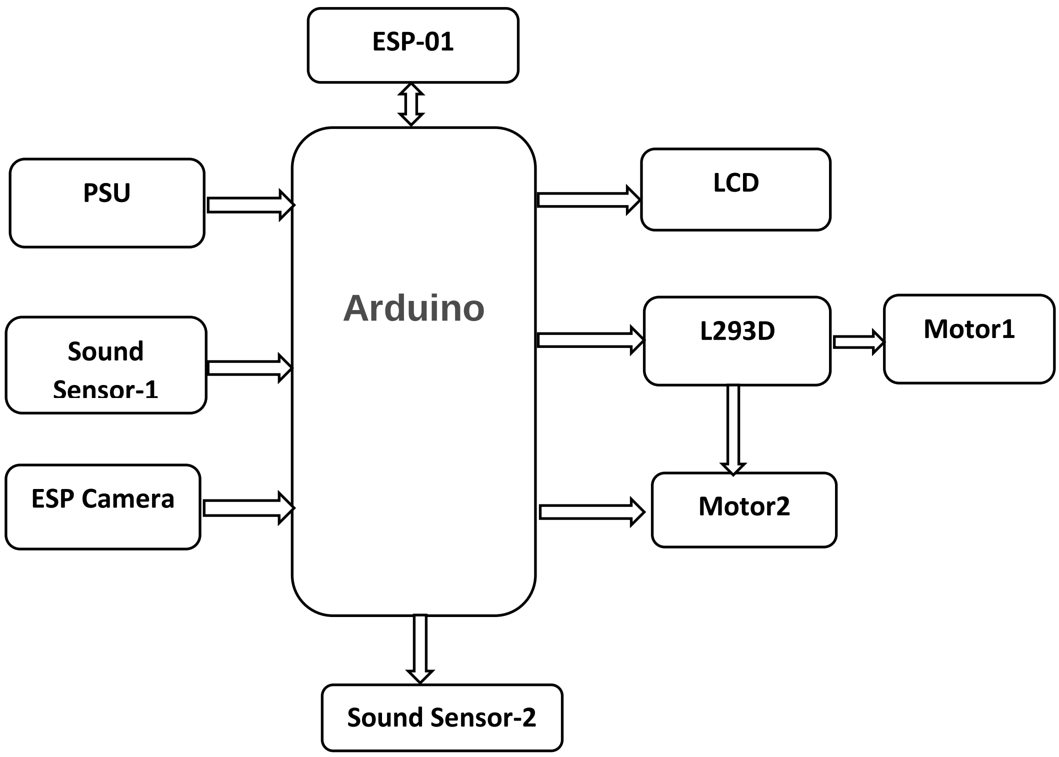


Fig.3.1: Block Diagram of The Project

The block diagram of women Safety patrolling Robot Using IOT is shown in above figure. The blocks of the circuit are

- Arduino UNO
- ESP-01(Wi-Fi Module)
- ESP Camera
- Sound Sensor -1
- Sound Sensor -2
- LCD
- L293D Motor Driver
- Motor-1
- Motor-2
- Power Supply Unit

3.2 SCHEMATIC EXPLANATION

3.2.1 Interfacing Between LCD and Arduino

The following circuit diagram shoes the liquid crystal display with Arduino module. From the circuit diagram, we can observe that the RS pin of the LCD is connected to the pin 7 of the Arduino. The LCD of R/W pin is connected to the ground. The pin 6 of the Arduino is connected to the enable signal pin of LCD module. The LCD module & Arduino module are interfaced with the 4-bit mode in this project. Hence there are four input lines which are DB4 to DB7 of the LCD. This process very simple, it requires fewer connection cables and also we can utilize the most potential of the LCD module.

The digital input lines (DB4-DB7) are interfaced with the Arduino pins from 2-5. To adjust the contrast of the display here we are using a 10K potentiometer. The current through the back LED light is from the 560-ohm resistor. The external power jack is provided by the board to the Arduino. Using the PC through the USB port the Arduino can power. Some parts of the circuit can require the +5V power supply it is taken from the 5V source on the Arduino board.

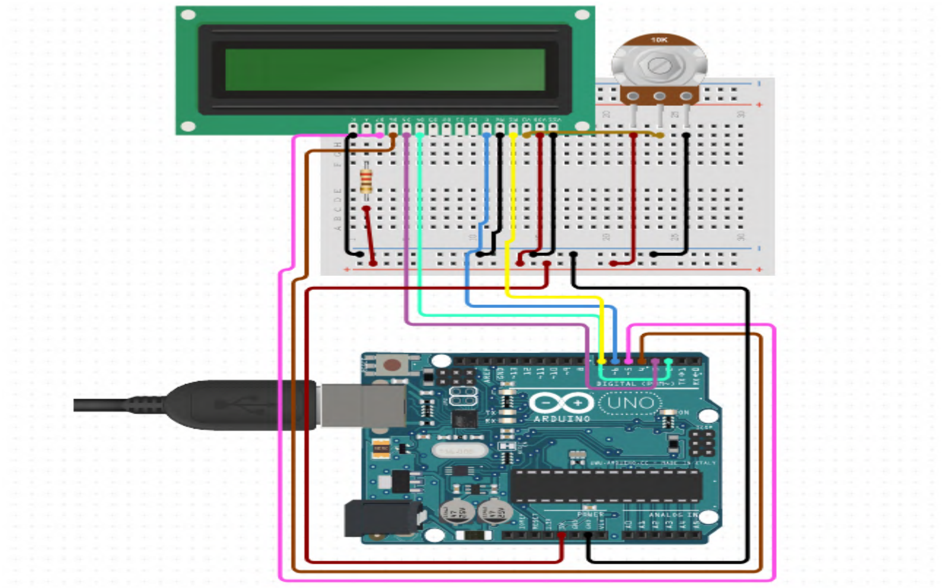


Fig.3.2: Interfacing Between LCD and Arduino

3.2.2 Interfacing Between Soundsensors and Arduino

The following figure shows the interfacing of sound sensors with Arduino, The Sound Sensor used in this project is shown in the image below .It consists of a microphone, a voltage comparator IC (LM393), a potentiometer, a transistor, couple of LEDS and a few other passive components (resistors and capacitors).

The VCC pin sound sensor is connected to 3.3v or 5v and GND pin of sensor to ground pin, the sound sensor-1 output is connected to Analog pin of A4 and sound sensor-2 output pin is connected to Analog pin of A5,when
the sound sensor will produce a logic LOW when sound is detected and a logic HIGH when there is no sound.

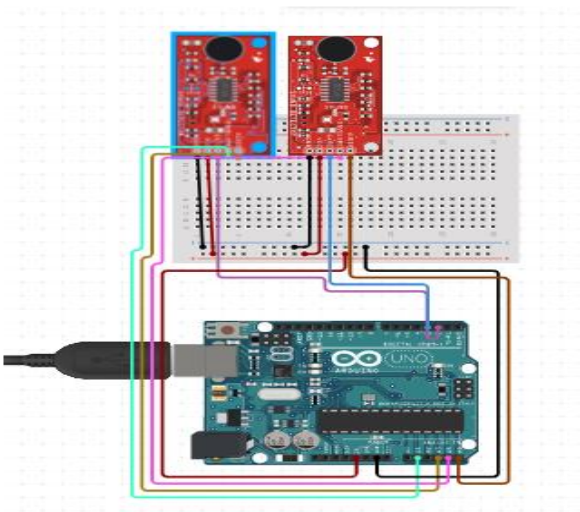


Fig.3.3: Interfacing Between Sound Sensors and Arduino

3.2.3 Interfacing between Camera and Wi-Fi Module

The UOT pin of camera is connected to receiver of Wi-Fi module, and UOR pin of ESP32-cam is connected with transmitter of Wi-Fi.

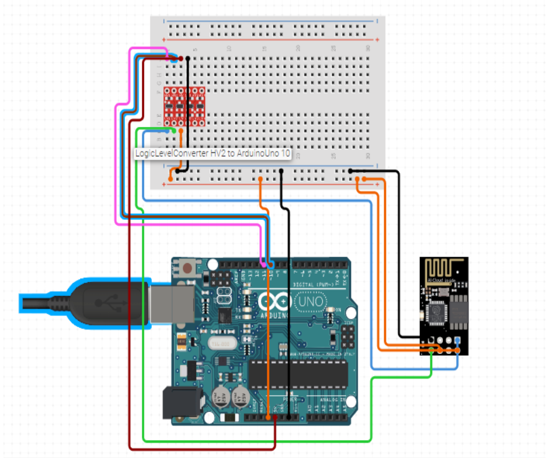


Fig.3.4: Interfacing Between Wi-Fi Module And Arduino

Fig.3.5: Wi-Fi Module

3.2.4 Interfacing between Arduino and L293D

4-7 numbered digital pins act as output pins from Arduino are connected to the motor driver module i.e L293D as input and output of the module is connected to motors for forward or backward movements,

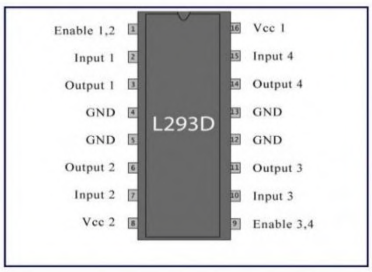


Fig.3.6: L293D Motor Driver

All the Ground pins should be grounded. There are two power pins for this IC, one is the Vss (Vcc1) which provides the voltage for the IC to work, this must be connected to +5V. The other is Vs (Vcc2) which provides voltage for the motors to run, based on the specification of your motor you can connect this pin to anywhere between 4.5V to 36V, here I have connected to +12V.

From the figure Enable pins (Enable 1, 2 and Enable 3, 4) are used to Enable Input pins for Motor 1 and Motor 2 respectively. Since in most cases we will be using both the motors both the pins are held high by default by connecting to +5V supply. The input pins Input 1, 2 which are digital pins 4,5 from Arduino are used to control the motor 1 and Input pins 3, 4 which are digital pins 6,7 from Arduino are used to control the Motor 2. The input pins are connected to the any Digital circuit or microcontroller to control the speed and direction of the motor. You can toggle the input pins based on the following table to control your motor.

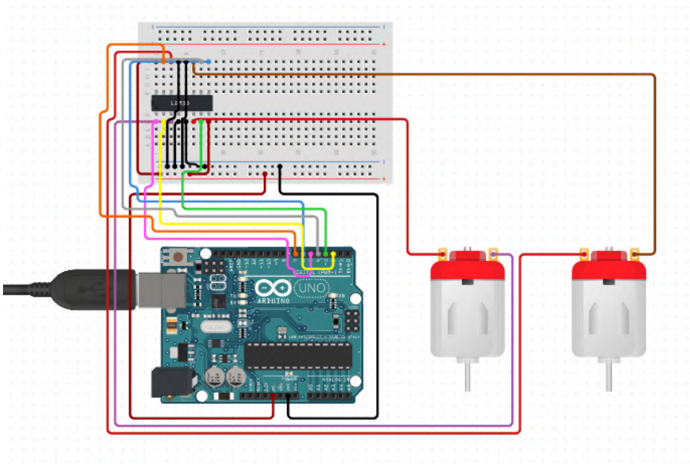


Fig.3.7: Interfacing Between Arduino and L293D

3.2.5 Interfacing between Components

we used 8-13 numbered digital pins of Arduino to connect to LCD to display acknowledgment messages, 4-7 numbered digital pins act as output pins from Arduino are connected to the motor driver module i.e L293D as input and output of the module is connected to motors for forward or backward movements, A4 and A5 Analog pins of Arduino act as input and is connected to two Sound Sensors which takes sound as Input and then we apply Digital Read Function to that input then ESP-cam is connected with Wi-Fi module is ready to stream .

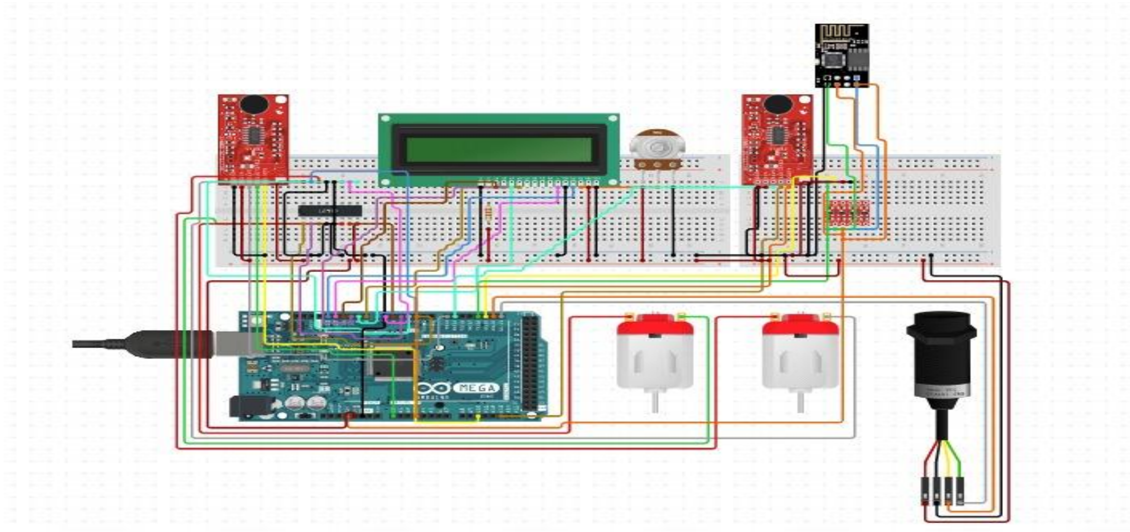


Fig.3.8: Schematic Diagram of All Components

3.2.6 WORKING:

we used Arduino UNO installed with ESP32 camera and Sound Sensors, we used 8-13 numbered digital pins of Arduino to connect to LCD to display acknowledgment messages, 4-7 numbered digital pins act as output pins from Arduino are connected to the motor driver module i.e L293D as input and output of the module is connected to motors for forward or backward movements,

A4 and A5 Analog pins of Arduino act as input and is connected to two Sound Sensors which takes sound as Input and then we apply digital Read Function to that input.

when we supply power to the robot initialization of all the inputs was done and displays "Welcome" message in LCD, ESP32 Camera starts to send live streaming of robot premises over the internet to the IOT web server by using ESP-01 Wi-Fi Module.

It uses a predefined line to follow its path while patrolling. It stops at particular points and moves to the next points if a specific sound command is detected, here we use the loud sounds for the robot to respond. if the surrounding is quiet it displays a message "Everything Is Fine" on LCD,

If the sound is detected the digital read function in the program reads input as High, displays a message "LOCATION 1:ALERT" if sound sensor-1 receives the sound and moves to left and similar with the right sound sensor

While it moves accordingly with the sound it also continuously sends the live video of the surroundings to the IOT web server, this robot still works fine at night as we use a night vision camera therefore patrolling can be done even at night.

Live videos are monitored by a human in the IOT server by entering the login details, any person who is in the local range can monitor the videos.

CHAPTER-4

HARDWARE AND SOFTWARE COMPONENTS

MICRO CONTROLLER UNIT

4.1 ATMEGA328

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 Analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The ATmega8 microcontroller contains 32 general purpose working registers. As shown in the below figure these registers are directly connected to ALU. Two registers can carry one single instruction consequently in one clock cycle.

Specifications:

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 Ma
DC Current on 3.3V Pin	50 Ma
Flash Memory	32 KB (0.5 KB is used for Boot loader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

Table.4.1: Atmega328 Specifications

4.1.1 Arduino:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

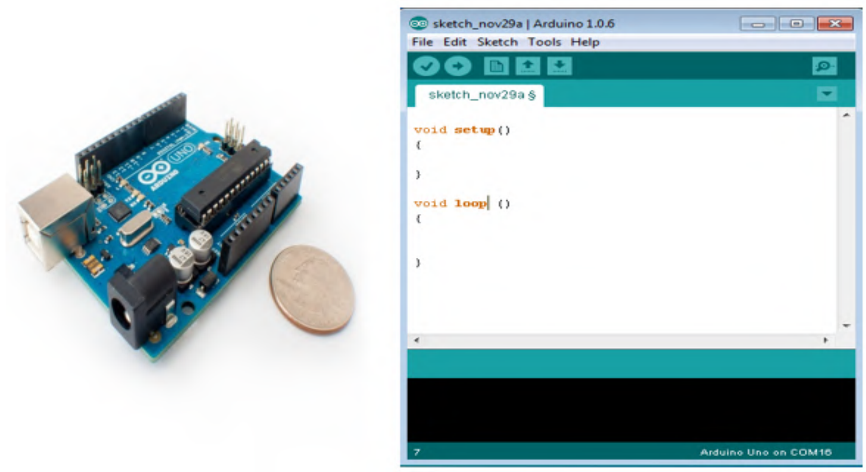


Fig.4.1: Arduino Uno

4.1.2 Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

ATMEGA328:

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header

Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compati

LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table.4.2: Arduino Boards Based On ATMEGA328 Microcontroller

ATMEGA32u4

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table.4.3: Arduino Boards Based On ATMEGA32u4 Microcontroller

ATMEGA2560

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16 U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table.4.4: Arduino Boards Based On ATMEGA2560 Microcontroller

AT91SAM3X8E

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table.4.5: Arduino Boards Based On AT91SAM3X8E Microcontroller

4.1.3 Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

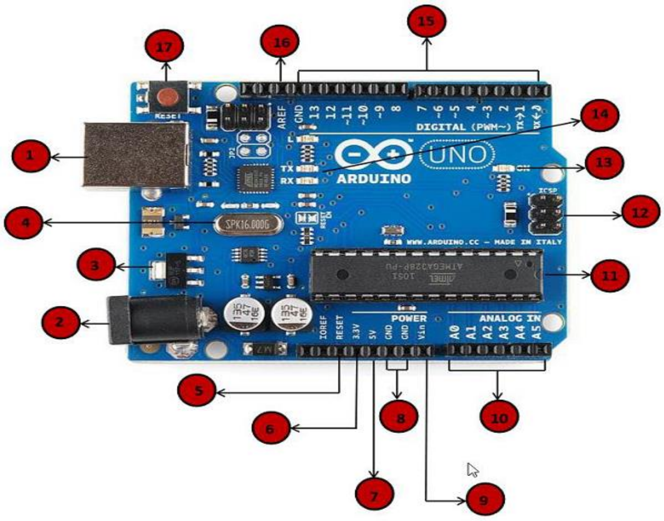


Fig.4.2: Arduino Uno Board Pins

Pin number and Pin Use

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

<table>
<tr><td>4</td><td>Crystal Oscillator
The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.</td></tr>
<tr><td>5, 17</td><td>Arduino Reset
You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).</td></tr>
<tr><td>6, 7, 8, 9</td><td>Pins (3.3, 5, GND, Vin)
3.3V (6) − Supply 3.3 output volt
5V (7) − Supply 5 output volt
Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.
GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit.
Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.</td></tr>
<tr><td>10</td><td>Analog pins
The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.</td></tr>
<tr><td>11</td><td>Main microcontroller
Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC.</td></tr>
</table>

	For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Table.4.6: Pin Explanation of Arduino

4.1.4 Arduino Family

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

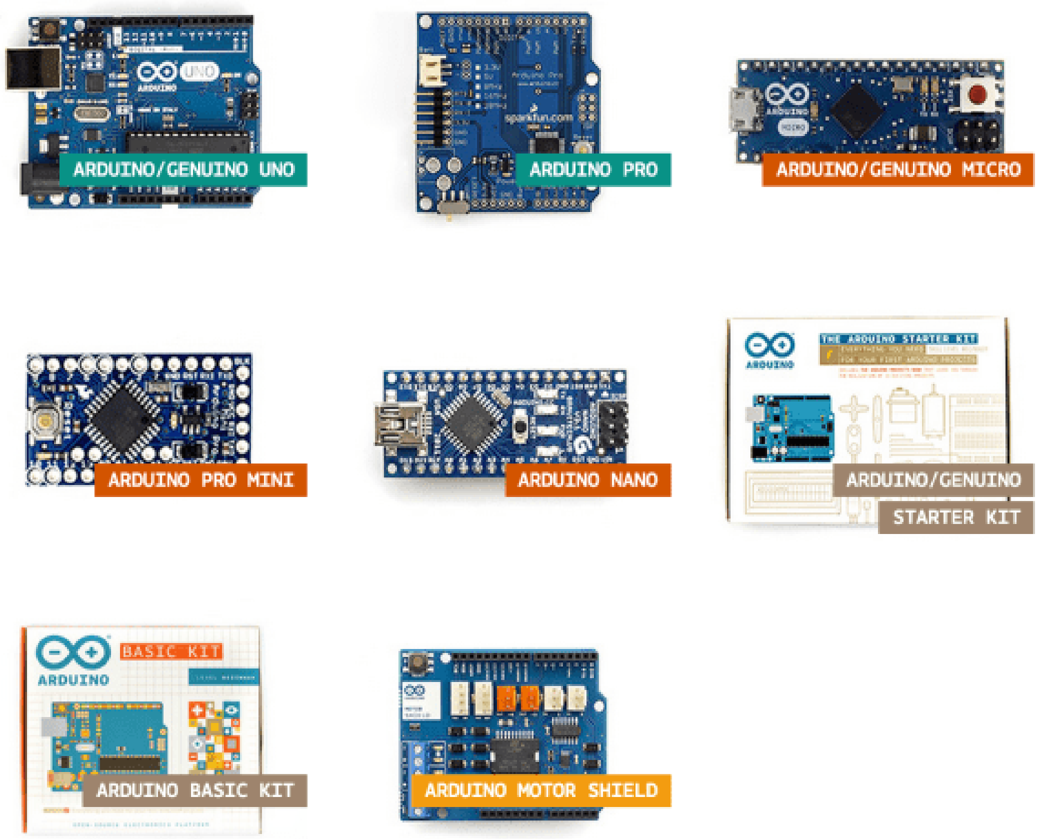


Fig.4.3: Arduino Family

4.1.5 Shields

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

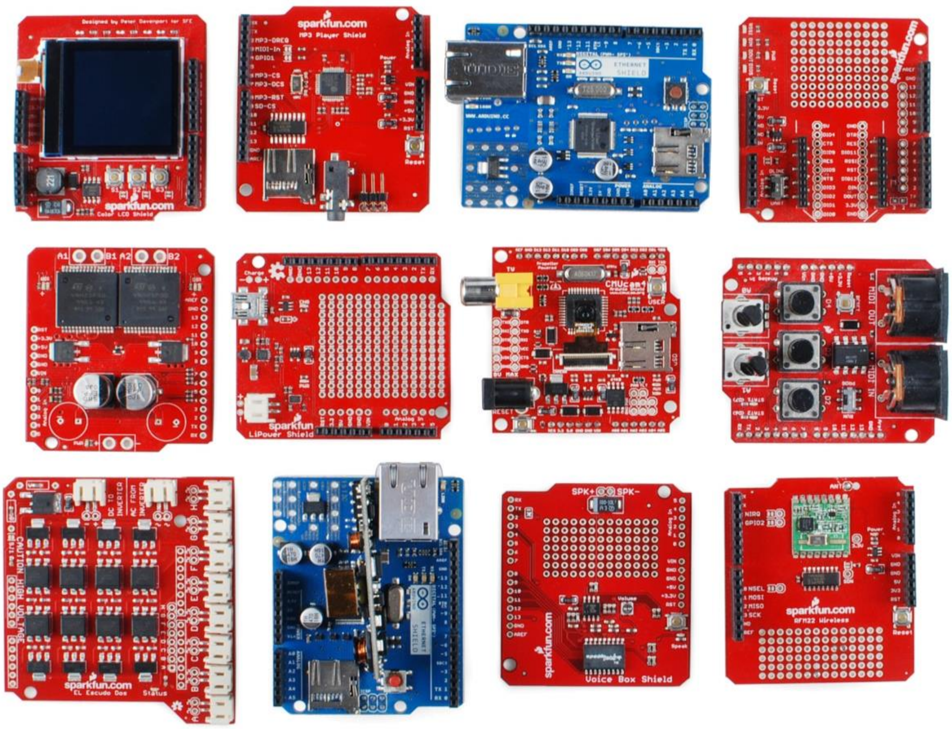


Fig.4.4: Arduino Shields

4.1.6 Pin Description Of Atmega328

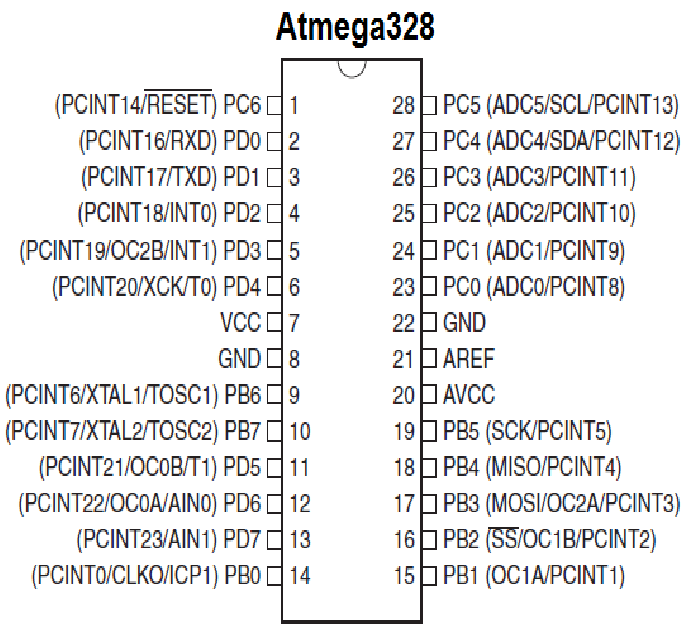


Fig.4.5: Pin Description Of ATMEGA328

4.1.7 Advantages Of Arduino

- It is cheap
- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
- For beginners, it is very simple to use.

4.1.8 Applications

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

4.2 POWER SUPPLY UNIT

4.2.1 Introduction:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices.

A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as “Regulated D.C Power Supply”.

- **Block Diagram Of Power Supply:**

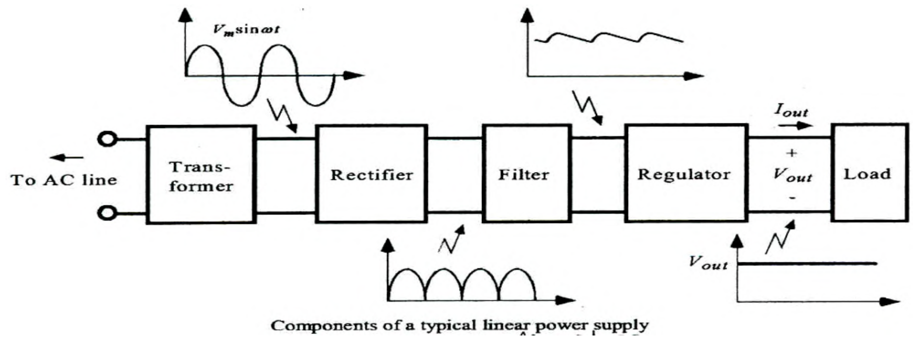


Fig.4.6: (a) Block Diagram of Power Supply

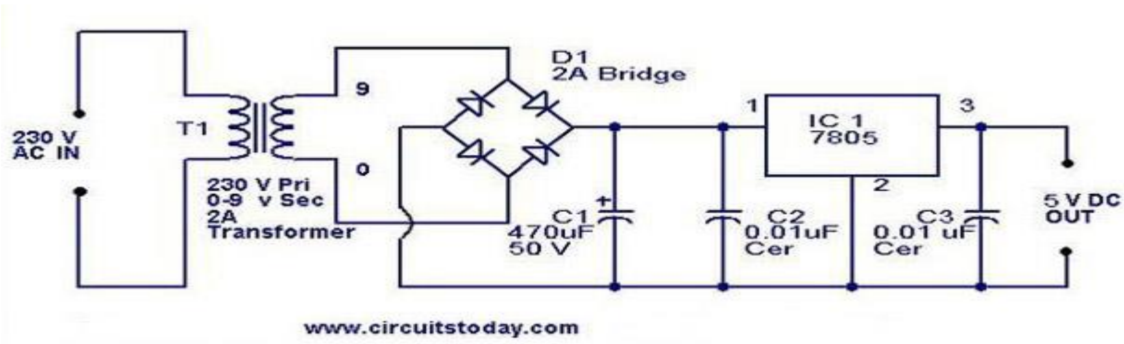


Fig.4.6: (b) Schematic Diagram of Power Supply

- **Description Of Power Supply:**

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a different form of energy – such as solar, mechanical, or chemical - into electrical energy.

A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered.

For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

4.2.2 Transformer:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

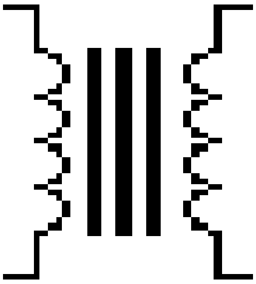


Fig.4.7: (a) Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

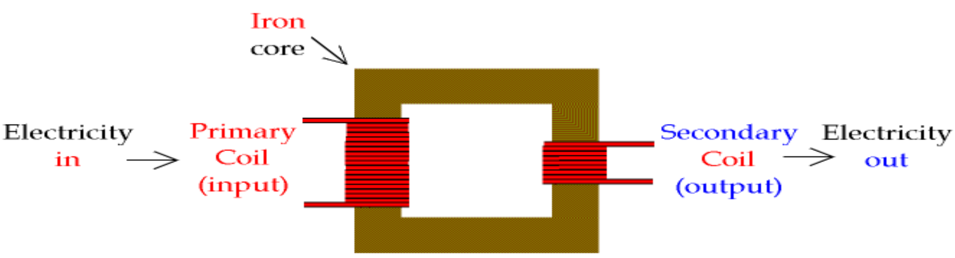


Fig.4.7: (b) Transformer

- **Basic Principle Of Transformer:**

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

From Faraday's Law

$$\frac{V_S}{V_P} = \frac{N_S}{N_P}$$

For ideal transformer

The voltage ratio is equal to the turns ratio, and power in equals power out.

From conservation of energy

$$P_P = V_P I_P = V_S I_S = P_S$$

- **Working Of Transformer:**

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below.

There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

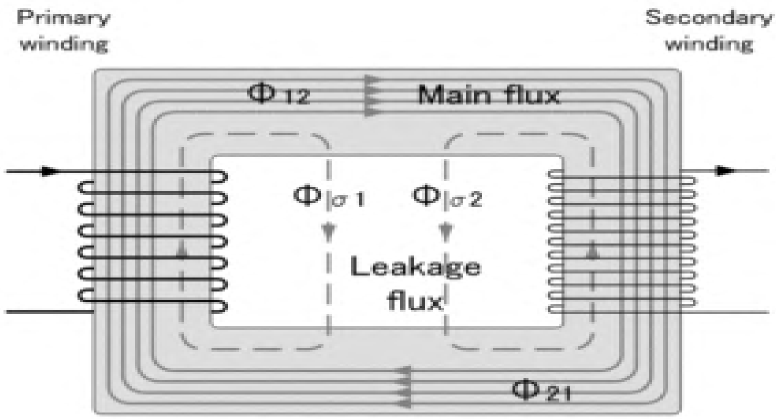


Fig.4.8: Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced EMF. The iron core is laminated to prevent 'eddy currents' flowing in the core.

These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

- ## Classification Of Transformer:

➢ Step-Up Transformer
➢ Step-Down Transformer

o Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage

applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn's ratio of the two sets of windings determines the amount of voltage transformation.

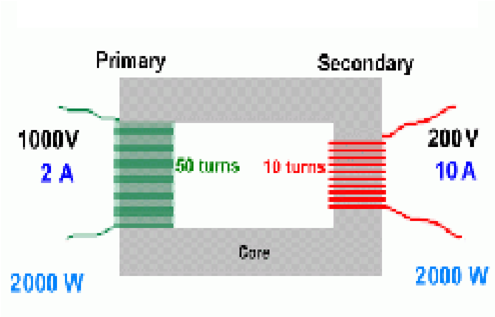


Fig.4.9: (a) Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.**Step down transformers can be considered nothing more than a voltage ratio device.**

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated.

Because of this it is possible to do some non-standard applications using standard transformer.single phase steps down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase steps up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.

- **Step-Up Transformer:**

A step up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply. A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

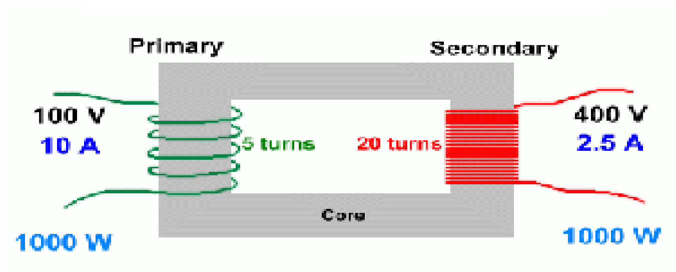


Fig.4.9: (b) Step-Up Transformer

- **Diodes:**

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

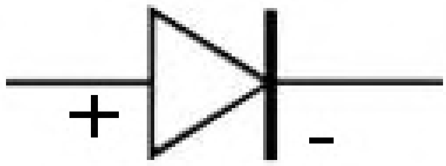


Fig.4.10: Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

- **Rectifier:**

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

- **The Half-Wave Rectifier:**

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure.

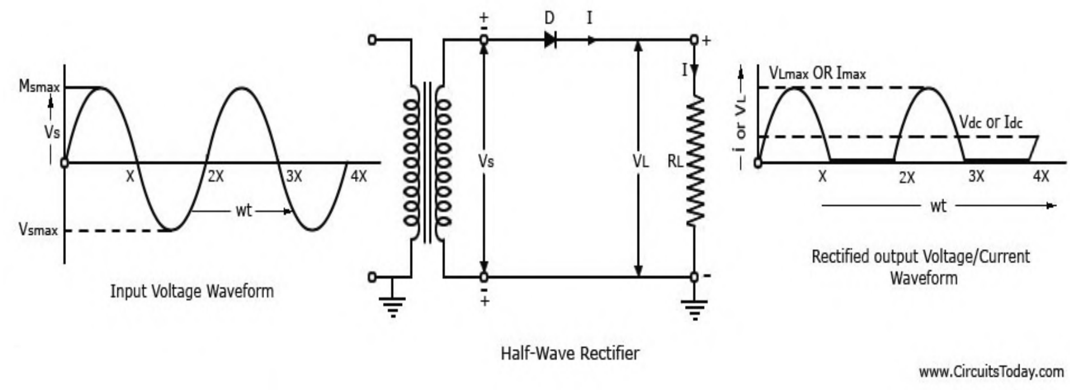


Fig.4.11: (a) Half Wave Rectifier

Figure 2 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

- **The Full-Wave Rectifier:**

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC.

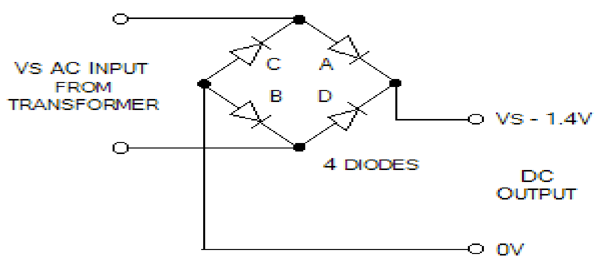


Fig.4.11: (b) Full-Wave Rectifier

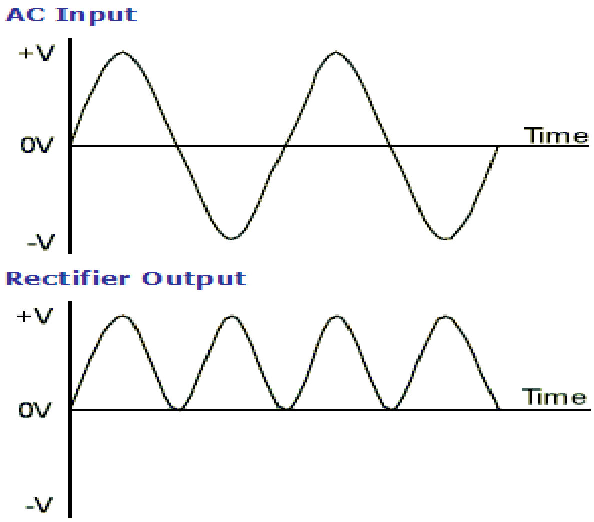


Fig.4.11: (c) Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased. While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

- **Capacitor Filter:**

The capacitor-input filter, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

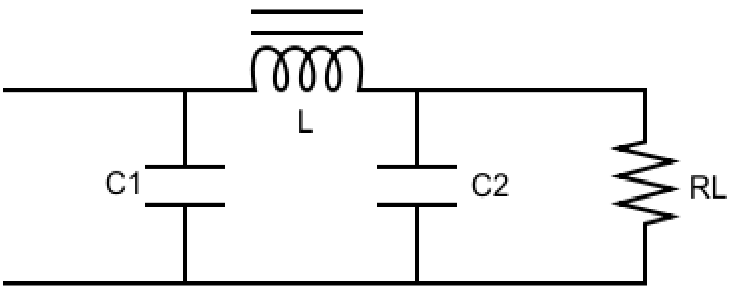


Fig.4.12: Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result, the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result, only the DC component appears across the load RL.

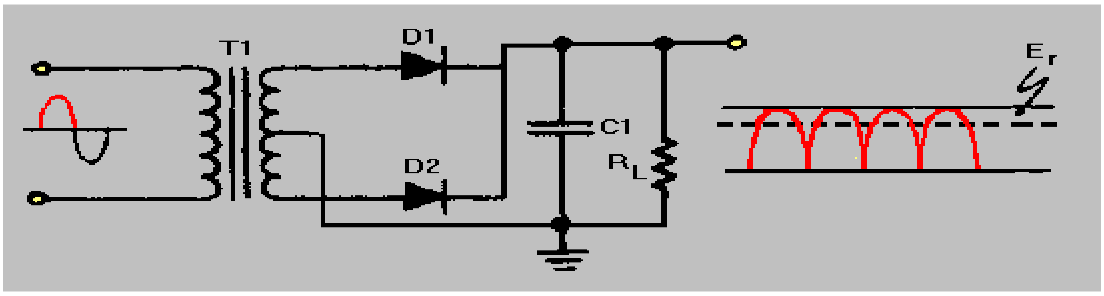


Fig.4.13: Centered Tapped Full-Wave Rectifier with a Capacitor Filter

- **Voltage Regulator:**

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages.

There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx' indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx' indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consist the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

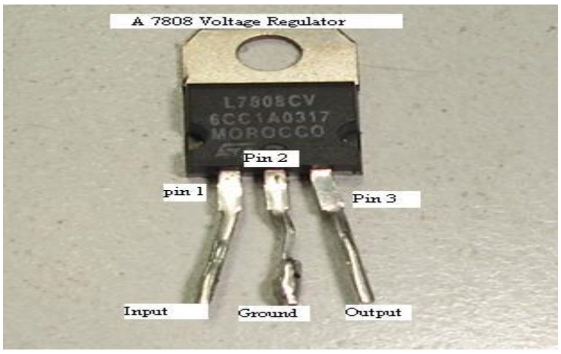


Fig.4.14: Regulator

4.3 WIFI MODULE:

The ESP8266 is a low-cost Wi-Fi module that can be integrated easily into IOT devices. We've featured several projects using this module, such as How To Make Smart Home Electronics: A Smart Mailbox and How To Read Your Arduino's Mind: Building A Childproof Lock. This tutorial will walk you through setting up ESP8266 Wi-Fi module which can be used with Arduino. The ESP8266 comes in many models with different functionalities. We'll be focusing on the ESP8266 ESP-01 module, the most common and basic one available.

4.3.1What is ESP8266?

The ESP8266 is a small Wi-Fi module built around the ESP8266 chip that can connect your microcontroller to the internet wirelessly for a very small cost. It can be a great option for Internet of Things (IOT) projects, but can be difficult to work with for beginner hobbyists who do not have prior experience with the module. In this tutorial, we hope to show you how to interface the ESP8266 with an Arduino and perform some basic functions like connecting it to a Wi-Fi network.

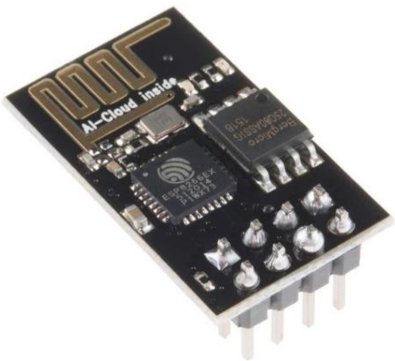


Fig.4.15: ESP8266 ESP-01 Module

ESP-01 Features – Spark fun**:**

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 1MB Flash Memory
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)

The first feature to notice about the ESP8266 is its awkwardly spaced header pins. The module has 8 pins that serve different functions, but they are packed in a 4×2 arrangement that makes plugging the module into a breadboard impossible. This means that to prototype projects on a breadboard, you'll need male-female jumper wires to connect the pins on the ESP8266 to rows on the breadboard. If you'd like to make your prototyping more compact, you can also purchase breadboard breakouts for the ESP8266 such as this one. For prototyping, I chose to just use jumper wires.

The pinout for the ESP8266's pins are according to the following diagram:

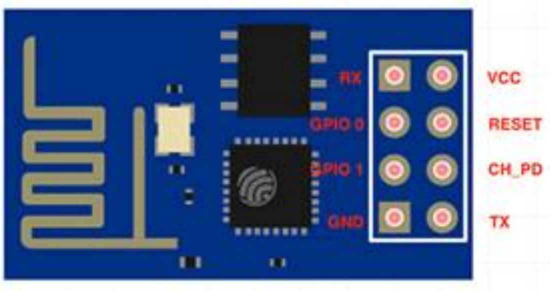


Fig.4.16: ESP8266 Pinout

RX: UART serial communication receive pin

GPIO 0: GPIO pin (unused in this project)

GPIO 1: GPIO pin (unused in this project)

GND: Connection to Ground

VCC: Connection to 3.3V Vcc(Vcc cannot exceed 3.3V!)

RESET: Reset pin (pull down to reset)

CH_PD: Chip enable and power down pin

TX: UART serial communication transmit pin

Note that the maximum voltage input for the ESP8266 is 3.3V. Any input voltage greater than 3.3V will damage the module! To program settings on the ESP8266, we'll first need to connect it to a serial terminal on a computer through which we can send it special commands. Settings that we'll have to program include, for example, the SSID and password for the wifi network the module will be connected to.

To connect the ESP8266 to a computer and configure its settings, we'll need a USB to serial adapter with 3.3V logic, along with a serial terminal program.

Fortunately for us, we have the Arduino and the Arduino IDE's serial monitor! This means that we'll just have to connect the ESP8266 module to the Arduino and upload a custom sketch to the Arduino.

Use the following diagram to connect the ESP8266 module to the Arduino:

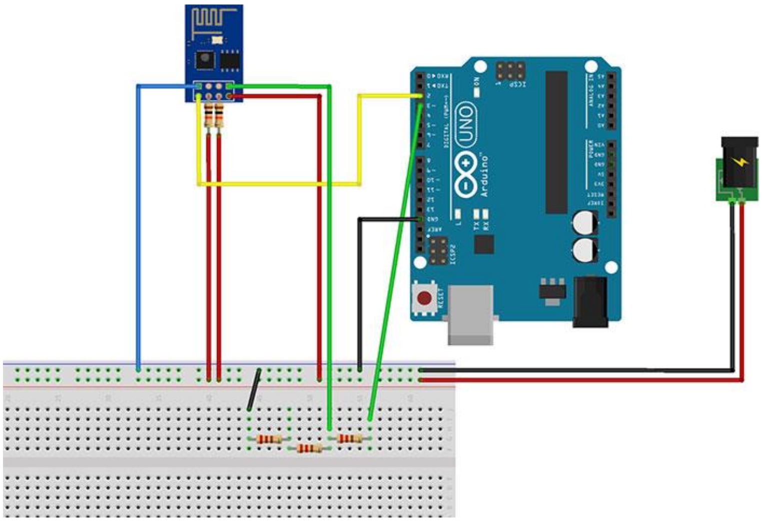


Fig.4.17: Connection Between ESP8266 And Arduino Uno

You will need the following parts:

- Arduino Uno or similar module
- ESP8266 ESP-01 module
- 2 x 1kΩ resistors for CH_PD and RESET pull-up
- 3 x 220Ω resistors for serial line voltage divider
- 3.3V regulated power supply for ESP8266 (you cannot plug the power line into the Arduino's 3.3V out pin because the ESP8266 draws a lot of current!)
    - You can use something like this: https://www.sparkfun.com/products/114
- Breadboard and jumper wires

A couple of features of this circuit stand out immediately.

First, the ESP8266 module is powered by a 3.3V regulated power supply. As described earlier, the ESP8266 has a maximum voltage input rating of 3.3 volts.

Using a power supply rated higher than this recommended 3.3 volts could fry the module! When connecting the ESP8266 to this power supply, You should not only connect the VCC and Ground pins to the power supply lines, but also connect the power supply's ground line to the Arduino's ground pin. Since the Arduino will be powered by the USB connection to the laptop, creating a common ground essentially creates a common reference (you can think of it as a baseline) to compare voltages and thereby interpret digital high and low signals.

Second, the receive (RX) line for the ESP8266 module is connected to the output of a resistor voltage-divider circuit. We do this to shift the serial communication logic level (the highs and lows of the digital signals that make up the serial communications) from a logic high of 5 volts on the Arduino to a logic high of 3.3 volts on the ESP8266 module. Again, the ESP8266 is specified with 3.3V logic, so connecting the module's receive line directly to the Arduino's transmit line could damage the device.

We do not need to shift levels on the ESP8266's transmit line because the module's 3.3V logic high is a high enough voltage to also register as a logic high on the Arduino.

Finally, there are two pull-up resistors on the ESP8266's CH_PD and RESET pins. These are specified by the module's creators to ensure that the device functions properly while it is being used. To reset the ESP8266 or disable it, these pins must be pulled down, but since we do not want those processes to occur, we will instead pull these pins high through some 10kΩ pull-up resistors.

4.3.2 ESP32 Camera

The ESP32-CAM is a very small camera module with the ESP32-S chip that costs approximately $10. Besides the OV2640 camera, and several GPIOs to connect peripherals, it also features a microSD card slot that can be useful to store images taken with the camera or to store files to serve to clients.

Step 1: Components required

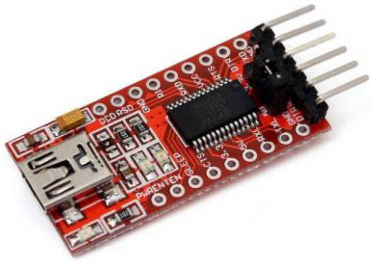


Fig.4.18: (a) WIFI Module

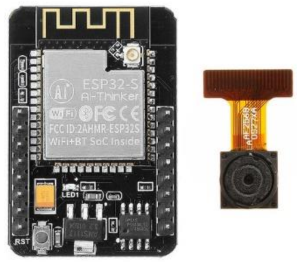


Fig.4.18: (b) ESP32-CAM

Step 2: Pin Configuration & Features

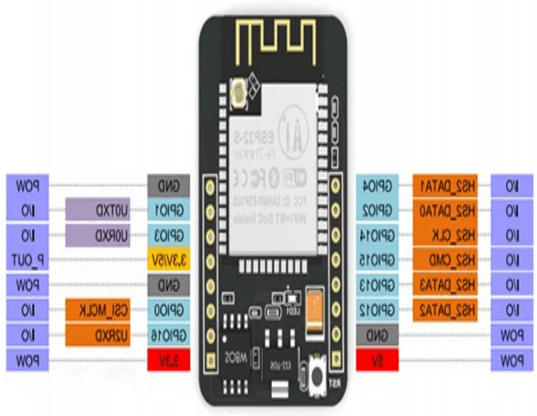


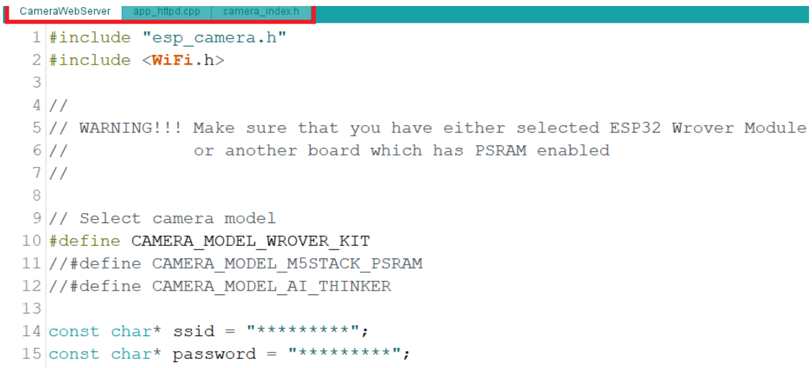
Fig.4.19: ESP32-CAM Pin Configuration

The smallest 802.11b/g/n Wi-Fi BT SoC module Low power 32-bit CPU, can also serve the application processor Up to 160MHz clock speed, summary computing power up to 600 DMIPS Built-in 520 KB SRAM,

- external 4MPSRAM
- Supports UART/SPI/I2C/PWM/ADC/DAC
- Support OV2640 and OV7670 cameras, built-in flash lamp
- Support image Wi-Fi upload

- Support TF card Supports multiple sleep modes
- Embedded Lwip and Free RTOS Supports STA/AP/STA+AP operation mode
- Support Smart Config/AirKiss technology
- Support for serial port local and remote firmware upgrades (FOTA)

Step 3: Code




```
#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! Make sure that you have either selected ESP32 Wrover Module
//            or another board which has PSRAM enabled
//

// Select camera model
#define CAMERA_MODEL_WROVER_KIT
//#define CAMERA_MODEL_M5STACK_PSRAM
//#define CAMERA_MODEL_AI_THINKER

const char* ssid = "*********";
const char* password = "*********";
```

Fig.4.20: Arduino Code For ESP32-CAM

In your Arduino IDE, go to File > Examples > ESP32 > Camera and open the Camera Web Server example.

Step 5: Schematics for Programming the Board

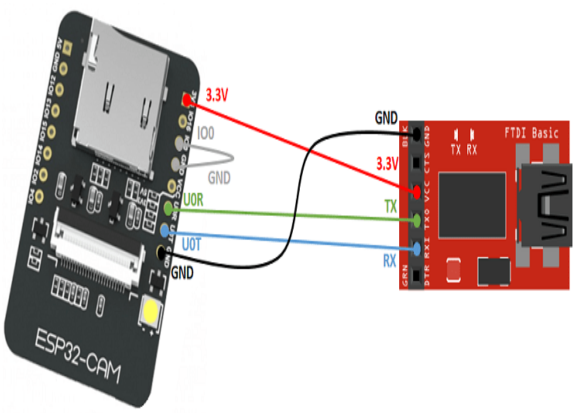


Fig.4.21: Connecting between Wi-Fi-Module with ESP32-CAM

The ESP32-CAM doesn’t have a USB connector, so you need to use FTDI to upload code through the U0R and U0T pins (serial pins) in ESP32 CAM Board.

Step 6: Uploading of the Code

Before uploading the code, you need to input your Wi-Fi credentials in the following part of **code**:

const char* ssid = "REPLACE_WITH_YOUR_SSID";

const char* password = "REPLACE_WITH_YOUR_PASSWORD";

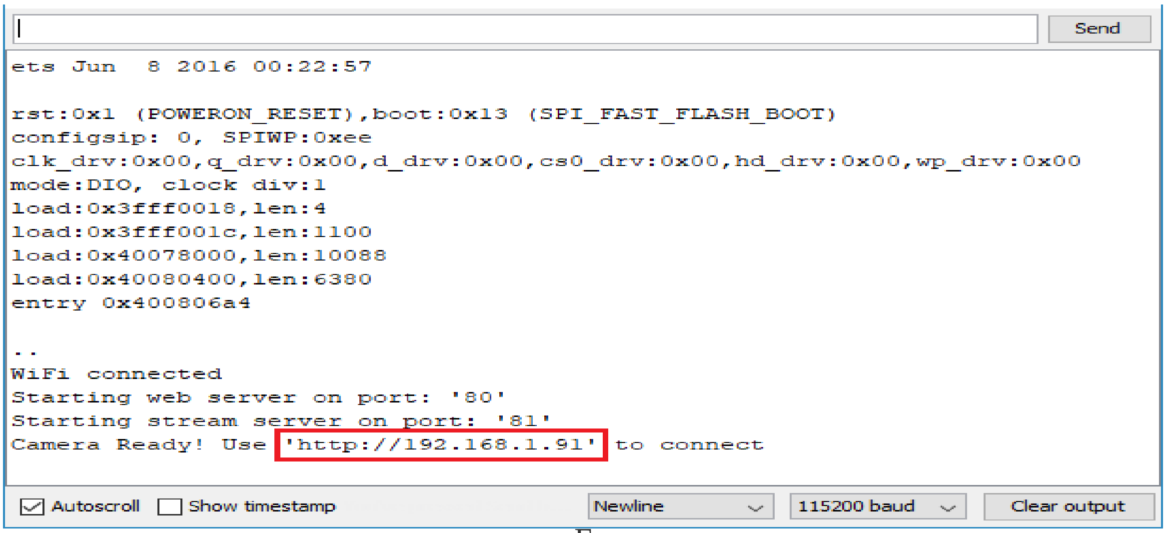
and make sure you select the right camera module. As here we’re using the AI-THINKER Model so select the following So, comment all the other models and uncomment this one:

#define CAMERA_MODEL_AI_THINKER

Follow these steps to upload the code :

Go to Tools > Board and select ESP32 wrover Module Go to Tools > Port and select the COM port the ESP32 is connected to In Tools > Partition Scheme, select “Huge APP (3MB No OTA)“Then, click the upload button to upload the code.

Step 7: Getting IP From Serial Monitor



F

FF

Fig.4.22: Getting IP Address From Serial Monitor

Remove the jumper connected between GPIO0 & GND then, Open the Serial Monitor with the baud rate : 115200. Press the ESP32-CAM on-board Reset button and wait for the IP to appear and wait for few seconds and then hit reset again.

4.4 16 * 2 ALPHANUMERIC LCD

4.4.1 Description

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

4.4.2 Block diagram

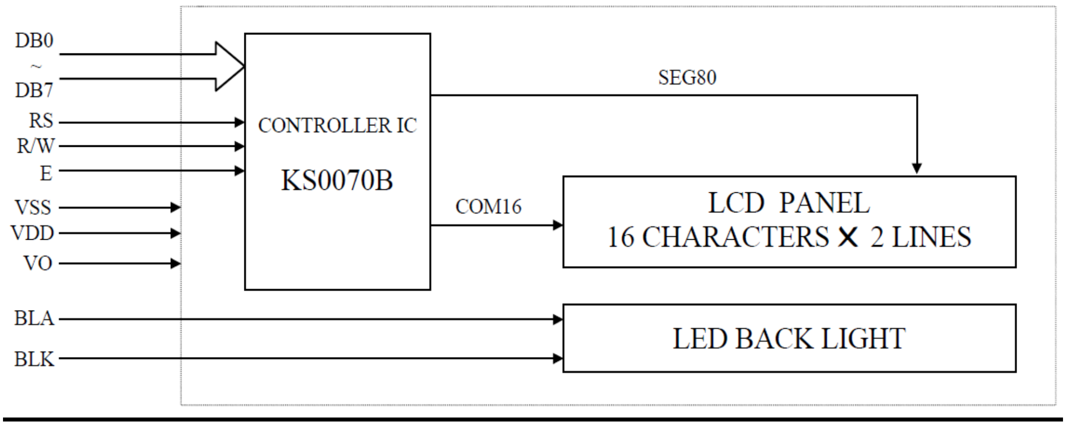


Fig.4.23: Block Diagram Of LCD Display

- **Display Data Ram**

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD.

For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.

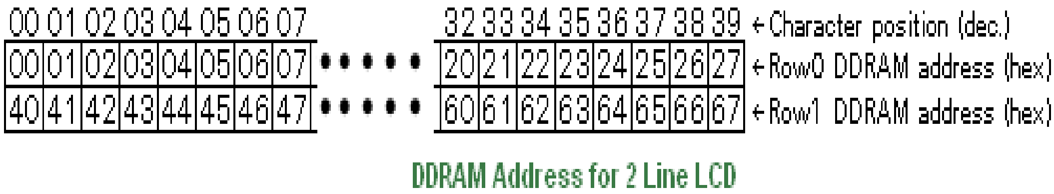


Fig.4.24: DDRAM Address of 2 Line LCD

- **Character Generator Rom**

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM.

- **Busy Flag**

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0).

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

- **Instruction Register (IR) And Data Register (DR)**

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD.

When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD

Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

4.4.3 16 x 2 Alphanumeric LCD Module Features

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

- **Schematic**

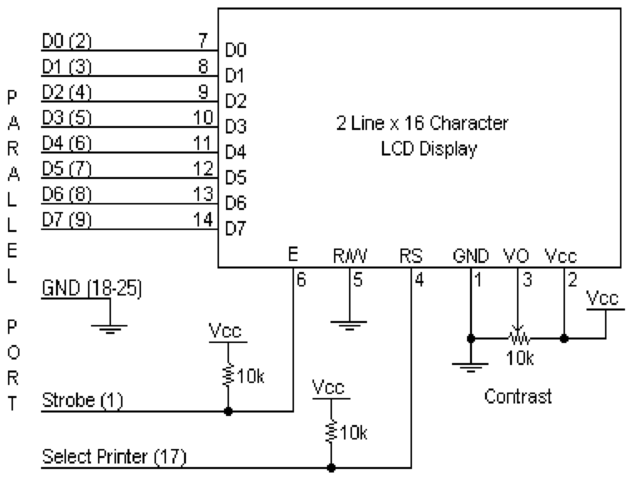


Fig.4.25: Schematic of 2 x16 Character LCD Display

- ## Specifications

Connector Pin Assignment:

Pin	*Symbol*	*Function*	*Pin*	*Symbol*	*Function*
14	DB7	Data Bus Line	6	E	Enable
13	DB6	Data Bus Line	5	R/W	Read/Write
12	DB5	Data Bus Line	4	RS	Register select input
11	DB4	Data Bus Line	3	Vcontrast	Contrast (0V for max contrast)
10	DB3	Data Bus Line	2	**GND (0V)**	Power supply
9	DB2	Data Bus Line	1	**Vcc +5V**	Power supply
8	DB1	Data Bus Line			
7	DB0	Data Bus Line			

Note1: Pin 1 is +5V and pin 2 is GND! This is different from most other HD44780 compatible models.

Note2: Pin 14 is marked with a “1” on the back of the PCB. Do not get confused by this.

Table.4.7: Connector Pin Assignment

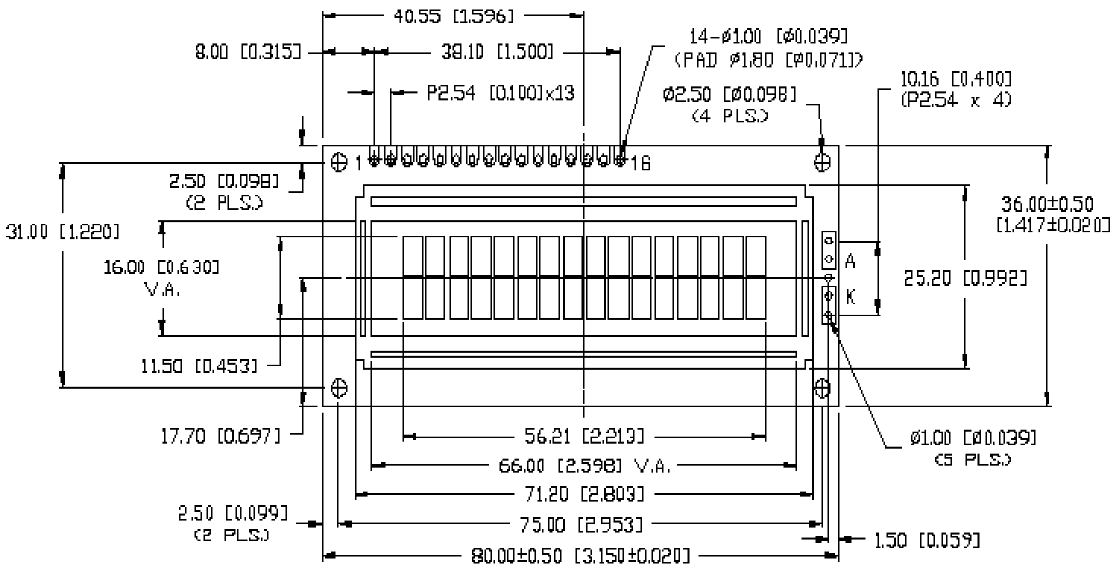


Fig.4.26: Internal Voltage and Current Distribution

- ## Circuit Description

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output.

While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly. The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

4.5 L293D

L293D is basically a high current dual motor driver/controller Integrated Circuit (IC). It is able to drive load having current up to 1A at the voltage ranging from 4.5V to 36V. Motor driver usually act as current amplifier because they receive a low current signal as an input and provides high current signal at the output.

Motors usually operates on this higher current. L-293D has to Built in H-Bridge driver circuits and is able to control two DC motors at a time in both clockwise and counter clockwise direction. It has two enable pins and they should be kept high in order to control the motor. By changing the polarity of applied signal motor can be rotated in either clockwise or counter clockwise direction. If L 293D enable pin is high, its corresponding driver will provide the desired out. If the enable pin is low, there will be no output. L-293D has different features including internal ESD protection, large voltage supply range, large output current per channel, high noise immunity input etc. L 293D plays a vital role in electronics era and has several different applications e.g relay drivers, DC motor drivers, stepping motor drivers etc. The further detail about L 293D motor driver/controller will be given later in this tutorial.

4.5.1 Introduction to L293D

L293D is basically a motor driver or controller. It has two built in H-bridge circuits which are able to control two DC motors simultaneously in both clockwise and counter clockwise direction. It acts as an current high amplifier because it take low current signal at its input and provides higher current signal at the output in order to drives different load e.g stepper motor & DC motors. Its features include large input voltage supply range, large output current high noise immunity input signals etc.

Its common real life applications include stepping motor drivers, relay drivers, DC motor drivers etc. L-293D motor driver/controller is shown in the figure given below.

Fig.4.27: L293D

1. L293D Pins

- L-293D has sixteen (16) pins, having different individual functions.
- All of these sixteen pins along with their serial no, are given in the table shown below

2. L293D Pin Functions

- L 293D each pin has different tasks to perform when they are in working condition.
- The tasks assigned to each pin are given in the table shown below.

L293D Pins	
Pin. No	Pin Name
1	Enable 1,2
2	Input 1
3	Output 1
4	GND
5	GND
6	Output 2
7	Input 2
8	V_{CC2}
9	Enable 3,4
10	Input 3
11	Output 3
12	GND
13	GND
14	Output put 4
15	Input 4
16	V_{CC1}

Table.4.8: L293D Pin

3. L293D Pinout

- Pinout diagram of any device presents the pins configuration through a completely labelled diagram.
- L293D pinout diagram is shown in the figure given below.

Fig.4.28: L293D Pin Configuration

4.6 MOTORS

4.6.1 Definition

Motor is a device that creates motion, not an engine; it usually refers to either an electrical motor or an internal combustion engine. It may also refer to:

Electric motor, a machine that converts electricity into a mechanical motion

- AC motor, an electric motor that is driven by alternating current
  - Synchronous motor, an alternating current motor distinguished by a rotor spinning with coils passing magnets at the same rate as the alternating current and resulting magnetic field which drives it
  - Induction motor, also called a squirrel-cage motor, a type of asynchronous alternating current motor where power is supplied to the rotating device by means of electromagnetic induction

- DC motor, an electric motor that runs on direct current electricity
  - Brushed DC electric motor, an internally commutated electric motor designed to be run from a direct current power source
  - Brushless DC motor, a synchronous electric motor which is powered by direct current electricity and has an electronically controlled commutation system, instead of a mechanical commutation system based on brushes

- Electrostatic motor, a type of electric motor based on the attraction and repulsion of electric charge
- Servo motor, an electric motor that operates a servo, commonly used in robotics
- Internal fan-cooled electric motor, an electric motor that is self-cooled by a fan, typically used for motors with a high energy density

4.6.2 Types of Motors

Industrial motors come in a variety of basic types. These variations are suitable for many different applications. Naturally, some types of motors are more suited for certain applications than other motor types are. This document will hopefully give some guidance in selecting these motors.

1. **AC Motors**

The most common and simple industrial motor is the three phase AC induction motor, sometimes known as the "squirrel cage" motor. Substantial information can be found about any motor by checking its (nameplate).

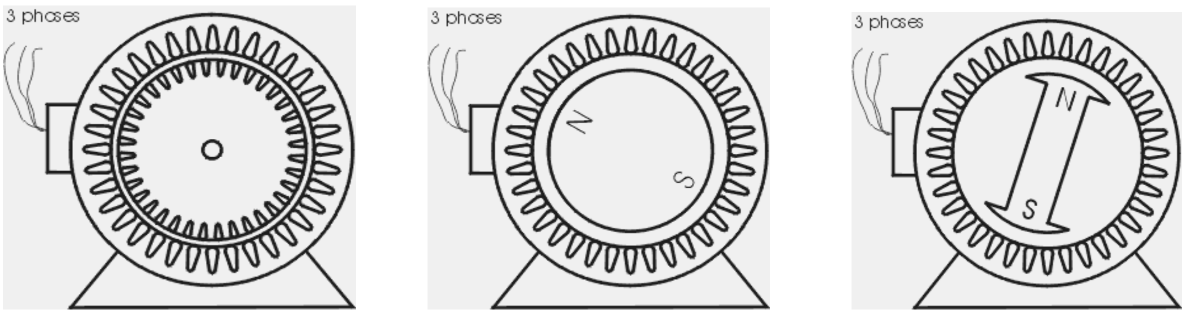


Fig.4.29: 3 Phase AC Induction Motor

- **Advantages**
    - Simple Design
    - Low Cost
    - Reliable Operation
    - Easily Found Replacements
    - Variety of Mounting Styles
    - Many Different Environmental Enclosures

Simple Design

The simple design of the AC motor -- simply a series of three windings in the exterior (stator) section with a simple rotating section (rotor). The changing field caused by the 50 or 60 Hertz AC line voltage causes the rotor to rotate around the axis of the motor.

The speed of the AC motor depends only on three variables:

1. The fixed number of winding sets (known as poles) built into the motor, which determines the motor's base speed.
2. The frequency of the AC line voltage. Variable speed drives change this frequency to change the speed of the motor.
3. The amount of torque loading on the motor, which causes slip.

Low Cost

The AC motor has the advantage of being the lowest cost motor for applications requiring more than about 1/2 hp (325 watts) of power. This is due to the simple design of the motor. For this reason, AC motors are overwhelmingly preferred for fixed speed applications in industrial applications and for commercial and domestic applications where AC line power can be easily attached. Over 90% of all motors are AC induction motors.

They are found in air conditioners, washers, dryers, industrial machinery, fans, blowers, vacuum cleaners, and many, many other applications.

- **Reliable Operation**

The simple design of the AC motor results in extremely reliable, low maintenance operation. Unlike the DC motor, there are no brushes to replace. If run in the appropriate environment for its enclosure, the AC motor can expect to need new bearings after several years of operation. If the application is well designed, an AC motor may not need new bearings for more than a decade.

- **Easily Found Replacements**

The wide use of the AC motor has resulted in easily found replacements. Many manufacturers adhere to either European (metric) or American (NEMA) standards. (For Replacement Motors) Variety of Mounting Styles

AC Motors are available in many different mounting styles such as:

- Foot Mount
- C-Face
- Large Flange
- Vertical
- Specialty

2. DC Motors

The brushed DC motor is one of the earliest motor designs. Today, it is the motor of choice in the majority of variable speed and torque control applications.

- **Advantages**
  - Easy to understand design
  - Easy to control speed
  - Easy to control torque
  - Simple, cheap drive design

- **Easy to understand design**

The design of the brushed DC motor is quite simple. A permanent magnetic field is created in the stator by either of two means:

- Permanent magnets
- Electro-magnetic windings

If the field is created by permanent magnets, the motor is said to be a "permanent magnet DC motor" (PMDC). If created by electromagnetic windings, the motor is often said to be a "shunt wound DC motor" (SWDC). Today, because of cost-effectiveness and reliability, the PMDC motor is the motor of choice for applications involving fractional horsepower DC motors, as well as most applications up to about three horsepower.

At five horsepower and greater, various forms of the shunt wound DC motor are most commonly used. This is because the electromagnetic windings are more cost effective than permanent magnets in this power range.

Caution: If a DC motor suffers a loss of field (if for example, the field power connections are broken), the DC motor will immediately begin to accelerate to the top speed which the loading will allow. This can result in the motor flying apart if the motor is lightly loaded. The possible loss of field must be accounted for, particularly with shunt wound DC motors.

Opposing the stator field is the armature field, which is generated by a changing electromagnetic flux coming from windings located on the rotor. The magnetic poles of the armature field will attempt to line up with the opposite magnetic poles generated by the stator field. If we stopped the design at this point, the motor would spin until the poles were opposite one another, settle into place, and then stop -- which would make a pretty useless motor!

However, we are smarter than that. The section of the rotor where the electricity enters the rotor windings is called the commutator. The electricity is carried between the rotor and the stator by conductive graphite-copper brushes (mounted on the rotor) which contact rings on stator. Imagine power is supplied:

The motor rotates toward the pole alignment point. Just as the motor would get to this point, the brushes jump across a gap in the stator rings.

Momentum carries the motor forward over this gap. When the brushes get to the other side of the gap, they contact the stator rings again and -- the polarity of the voltage is reversed in this set of rings! The motor begins accelerating again, this time trying to get to the opposite set of poles. (The momentum has carried the motor past the original pole alignment point.) This continues as the motor rotates.

In most DC motors, several sets of windings or permanent magnets are present to smooth out the motion.

- **Easy to control speed**

Controlling the speed of a brushed DC motor is simple. The higher the armature voltage, the faster the rotation. This relationship is linear to the motor's maximum speed.

The maximum armature voltage which corresponds to a motor's rated speed (these motors are usually given a rated speed and a maximum speed, such as 1750/2000 rpm) are available in certain standard voltages, which roughly increase in conjunction with horsepower. Thus, the smallest industrial motors are rated 90 VDC and 180 VDC. Larger units are rated at 250 VDC and sometimes higher.

Specialty motors for use in mobile applications are rated 12, 24, or 48 VDC. Other tiny motors may be rated 5 VDC.

Most industrial DC motors will operate reliably over a speed range of about 20:1 -- down to about 5-7% of base speed. This is much better performance than the comparable AC motor. This is partly due to the simplicity of control, but is also partly due to the fact that most industrial DC motors are designed with variable speed operation in mind, and have added heat dissipation features which allow lower operating speeds.

- **Easy to control torque**

In a brushed DC motor, torque control is also simple, since output torque is proportional to current. If you limit the current, you have just limited the torque which the motor can achieve. This makes this motor ideal for delicate applications such as textile manufacturing.

- **Simple, cheap drive design**

The result of this design is that variable speed or variable torque electronics are easy to design and manufacture. Varying the speed of a brushed DC motor requires little more than a large enough potentiometer. In practice, these have been replaced for all but sub-fractional horsepower applications by the SCR and PWM drives, which offer relatively precisely control voltage and current. Common DC drives are available at the low end (up to 2 horsepower) for under US$100 -- and sometimes under US$50 if precision is not important.

Large DC drives are available up to hundreds of horsepower. However, over about 10 horsepower careful consideration should be given to the price/performance trade offs with AC inverter systems, since the AC systems show a price advantage in the larger systems. (But they may not be capable of the application's performance requirments).

- **Disadvantages**
  - Expensive to produce
  - Can't reliably control at lowest speeds
  - Physically larger
  - High maintenance
  - Dust

4.6.3 WORKING OF DC MOTOR

In any electric motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. As you are well aware of from playing with magnets as a kid, opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion.

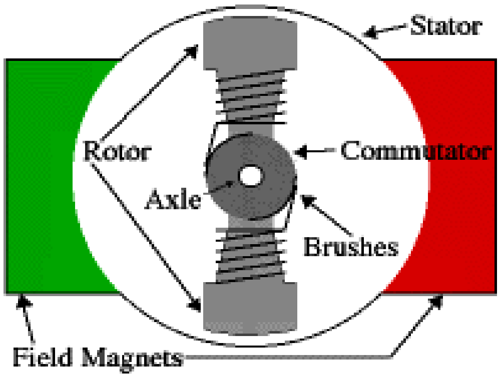


Fig.4.30: Internal Components of Motor

- **Principle**

When a rectangular coil carrying current is placed in a magnetic field, a torque acts on the coil which rotates it continuously. When the coil rotates, the shaft attached to it also rotates and thus it is able to do mechanical work.

Every DC motor has six basic parts -- axle, rotor (a.k.a., armature), stator, commutator, field magnet(s), and brushes. In most common DC motors (and all that BEAMers will see), the external magnetic field is produced by high-strength permanent magnets[1]. The stator is the stationary part of the motor -- this includes the motor casing, as well as two or more permanent magnet pole pieces. The rotor (together with the axle and attached commutator) rotate with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a common motor layout -- with the rotor inside the stator (field) magnets.

The geometry of the brushes, commentator contacts, and rotor windings are such that when power is applied, the polarities of the energized winding and the stator magnet(s) are misaligned, and the rotor will rotate until it is almost aligned with the stator's field magnets. As the rotor reaches alignment, the brushes move to the next commentator contacts, and energize the next winding. Given our example two-pole motor, the rotation reverses the direction of current through the rotor winding, leading to a "flip" of the rotor's magnetic field, driving it to continue rotating.

In real life, though, DC motors will always have more than two poles (three is a very common number). In particular, this avoids "dead spots" in the commutator.

You can imagine how with our example two-pole motor, if the rotor is exactly at the middle of its rotation (perfectly aligned with the field magnets), it will get "stuck" there. Meanwhile, with a two-pole motor, there is a moment where the commutator shorts out the power supply (i.e., both brushes touch both commutator contacts simultaneously). This would be bad for the power supply, waste energy, and damage motor components as well. Yet another disadvantage of such a simple motor is that it would exhibit a high amount of torque "ripple" (the amount of torque it could produce is cyclic with the position of the rotor).

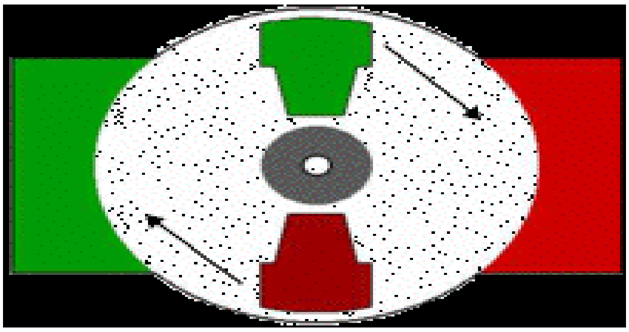


Fig.4.31: Rotation Of Two Polar Motor

- **Construction and Working**

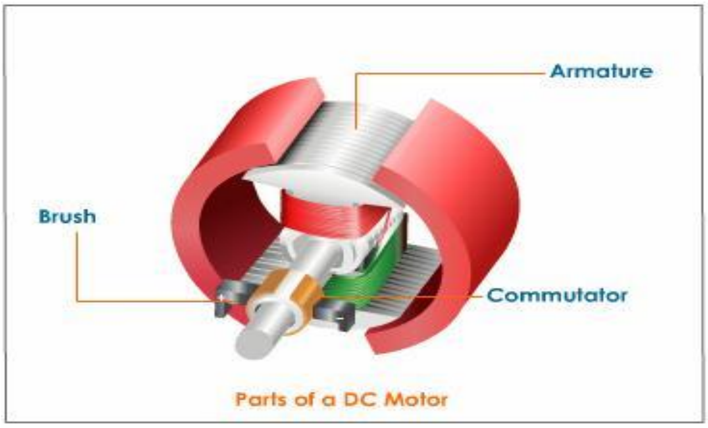


Fig.4.32: Parts of DC Motor

4.6.4 Parts of a DC Motor

- **Armature**

A D.C. motor consists of a rectangular coil made of insulated copper wire wound on a soft iron core. This coil wound on the soft iron core forms the armature. The coil is mounted on an axle and is placed between the cylindrical concave poles of a magnet.

- **Commutator**

A commutator is used to reverse the direction of flow of current. Commutator is a copper ring split into two parts C_1 and C_2. The split rings are insulated form each other and mounted on the axle of the motor. The two ends of the coil are soldered to these rings. They rotate along with the coil. Commutator rings are connected to a battery. The wires from the battery are not connected to the rings but to the brushes which are in contact with the rings.

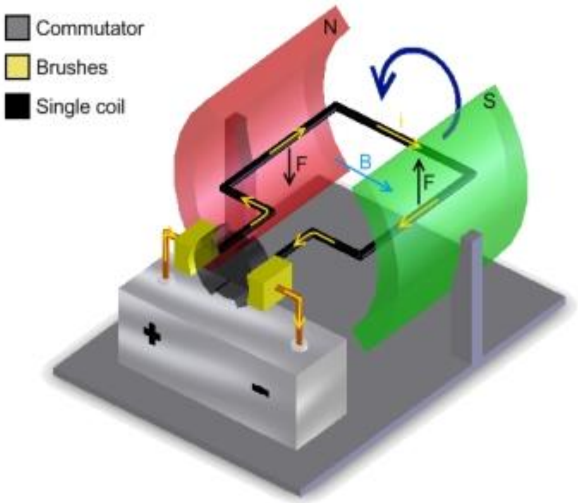


Fig.4.33: Commutator

- **Brushes**

Two small strips of carbon, known as brushes press slightly against the two split rings, and the split rings rotate between the brushes.

The carbon brushes are connected to a D.C. source.

- **Working of a DC Motor**

When the coil is powered, a magnetic field is generated around the armature. The left side of the armature is pushed away from the left magnet and drawn towards the right, causing rotation.

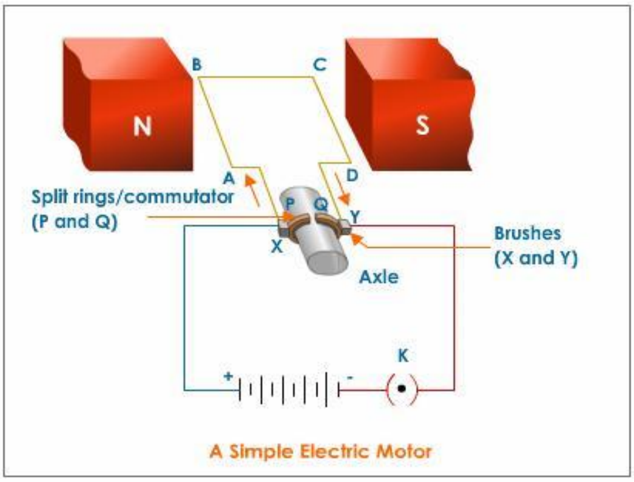


Fig.4.34: Simple Electric Motor

When the coil turns through 90^0, the brushes lose contact with the commutator and the current stops flowing through the coil.

However the coil keeps turning because of its own momentum. Now when the coil turns through 180^0, the sides get interchanged. As a result the commutator ring C_1 is now in contact with brush B_2 and commutator ring C_2 is in contact with brush B_1. Therefore, the current continues to flow in the same direction.

4.6.5 Parameters of the Dc Motors

- Direction of rotation
- Motor Speed
- Motor Torque
- Motor Start and Stop

▪ Direction of Rotation

A DC Motor has two wires. We can call them the positive terminal and the negative terminal, although these are pretty much arbitrary names (unlike a battery where these polarities are vital and not to be mixed!). On a motor, we say that when the + wire is connected to + terminal on a power source, and the - wire is connected to the - terminal source on the same power source, the motor rotates clockwise (if you are looking towards the motor shaft). If you reverse the wire polarities so that each wire is connected to the opposing power supply terminal, then the motor rotates counter clockwise. Notice this is just an arbitrary selection and that some motor manufacturers could easily choose the opposing convention. As long as you know what rotation you get with one polarity, you can always connect in such a fashion that you get the direction that you want on a per polarity basis.

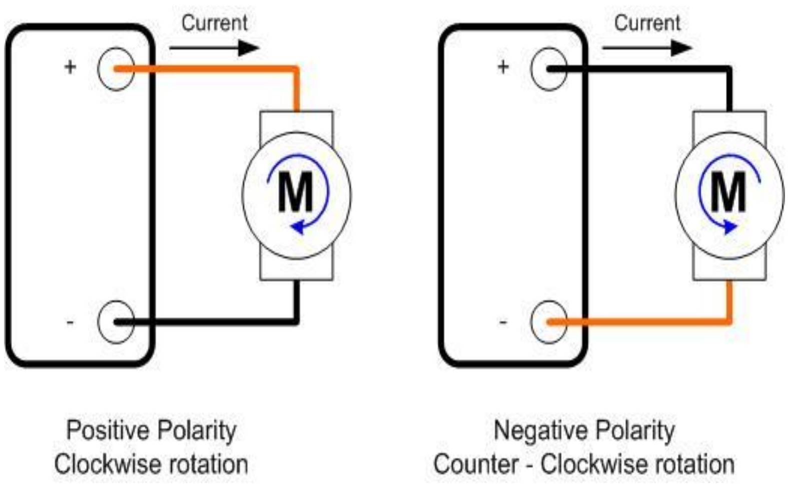


Fig.4.35: Motor rotation with Polarity

▪ DC Motor Rotation vs Polarity

Facts:

- DC Motor rotation has nothing to do with the voltage magnitude or the current magnitude flowing through the motor.
- DC Motor rotation does have to do with the voltage polarity and the direction of the current flow.

- **DC Motor Speed**

Whereas the voltage polarity controls DC motor rotation, voltage magnitude controls motor speed. Think of the voltage applied as a facilitator for the strengthening of the magnetic field. In other words, the higher the voltage, the quicker will the magnetic field become strong. Remember that a DC motor has an electromagnet and a series of permanent magnets. The applied voltage generates a magnetic field on the electromagnet portion. This electromagnet field is made to oppose the permanent magnet field. If the electromagnet field is very strong, then both magnetic entities will try to repel each other from one side, as well as attract each other from the other side. The stronger the induced magnetic field, the quicker will this separation/attraction will try to take place. As a result, motor speed is directly proportional to applied voltage.

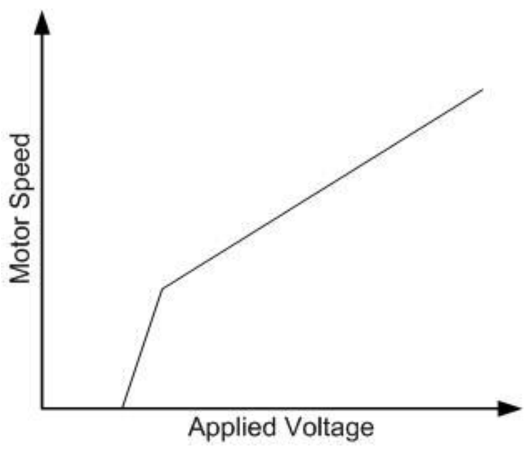


Fig.4.36: Motor Speed Vs Applied Voltage

- **Motor Speed Curve**

One aspect to have in mind is that the motor speed is not entirely lineal. Each motor will have their own voltage/speed curve. One thing I can guarantee from each motor is that at very low voltages, the motor will simply not move. This is because the magnetic field strength is not enough to overcome friction.

Once friction is overcome, motor speed will start to increase as voltage increase. The following video shows the concept of speed control and offers some ideas on how this can be achieved.

- ## Motor Torque

In the previous segment I kind of described speed as having to do with the strength of the magnetic field, but this is in reality misleading. Speed has to do with how fast the magnetic field is built and the attraction/repel forces are installed into the two magnetic structures. Motor strength, on the other hand, has to do with magnetic field strength. The stronger the electromagnet attracts the permanent magnet, the more force is exerted on the motor load.

Per example, imagine a motor trying to lift 10 pounds of weight. This is a force that when multiplied by a distance (how much from the ground we are lifting the load) results in WORK. This WORK when exerted through a predetermined amount of time (for how long we are lifting the weight) gives us power. But whatever power came in, must come out as energy cannot be created or destroyed. So that you know, the power that we are supplying to the motor is computed by

$$\mathbf{P = IV}$$

Where P is power, I is motor current and V is motor voltage. Hence, if the voltage (motor speed) is maintained constant, how much load we are moving must come from the current.As you increase load (or torque requirements) current must also increase.

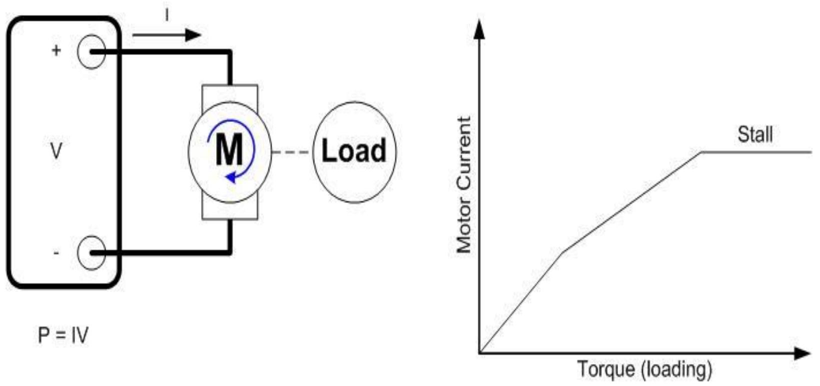


Fig.4.37: Torque Vs Motor Current

- ## Motor Loading

One aspect about DC motors which we must not forget is that loading or increase of torque cannot be infinite as there is a point in which the motor simply cannot move. When this happens, we call this loading “Stalling Torque”. At the same time this is the maximum amount of current the motor will see, and it is refer to Stalling Current. Stalling deserves a full chapter as this is a very important scenario that will define a great deal of the controller to be used. I promise I will later write a post on stalling and its intricacies.

- ## Motor Start and Stop

You are already well versed on how to control the motor speed, the motor torque and the motor direction of rotation. But this is all fine and dandy as long as the motor is actually moving. How about starting it and stopping it? Are these trivial matters? Can we just ignore them or should we be careful about these aspects as well? You bet we should!

Starting a motor is a very hazardous moment for the system. Since you have an inductance whose energy storage capacity is basically empty, the motor will first act as an inductor. In a sense, it should not worry us too much because current cannot change abruptly in an inductor, but the truth of the matter is that this is one of the instances in which you will see the highest currents flowing into the motor. The start is not necessarily bad for the motor itself as in fact the motor can easily take this Inrush Current. The power stage, on the other hand and if not properly designed for, may take a beating.

Once the motor has started, the motor current will go down from inrush levels to whatever load the motor is at. Per example, if the motor is moving a few gears, current will be proportional to that load and according to torque/current curves.

Stopping the motor is not as harsh as starting. In fact, stopping is pretty much a breeze. What we do need to concern ourselves is with how we want the motor to stop. Do we want it to coast down as energy is spent in the loop, or do we want the rotor to stop as fast as possible? If the later is the option, then we need braking. Braking is easily accomplished by shorting the motor outputs.

The reason why the motor stops so fast is because as a short is applied to the motor terminals, the Back EMF is shorted. Because Back EMF is directly proportional to speed, making Back EMF = 0, also means making speed = 0.

4.6.6 MOTOR DRIVER CIRCUIT

The name "H-Bridge" is derived from the actual shape of the switching circuit which control the motion of the motor. It is also known as "Full Bridge". Basically there are four switching elements in the H-Bridge as shown in the figure below.

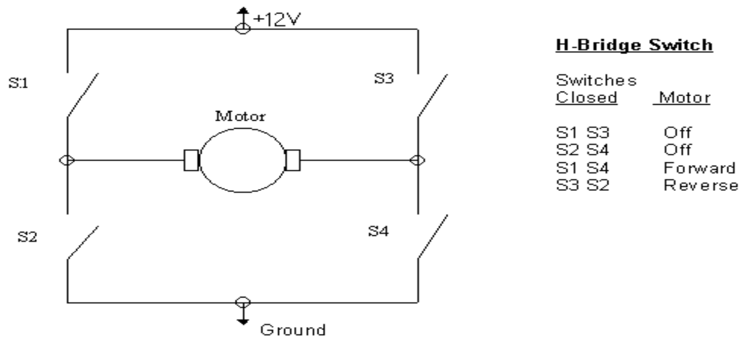


Fig.4.38: (a) H-Bridge

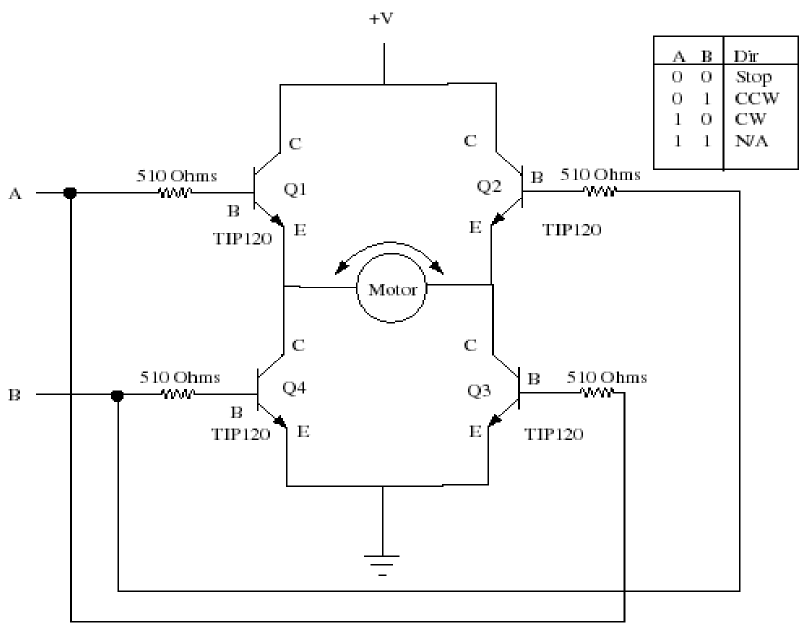


Fig.4.38: (b) Motor Driver Circuit

As you can see in the figure above there are four switching elements named as "High side left", "High side right", "Low side right", "Low side left". When these switches are turned on in pairs motor changes its direction accordingly. Like, if we switch on High side left and Low side right then motor rotate in forward direction, as current flows from Power supply

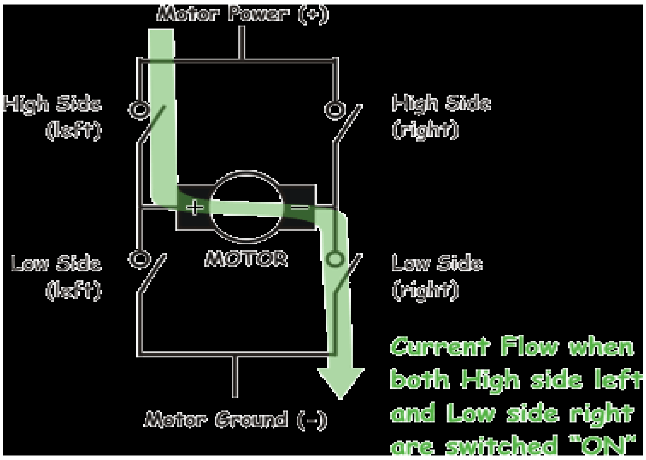


Fig.4.39: Current in Motor Circuit

Similarly, when you switch on low side left and high side right, the current flows in opposite direction and motor rotates in backward direction. This is the basic working of H-Bridge. We can also make a small truth table according to the switching of H-Bridge explained above.

Truth Table

High Left	High Right	Low Left	Low Right	Description
On	Off	Off	On	Motor runs clockwise
Off	On	On	Off	Motor runs anti-clockwise
On	On	Off	Off	Motor stops or decelerates
Off	Off	On	On	Motor stops or decelerates

Table.4.9: Truth Tables

As already said, H-bridge can be made with the help of transistors as well as MOSFETs, the only thing is the power handling capacity of the circuit. If motors are needed to run with high current then lot of dissipation is there. So head sinks are needed to cool the circuit.

Now you might be thinking why I did not discuss the cases like High side left on and Low side left on or high side right on and low side right on. Clearly seen in the diagram, you don't want to burn your power supply by shorting them. So that is why those combinations are not discussed in the truth table.

4.7 SOUND SENSOR:

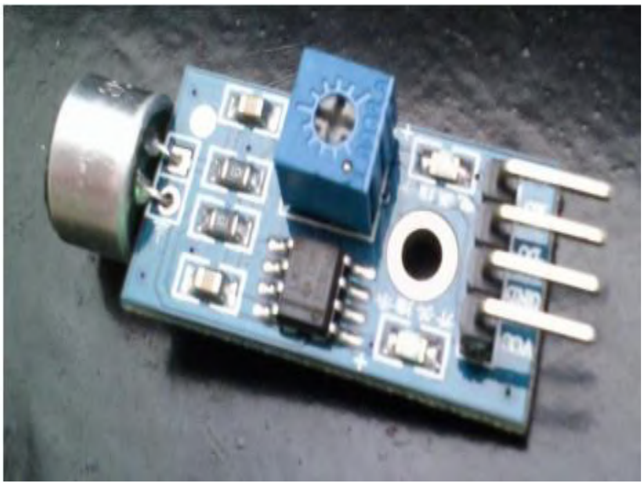


Fig.4.40: Sound Sensor

4.7.1 Description

The sound sensor module provides an easy way to detect sound and is generally used for detecting sound intensity. This module can be used for security, switch, and monitoring applications. Its accuracy can be easily adjusted for the convenience of usage. It uses a microphone which supplies the input to an amplifier, peak detector and buffer. When the sensor detects a sound, it processes an output signal voltage which is sent to a microcontroller then performs necessary processing.

Sound Sensor is designed to be used with a PASCO computer interface to make measurements of relative intensity of sound. The sensing element of the Sound Sensor is an electret condenser microphone, which consists of an electret membrane, metal electrode, and field effect transistor that are in an efficient configuration yielding superior signal to-noise ratios (>60 dB) and excellent frequency response (20 to 16,000 Hz).

Two stages of amplification are provided to condition the low-level signal from the microphone for input into the Science Workshop computer interface.

The output from the sensor is bipolar and ranges between ±10 volts. When the sensitivity is raised to high in Science Workshop (700 interface only), the Sound Sensor detects voltage levels as low as 0.0005 volts, corresponding to sound levels that are barely audible to the human ear.

Sound levels ranging from classroom background noise (45 dB) to levels exceeding 100 dB are easily detected with the Sound Sensor. The Oscilloscope and Fast Fourier Transform (FFT) functions of Science Workshop may be used effectively Equipment Included:

• Sound sensor in sensor box

• 6-foot cable with 8-pin DIN connectors

- **Additional Equipment Required**:

• Any PASCO Science Workshop™ computer interface (300, 500, or 700 series for Macintosh or Windows) or the 6500 series interface for DOS with the Sound Sensor: the Oscilloscope allows the student to view the Sound Sensor output directly, and the FFT function will transform the time domain signal from the sensor to a frequency domain display. These two functions allow the student to investigate the frequency composition of sound produced by the human voice, a tuning fork, or loud speaker driven by a complex waveform such as a square wave. Both the Oscilloscope and FFT functions may be used simultaneously. The Sound Sensor can be plugged directly into any PASCO computer interface box or can be connected to the interface box using the supplied cable with 8-pin DIN connectors. SOUND SENSOR CI-6506B aperture for sensor Sound Sensor interface cable with DIN connectors DIN connector to computer interface 012-06296B 2 Sound Sensor ② Open the Experiment Setup window in Science Workshop. Click and drag the analog plug icon to the analog channel icon that matches the analog port you are using for the Sound Sensor (Figure 2). Figure 2. Activating the analog channel in Science Workshop. Setup Procedure A B Plug into any analog channel. Plug into any analog channel. Figure 1. Connecting the Sound Sensor into the computer interface. ③ Select "Sound Sensor" from the drop-down menu (Figure 3). ④ Open a display window, such as the Oscilloscope display, by dragging and dropping the appropriate display icon to the Sound Sensor icon (Figure 4). Figure 3. Setting up the Sound Sensor in Science Workshop. Additional Equipment Suggested:

• Tuning Forks (SE-9325 or SE-9326) • Organ Pipe with Sliding Piston (SF-9328)

• Acoustics Demonstration Disk (SF-9410) ① Connect the Sound Sensor and any Analog channel on the computer interface box with interface cable (Figure 1A), or insert the DIN plug of the Sound Sensor into the jack of any Analog channel on the computer interface box (Figure 1B). The sound level, which is measured in decibels can be calculated from amplitude of the voltage.

The formula is.

B = 10 * log(A^2/A(0)^2)

Where A is the amplitude of the voltage and A(0) is the theoretical amplitude for a sound level of 0 dB. At a frequency of 440 Hz, A(0) = 10 ^ -4 volts approximately. To have Data Studio perform this calculation, click on th Calculate button and type the following equation: sound level = 10*log(amplitude (10,10,.001, v^2)/(10^- 8). In the Variables section of the Calculator, define v as Voltage, Ch A. This equation is optimized for a sine wave at 440 Hz, so you may need to adjust the third argument of the amplitude function (it should be about equal to the period of the wave). The value of 10^-4 volts is also optimized for 440 Hz. Since the response of the sound sensor varies with frequency, you may want to calibrate the equation for a different frequency. To do so, you would need a stand-alone sound level meter. Measure the sound level of as sound (B) with the meter, and the voltage amplitude (A) with the sound sensor. Use these values to solve for A (0) in the first equation above.

ON ANALOG CHANNELS B DIGITAL CHANNELS Science Workshop™ SOUND SENSOR CI-6506B 012-06296B Sound Sensor 3 Calibration and Sensitivity Adjustment in Science Workshop To open the Sound Sensor's calibration window, double-click on the Sound Sensor icon in the Experiment Setup window . The sensitivity (gain) of channels A and B of the 300, 500, and 700 interfaces in Science Workshop can be adjusted by selecting the appropriate level from the drop-down menu .If the Sound Sensor is producing voltage above 1 V, the sensitivity should be set to Low (1x). If the Sound Sensor is producing voltages between +1 V and -1 V, select Medium (10x). With the Science Workshop 700 interface, the sensitivity can be set to High (100x) if the voltage is very low (between +0.1 V and -0.1 V).

4.7.2 Specifications

- Operating voltage 3.3V-5V
- Output model: digital switch outputs (0 and 1, high or low level)
- With a mounting screw hole
- PCB size: 3.4cm * 1.6cm

4.7.3 Schematic Diagram

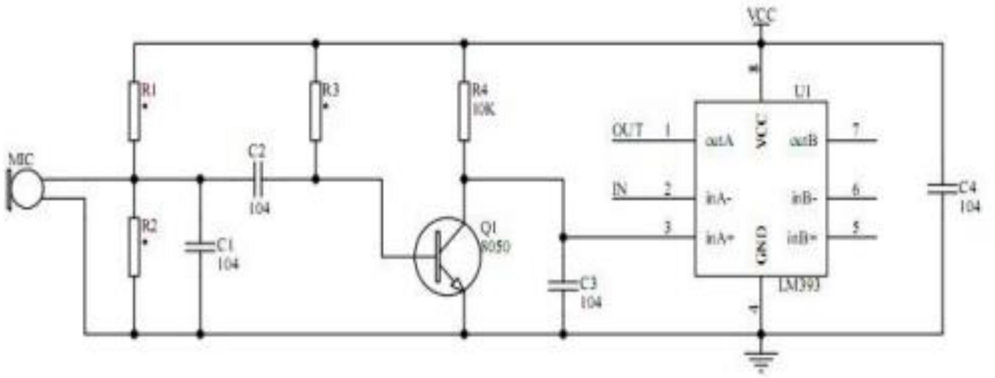


Fig.4.41: Schematic Diagram Of Sound Sensor

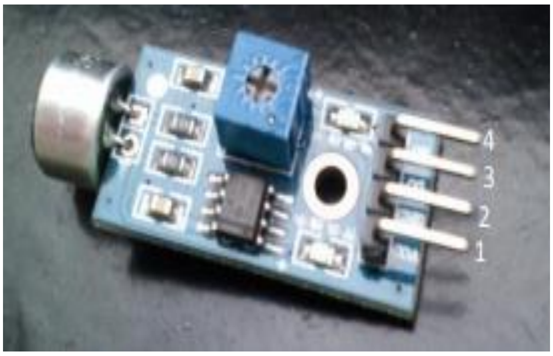


Fig.4.42: Sound Sensor

1. VCC: 3.3V-5V DC
2. GND: Ground
3. DO: Digital output
4. AO: Analog output 1 2 3 4

4.7.4 Wiring Diagram

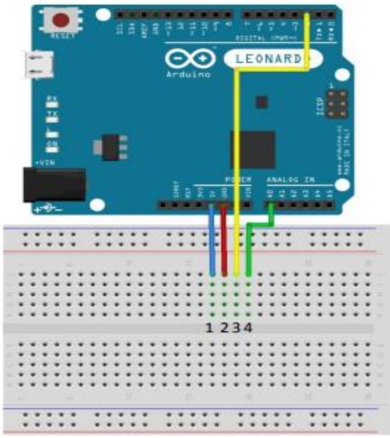


Fig.4.43: Wiring Diagram

```
void setup(){
  Serial.begin(9600);
  pinMode(2, INPUT);
}

void loop()
{
  if(digitalRead(2) == 0) Serial.println("no sound detected");
  else Serial.println("sound detected");
  delay(250);
}
```

Fig.4.44: Simple Code For Sound Detection Using Sound Sensor

4.7.5 How to test

The components to be used are: – Microcontroller (any compatible arduino) – Sound sensor module – 1 Pin M-M connectors – Breadboard – USB cable 1. Connect the components based on the figure shown in the wiring diagram using a M-M pin connector.

VCC pin is connected to the 3.3V or 5V power supply, GND pin is connected to the GND, DO pin is connected to a digital I/O pin and AO pin is connected to an analog pin. Pin number will be based on the actual program code. 2. After hardware connection, insert the sample sketch into the Arduino IDE. 3. Using a USB cable, connect the ports from the microcontroller to the computer. 4. Upload the program. 5. See the results in the serial monitor.

4.7.6 Testing Results

The figure below shows when the module was not subjected to sound.

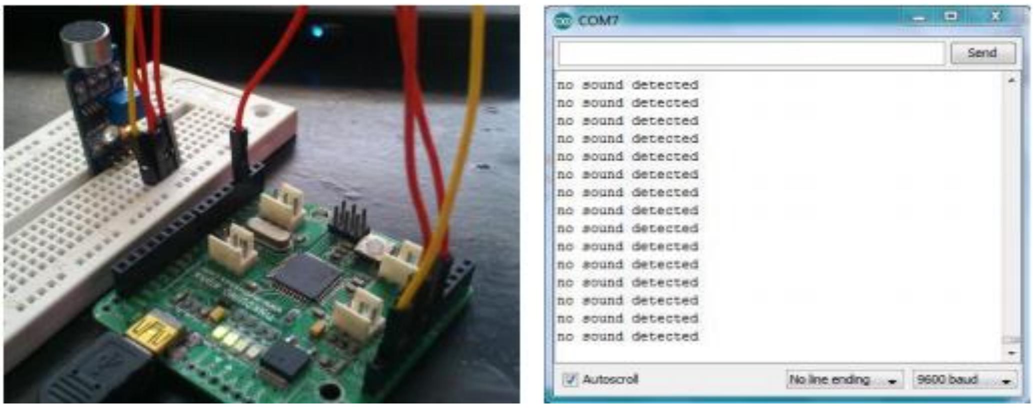


Fig.4.45: Output When Sound Is Not Detected

The figure below shows when the module was subjected to sound. Note that the red LED should also light up when sound is detected

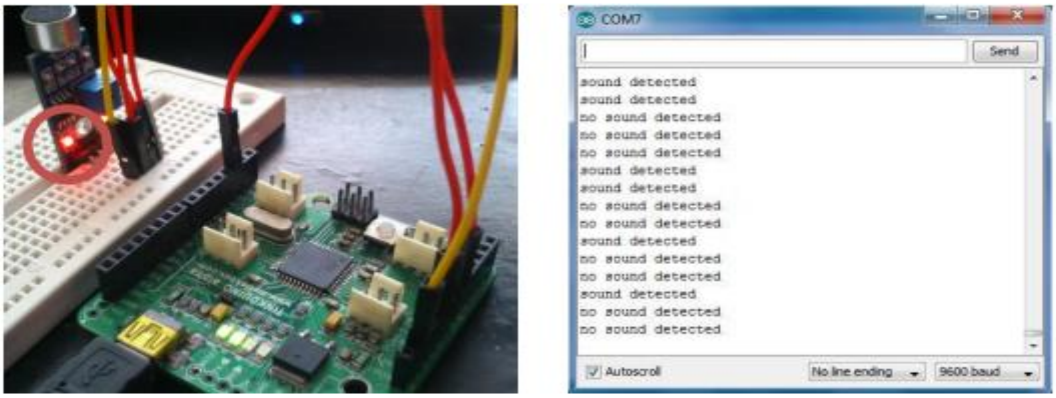


Fig.4.46: Output When Sound Is Detected

4.8 SOFTWARE EXPLANATION:

- **Software Requirements**
  - Proteus simulation
  - Arduino software
  - Programming language

4.8.1 Arduino software:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IOT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

4.8.2 Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux.

Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments.

Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handy board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version

- Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money
- Getting Started with Arduino and Genuino products:-

Install the Arduino Software (IDE) on Windows PCs-

…This document explains how to install the Arduino Software (IDE) on Windows machines.

- Download the Arduino Software(IDE)
- Proceed with board specific instructions.

4.8.3 How to Download the Arduino Software (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

- **Installation:**

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1

First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.4.47: (a) Standard USB Cable

Fig.4.47: (b) USB cable A to Mini-B

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

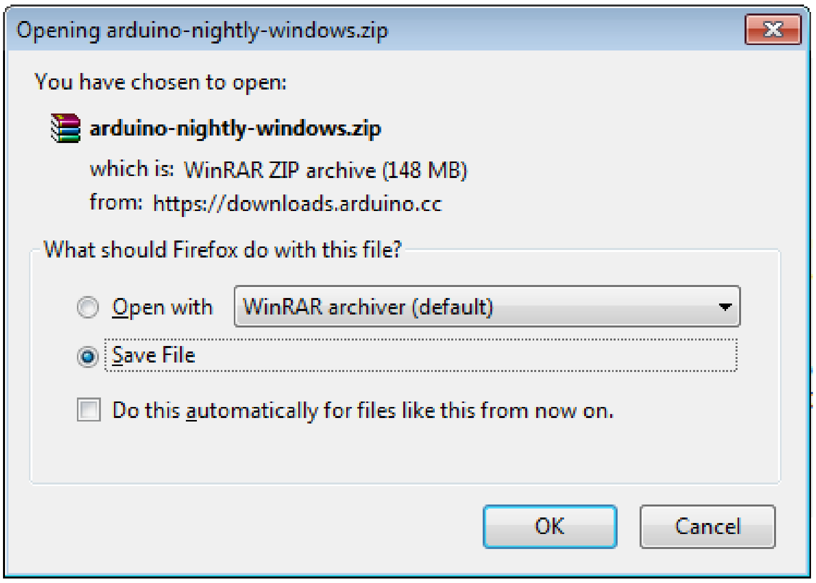


Fig.4.48: Process of Downloading the Arduino

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply.

If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe).

Double-click the icon to start the IDE.

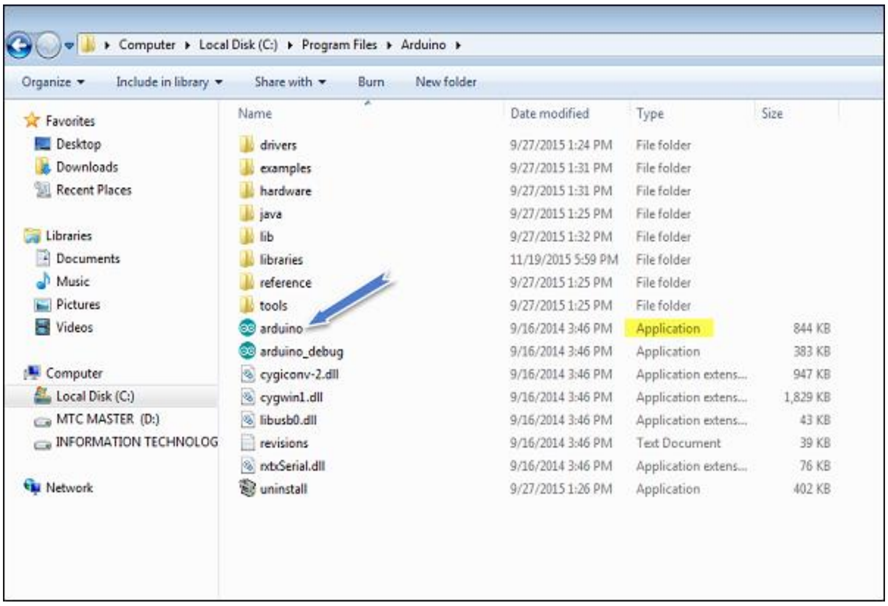


Fig.4.49: Launching Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.

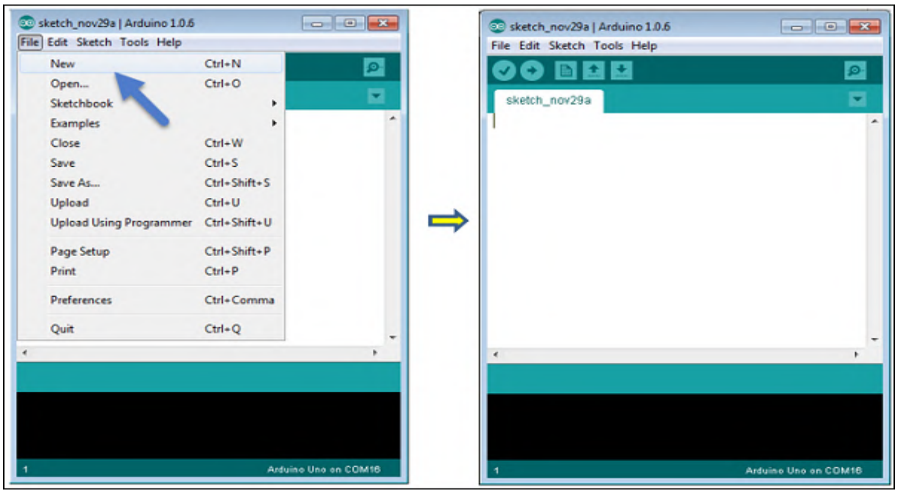


Fig.4.50: Opening Editor Window in IDE

To open an existing project example, select File → Example → Basics → Blink.

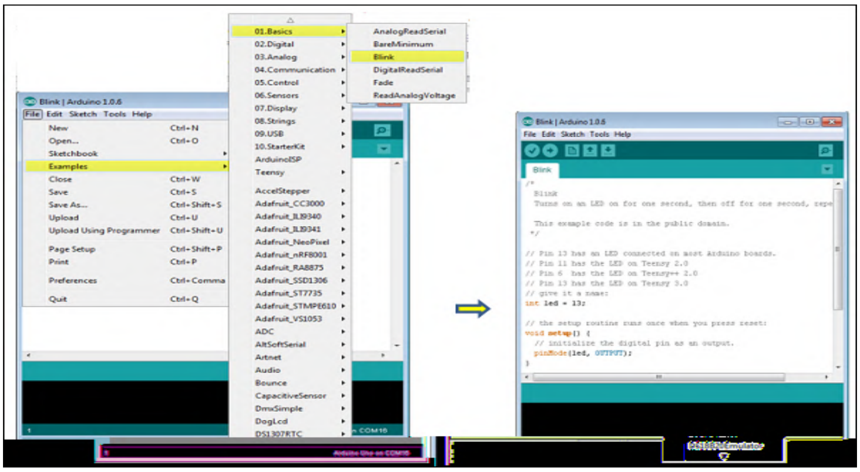


Fig.4.51: Opening Existing File in IDE

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

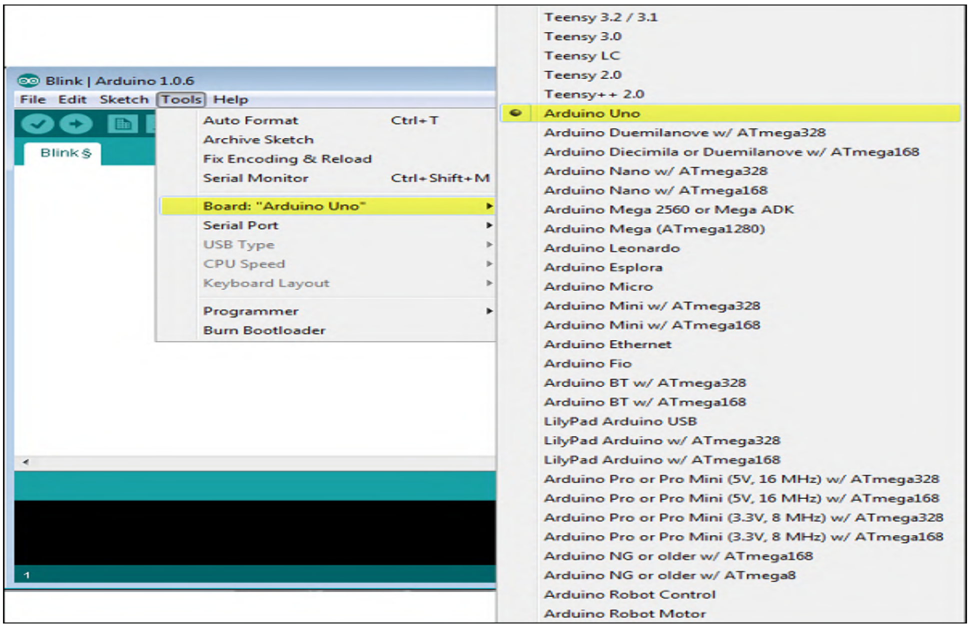


Fig.4.52: Selecting Arduino Board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports).

To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

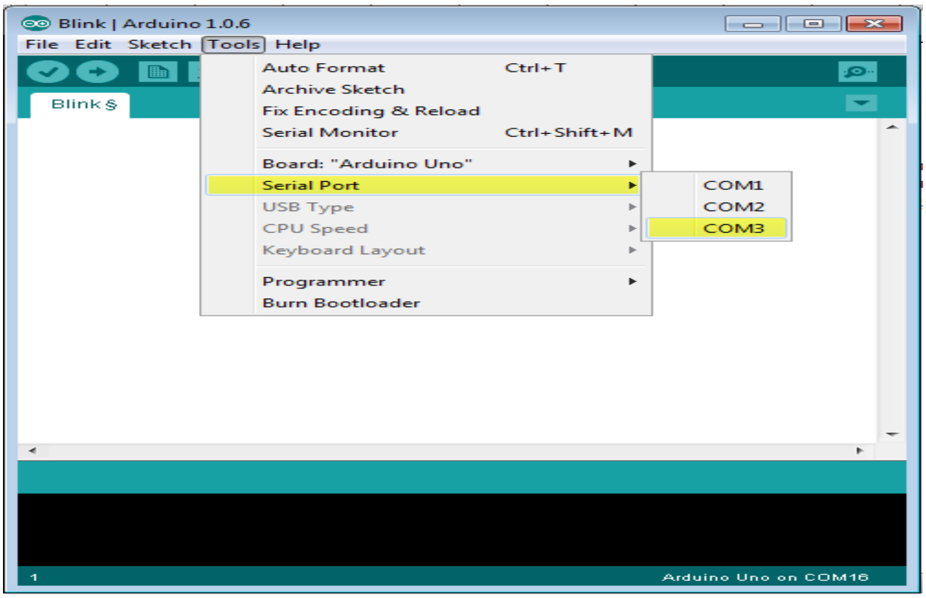


Fig.4.53: Selecting Serial Port

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

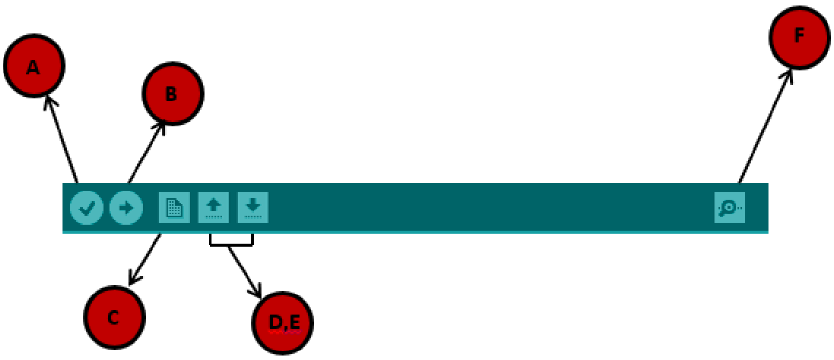


Fig.4.54: IDE Tool Bar

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

4.8.4 Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24 V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

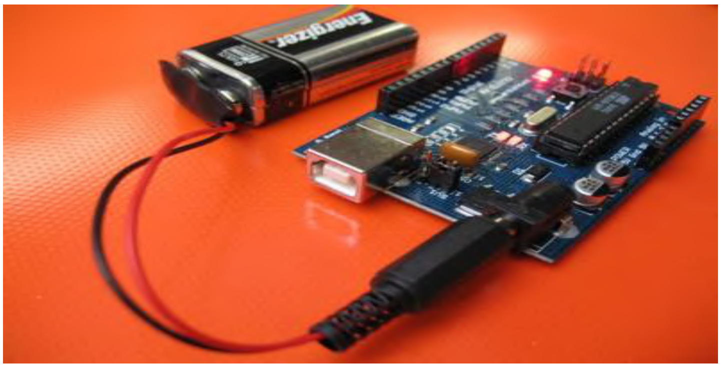


Fig.4.55: Arduino with Battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called “sketches”, but here we will just call them programs.

In the editing window that comes up, enter the following program show in Figure:4.66, paying attention to where semi-colons appear at the end of command lines.

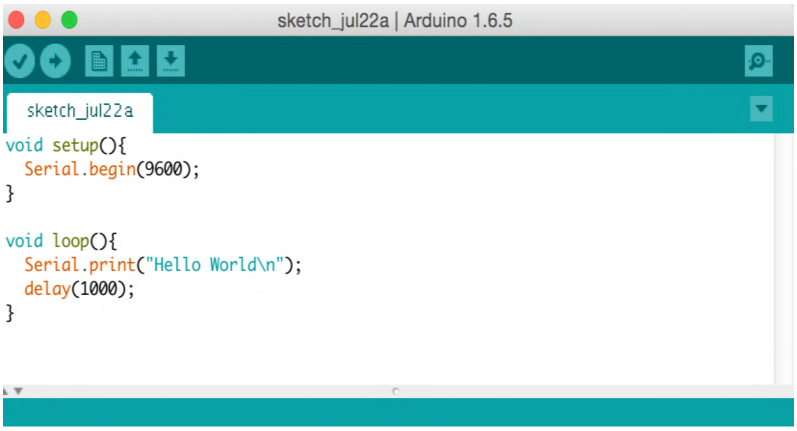


Fig.4.56: Writing the Program in IDE

4.9 MC Programming Language: Embedded C

This is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements. Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems.

There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

4.10 Proteus:

4.10.1 Proteus:

Proteus is a simulation and design software tool developed by Lab Centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

4.10.2 About Proteus

It is a software suite containing schematic, simulation as well as PCB designing.ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

4.10.3 Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components.

It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

4.10.4 Starting New Design

Step 1: Open ISIS software and select new design in File menu

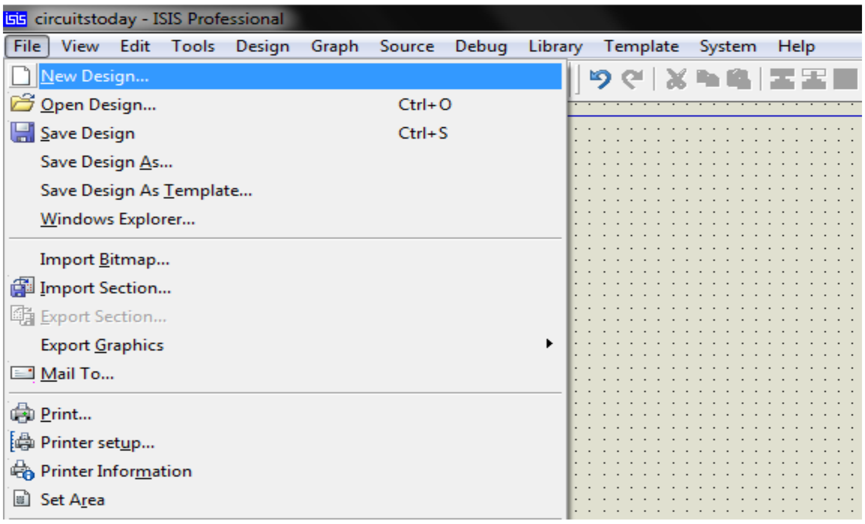


Fig.4.57: Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

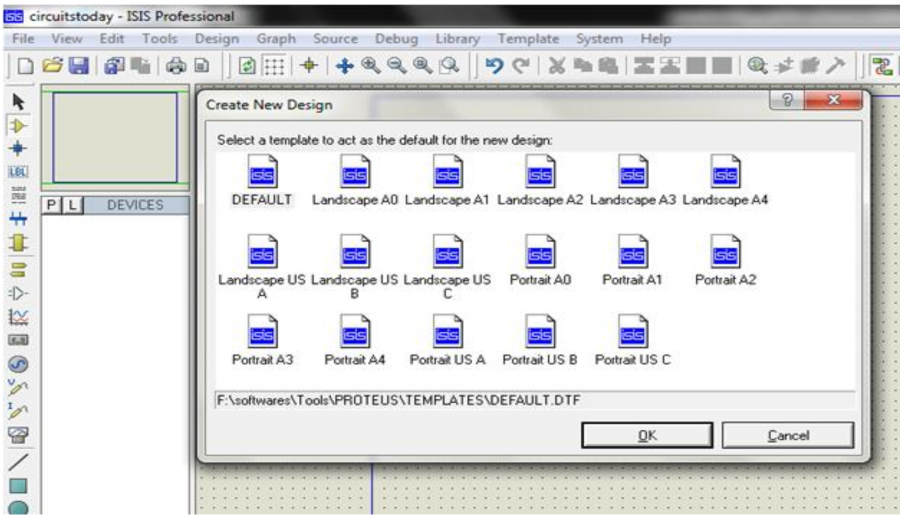


Fig.4.58: Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

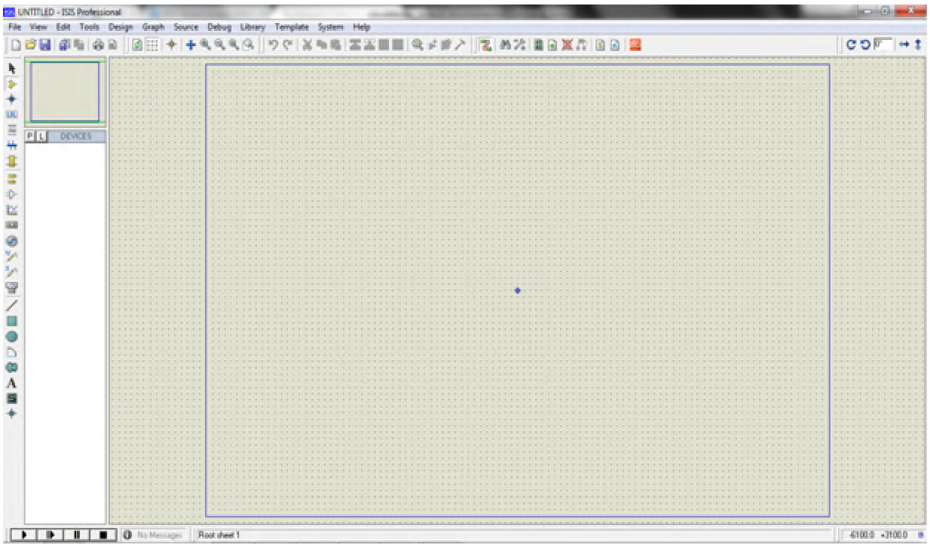


Fig.4.59: Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

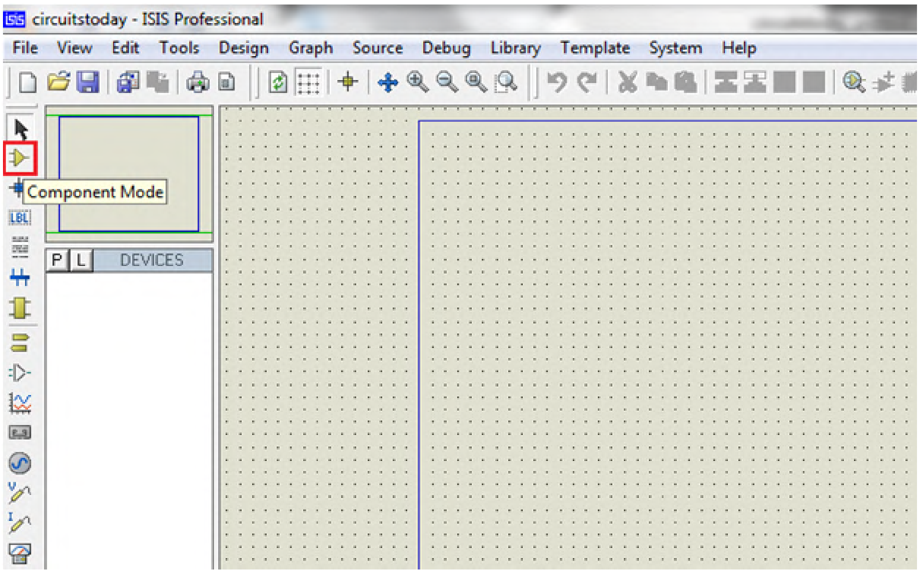


Fig.4.60: Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

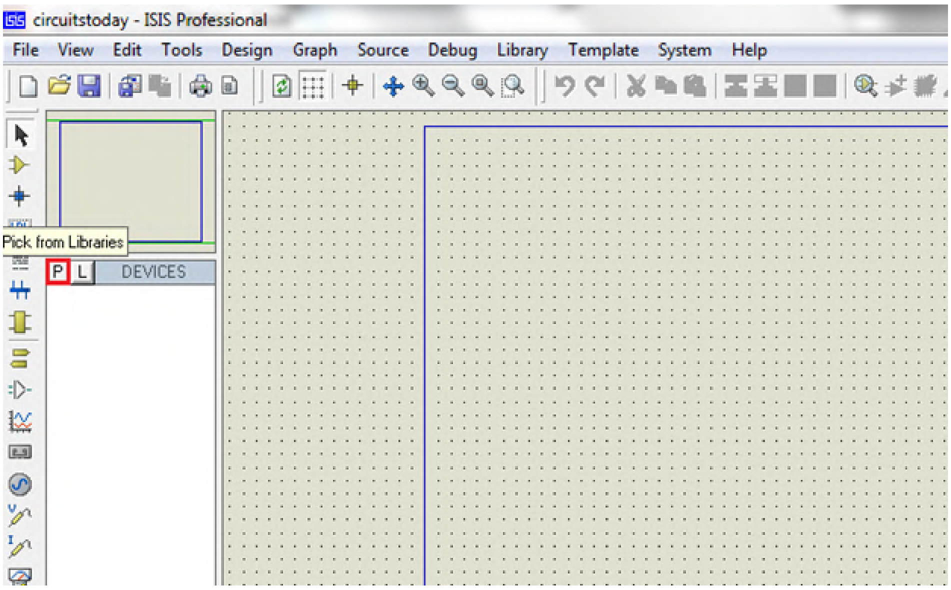


Fig.4.61: Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

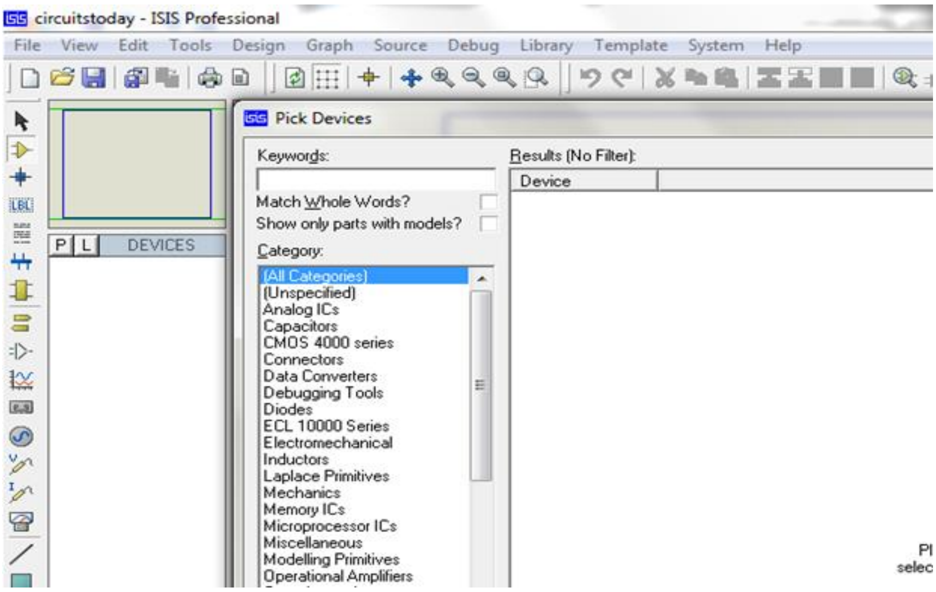


Fig.4.62: Keywords Textbox

Example shows selection of push button. Select the components accordingly.

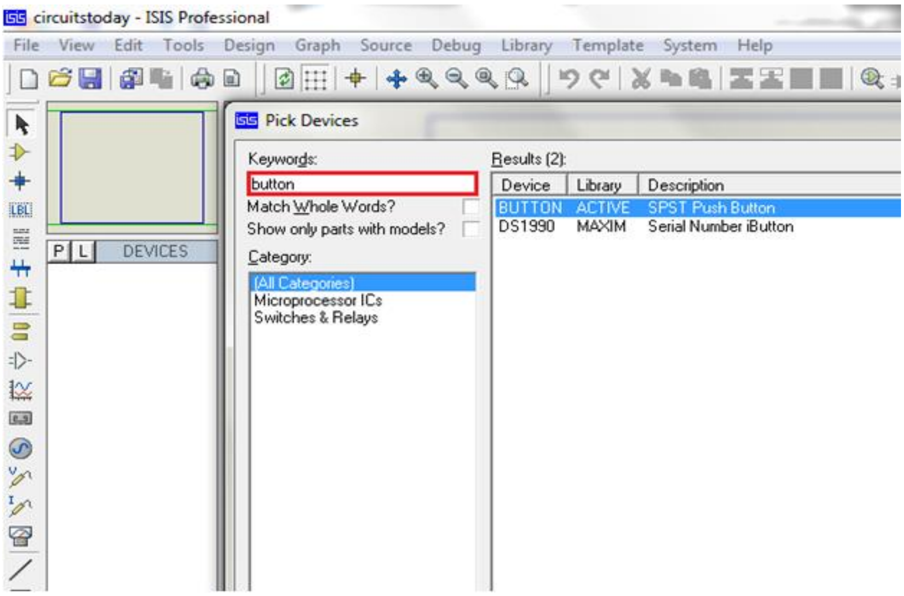


Fig.4.63: Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

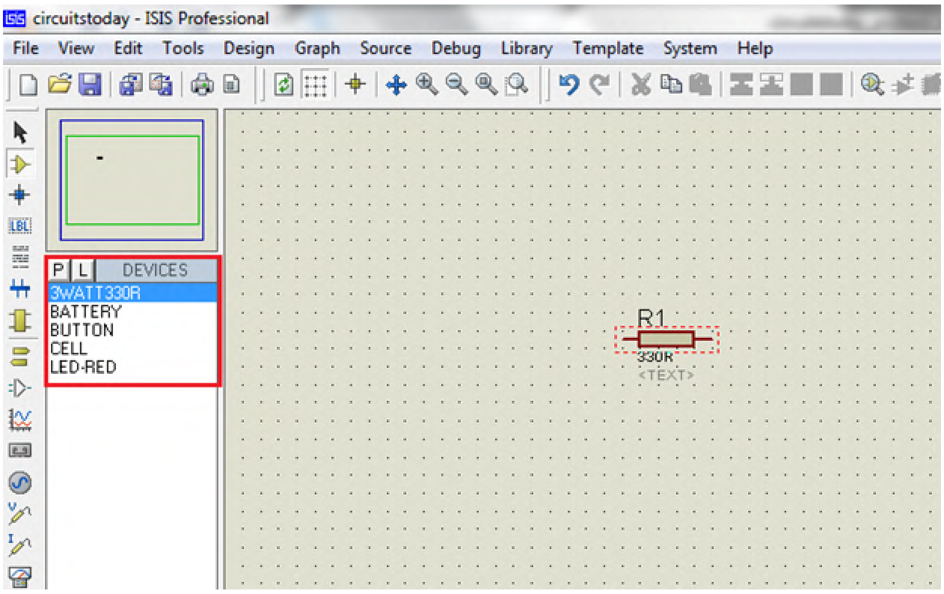


Fig.4.64: Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

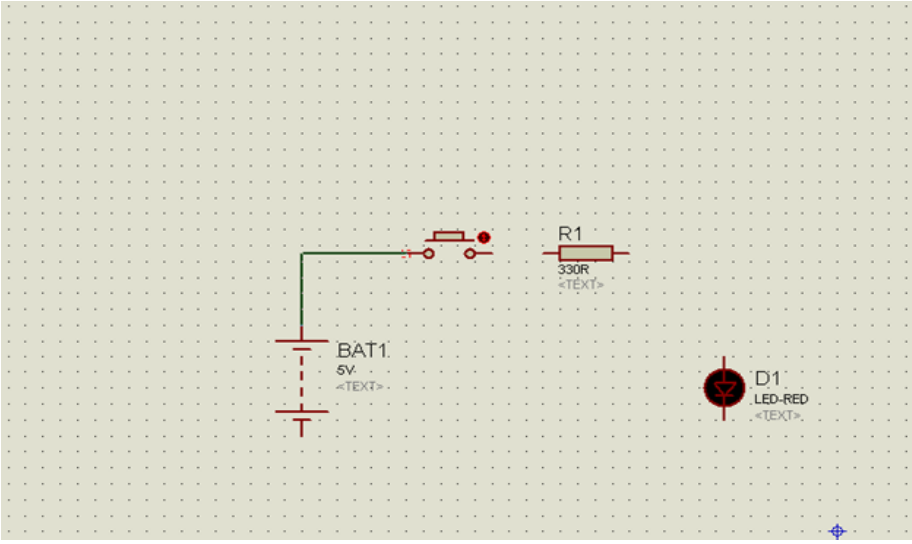


Fig.4.65: Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

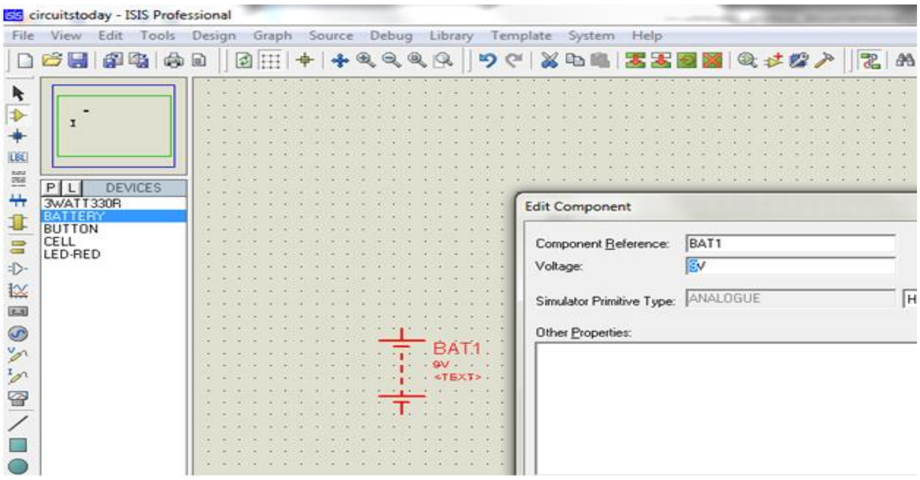


Fig.4.66: Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

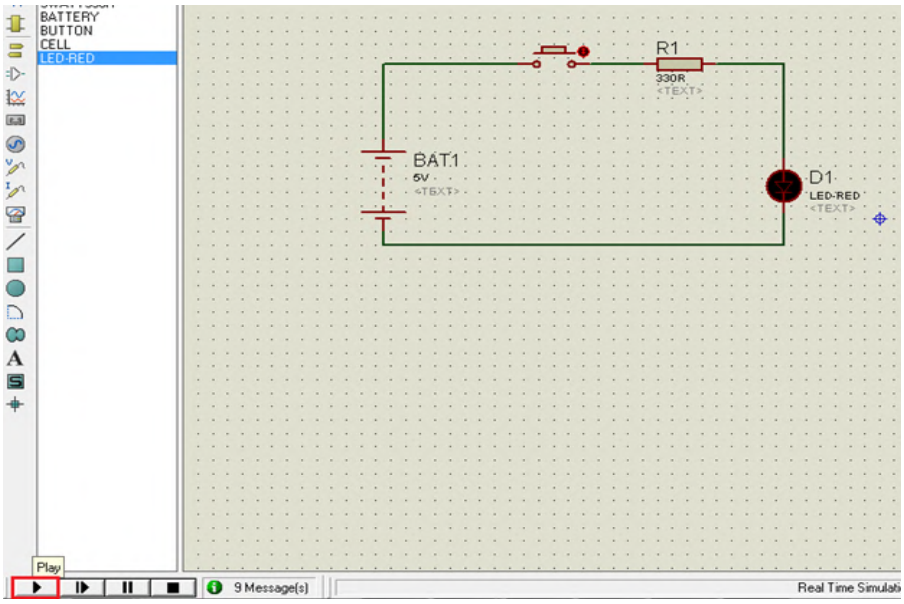


Fig.4.67: Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

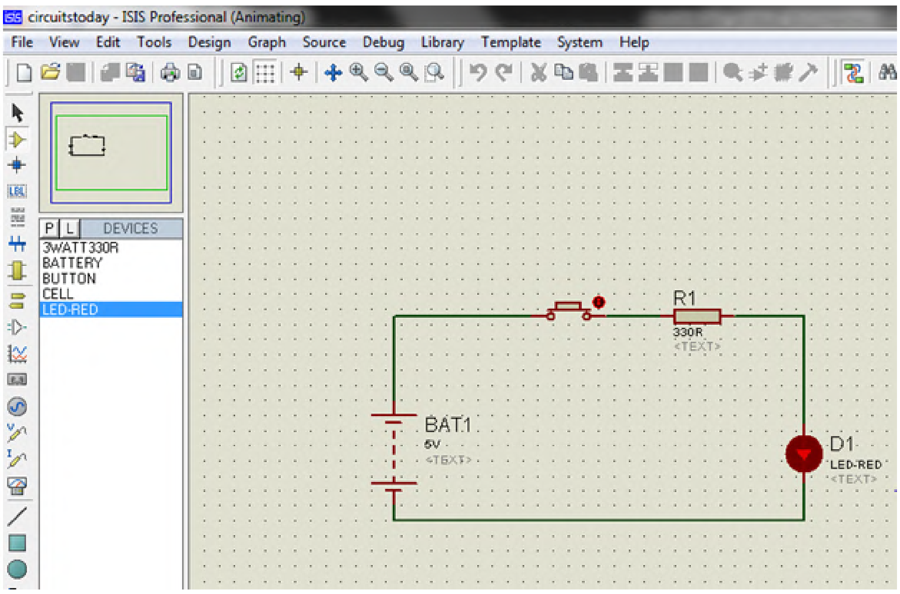


Fig.4.68: Simulation Animating

Simulation can be stepped, paused or stopped at any time.

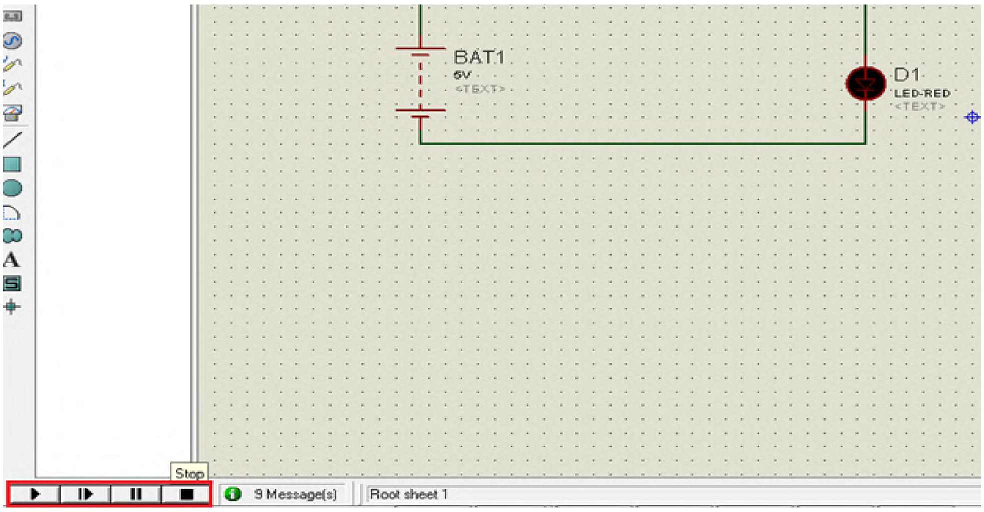


Fig.4.69: Simulation Step-Pause-Stop Buttons

4.10.4 Project Code:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(13,12,11,10,9,8);
int sw1=4,sw2=5,sw3=6,sw4=7;
int v1=A4,v2=A5;

void setup()
{
  Serial.begin(9600);

  pinMode(sw1,OUTPUT);
  pinMode(sw2,OUTPUT);
  pinMode(sw3,OUTPUT);
  pinMode(sw4,OUTPUT);
  pinMode(v1,INPUT);
  pinMode(v2,INPUT);
```

```
   digitalWrite(sw1,LOW);
   digitalWrite(sw2,LOW);
   digitalWrite(sw3,LOW);
   digitalWrite(sw4,LOW);

   lcd.begin(16,2);
   lcd.clear();
   lcd.setCursor(0,0);
   lcd.print("***WELCOME***");

   Serial.begin(9600);
   delay(3000);
}

void loop() {
    int v1d = digitalRead(v1);
    int v2d = digitalRead(v2);

    lcd.clear();lcd.setCursor(0,1);lcd.print("S1:");lcd.print(v1d);
    lcd.setCursor(8,1);lcd.print("S2:");lcd.print(v2d);delay(2000);

 if (v1d == LOW)
   {
   lcd.clear();lcd.setCursor(0,1);lcd.print("LOCATION 1: ALERT");
   digitalWrite(sw1,HIGH);digitalWrite(sw3,HIGH);
   digitalWrite(sw2,LOW);digitalWrite(sw4,LOW);
   Serial.print("*1");
   delay(2000);
   digitalWrite(sw1,LOW);digitalWrite(sw3,LOW);
   digitalWrite(sw2,LOW);digitalWrite(sw4,LOW);
   }
```

```
else if (v2d == LOW)
   {
   lcd.clear();lcd.setCursor(0,1);lcd.print("LOCATION 2: ALERT");
   digitalWrite(sw2,HIGH);digitalWrite(sw4,HIGH);
   digitalWrite(sw1,LOW);digitalWrite(sw3,LOW);
   Serial.print("*1");
   delay(2000);
   digitalWrite(sw1,LOW);digitalWrite(sw3,LOW);
   digitalWrite(sw2,LOW);digitalWrite(sw4,LOW);
   }

else
   {
   lcd.clear();lcd.setCursor(0,1);lcd.print("EVRYTHING FINE...");
   digitalWrite(sw1,LOW);digitalWrite(sw3,LOW);
   digitalWrite(sw2,LOW);digitalWrite(sw4,LOW);
   delay(1000);
   }

}
```

CHAPTER-5

RESULTS & OUTPUTS

5.1 RESULT:

The outcome of the project "women safety patrolling robot using IOT " after aligning all components is shown in figure.5.1 and figure.5.2, all the components are placed over the robot chassis, a caster wheel is placed for comfortable movements in any directions.

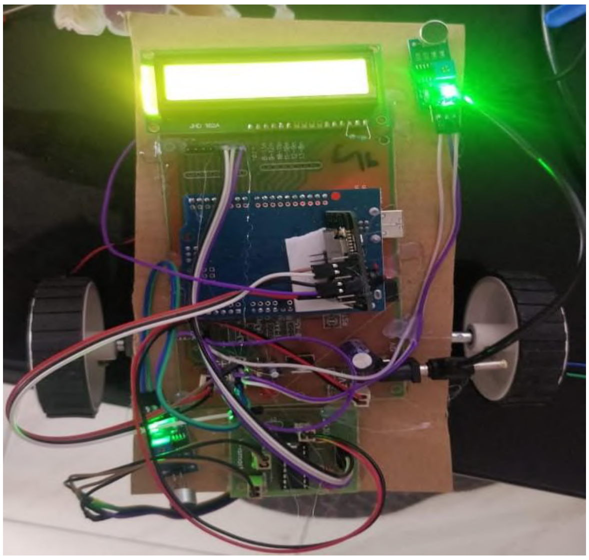


Fig.5.1: Women Safety Patrolling Robot Top View

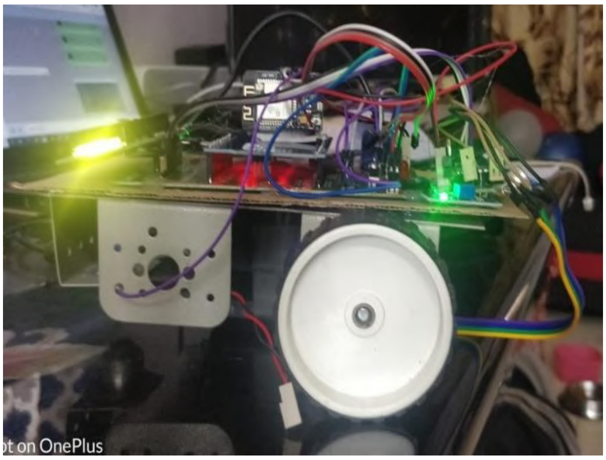


Fig.5.2: Women Safety Patrolling Robot Side View

Initially, the robot is set to ON by using a power supply unit of a 12Volts battery and regulated to a voltage of 5 to power the Arduino

Initially, the threshold detection level of the sound sensor is set to an intensity level of 540 by rotating the potentiometer of the sound sensor, and ESP32-CAM which is of 2MP 150 degrees wide-angle view is adjusted properly to cover the area properly and it is set to work properly.

After the kit is ON, ESP32-CAM continuously monitors the robot premises and sends the live streaming to the server which is observed in figure 5.3.as well as a message "Everything is Fine" is displayed on LCD, when no sound is detected and is shown in figure 5.4

The outputs of the projects are shown below.

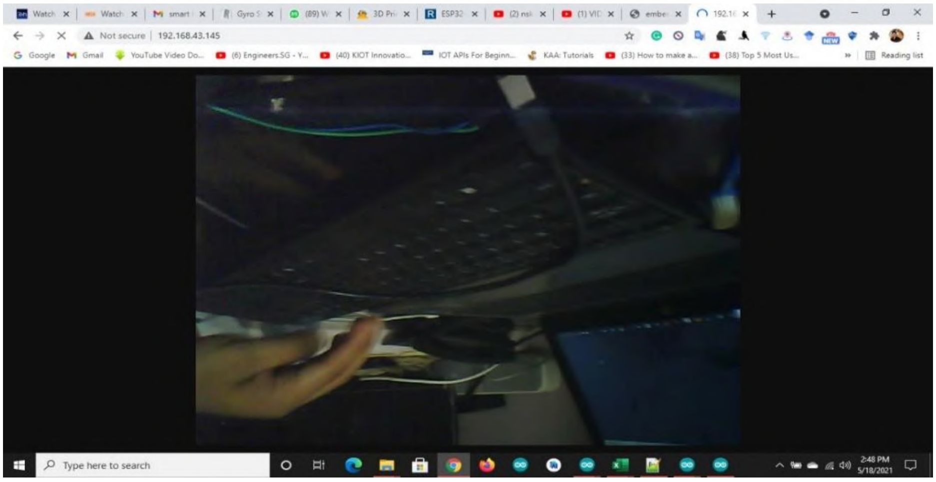


Fig.5.3: Screenshot Of Live Video Monitoring With ESP32 Camera Module Over Internet

The live streaming of robot premises can be observed by entering the IP address provided to the ESP32-CAM in the IOT web server, here we used “Embeddedspot.top” website for monitoring.

Fig.5.4: Displays When No Sound Is Detected

Fig.5.5: Displays If Sound Sensor1 Detects The Sound

Fig.5.6: Displays If Sound Sensor2 Is Detects The Sound

Whenever a Sound is detected in a range of 0.5meters, and if sound sensor-1 detects the sound, the output of the sound sensor becomes low and it is monitored by Arduino displays a message of "Location Alert: 1" on LCD and sets the output pins sw1 and sw2 to high such that robot moves left, similarly with the sound sensor-2 the LCD messages can be seen in Figures 5.5 and 5.6.

CHAPTER-6

APPLICATIONS, ADVANTAGES AND LIMITATIONS

6.1 APPLICATIONS OF THE PROJECT:

- By joining camera highlights with the robot we can undoubtedly screen indoor just as outside areas during daytime and around evening time.
- Remote regions can likewise be investigated.
- Used to record and send video yield of the necessary condition, expels human nearness of observing.
- Used for the safety of children.
- Used for the safety of physically challenged people
- Used as a legal evidence of crime with exact location information for prosecution

6.2 ADVANTAGES OF THE PROJECT:

- Easy and fast to install- These system will be easy to handle.
- Low cost with high performance- The device will be in a low cost which will work with a good performance.
- Environmental friendly system- The system will not harmful for the surroundings.
- Involvement of human force is reduced up to some extent.
- Patrolling is done autonomously
- It has a scope to catch the culprits as it works like movable CCTV

6.3 LIMITATIONS:

- Need a Human to Monitor the Video and Location
- Limited to local server.
- Obstacles in the path can be distractors.

CHAPTER-7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION:

Patrolling Robots are designed to help women from the dangerous situations. Women can rescue and protect themselves in any circumstances, so that she will never feel helpless at any sort locale and can protect herself even at late night. These techniques will also help police to arrest and search for the culprits.

7.2 FUTURE SCOPE:

- One can implement a few more sensors and connect it to the satellite as a global feature of this system.
- Adding more sensor to monitor other environmental parameters such as CO2,Pressure and Oxygen Sensor
- In aircraft, navigation and military there is a great scope of this real-time system.
- One can add GPS tracking and face detection algorithms to efficiently use the robot.
- It can also be implemented in hospitals or medical institutes for the research & study in “Effect of Weather on Health and Diseases”, hence to provide better precaution alerts.

REFERENCES

[1] “Review of human detection techniques in night vision” in 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)

[2] Eun Som Jeon et al., "Human detection based on the generation of a background image by using a far-infrared light camera", Sensors, vol. 15, no. 3, pp. 6763-6788, 2015.

[3] Dushyant Kumar Singh, Dharmender Singh Kushwaha, "automatic intruder combat system: a way to smart border surveillance", Defence Science Journal, vol. 67, no. 1, pp

[4] Vigneshwari S, Aramudhan M. Social information retrieval based on semantic annotation and hashing upon the multiple ontologies. Indian Journal of Science and Technology.2015Jan; 8(2):103–7.

[5] Chand D, Nayak S, Bhat KS, Parikh S. A mobile application for Women’s Safety:

WoS App. 2015 IEEE Region 10 Conference TENCON; Macao. 2015 Nov 1-4. p. 1–5.

[6] Sethuraman R, Sasiprabha T, Sandhya A. An effective QoS based web service composition algorithm for integration of travel and tourism resources. Procedia Computer Science.2015; 48:541–7.

[7] Gowri S, Anandha Mala GS. Eicacious IR system for investigation in textual data. Indian Journal of Science and Technology. 2015 Jun; 8(12):1–7.

[8] George R, Anjaly Cherian V, Antony A, et al. An intelligent security system for violence against women in public places.

[9] Chand D, Nayak S, Bhat KS, Parikh S. A mobile application for Women’s Safety: WoS App. 2015 IEEE Region 10 Conference TENCON; Macao. 2015 Nov 1-4. p. 1–5.

[10] Arias, Orlando, et al. "Privacy and security in internet of things and wearable devices."IEEE Transactions on Multi-Scale Computing Systems1.2 (2015): 99-109.

[11] Gadhave, Swapnali N., et al. "Electronic Jacket For Women Safety."IRJET(2017).

[12]George, Remya, et al. "An intelligent security system for violence against women in public places."International Journal of Engineering and Advanced Technology3.4

(2014).

[13]Helen, A., et al. "A smart watch for women security based on iot concept „watch me‟."2017 2nd International Conference on Computing and Communications Technologies (ICCCT). IEEE, 2017.

[14] .Hyndavi, V., N. Sai Nikhita, and S. Rakesh. "Smart wearable device for women safety using IoT."2020 5th International Conference on Communication and Electronics Systems (ICCES). IEEE, 2020.

[15] Jain, R. A., et al. "Women‟s safety using IOT"International Research Journal of Engineering and Technology (IRJET)4.05 (2017): 2336-2338.

[16] Miriyala, Geetha Pratyusha, et al. "Smart intelligent security system forwomen."International Journal of Electronics and Communication Engineering & Technology (IJECET)7.2 (2016): 41-46.

[16] More, S. A., et al. "Smart band for women security based on internet of things (IOT)."International journal of research in science and engineering 6 (2017): 840-845.

[17] Nguyen, Hung, et al. "Using inertial sensors to automatically detect and segment activities of daily living in people with Parkinson‟s disease."IEEE Transactions on Neural Systems and Rehabilitation Engineering26.1 (2017): 197-204.

[18] Paradkar, Abhijit, and Deepak Sharma. "All in one intelligent safety system for women security."International journal of computer applications130.11 (2015): 33-40.

[19] Sharma, S., Ayaz, F., Sharma, R., Jain, D. and Student, B.E. “IoT Based Women Safety Device using ARM7.”IJESC,7(5) (2017): 11465-11466.

A

MAJOR PROJECT REPORT

on

FINGERPRINT BASED EXAMHALL AUTENTICATION

Submitted by

1) Ms. B. Reethika (17K81A04G9)
2) Ms. G. Samyuktha Reddy (17K81A04E6)
3) Ms. V. Shivani reddy (17K81A04H5)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

G.Vinatha

Asst.Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500100

NBA &NAAC A+ Accredited

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE
UGC AUTONOMOUS
Dhulapally, Secunderabad-500 100
NBA &NAAC A+ Accredited

BONAFIDE CERTIFICATE

This is to certify that the project entitled "Recursive Block Based Key Point Matching Copy Move Image Forgery Detection" is being submitted by **1.Ms. B. Reethika 17K81A04G9, 2.Ms. G. Samyuktha Reddy 17K81A04E6 3.Ms. V. Shivani Reddy 17K81A04H5** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodiedin this report have been verified and found satisfactory.

Internal Guide	**Head of the Department**
G.Vinatha Department of Electronics and Communication Engineering	Dr. Hari Krishna Department of Electronics and Communication Engineering

Internal Examiner	**External Examiner**

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT B REETHIKA WITH ROLL NO.17K81A04G9, G SAMYUKTHA REDDY WITH ROLL NO.17K81A04E6, V SHIVANI REDDY WITH ROLL NO.17K81A04H5, OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "FINGERPRINT BASED EXAMHALL AUTENTICATION" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014

Email : contact@lasyainfotech.com, ov@lasyainfotech.com

Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of ‘**Bachelor of Technology in Department of Electronics and Communication Engineering**’, session: 2017-2021, St. Martin’s Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project work entitled ‘**Fingerprint Based Exam Hall Authentication**’ is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

B.Reethika	G.Samyuktha Reddy	V.Shivani Reddy
17K81A04G9	17K81A04E6	17K81A04H5

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success**.**

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr. P. Santosh Kumar Patra**, our beloved **Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B. Hari Krishna**, **Head & Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our department **Project Coordinator, Dr. A. Anand Kumar, Professor** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our **Class Project Coordinator, Mr. K. Nishakar,** Department of ECE and our project **Internal guide, Mrs. G. Vinatha, Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level **Project Coordinators Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. B. Reethika
2. G.Samyuktha Reddy
3. V.Shivani Reddy

ABSTRACT

Authentication has always been a major challenge in all types of examination. Verification of the authentic candidate is not an easy task, and also it consumes a lot of time and process. This led to the design of Fingerprint based exam hall authentication system that is designed to pass only users verified by their fingerprint scan and block non verified users. Formal examination can rightly be defined as the assessment of a person‘s Performance, when confronted with a series of questions, problems, or tasks set him/her, in order to ascertain the amount of knowledge that he has acquired, the extent to which he is able to utilize it, or the quality and effectiveness of the skills he/she has developed. During the 19th century, formal written examinations became regular in universities, schools, and other educational institutions. Examinations were also increasingly employed for the selection of recruits to the civil service, and the professions, and to posts in industry and commence.

Over the ages, standardized testing has been the most common methodology, yet the validity and credibility of the expanded range of contemporary assessment techniques have been called into question. There are two types of systems that help automatically establish the identity of a person: 1) Authentication (verification) systems and 2) Identification systems. In a verification system, a person desired to be identified submits an identity claim to the system, usually via a magnetic stripe card, login name, smart card, etc., and the system either rejects or accepts the submitted claim of identity (Am I who I claim I am?). In an identification system, the system establishes a subject‘s identity (or fails if the subject is not enrolled in the system database) without the subject‘s having to claim an identity (Who am I?). The topic of this paper is channel towards the development of examination impersonation elimination system and this system would strictly do with the unique feature of identification by means of finger print. A verification system based on fingerprints, and the terms verification, authentication, and identification are used in a loose sense and synonymously.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.0 OVERVEIW OF THE PROJECT:

Biometrics is the science and technology of measuring and analyzing biological data. Biometrics refers to technologies that measure and analyze human body characteristics, such as DNA, fingerprints, eye retinas and irises, voice patterns, facial patterns and hand measurements, for authentication purposes. The field of biometrics was formed and has since expanded on to many types of physical identification. Among the several human fingerprints remain a very common identifier and the biometric method of choice among law enforcement. These concepts of human identification have lead to the development of fingerprint scanners that serve to quickly identify individuals and assign access privileges. The basic point of these devices is also to examine the fingerprint data of an individual and compare it to a database of other fingerprints . In our project we have used fingerprint for the purpose of voter identification or authentication. As the thumb impression of every individual is unique, it helps in minimizing the error. A database is created containing the fingerprint images of all the voters as required. Illegal votes and repetition of votes is checked for in this system with accurate coding. Hence with the application of this fingerprint based EVM system elections could be made fair and free from rigging. Further that the elections would are no longer a tedious and expensive job.

1.1 OBJECTIVES OF STUDY:

Fingerprint based security system is the most secured system as compared to other systems. Reason is that RFID card or Keys of lock can be stolen, password may be leaked. However, thumbnail of every human being is unique, so lock will not open unless the same person is present to give the impression of fingerprint. This system was been designed to control thefts in private areas and to provide more security and authenticity to the users, as we are using finger prints for identifying the persons this was one of the most efficient way to identify people because the fingerprints will acts as the unique IDs for persons.

The development steps or method that outlines the way to achieve certain goal or in this case design and build a prototype Fingerprint based exam hall authentication. In other words, it is a constructive framework. Below in figure , methodology flow block diagram of this project is shown.

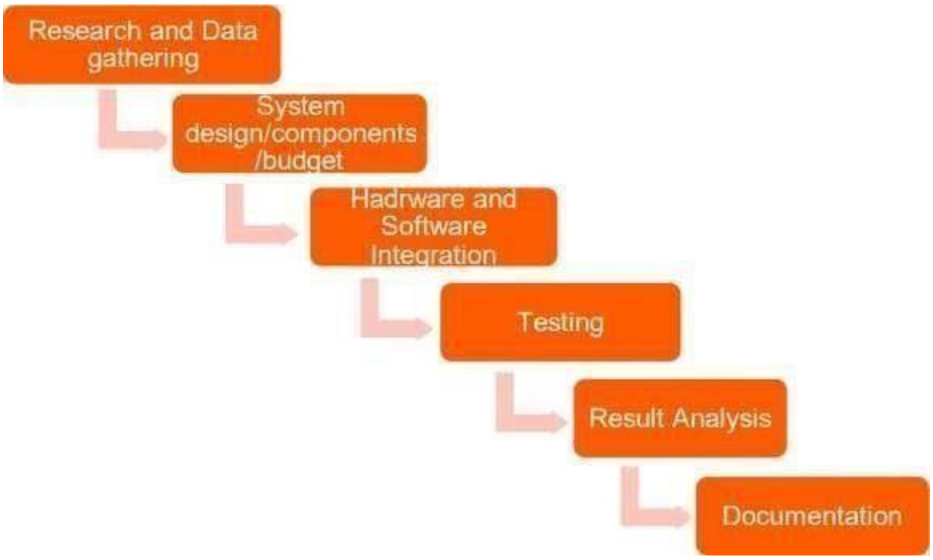


Fig 1.1: Flow diagram methodology

1.2 SCOPE OF THE STUDY:

The world is going through a global crisis due to the contemporary COVID 19 pandemic situations. In these taxing times, students have been severely affected. So our government has not compromised their future and has decided to conduct all competitive examinations. Thus, it becomes crucial for the students to register for the exams. However, the traditional exam hall authentication system does not assure that the students would not be exposed to the infection. So it is important to come up with a solution for this scenario. This paper presents a model in which the student's registration and authentication happen using the fingerprint sensor and Atmega328 microcontroller in addition to recording the student's body temperature by the LM35 temperature sensor. The data is transferred to a cloud server where it can be monitored and analyzed. The system integrates different hardware components like microcontroller, fingerprint module, LEDs, switches that facilitates a flawless authentication. For the implementation of this system, DY50 fingerprint sensor is used to take user finger print image and store in internal memory, these images are further processed and analyzed using Arduino. The user interface is implemented using LCD screen, which is mainly used to print user instructions during the execution of the process and the result.

1.3 MATERIALS REQUIREMENT:

1.3.1 HARDWARE COMPONENTS:

- ARDUINO UNO (ATMEGA328) MICROCONTROLLER BOARD

Fig:1.2 Arduino Uno board

- FINGER PRINT SCANNER ALONG WITH 3 PUSHBUTTONS

Fig:1.3 Fingerprint scanner

- LCD DISPLAY (16*2)

Fig:1.4 LCD Display

- POTENTIOMETER

Fig:1.5 POTENTIOMETER

- MB27L-12V PIEZO BUZZER

Fig:1.6 Piezo buzzer

1.3.2 SOFTWARE REQUIREMENTS:

1.3.2.1 ARDUINO IDE SOFTWARE:

This software is basically used for giving commands to ARDUNO Microcontroller**.**

STEPS FOR INSTALLING ARDUNO IDE SOFTWARE:

1. Download Arduino IDE Software in your PC.

2. When the download finishes, proceed with the installation.

3. Allow the driver installation process when you get a warning from the operating system.

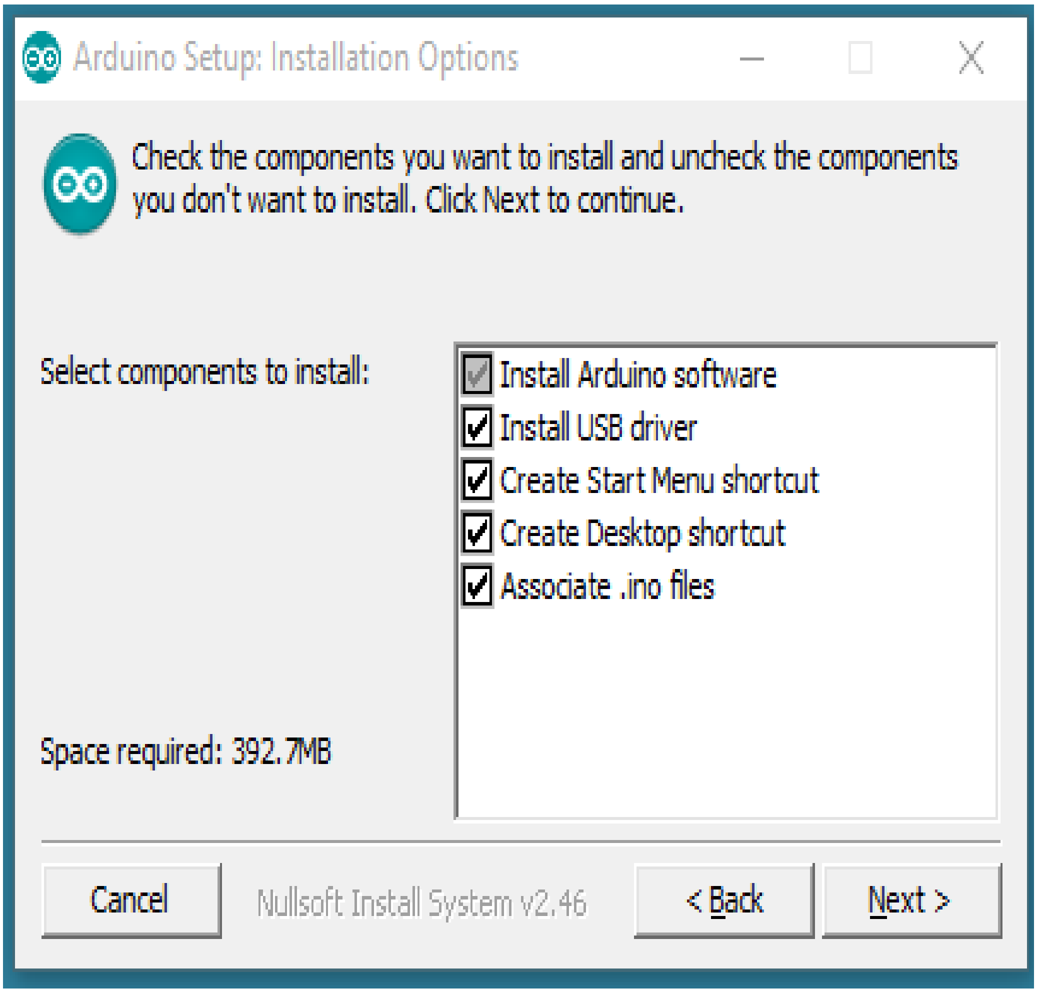


Fig:1.7 Installation process of driver

4. Choose the components to install

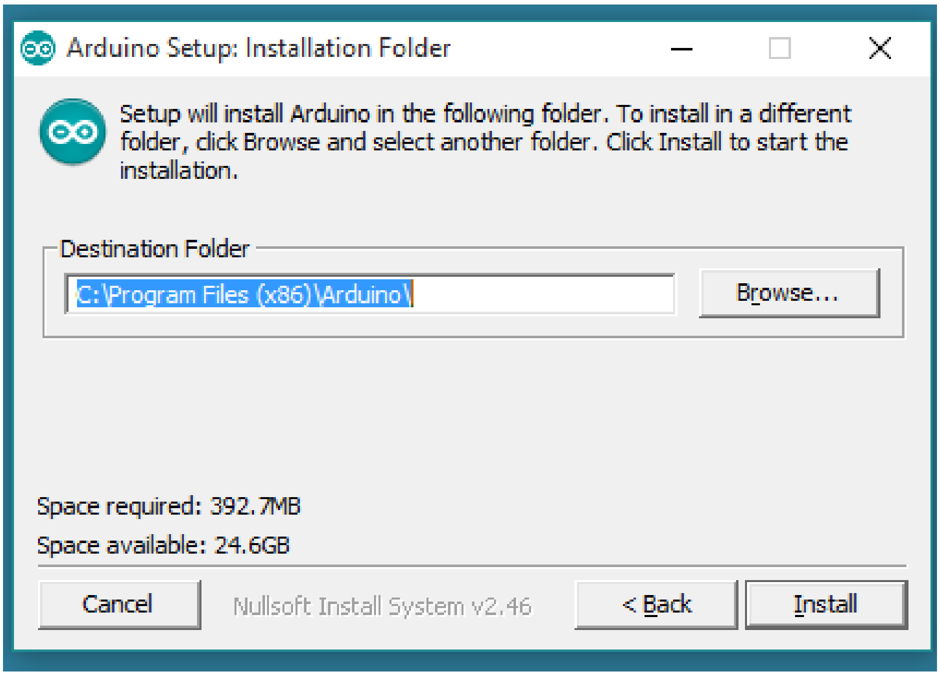


Fig 1.8 Setting destination folder

5. Choose the installation directory (we suggest to keep the default one)

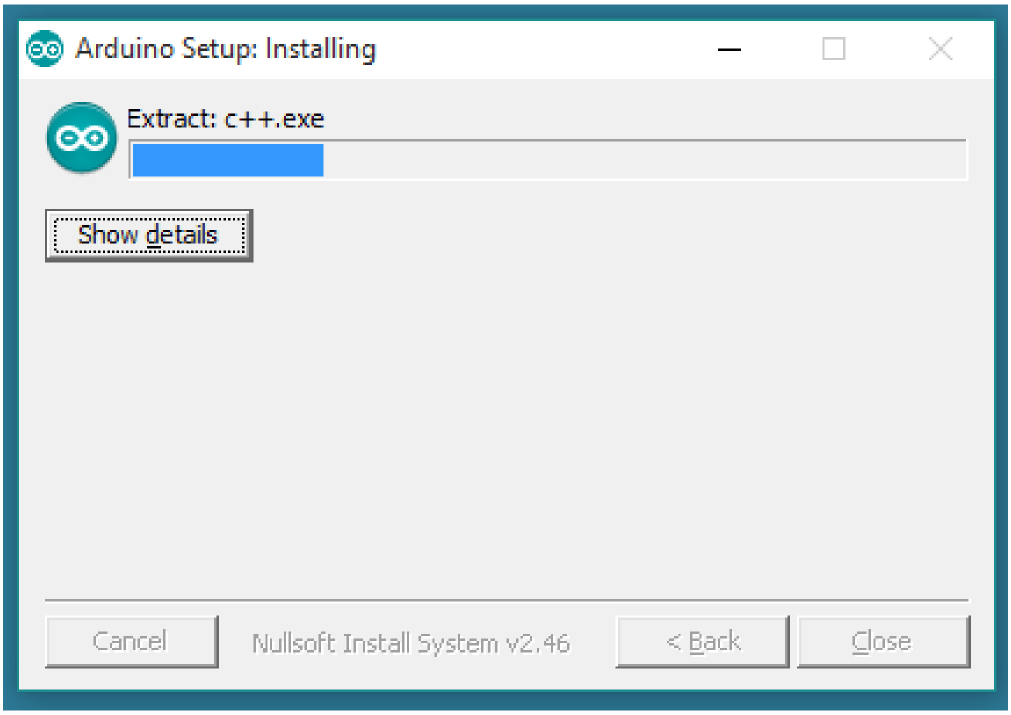


Fig:1.9 Installing directory

6. Successfully completed Installation of Software

7.Writing code on IDE platform

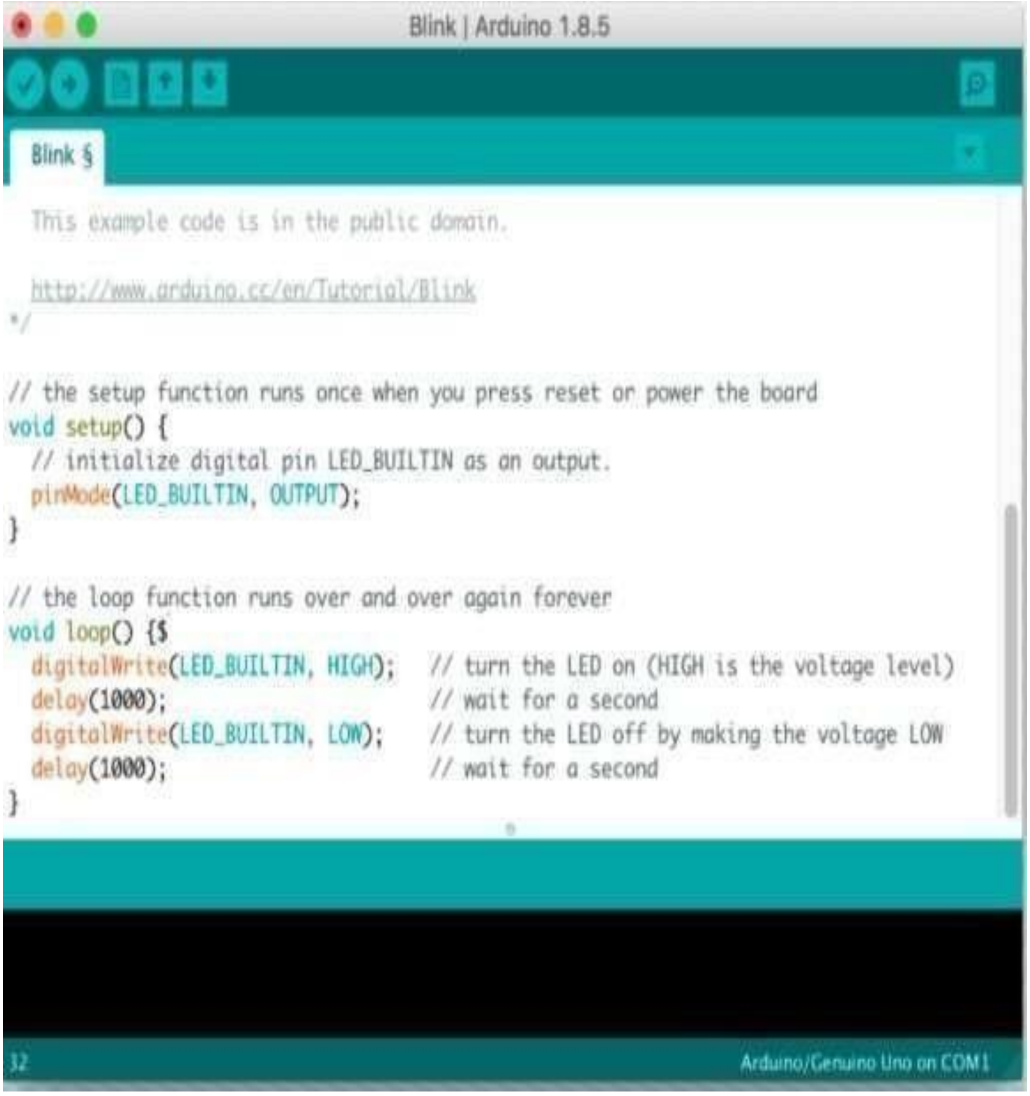


Fig:1.10 Source code written on IDE platform

1.3.2 PROTEUS SOFTWARE:

Through this software Schematic implementation of the embedded system will be done**.**

The topic of this paper is channel of this development of impersonation elimination system and this system would strictly with the subject having to claim as identify. The topic of this paper is channeled towards the unique feature of identification of fingerprint. A verification system based on the fingerprint and the term verification.

STEPS FOR DESIGNING SYSTEM ON PROTEUS DESIGN SUITE:

Step 1**:** Open ISIS software and select new design in File menu

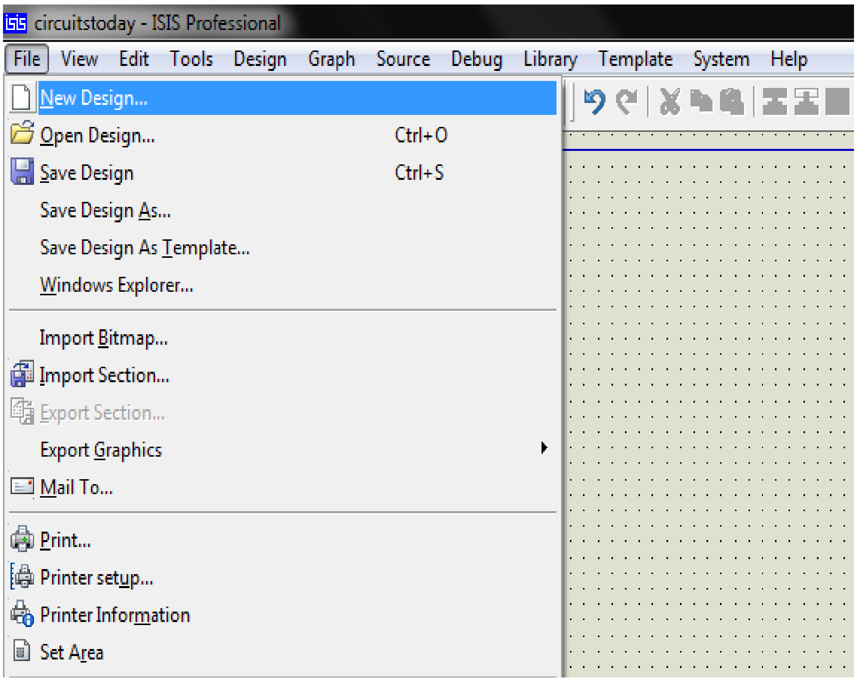


Fig:1.11 Simulation Procedure 1

Step 2**:** A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

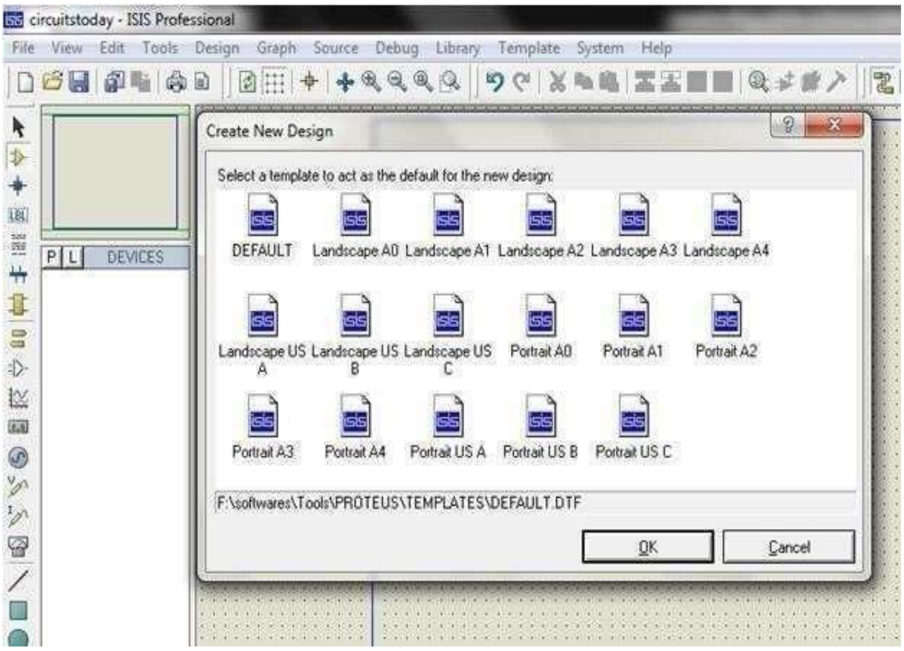


Fig:1.12 Simulation Procedure 2

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design.

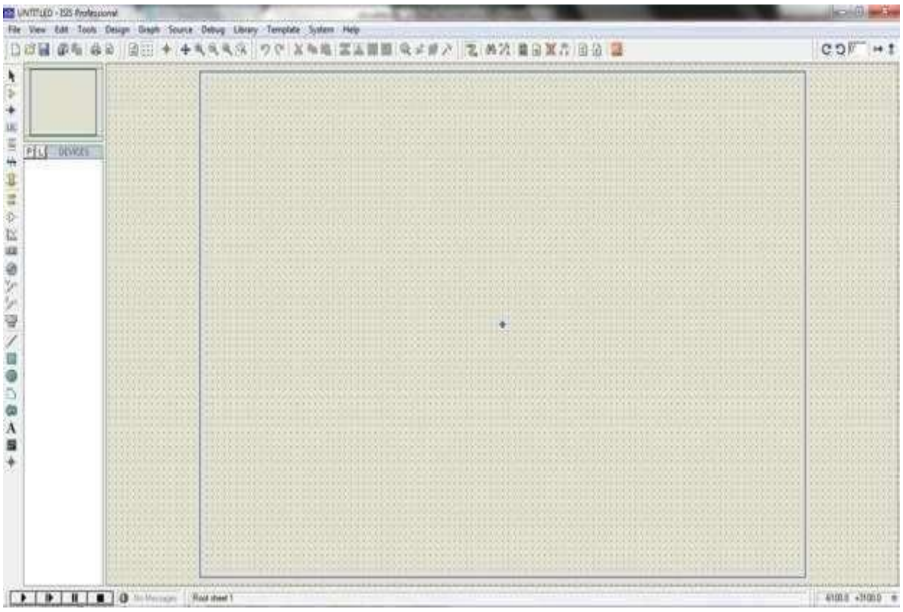


Fig:1.13 Simulation Procedure 3

Step 4: To Select components, Click on the component mode button.

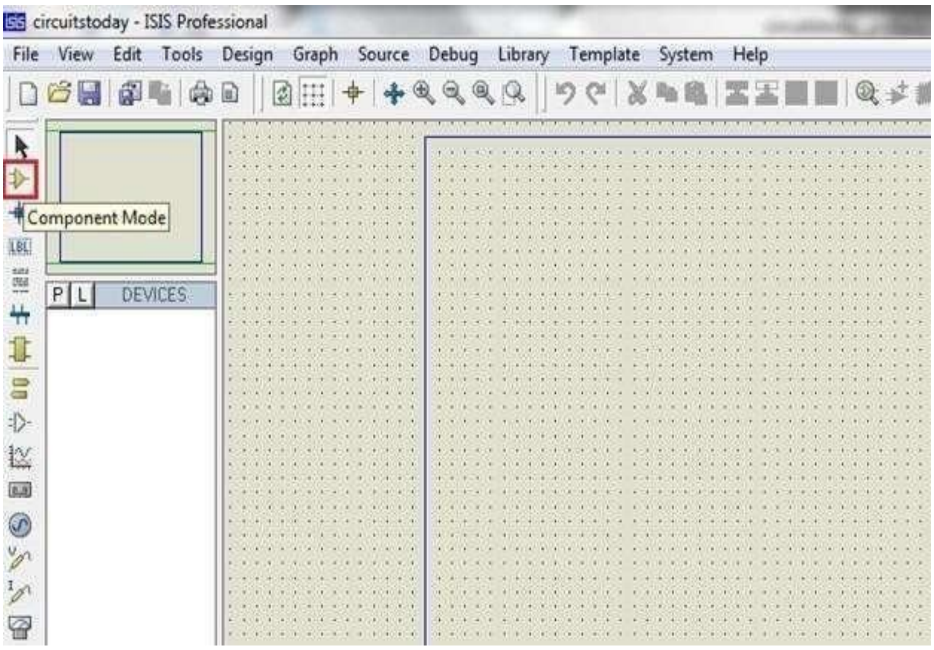


Fig:1.14 Simulation Procedure 4

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

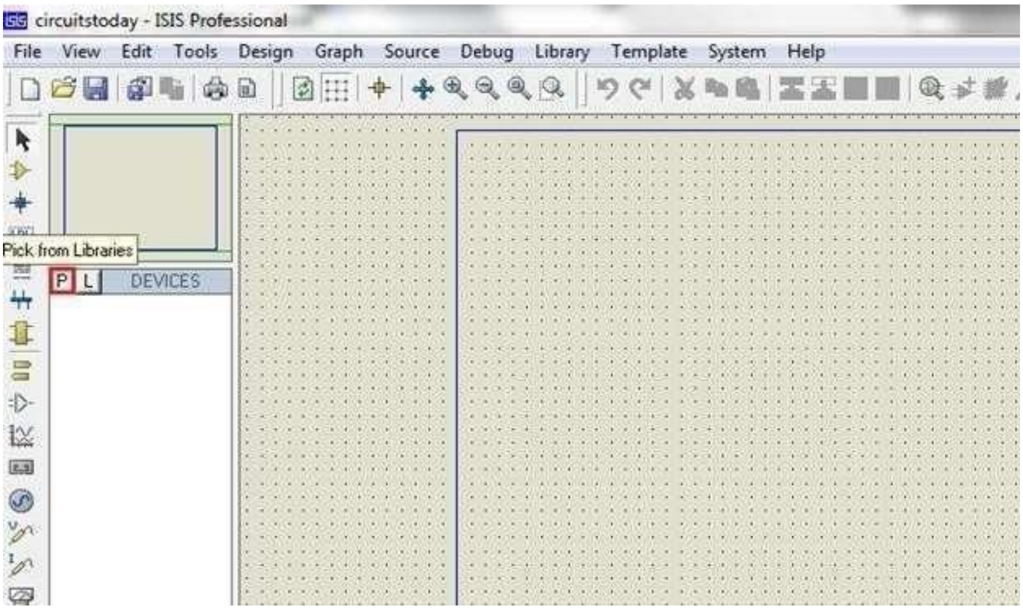


Fig:1.15 Simulation Procedure 5

Step 6: Select the components from categories or type the part name in Keyword's text box.

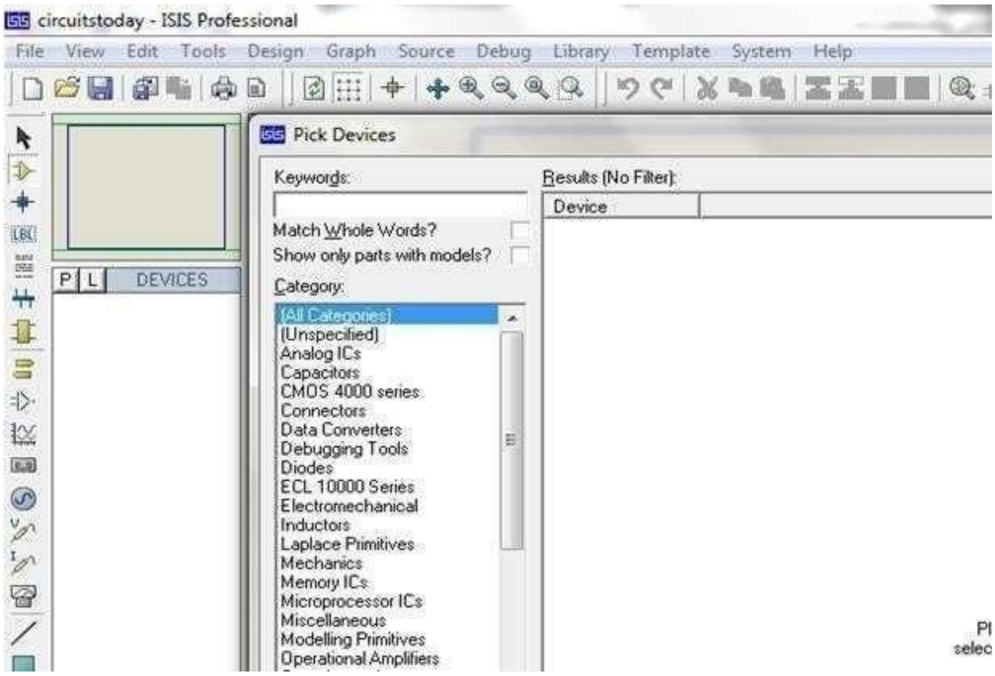


Fig:1.16 Simulation Procedure 6

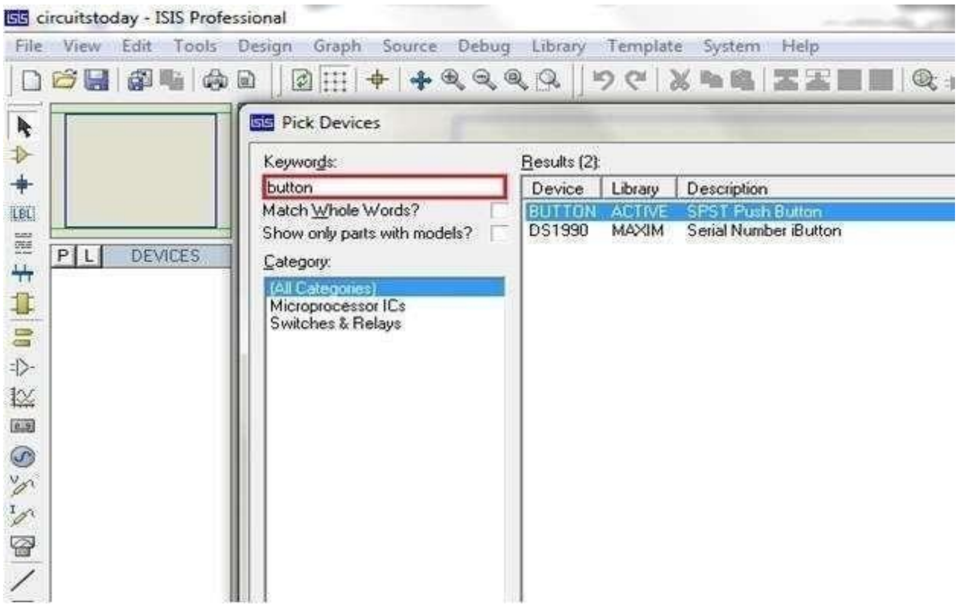


Fig:1.17 Simulation Procedure 7

Step 7**:** The selected components will appear in the devices list. Select the component and place it in the design sheet by left-cl.

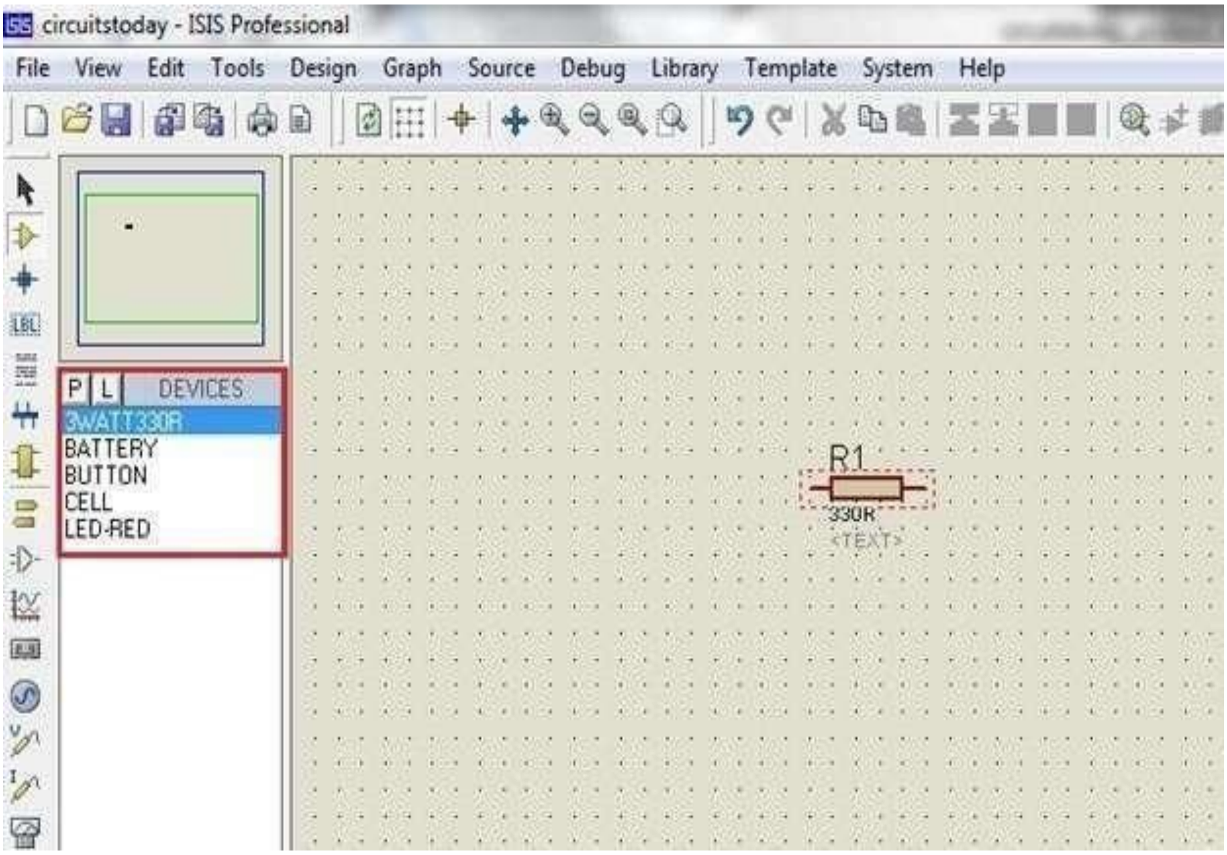


Fig:1.18 Simulation Procedure 8

Step 8: Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

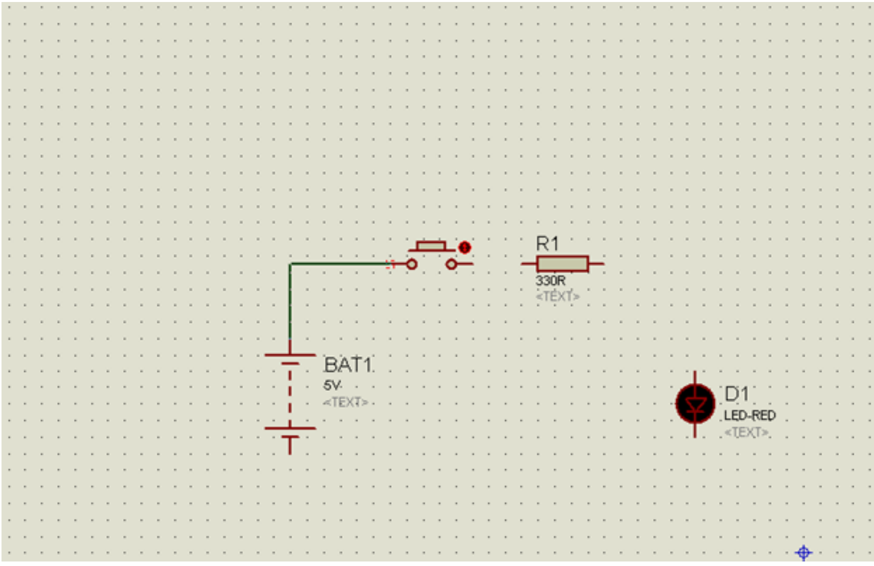


Fig:1.19 Simulation Procedure 9

Step 9: Double click on the component to edit the properties of the components and click on Ok.

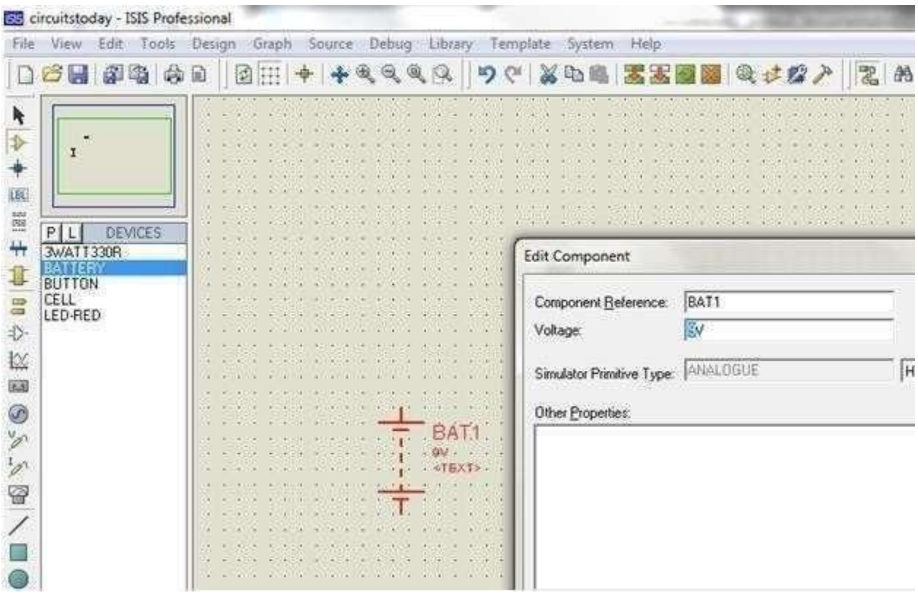


Fig.1.20 Simulation Procedure 10

Step 10: In this example simulation, the button is depressed during simulation by clicking on it to make LED

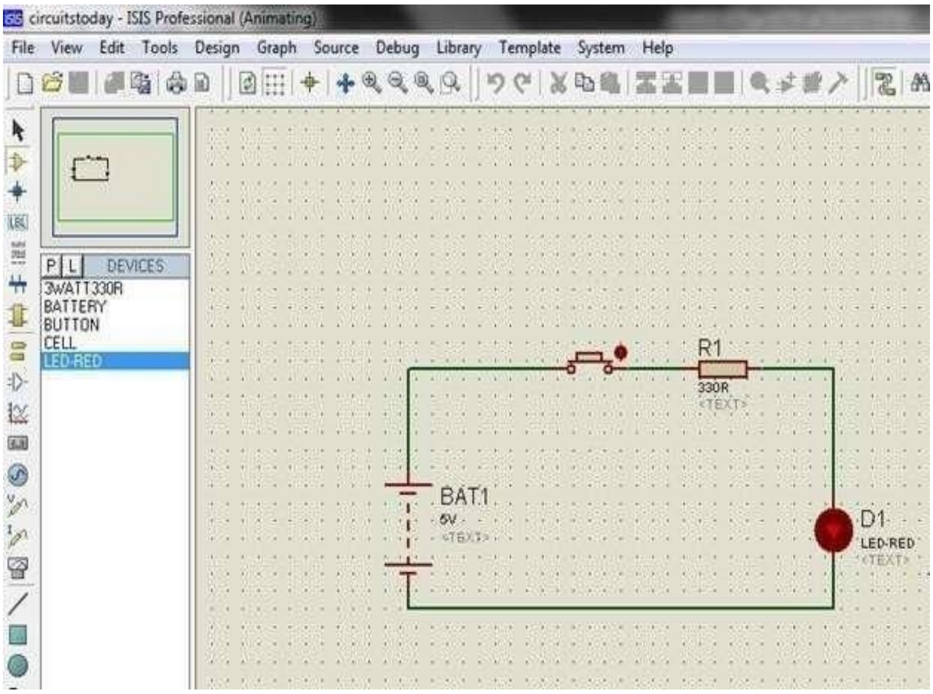


Fig:1.21 Simulation Procedure 11

Step 11: Simulation can be stepped, paused or stopped at any time.

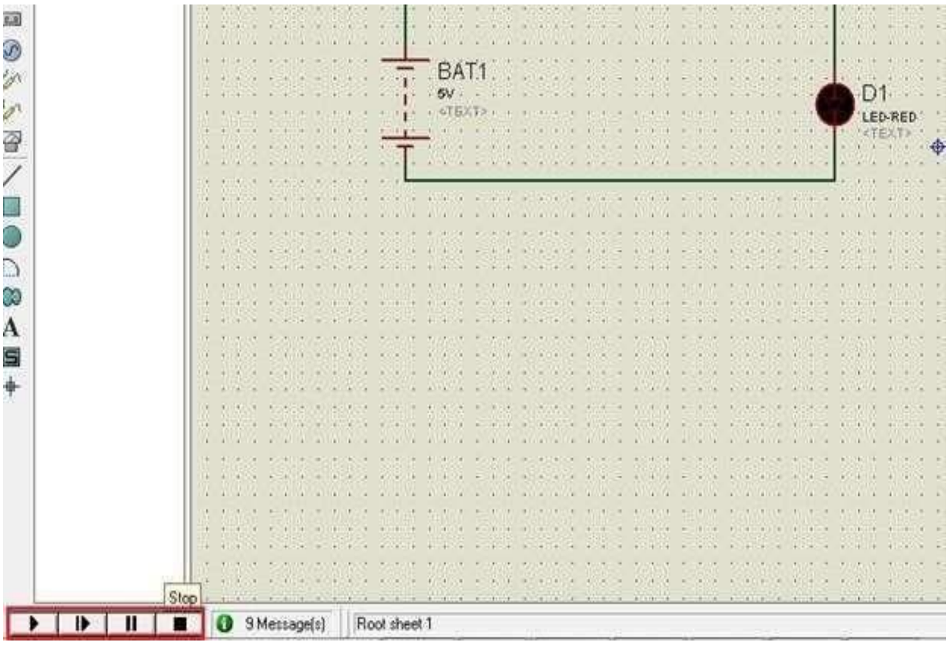


Fig:1.22 Simulation Procedure 12

1.4 PROCURMENT OF EQUIPMENT:

Fingerprints—the small ridges of skin on the fingertips which may enhance texture perception and grip friction—are unique to each person and have been used in identification for more than 100 years. Identification, the process of comparing a set of prints or a single print to others in a database, is used in many aspects of law enforcement and counterterrorism operations and investigations. Fingerprints may be compared to local and national databases for criminal history or to watch lists for border security. Latent prints from crime scenes are used in forensic investigations. Fingerprints can also be used to identify victims and for access control. Each application involves different processes for fingerprint capture and database searching and matching.

1.4.1 Components of Fingerprint Identification Systems

Fig:1.23 Finger print capture device

Fingerprint Capture the first step in fingerprint identification is obtaining an image of prints interest. Fingerprints deliberately collected from an individual are called exemplar prints, and may include simultaneous four finger images or rolled prints, which capture an image of each side the fingernail to the other. Exemplar prints have historically been obtained using paper and ink, which must then be optically scanned for database input. Live scanners electronically capture and store a digital image, eliminating paper and ink. Digitized prints are transmitted electronically for AFIS searches. Most current live scanners are based on optical sensors. Emerging technologies include thermal, capacitive, and ultrasound sensors, and some have to use a striped pattern projected onto the fingertip to obtain a three- dimensional image.

Minutiae Encoding

The second step in fingerprint identification is to extract the fingerprints which are identifiable characteristics to be used in a database search. Fingerprints can be sorted into general classification by pattern type, such as arch, loop, or whorl, but minute details such as the ridge endings or burifications are unique to each individual. Software identifies these minutiae and converts them into a system of numerical values. These minutiae templates are stored and used in searches instead of the whole image to reduce data and time requirements.

Recognition Software

An AFIS search involves the interaction of various databases where the minutiae are stored. Depending on the application, proprietary algorithms search segments of the database. For example, software may search the minutiae of two fingerprints of each subject, such as two index fingers, two thumbs, or a combination of finger and thumb. In other cases it may be necessary to search a database of all ten fingers. The simplest type of search is an identity verification used for access control or credentialing applications. Another identifier, such as a username, is submitted with a single fingerprint to specify who is attempting the authentication.

Certifications and Evaluations

A wide variety of fingerprint capture equipment and proprietary matching algorithms are commercially available. Interoperability is critical to the identification process and is achieved through standards and certification. The FBI has established image quality specifications for fingerprint images as well as data compression requirements for storage and transmission. The FBI certifies equipment compatible with the IAFIS, including live scanners, card scanners, and printers. Certified products are listed at https://www.fbibiospecs.org/IAFIS/Default.aspx. The National Institute of Standards and Technology (NIST) leads developments in automating fingerprint identification and conducts evaluations of hardware and software. Some examples are fast Capture, an initiative to improve technology to capture 10 rolled equivalent prints in less than 15 seconds in a portable, rugged system; the Minutiae Interoperability Exchange (MINEX), which tests the use of a standardized minutiae template, rather than entire fingerprint images, for exchange of data between 1:1 matching systems; and the Fingerprint Vendor Technology evaluation 2012, which assesses the performance of 1:N matching algorithms based on proprietary fingerprint templates using a database of millions of fingerprint sets, including live scan and scanned inked cards.

Standards

Two standards are currently used in FBI certification. The image quality specifications for interoperability with IAFIS for fingerprint scanners and printers are contained in Appendix F of IAFIS- DOC-01078-9.3 Criminal Justice Information Services Electronic Biometric Transmission Specification. Fingerprint verification using 1:1 matching is the focus of the standard PIV-071006.

Development of national and international standards for fingerprint technology is ongoing, and many other standards are available. For example, ANSI/INCITS 381-2004 specifies data compression for image storage and transmission, and ANSI/INCITS 378-2009 defines a fingerprint standard minutiae template.

1.5 ORGANISATION OF THE CHAPTERS

The Overall Overview of Documentation is:

1.5.1 INTRODUCTION

1.5.2 LITERATURE SURVEY

1.5.3 PROJECT DESIGN

1.5.4 PROJECT IMPLEMENTATION

1.5.5 PROJECT TESTING

1.5.6 CONCLUSION AND FUTURE ENHANCEMENT

CHAPTER 2

LITERATURE SURVEY

This chapter gives a brief overview of previous works done related to Fingerprint system.

2.0 INTRODUCTION:

The main idea of our project is to secure the places by using fingerprint sensor used to scan the finger of n the persons, the authorized people have to register their fingerprints to the system by using fingerprint sensor and each person is provided a particular ID which is stored in the data, Authorized people are the only eligible for vote, while the remaining people can't. For more security, the project is done along with buzzer indicator.

2.1 RELATED WORK:

Biometrics is a way used to recognize a person based on his physical nature. The fingerprint, iris, face, voice, etc. are the mainly used biometrics to recognize a person. There are two key functions for biometrics, first is one to one matching and other is one too many matchings. In one to many matching the biometric sample is compared with the already stored samples. In one to one matching, it compares with the previously stored sample. Biometric method results in a faster security, and more convenient method for user verification. Biometric method is better than password security. Fingerprint is unique for each individual so it can be used as a mark of signature, verification and authentication. Fingerprint is the biometric which is used in this project. Finger-print will be different for each individual. In this project, fingerprint is used for the authentication of the user on his fingerprint image. Finger-print matching can be divided into three types: correlation based matching, minutiae-based matching, pattern- based (or image- based) matching. In correlation based matching, two fingerprint images are superimposed and therefore the correlation between corresponding pixels is computed for various alignments. In minutiae- based matching, minutiae from the two fingerprints are extracted and stored in a two-dimensional plane as a set. This matching method consists of finding the alignment between the template and the input minutiae sets that result in the maximum number of minutiae pairings. In pattern-based (or image-based) matching method it compares with stored template and the candidate's fingerprint. This requires that the images to be aligned within the same orientation. To do this, the algorithm finds a central point within the fingerprint image and centers there on. In a pattern- based algorithm, the template contains the sort, size, and orientation of patterns within the aligned fingerprint image. Almost all the sectors are storing data digitally. To create digital India, most of the tasks are made through on-line. and implementation of ID authentication system based on Fingerprint Identification is designed, and finally. the problem at hand, which is exam student impersonation and corruption is Tertial Education will be demolished and to demonstrate that anything can be achieved using computer technology in this fast growing Information Technology and computer communication world . Simulation phase of this project has been achieved, with the following devices: Microcontroller (Atmega 328), LCD display device (LM016L), regulated DC power supply, Keypad-phone, Virtual Terminal, X1 Crystal clocking, Variable Resistor, and Capacitors. Regulated DC power supply is used for powering the devices on the system and it has to be regulated to avoid too much power to burn the device and to be too low for it to be insufficient for the devices operations. Microcontroller, this is the heart of the whole system, it is there to facilitate the connection between devices, control the operation of every device connected on this system, it has clocking equipment that must be synchronised with the crystal clocking device to ensure precise operation of the system. Keypad, this device is used as a input device used by the user (student) when they are entering their password during authentication process. LCD display is used as the output device of the whole system because, every message that the system need to convey to the user, is display on here, starting from displaying the name of the system (Fingerprint based exam hall authentication), asking the user to place their fingers (Put your finger), displaying the results of the authentication (Enter the passkey or Access denied student not registered) and the final display if the user's passkey is correct (Access granted). Variable resistor, this is to give brightness to the LCD display. Crystal Clocking, this has the simple task, which is to provide clocking to the system, clocking which is synchronised with the clock of the microcontroller.

Virtual Terminal is used as input device as well to input the student number of the student when the system ask the user to put their finger, because there is no fingerprint scanner on the system, so this device is used in the simulation phase. And the final device are the capacitors, which are used to prevent the unwanted current from entering the system.

Although it was not an easy job, to prove that the proposed system is working according to how the real systems work. This was proved by implementing the project in simulation but its working according to the way it is intended to work. By simulating this project one will see the display on the LCD, the system writing the name of the project "Fingerprint based exam hall Authentication", few seconds later it will display the following message: "PUT YOUR FINGER" this is just to tell the subject (student) to be authenticated to put her/his finger on the fingerprint scanner for it to be able to read his/her fingerprints, after that if the student is authentic then the following message will be displayed: "ENTER YOUR PASSKEY", this is a password that every registered student got during registration/enrolment phase.

2.2 EXISTING METHODOLOGY:

Paper Ballots were used in India before 1997 for conducting the public elections. The existing system that we are using right now is very inefficient, time consuming and requires lot of patience to handle the situation. It requires man time to complete this process. For entering an exam hall the teacher should check the students ID and then make sure the person is suitable or not through ID cards, due to this there is lot of possibility of malpractices. Sometimes there might be a chance that the teacher or the guardian of the exam hall might not check the ID cards and because of that some other person might write the exam instead of the main person. This process that we are using right now is very time consuming, checking each and every student details is not a very easy task. The system that we using now currently are not secured because it requires student to provide certain physical documents such as student cards, examination application, etc. This kind of methods can be forged by almost anyone in the fast growing computer technology word.

2.3 PROPOSED METHODOLOGY:

The purpose of this project is to use the fingerprint identification to authenticate the identity of a private and therefore the main aim of this project is to differentiate between a licensed person (student) and an imposter before entering the examination hall. Here we proposed a finger print based exam hall authentication system. The system is meant to permit only users verified by their fingerprint scan and doesn't allow non verified users. Our system consists of a fingerprint scanner connected to microcontroller circuit. In registration mode the system allows to register user's information and save their identity with respective ID numbers within the system memory. After registration the person must scan his/her finger with the help of the scanner. The microcontroller checks the person's fingerprint. If the fingerprint valid the motor driver send a massage to open a door. This ensures only authorized users are allowed to enter the examination section and unauthorized users aren't allowed to enter the room.

CHAPTER 3

PROJECT DESIGN

3.0 OVERVEIW OF THE SYSTEM DESIGN

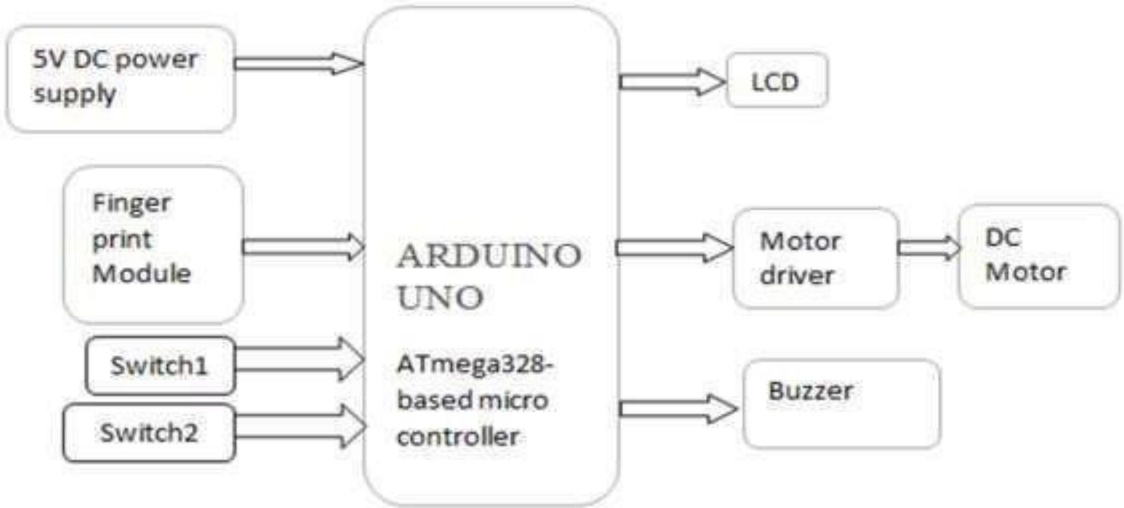


Fig:3.1 Block Diagram of finger print based exam hall authentication

3.1 SCHEMATIC DIAGRAM:

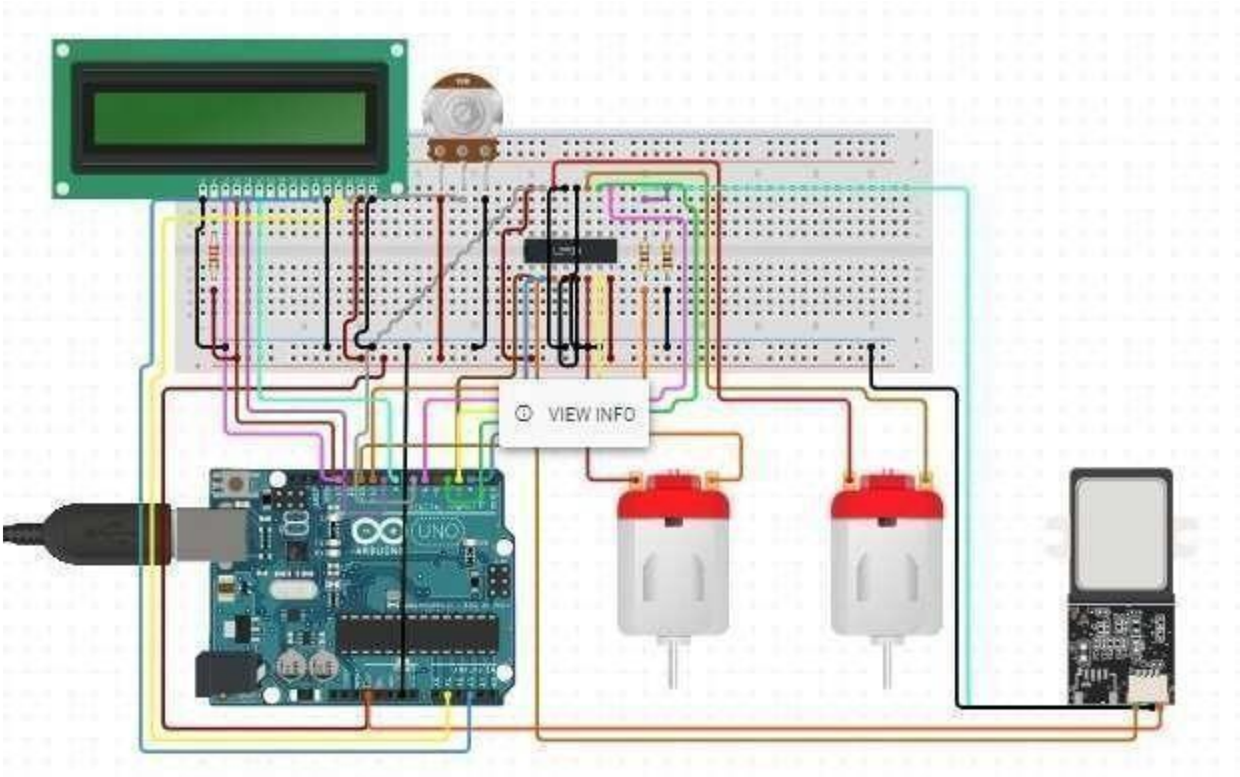


Fig:3.2 Simulation layout

The above schematic diagram of Finger Print Based exam hall authentication explains the interfacing section of each component with Arduino uno and input output modules

3.2 DEFINE THE MODULES:

3.2.1 FINGER PRINT MODULE:

Fingerprint Module consists of optical fingerprint sensor, high-speed DSP processor, high performance fingerprint alignment algorithm, high-capacity FLASH chips and other hardware and software composition, stable performance, simple structure, with fingerprint entry, image processing, fingerprint matching, search and template storage and other functions. Fingerprint module has two interface TTL UART and USB2.0, USB2.0 interface are often connected to the computer; RS232 interface may be a TTL level, the default baud is 57600, can be changed, ask a communication protocol, microcontrollers like ARM, DSP and other serial devices with a connection, 3.3V- 5V microcontroller are often connected directly.

Fig 3.3 : Fingerprint module

Specifications:

Here's the specifications of the fingerprint sensor module we're using (you should check your sensor datasheet or the specifications provided by your supplier – they shouldn't be much different than these):

- Voltage supply: DC 3.6 to 6.0V
- Current supply: <120mA
- Backlight color: green
- Interface: UART
- Bad rate: 9600
- Safety level: five (from low to high: 1,2,3,4,5)
- False Accept Rate (FAR): <0.001% (security level 3)
- False Reject Rate (FRR): <1.0% (security level 3)
- Able to store 127 different fingerprints

3.2.2 LCDMODULE:

An LCD (Liquid Crystal Display) screen is an electronic display module and has a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. This LCD has two registers, namely, Command and Data.

Command register stores various commands given to the display. Data register stores data to be displayed. The process of controlling the display involves putting the data that form the image of what you want to display into the data registers, then putting instructions in the instruction register. In your arduino project Liquid Crystal Library simplifies this for you so you don't need to know the low-level instructions. Contrast of the display can be adjusted by adjusting the potentiometer to be connected across VEE pin.

3.3 MODULES FUNCTIONALITIES:

3.3.1 Arduino

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are :

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

3.3.1.1 Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino

IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Board Name	**Operation volt**	**Clock speed**	**digital i/o**	**Analog inputs**	**PWM**	**UART**	**Programming Interface**
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header

Table:3.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
Lily pad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table:3.2 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table:3.3 Arduino boards based on ATMEGA2560 microcontroller

Operating Volt	Board Name	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
3.3V	Arduino Mega 2560R3	84MHz	54	12	12	4	USB native

Table:3.4 Arduino boards based on AT91SAM3X8E microcontroller

3.3.1.2 Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common

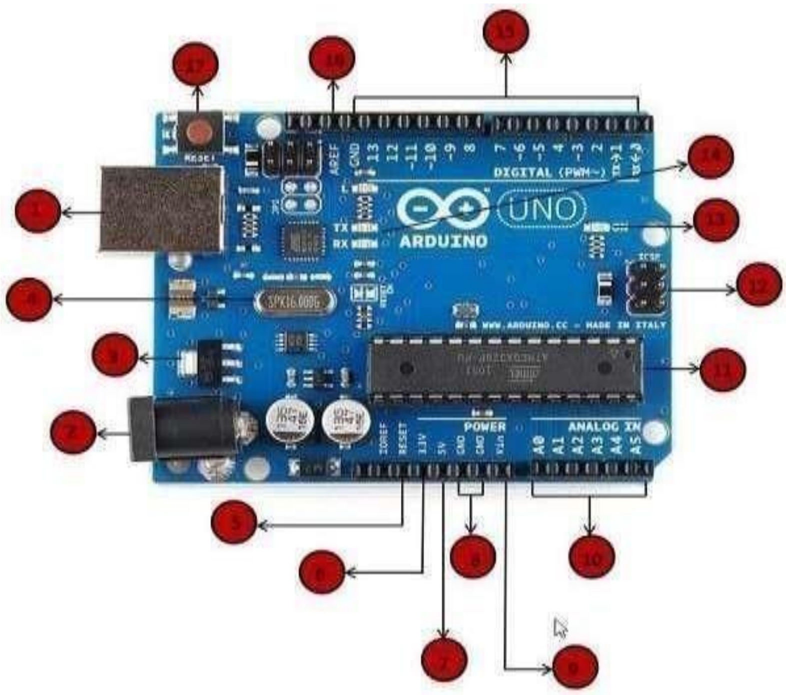


Fig:3.4 Board description of Arduino uno

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt Most of the components used with Arduino board works fine with 3.3 volt and 5 volts. GND (8) (Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.

10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

3.4 Liquid Crystal Display

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to a controller is an LCD display. Some of the most common LCDs connected to the controllers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines (RS RW D7 D6 D5 D6).

When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller

3.4.1 Description of 16x2:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all-Parallel Ports. It however doesn't show the use of the Status Port as an input. So, what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

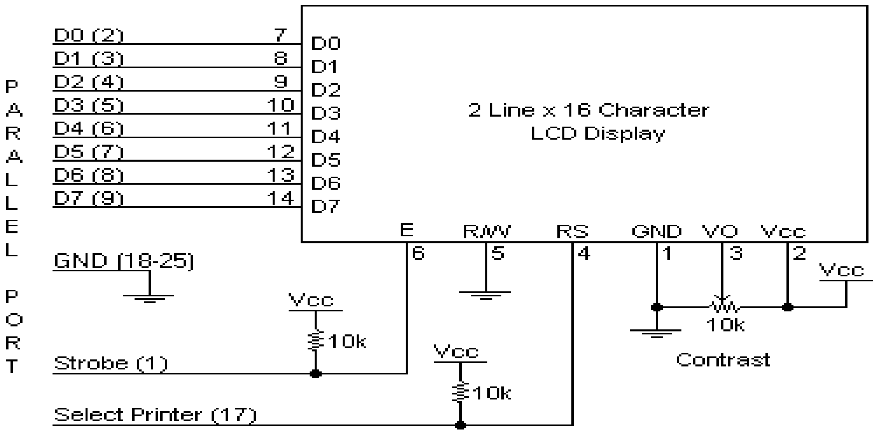


Fig 3.5: Schematic diagram

- Above is the quite simple schematic. The LCD panel's *Enable* and *Register Select* is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore, by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

- We make no effort to place the Data bus into reverse direction. Therefore, we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result, we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

- The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. You can use a bench power supply set to 5v or use an onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.

3.4.2 16 x 2 Alphanumeric LCD Module Features:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area.
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line.
- Can display 224 different symbols.
- Low power consumption (1 mA typical).
- Powerful command set and user-produced characters.
- TTL and CMOS compatible.
- Connector for standard 0.1-pitch pin headers.

Pin	Symbol	Level	Function
1	V_{SS}	-	Power, GND
2	V_{DD}	-	Power, 5V
3	Vo	-	Power, for LCD Drive
4	RS	H/ L	Register Select Signal H: Data Input L: Instruction Input
5	R/W	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)
6	E	H,H>L	Enable

Table:3.5 16x2 LCD Features

3.4.3 FEATURES:

- 5 x 8 dots with cursor
- Built-in controller (KS 0066 or Equivalent)
- + 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)
- N.V. optional for + 3V power supply

Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are:

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

Table:3.5 Data Location in LCD

3.4.4 Figure Address locations for a 1x16 line LCD

Even limited to character-based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8,16,20,24,32 and 40 characters are all standard, in one-, two- and four-line versions.

Several different LC technologies exists. “supertwist” types, for example, offer Improvedcontrast and viewing angle over the older “twisted nematic” types. Some modules are available with back lighting, so that they can be viewed in dimly-lit conditions. The back lighting may be either “electro-luminescent”, requiring a high voltage inverter circuit, or simple LED illumination.

3.4.5 Pin Description In LCD:

Most LCDs with 1 controller have 14 Pins and LCDs with 2 controllers has 16 Pins (two pins are extra in both for back-light LED connections).

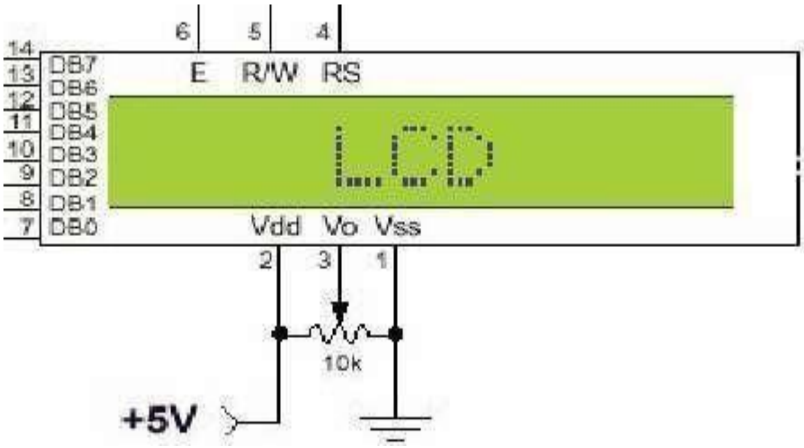


Fig:3.6 Pin Description of 1x16 lines

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Table 3.7 Pin specifications

3.4.6 CONTROL LINES:

EN: Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS: Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW: Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands, so RW will almost always be low. Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5,DB6,and DV.

Logic status on control lines

- E - 0 Access to LCD disabled.
- 1 Access to LCD enabled.
- R/W - 0 Writing data to LCD.
- 1 Reading data from LCD.
- RS - 0 Instructions.
- 1 Character.

Writing data to the LCD

- Set R/W bit to low.
- Set RS bit to logic 0 or 1 (instruction or character).
- Set data to data lines (if it is writing).
- Set E line to high.
- Set E line to low.

Read data from data lines (if it is reading) on LCD

- Set R/W bit to high.
- Set RS bit to logic 0 or 1 (instruction or character).
- Set data to data lines (if it is writing).
- Set E line to high.
- Set E line to low.

Entering Text

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the d.i.l. switch pack with a of re-wiring is necessary.

LCD Commands

There are some present commands instructions in LCD, which we need to send to LCD through some microcontroller. Some important command instructions are given below

Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left
0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

Table:3.8 LCD Commands

3.5 Fingerprint Sensor

The fingerprint sensor can read different fingerprints and store in its own flash memory. The sensor can perform three functions namely Add (Enroll), Empty Database or search database and return the ID of stored fingerprint. Any of three functions can be called simply by making the pin low of the sensor or pressing onboard three switches. The response is either error or ok which is indicated by onboard LED. The response is also returned as single serial data byte.

The return byte is a valid ID or error code. The response byte is a single byte at 9600 bps thus making whole sensor very easy to use. We have provided indicating LEDs and function switch already so it’s ready to use when you receive it. Just give power and start using the sensor using onboard switches. Then you can move on making external application using these functions.

3.5.1 Inputs and Outputs of Sensor

Input: Two ways to trigger the function of fingerprint sensor

- Onboard switch: Add, Empty or Search.
- Make pin low from external microcontroller for 5ms as per function required to be executed.

Output: Two ways to monitor output response after a function is executed

- Onboard LEDs: ERROR or OK.
- Read byte after executing function.

3.5.2 Types of Sensor Function

There are namely three functions you can call for the fingerprint sensor. We will see each in brief.

Add (Enroll) Function: Adds a fingerprint to database and return a byte of newly added ID. Return values are from 0x00 to 0xFE. In case of error like no finger placed, return code is 0xFF. Here 0xFF means error executing function

Search Function: When a finger is put and search function is called, it returns a matching ID if found in its existing memory. Return values are from 0x00 to 0xFE. In case of error like no finger placed, return code is 0xFF. Here 0xFF means error executing function.

Empty Function: When you wish to empty all fingerprint, data stored on sensor you can use this function. After executing this function, you will get 0xCC as OK or 0xFF in case of error.

3.5.3 Fingerprint Recognition Sensor Module

The sensor is a solid-state fingerprint sensor that reliably captures fingerprint information. It is designed to integrate into devices for improved security and convenience. The sensor provides are liable, quick and user-friendly alternative to passwords, PIN's and other forms of user authentication. This fingerprint scanner is capable of gathering and storing unique finger prints. Simply hold your finger on the optical scanner, query the device over serial, and you will be issued a unique ID. Use that ID within your embedded system to determine access levels, time clocks, door locks, etc. Unit includes 4 pin connector cable to connect and read to controller. The outputs are TTL level serial data. A biometric sensor, fingerprint sensor to be specific, also known as the fingerprint reader, is a fingerprint image capture device, the very front end of the biometric fingerprint identification/verification module. The fingerprint sensor captures the fingerprint images, matches the uniqueness of each print read by the sensor and compares it to the one stored in its module or local system database.

It consists of optical fingerprint sensor, high performance DSP processor and Flash. It boasts of functions such as fingerprint enrollment, fingerprint deletion, fingerprint verification, fingerprint upload, fingerprint download, etc.

3.5.4 Applications

- Computer peripherals – improves security and convenience
- Transportation systems – validation of operators, drivers and inspectors
- Medical equipment – authorization of operator or technician
- Physical access systems – approval for entry
- Kiosks and vending machines – confirmation of person receiving the selection
- Point of Sale terminals – authentication of tellers and cashiers

3.5.5 Features

- Rugged, solid-state optical fingerprint sensor
- High resolution 500 DPI imager
- Adapts to wet/dry fingers
- Simple Interfacing protocol

Fig:3.7 Fingerprint sensor

3.5.6 Specifications of Sensor

- Fingerprint enrollment time <250ms
- Fingerprint search time <1s（100 fingerprint, average value in test） Resolutions 500 DPI
- Security level 5, Adjustable Capacity 768 Templates
- FAR <0.0001%

- FRR <0.01%
- Power Supply 4.3V to 6V Working Current <80mA Peak Current <90mA Communication Interface TTL
- Communication Baud Rate 57600 bps Working Temperature -10 deg C to +40 deg C
- Working Humidity 40% RH to 85% TH (no dew) Module Dimensions 60x21x25 mm (LxWxH)

3.6 BUZZER:

3.6.1 Types of Buzzer

Magnetic Transducer

Magnetic transducers contain a magnetic circuit consisting of a iron core with a wound coil and a yoke plate, a permanent magnet and a vibrating diaphragm with a movable iron piece. The diaphragm is slightly pulled towards the top of the core by the magnet's magnetic field. When a positive AC signal is applied, the current flowing through the excitation coil produces a fluctuating magnetic field, which causes the diaphragm to vibrate up and down, thus vibrating air. Resonance amplifies vibration through resonator consisting of sound hole(s) and cavity and produces a loud sound.

Magnetic Buzzer (Sounder)

Buzzers like the TMB-series are magnetic audible signal devices with built-in oscillating circuits. The construction combines an oscillation circuit unit with a detection coil, a drive coil and a magnetic transducer. Transistors, resistors, diodes and other small devices act as circuit devices for driving sound generators. With the application of voltage, current flows to the drive coil on primary side and to the detection coil on the secondary side. The amplification circuit, including the transistor and the feedback circuit, causes vibration. The oscillation current excites the coil and the unit generates an AC magnetic field corresponding to an oscillation frequency. This AC magnetic field magnetizes the yoke comprising the magnetic circuit. The oscillation from the intermittent magnetization prompts the vibration diaphragm to vibrate up and down, generating buzzer sounds through the resonator.

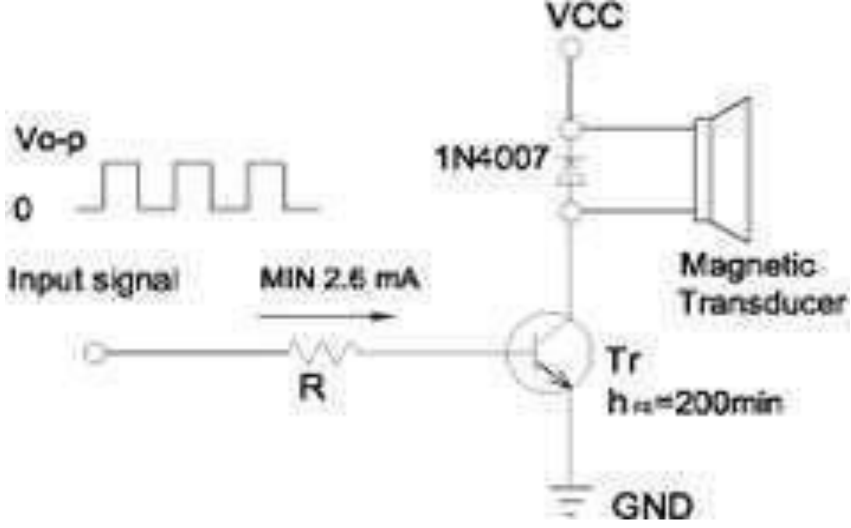


Fig:3.8 Driving Circuit for Magnetic Transducer

3.6.2 Specifications of buzzer:

- **Rated Voltage**: A magnetic buzzer is driven by 1/2 square waves (V o-p). **Operating Voltage**: For normal operating. But it is not guaranteed to make the minimum Sound Pressure Level (SPL) under the rated voltage.
- **Consumption Current**: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.
- **Direct Current Resistance**: The direct current resistance is measured by ammeter directly. **Sound Output**: The sound output is measured by decibel meter. Applying rated voltage and 1/2 square waves, and the distance of 10 cm.
- **Rated Frequency**: A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.
- **Operating Temp**: Keep working well between -30°C and +70°C.
- **Driving methods**: AX series with built drive circuit will be the best choice when we cannot provide frequency signal to a buzzer, it only needs direct current.
- **Dimension**: Dimension affects frequency, small size result in high frequency.
- **Voltage**: Depend on V o-p (1/2 square waves)
- **Fixed methods**: From the highest cost to the lowest- DIP, wires/ connector, SMD.

3.6.3 How to choose a buzzer

There are many different kinds of buzzer to choose, first we need to know a few parameters, such as voltage, current, drive method, dimension, mounting type, and the most important thing is how much SPL and frequency we want.

Operating voltage: Normally, the operating voltage for a magnetic buzzer is from 1.5V to 24V, for a piezo buzzer is from 3V to 220V. However, in order to get enough SPL, we suggest giving at least 9V to drive a piezo buzzer.

Consumption current: According to the different voltage, the consumption current of a magnetic buzzer is from dozens to hundreds of mill amperes; oppositely, the piezo type much more electricity, only needs a few mill amperes, and consumes three times current when the buzzer starts to work.

Driving method: Both magnetic and piezo buzzer have self-drive type to choose. Because of the internal set drive circuit, the self-drive buzzer can emit sound as long as connecting with the direct current. Due to the different work principle, the magnetic buzzer needs to be driven by 1/2 square waves, and the piezo buzzer need square waves to get better sound output.

Dimension: The dimension of the buzzer affects its SPL and the frequency, the dimension of the magnetic buzzer is from 7 mm to 25 mm; the piezo buzzer is from 12 mm to 50 mm, or even bigger.

Connecting way: Dip type, Wire type, SMD type, and screwed type for big piezo buzzer are usually seen.

Sound Pressure Level (SPL): Buzzer is usually tested the SPL at the distance of 10 cm, if distance double, the SPL will decay about 6 dB; oppositely, the SPL will increase 6 dB when the distance is shortened by one time. The SPL of the magnetic buzzer can reach to around 85 dB/ 10 cm; the piezo buzzer can be designed to emit very loud sound, for example, the common siren, are mostly made of silicon.

Fig:3.9 Buzzer

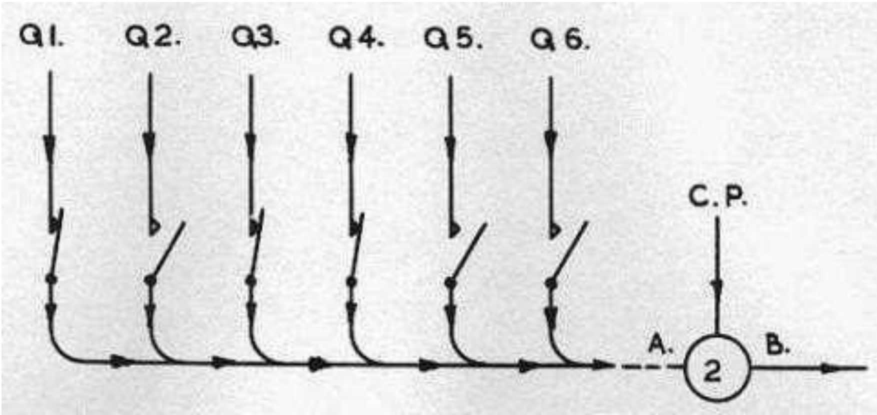


Fig:3.10 Pin Diagram of buzzer

3.7 Switches

A switch is an electrical component that can break an electrical circuit, interrupting the current or diverting it from one conductor to another. The most familiar form of switch is a manually operated electromechanical device with one or more sets of electrical contacts. Each set of contacts can be in one of two states: either 'closed' meaning the contacts are touching and electricity can flow between them, or 'open', meaning the contacts are separated and non-conducting.

Fig:3.11 Switches in a row

Switches with larger numbers of poles or throws can be described by replacing the "S" or "D" with a number or in some cases the letter "T" (for "triple"). In the rest of this article the terms *SPST*, SPDT and intermediate will be used to avoid the ambiguity in the use of the word "way".

3.7.1 Working of Switch

A pair of contacts is said to be "closed" when current can flow from one to the other. When the contacts are separated by an insulating air gap, an air space, they are said to be "open", and no current can flow at typical voltages.

Switches are classified according to the arrangement of their contacts in electronics. Electricians installing building wiring use different nomenclature, such as "one-way", "two- way", "three-way" and "four-way" switches

In a push-button type switch, in which the contacts remain in one state unless actuated, the contacts can either be normally open (abbreviated "no." or "no") until closed by operation of the switch, or normally closed (″n.c. or "nc″) and opened by the switch action. A switch with both types of contact is called a changeover switch. These may be "make-before- break" which momentarily connect both circuits, or may be "break-before-make" which interrupts one circuit before closing the other

3.7.2 APPLICATIONS

- They are used for the given instructions to the required device like robot, driving the motor, etc.
- Switches can arrange in the matrix from or in the required from according to the application.
- **Reset circuit of the controller, motor also has the switches for its particular functioning.**

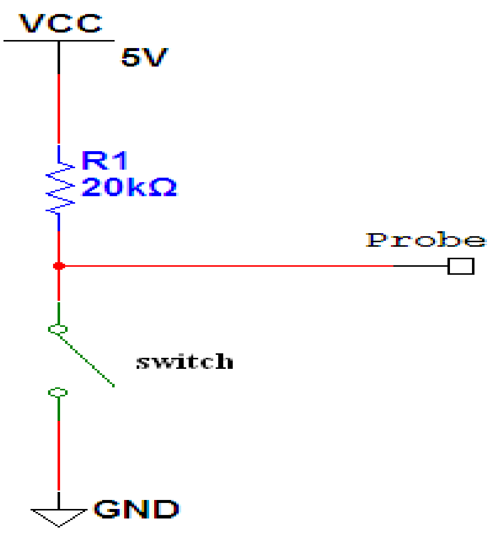


Fig:3.12 Circuit of Switch

From the above circuit it explains the working of the switch when supply is given to the circuit the resistor takes the current and oppose to the certain extend until the switch is pressed. These switches can be connected parallel or serial in required manner.

We can have resistor, capacitor, light resistor diode etc. as the combination of the switch. For example, w the reset circuit in which we the resistor and capacitor as the combination.

From the above circuit it explains the working of the switch when supply is given to the circuit the resistor takes the current and the oppose to the certain extend until the switch is pressed. These switches can be connected parallel or serial in required manner.

CHAPTER 4
PROJECTIMPLEMENTATION

4.0 IMPLEMENTATION STAGES:

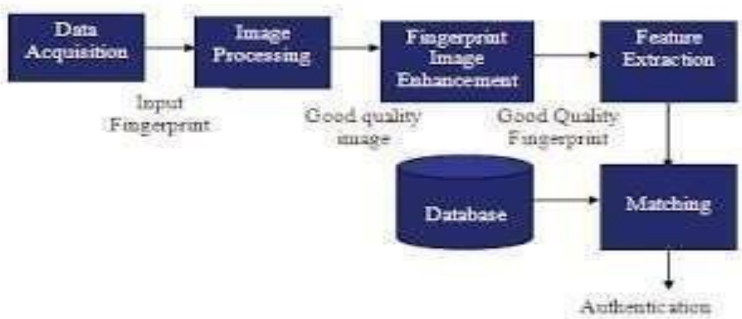


Fig:4.1 Implementation of Finger print based exam hall authentication

1. Algorithm of Fingerprint Based Voting System:
Step 1: Start

Step 2: Users can register by giving their fingerprints. We use a microcontroller to store them.

Step 3: Scan your finger

Step 4: If match found go to step 5 else return to step3

Step 5: user checks their detail

Step 6: enrolled

Step 7: cast your fingerprint

Step 8: verified

Step 9: Stop

2. Fingerprint Identification and Matching:

Fundamentally, we utilize advanced imaging innovation in securing, putting away, and dissecting the unique mark information.

1. Acquiring Images:

In the image identification process, the first step is to obtain an image as, without it, we cannot perform further steps. To acquire an image of a fingerprint, the person needs to place their finger on the sensor. Once the voter places their finger, on one end of the prism, the total internal reflection occurs through which we can capture the image using the image sensor and lens from another end of the prism. But the image extracted in this step is unprocessed.

The position and placement of the finger play a prominent role in the process of capturing an impression. For intensifying the total internal reflection and capture a fingerprint of good quality with the image sensor, we need to make sure that the finger is placed correctly on the module.

2. Storing the images:

The unprocessed image acquired in the previous step is now processed using image segmentation.

Image Segmentation: The captured images may contain some redundant data and noise along with required data, so we use image segmentation in which we divide the image into many segments called pixels to remove the irrelevant data. To ease the process of image analysis, we use image segmentation.

We use the normalization of an image to get even pixels. Once the pixels are uniform, it results in the formation of an image, and then to reduce or remove the noise present in it, we use the Gabor filter. The thresholding technique is implemented on the filtered image to change it into a binary image, then we compare threshold values and pixel values, if the value is higher than the threshold value then we set the pixel value to 1 else 0. Next, to remove some pixels from the foreground, we implement the thinning process. Finally, all the segments form a single image.

3. Analyzing the Image:

Finally, it shows to whom the user cast the vote.

Using image analysis, we can retrieve all related data from the image for further use. Mostly some electrical machines are used to collect the required data. According to this project, the electrical machine is a fingerprint module through which we capture the fingerprint. The retrieved data from the image can be compared with data stored in different storage devices for identification, authentication, etc.

4.1 WORKING FLOW OF THE PROJECT:

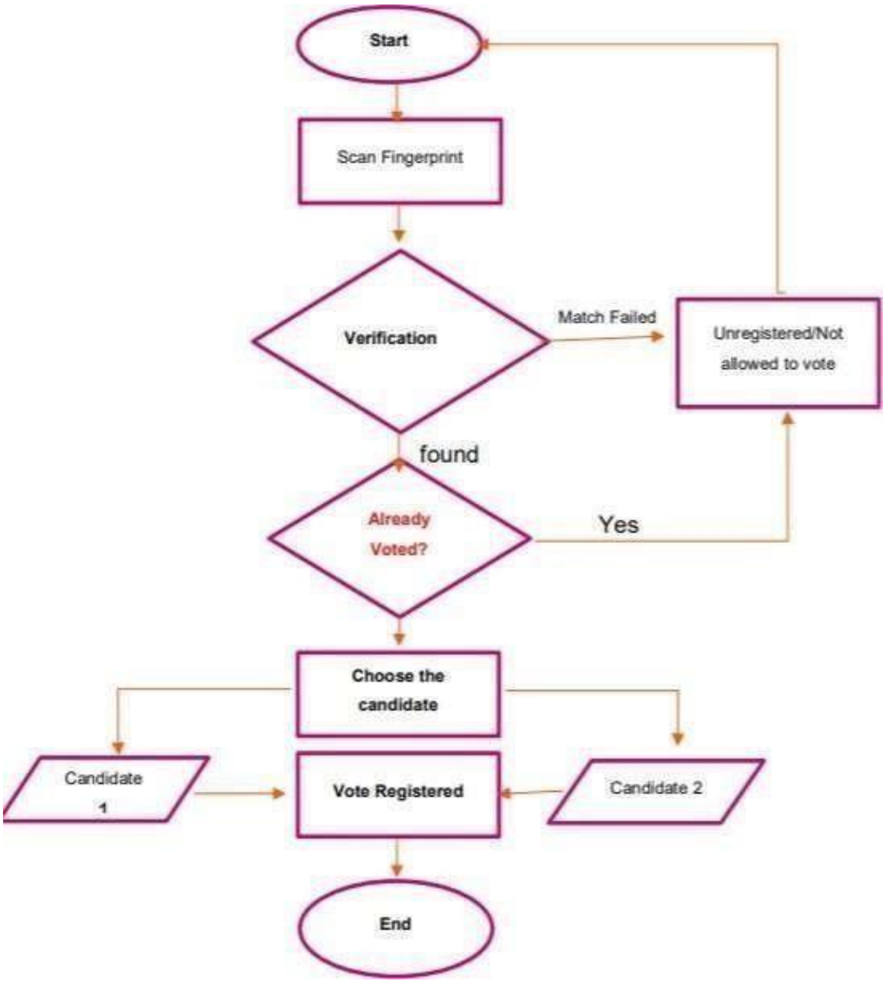


Fig:4.2 Working flow of Finger Print Based Exam Hall Authentication

4.2 EXPERIMENTAL RESULTS:

By simulating this project one will see the display on the LCD, the system writing the name of the project "Fingerprint based exam hall Authentication", few seconds later it will display the following message: "PUT YOUR FINGER" this is just to tell the subject (student) to be authenticated to put her/his finger on the fingerprint scanner for it to be able to read his/her fingerprints, after that if the student is authentic then the following message will be displayed: "ENTER YOUR PASSKEY", this is a password that every registered student got during registration/enrolment phase.

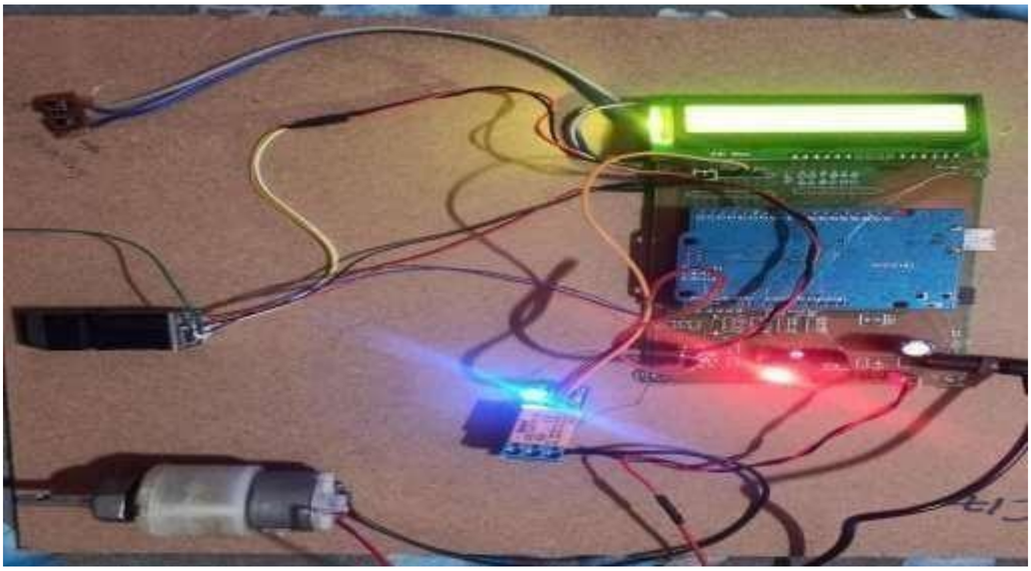


Fig 4.3:Overall design

4.2.1 INTERPRETING EXPERIMENTAL RESULTS:

Fig:4.4 Displays Commands in LCD

Fig 4.5 Finger print Scanner

Fig:4.6 Displayed on LCD “User not authorized”

Fig:4.7 It shows the result obtained after authentication

CHAPTER 5
ADVANTAGES AND LIMITATIONS

5.0 ADVANTAGES

Some advantages of using this system are discussed below:

- Safe: It is an utmost important that the process be secure and no one should be able to tamper with the result before, during authentication process. Since the system is not connected to internet, no online external influence can occur. Also, it is very easy for security personnel to secure the device if needed.
- Cost: The conventional paper system used in Nepal is very expensive because of papers and printings, transportation, staff expenses and it takes several days to count the votes. On the other hand, the biometric fingerprint system is cheap, compact and can store any amount of data with proper upgrade.
- Accuracy and Reliability: There is a very slim chance of errors happening from electronic system so, people can rely on results to be accurate.
- Time Saving: It takes weeks to organize the authentication. This time frame is not peaceful for that area so it is ideal to get result as soon as possible. With electronic device results can be produced is minutes rather than days or weeks

5.1 LIMITATIONS

- The collection of fingerprints of all students is little difficult.
- The personal details of the students does not verifying in this system other than fingerprint.
- Only fingerprints of the student is identified instead of identifying the person.

5.2 APPLICATIONS

This project can be used as authentication and it is a very simple and efficient way to organize a system.

1. Fingerprint is used to save time and get accurate results.
2. This system is trustable and very efficient.
3. This system is very useful in emergency cases.

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

6.0 CONCLUSION

The project Biometric Model for Examination impersonation and Biometric Access is a better substitute for the use of Identity card in verifying users ‘identity Experience has shown the porosity of Identity cards in uniquely identifying individual in the face of sophisticated Forgery technology. The naturalness in the use of fingerprint makes it a reliable access control technique. The fact that a user no longer needs to carry identity cards and other documents for identification explain the ease of use. The Exam hall authentication system using fingerprints. The implemented minutiae extraction is much more accurate and faster than our previous feature-extraction. In our proposed system accurately verify the fingerprint is valid user or not. If valid user then it allow attending the exam else not allowed. In this experimental result shows the proposed method is suitable for all the authentication based application and also it robust.

6.1 FUTURE SCOPE

- Number of active forging may decrease.
- It could be interfaced with the computer and results can be displayed quickly and easily or the result can be stored in the central server whose data can be stored in the backend servers.
- It could be interfaced with the printer so that the hardcopy of the data can be taken.
- It could make the results available at any corner of the world in minimum time.

REFERENCES

[1] A. K. Sharma, A. Raghuwanshi and V. K. Sharma, "Biometric System- A Review," International Journal of Computer Science and Information Technology, vol. 6, p. 1, 2015.

[2]B.Aleksandra, "handle/10024/44684," 2012. [Online]. Available: https://www.theseus.fi/handle/10024/44684.

[3] "technology," 2013. [Online]. Available: http://www.nitgen.com/eng/technology/technology_02.html. [Accessed 18 10 2019].

[4] "Adafruit," Adafruit, [Online]. Available: https://www.adafruit.com/product/751?gclid=CjwKEAj. [Accessed 06 11 2019].

[5] "arduino," 11 11 2019. [Online]. Available: https://store.arduino.cc/.

[6] B. Sigdel, "Thesis," Metropolia UAS, Helsinki, 2017.

[7] "software," 11 11 2019. [Online]. Available: https://www.arduino.cc/.

[8] "components101," [Online]. Available: https://components101.com/microcontrollers/arduino-uno. [Accessed 14 11 2019].

[9] Dejan, "lcd tutorials," [Online]. Available: https://howtomechatronics.com/tutorials/arduino/lcd-tutorial/. [Accessed 08 10 2019].

[10] U. Hussaini, "technobyte," [Online]. Available: https://www.technobyte.org. [Accessed 17 11 2019].

[11] "wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/USB. [Accessed 17 11 2019].

[12] 19 11 2019. [Online]. Available: https://en.wikipedia.org/wiki/Electoral_fraud. [Accessed 20 11 2019].

[13] P. Pokharel, "Study on the Election Campaign Finance of Local, Provincial, and Federal Elections in Nepal," Pradip Pokharel, Kathmandu, 2017.

[14] "Kathmandu Post," 14 05 2017. [Online]. Available: https://kathmandupost.com/national/2017/05/15/ballot-papers-confusing

APPENDICES

```
/*
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2,3);*/

#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);

char buff[200],k=0;
char res[130];


unsigned char enroll[12]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X03,0X01,0X00,0X05}; // ok
unsigned char generate_ch[13]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X04,0x02,0X01,0X00,0X08}; //ok
unsigned char generate_ch1[13]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X04,0x02,0X02,0X00,0X09}; //ok
unsigned char un_cmd[12]={0xef,0x01,0xff,0xff,0xff,0xff, 0x01,0x00,0x03,0x05,0x00,0x09 };
unsigned char store[12]={0xEF,0X01,0XFF,0XFF,0XFF,0XFF,0X01,0X00,0X06,0X06,0X02,0x00};      //ok
unsigned char identify[17]={0xef,0x01,0xff,0xff,0xff,0xff,0x01,0x00,0x08,0x1b,0x01,0x00,0x00,0x01,0x01,0x00,0x27};


void serialFlush(){
  while(Serial.available() > 0) {
  char t = Serial.read();
  }
}
int fpenroll(char);
int fpsearch();

int s1=5,s2=6,m1=4,buz=7;
void setup() {
  char ret;
  pinMode(s1,  INPUT_PULLUP);
  pinMode(s2,  INPUT_PULLUP);
  pinMode(m1, OUTPUT);
 //  pinMode(m2,  OUTPUT);
  pinMode(buz, OUTPUT);

     digitalWrite(buz,HIGH);
     digitalWrite(m1,LOW);
    // digitalWrite(m2,LOW);
     Serial.begin(9600);
 // mySerial.begin(57600);//57600


  lcd.begin(16,2);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("FINGERPRINT BASED EXAM HALL AUTENTICATION");
  delay(2000);

}
 int err =0;
 int idk = 0,eid=0;
void loop()
{
  //digitalWrite(buz,LOW);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("PUT UR FINGER"); delay(1000);
if(digitalRead(s1) == 0)
{
  //Serial.println("Enrolling");
  lcd.clear();lcd.setCursor(0, 0);lcd.print("ENROLLING..");
  if(fpenroll(eid) == -1)
  {
   // Serial.print("Enroll failed:");Serial.print(err);Serial.println("");
    err=0;
    lcd.clear();lcd.setCursor(0, 0);lcd.print("ENROLL FAILED");
    digitalWrite(buz,LOW);delay(1000);digitalWrite(buz,HIGH);
  }
```

```
  else
  {
   lcd.clear();lcd.setCursor(0, 0);lcd.print("ENROLLED:");lcd.print((int)eid);
   //Serial.print("Enroll Success to id:");Serial.print((int)eid);Serial.println("");
   //Serial.print("*E");Serial.print((int)eid);Serial.println("#");
   eid++;
  }
  delay(2000);
 // lcd.clear();lcd.setCursor(0, 0);lcd.print("SELECT OPTION");
}
if(digitalRead(s2) == 0)//identify
{
  lcd.clear();lcd.setCursor(0, 0);lcd.print("SEARCHING..");
   idk = fpsearch();
   if(idk == -1)
   {
    err=0;lcd.clear();lcd.setCursor(0, 0);lcd.print("UNAUTHOISED ");
    lcd.clear();lcd.setCursor(0, 1);lcd.print("PERSON");

      digitalWrite(buz,LOW);
      delay(400);
      digitalWrite(buz,HIGH);

   }
   else
   {
    lcd.clear();lcd.setCursor(0, 0);lcd.print("IDENTIFIED:");lcd.print((int)idk);

    if(idk == 0)
    {
      lcd.clear();lcd.setCursor(0, 0);lcd.print("AUTHORISED ");
      //lcd.clear();
      lcd.setCursor(0, 1);lcd.print("PERSON");
      digitalWrite(m1,HIGH);
      delay(2000);
      digitalWrite(m1,LOW);
    }
    if(idk == 1)
    {
       lcd.clear();lcd.setCursor(0, 0);lcd.print("AUTHORISED ");
      //lcd.clear();
      lcd.setCursor(0, 1);lcd.print("PERSON");
      digitalWrite(m1,HIGH);
      delay(2000);
      digitalWrite(m1,LOW);
    }
    if(idk == 2)
    {
      lcd.clear();lcd.setCursor(0, 0);lcd.print("AUTHORISED ");
      //lcd.clear();
      lcd.setCursor(0, 1);lcd.print("PERSON");
      digitalWrite(m1,HIGH);
      delay(2000);
      digitalWrite(m1,LOW);
    }
    if(idk == 3)
    {
      lcd.clear();lcd.setCursor(0, 0);lcd.print("AUTHORISED ");
      //lcd.clear();
      lcd.setCursor(0, 1);lcd.print("PERSON");
      digitalWrite(m1,HIGH);
      delay(2000);
      digitalWrite(m1,LOW);
    }
     }
   delay(2000);
   //lcd.clear();lcd.setCursor(0, 0);lcd.print("SELECT OPTION");

  }
```

```
}

int ct=0;
char dummy=0x0f;
int fpenroll(char id)
{

  serialFlush();
  //enroll buffer send 12 bytes
  for(int i =0;i<12;i++)
     Serial.write(enroll[i]);
  res[9] = 1;//
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){}
  else{err=1;return -1;}

  //generate ch buffer
  for(int i =0;i<13;i++)
     Serial.write(generate_ch[i]);
  res[9] = 1;
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){}
  else{err=2;return -1;}

 //enroll buffer send 12 bytes
  for(int i =0;i<12;i++)
     Serial.write(enroll[i]);
  res[9] = 1;//
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){}
  else{err=3;return -1;}

  //generate ch1 buffer
  for(int i =0;i<13;i++)
     Serial.write(generate_ch1[i]);
  res[9] = 1;
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){}
  else{err=4;return -1;}

  //uncmd buffer send 12 bytes
  for(int i =0;i<12;i++)
     Serial.write(un_cmd[i]);
  res[9] = 1;//
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){}
  else{err=5;return -1;}

  //store buffer send 12 bytes
  for(int i =0;i<12;i++)
     Serial.write(store[i]);
  dummy = 0x0f+id;
  Serial.write((uint8_t)id);
  Serial.write((uint8_t)0x00);
  Serial.write((uint8_t)dummy);
  res[9] = 1;//
  delay(1000);//wait some time to get replay from r305
  while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
  if(res[9] == 0){return id;}
  else{err=6;return -1;}
}
int fpsearch()
{
  ct=0;
```

```
 serialFlush();
//enroll buffer send 12 bytes
for(int i =0;i<12;i++)
    Serial.write(enroll[i]);
res[9] = 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=1;return -1;}


 //generate ch buffer
for(int i =0;i<13;i++)
    Serial.write(generate_ch[i]);
res[9] = 1;
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=2;return -1;}


//enroll buffer send 12 bytes
for(int i =0;i<17;i++)
    Serial.write(identify[i]);
res[9] = 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){return (int)res[11];}
else{err=1;return -1;}

}
```

```
if( count3 >= 1)
{
lcd.clear();lcd.setCursor(0, 0);lcd.print("DOUBLE VOTE TRYING !!! ");
digitalWrite(buz,LOW);
delay(400);
digitalWrite(buz,HIGH);

}
if(idk == 2 && count3==0)
{

if(digitalRead(s3) ==0)//trs
{
trs++;
lcd.clear();lcd.setCursor(0, 0);lcd.print("YOUR VOTE FOR TRS !!! ");delay(2000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("THANK YOU FOR YOUR VOTE!!! ");delay(2000);
}
if(digitalRead(s4) == 0)//BJP
{
bjp++;
lcd.clear();lcd.setCursor(0, 0);lcd.print("YOUR VOTE FOR BJP !!! ");delay(2000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("THANK YOU FOR YOUR VOTE!!! ");delay(2000);
}
count3++;
}
delay(2000);
}
if(digitalRead(s5) == 0)//result
{
lcd.clear();lcd.setCursor(0, 0);lcd.print("RESULTS !!! ");
lcd.setCursor(0, 1);lcd.print("TRS:");lcd.print(trs);
lcd.setCursor(8, 1);lcd.print("BJP:");lcd.print(bjp);
delay(4000);

}
}

int ct=0;
char dummy=0x0f; int
fpenroll(char id)
{

serialFlush();
        //enroll buffer send 12 bytes
```

```
Serial.write(enroll[i]); res[9]
= 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=1;return -1;}

//generate ch buffer
for(int i =0;i<13;i++)
Serial.write(generate_ch[i]); res[9]
= 1;
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=2;return -1;}

//enroll buffer send 12 bytes
for(int i =0;i<12;i++)
Serial.write(enroll[i]); res[9]
= 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=3;return -1;}

//generate ch1 buffer
for(int i =0;i<13;i++)
Serial.write(generate_ch1[i]); res[9]
= 1;
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=4;return -1;}

//uncmd buffer send 12 bytes
for(int i =0;i<12;i++)
Serial.write(un_cmd[i]); res[9]
= 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=5;return -1;}

//store buffer send 12 bytes
```

```
Serial.write(store[i]); dummy =
0x0f+id; Serial.write((uint8_t)id);
Serial.write((uint8_t)0x00);
Serial.write((uint8_t)dummy);
res[9] = 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){return id;}
else{err=6;return -1;}
}
int fpsearch()
{
ct=0;
serialFlush();
//enroll buffer send 12 bytes
for(int i =0;i<12;i++)
Serial.write(enroll[i]); res[9]
= 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=1;return
//generate ch buffer
for(int i =0;i<13;i++)
Serial.write(generate_ch[i]); res[9]
= 1;
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){}
else{err=2;return -1;}
//enroll buffer send 12 bytes
for(int i =0;i<17;i++)
Serial.write(identify[i]); res[9]
= 1;//
delay(1000);//wait some time to get replay from r305
while(Serial.available()){res[ct] = Serial.read();ct++;}ct=0;
if(res[9] == 0){return (int)res[11];}
else{err=1;return -1;}

}
```

A

MAJOR PROJECT REPORT

On

IOT BASED LIQUID LEVEL MONITORING SYSTEM

Submitted by

1)Ms.D.Jahnavi (17K81A04E2)

2)Ms.B.Divya (17K81A04C5)

3)Ms.R.Navitha (18K85A0431)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Ms. P. Pushpa

M.Tech

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST.MARTIN'S ENGINEERING COLLEGE

An Autonomous Institute

Dhulapally, Secunderabad – 500 100

NBA &NAAC A+ Accredited

JUNE 2021

St.MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

BONAFIDE CERTIFICATE

This is to certify that the project entitled **IOT Based Liquid Level Monitoring System**, is being submitted by **1.Ms. D. Jahnavi (17K81A04E2), 2. Ms. B. Divya (17K81A04C5), 3. Ms. R. Navitha (18K85A0431)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Internal Guide | **Head of the Department**

Ms.P. Pushpa
Department of Electronics and
communication engineering

Dr. Hari Krishna
Department of Electronics and
communication engineering

Internal Examiner | **External Examiner**

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **B.DIVYA** WITH ROLL NO. **17K81A04C5, D.JAHNAVI** WITH ROLL NO. **17K81A04E2**, **NAVITHA** WITH ROLL NO. **18K85A0431,** OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**IOT BASED LIQUID LEVEL MONITORING SYSTEM**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of **Bachelor of Technology in Department of Electronics and Communication Engineering**,session:2017–2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled '**IOT Based Liquid Level Monitoring System**' is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

D. Jahnavi	B. Divya	R. Navitha
17K81A04E2	17K81A04C5	18K85A0431

ACKNOWLEDGMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

We extended our deep sense of gratitude to Principal**, Dr. P. SANTOSH KUMAR PATRA**, St. Martin's Engineering College, Dhulapally, for permitting us to undertake this project.

We are also thankful to **Dr. B. Hari Krishna, Head & Professor**, of the department Electronics and Communication Engineering, St. Martin's Engineering College, Dhulapally, for his support and guidance throughout our project and as well as our **Project Coordinator** , **Dr. K. Nishakar**, **Professor** in Electronics and Communication Engineering, for his valuable support.

We would like to express our sincere gratitude and indebtedness to our **Project Supervisor, Ms. P. Pushpa**, **Assistant Professor**, Electronics and Communication Engineering , St. Martin's Engineering College, Dhulapally, for her support and guidance throughout our project.

Finally, we express thanks to all those who have helped us successfully to completing this project. Furthermore, we would like to thank our family and friends for their moral support and encouragement.

We express thanks to all those who have helped us in successfully completing the project.**.**

1. D. Jahnavi
2. B.Divya
3. R.Navitha

ABSTRACT

This project IOT Liquid Level Monitoring system is a very innovative system which will inform the users about the level of liquid and will prevent it from overflowing. To demonstrate this the system makes use of 4 containers. For this the system uses ultrasonic sensors placed over the containers to detect the liquid level and compare it with the container's depth. The system makes use of AVR family microcontroller, LCD screen, WiFi modem for sending data and a buzzer. The system is powered by a 12V transformer. The LCD screen is used to display the status of the level of liquid in the containers. Whereas a web page is built to show the status to the user monitoring it. The web page gives a graphical view of the containers and highlights the liquid level in color in order to show the level of liquid. The LCD screen shows the status of the liquid level. The system puts on the buzzer when the level of liquid collected crosses the set limit. Thus this system helps to prevent the wastage of water by informing about the liquid levels of the containers by providing graphical image of the containers via a web page.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

INTRODUCTION:

Imagine the usefulness of being able to switch on the air conditioner system ten minutes before you get home on a hot afternoon. How about security system to detect smoke, burglary attempts alerted to you. These imaginations build the idea of Water/Liquid Level Monitoring that has no end. The sophisticated liquid Level Monitoring systems of course record the usage of liquid. Technology nowadays has become an integrated part of people's lives. It has, and continues to influence many aspects of daily life and has allowed better social interaction, ease of transportation,

From the time of its introduction, the number of people that use mobile phones and the internet to communicate with other people has increased dramatically to become one of the major means of communication. People with the help of smart phones can now connect to the internet with out the need for a computer, while still offering the same functionality but through different means.With the introduction of advanced software and hardware devices, smart phones are now powerful devices and have become an important part of people’s daily lives. A major aspect is how the Smartphone is able to connect and communicate with other devices.

A field that is recently gaining popularity is Liquid Level Monitoring which can also use smart phones as information or functionality hubs. The present invention relates to the field of Internet of things (IOT) in Liquid Level Monitoring. The goal of this research is to develop a smart device for estimating accurately the liquid level and prevent overflowing and leakage. The recent advancements in the area of IoT has provided solutions to measure the quantity and level of liquid in liquid transmission system using sensor based wireless technology.

1.1 OBJECTIVE OF THE PROJECT

The main aim of this system is to monitor the liquid level at industrial areas so that they help in detecting the wastage of water and measures can be taken to avoid unnecessary overflowing of water in the areas where monitoring is a difficult task.

1.2 AIM OF THE PROJECT

The primary aim for doing our project is to present a handy module consisting of sensors and micro systems which will inform the users about the level of liquid and will prevent it from overflowing. To achieve this, the following objectives have to be completed.

- Studying the components and its working.
- Develop a kit with the all the components needed for its functioning.
- Studying the articles of various publishers.
- Designing and developing a system which can communicate to the server.
- Testing and maintaining the implemented system.

1.3 MOTIVATION OF THE PROJECT

Many essential liquids need to be measured and monitored as its leakage and overflowing might result in accidents and most probably disasters. So few industrial liquids are to be carefully monitored as they are costly, imported and requires high maintenance to avoid any liquid loss. This has motivated to develop Liquid Level Monitoring System Using Internet of Things. Nowadays, liquid level monitoring system plays a vital role in water, oil, automotive and also gas industries.Many industrial and process control applications involve the monitoring of the level of a liquid in a vessel, or storage tank.

1.4 EXISTENCE METHOD

Water Monitoring System has ultrasonic sensor and it is used to detect the level of the water. The level of the water can be measured by means of centimetre. The processor process the data from the level of water and send it to the Arduino and display the level using LCD display.

Ultrasonic sensor detects the level of water and displays the level on the LCD. When the detected level is indicated as “FULL” the buzzer will produce sound. This project saves time and money. Helps in preventing accidents that occur at railway bridges. The input to the Arduino is ultrasonic sensor. Water level will be detected by ultrasonic sensor. The outputs are buzzer and LCD.

1.5 PROPOSED METHOD

This project IOT Liquid Level Monitoring system is a very innovative system which will inform the users about the level of liquid and will prevent it from overflowing. To demonstrate this the system makes use of 4 containers. For this the system uses ultrasonic sensors placed over the containers to detect the liquid level and compare it with the container’s depth. The system makes use of AVR family microcontroller, LCD screen, WiFi modem for sending data and a buzzer. The system is powered by a 12V transformer. The LCD screen is used to display the status of the level of liquid in the containers. Whereas a web page is built to show the status to the user monitoring it. The web page gives a graphical view of the containers and highlights the liquid level in color in order to show the level of liquid. The LCD screen shows the status of the liquid level. The system puts on the buzzer when the level of liquid collected crosses the set limit.

1.6 LITERATURE SURVEY

Liquid Level Monitoring can be found in many areas since before.

Various types of Liquid Level Monitoring system based on the control are:

1. Individual systems: This is a very popular type of systems. Here the whole model is implemented on a targeted single source system.

2. Large control systems: Here the liquid level is implemented on a very large scale basis and huge amounts of sensors are used.

3. Central Control systems: Computerized systems programmed to handle all the functions of multiple utilities like air conditioning system or home entertainment systems, refrigerators all at the same instant regardless of your presence. Control system can be accessed through telephone or internet from any corner of the world.

Based on the method of automation various types are as follows:

1. Bluetooth based Liquid Level Monitoring: Here Bluetooth plays a major role in alerting the anomalies.

2. Remote Liquid Level Monitoring: In this, the system is controlled remotely.

3.Automatic Liquid Level Monitoring: The system is programmed to automatically perform some defined actions.

1.7 ORGANIZATIONAL THESIS

In this project, firstly we learn about the block diagram part to implement this application. This gives a very clear idea about the further procedure. Then we put forward towards the schematics of each and every module in the project. After having clear overview about the project with the help of schematics and block diagram, we go through the hardware modules in the project one after the other. Firstly, micro controller is the base module of the project. We interface other hardware modules with the help of micro controller only. Ultrasonic sensors are used for the identification of distance between the sensor and object, depth of the object, etc . In this project WiFi module is used to transfer the updated result to the server. LCD is used for the displaying of the data regarding the level of water identified . The distance of the water level is displayed on the LCD. When the level of the water identified is “FULL” the buzzer sets “ON”. Now, the software implementation is done using Arduino IDE. The results are followed by the conclusion, applications and future scope of the project.

CHAPTER 2

BLOCK DIAGRAM

2.1 BLOCK DIAGRAM

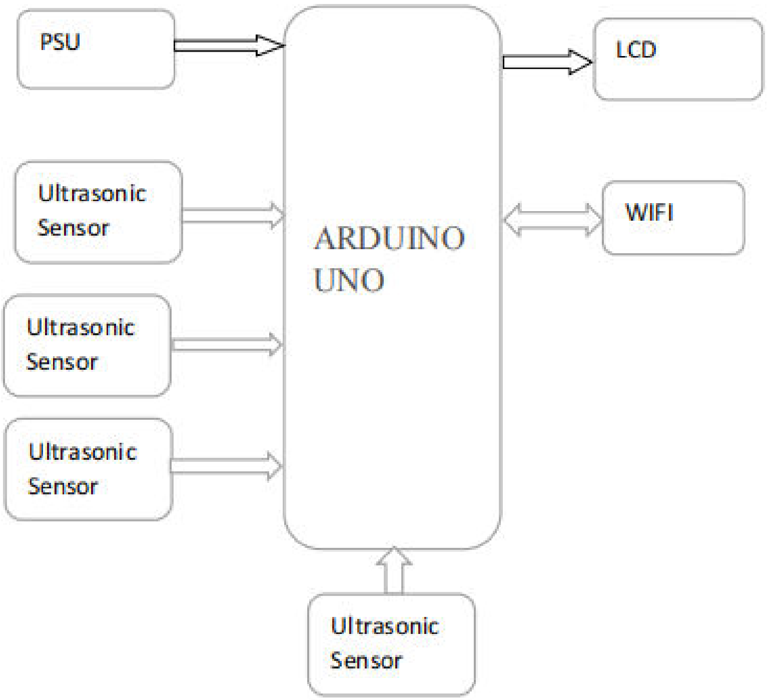


Figure:2.1 Block Diagram

MICROCONTROLLER

Arduino is open source software. Arduino UNO is a micro-controller, ATmega328P contain in it. It has 14 digital i/o pins and 6 analog input pin, a 16 MHz quartz crystal, a USB connection, a power jack and a reset button. Simply connect to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. Embedded C program is used for coding.

WIFI MODULE

The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all WiFi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

ULTRA SONIC SENSOR

Ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal.Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).In order to calculate the distance between the sensor and the object, the sensor measures the time it takes between the emission of the sound by the transmitter to its contact with the receiver. The formula for this calculation is **D = ½ T x C** (where D is the distance, T is the time, and C is the speed of sound ~ 343 meters/second)

LIQUID CRYSTAL DISPLAY(LCD)

Liquid-crystal display (**LCD**) is a flat panel display, electronic visual display that uses the light modulation properties of liquid crystals. Liquid crystals do not emit light directly. LCD’s are available to display arbitrary images or fixed images which can be displayed or hidden, such as preset words, digits, and 7-segment displays as in a digital clock. They use the same basic technology, except that arbitrary images are made up of a large number of small pixels, while other displays have larger element.

POWER SUPPLY

Every electrical and electronic device that we use in our day-to-day life will require a power supply. In general, we use an AC supply of 230V 50Hz, but this power has to be changed into the required form with required values or voltage range for providing power supply to different types of devices. There are various types of power electronic converters such as step-down converter, step-up converter, voltage stabilizer, AC to DC converter, DC to DC converter, DC to AC converter, and so on. For example, consider the microcontrollers that are used frequently for developing many embedded systems' based projects and kits used in real-time applications. These microcontrollers require a 5V DC supply, so the AC 230V needs to be converted into 5V DC using the step-down converter in their power supply circuit. Power supply circuit, the name itself indicates that this circuit is used to supply the power to other electrical and electronic circuits or devices. There are different types of power supply circuits based on the power they are used to provide for devices. For example, the micro-controller based circuits, usually the 5V DC regulated power supply circuits, are used, which can be designed using different techniques for converting the available 230V AC power to 5V DC power. Generally the converters with output voltage less than the input voltage are called as step-down converters.

BUZZER

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke. Nowadays, it is more popular to use a ceramic-based piezoelectric sounder like a Son alert which makes a high-pitched tone. Usually these were hooked up to "driver" circuits, which varied the pitch of the sound or pulsed the sound on and off.

2.2 SCHEMATIC DIAGRAM

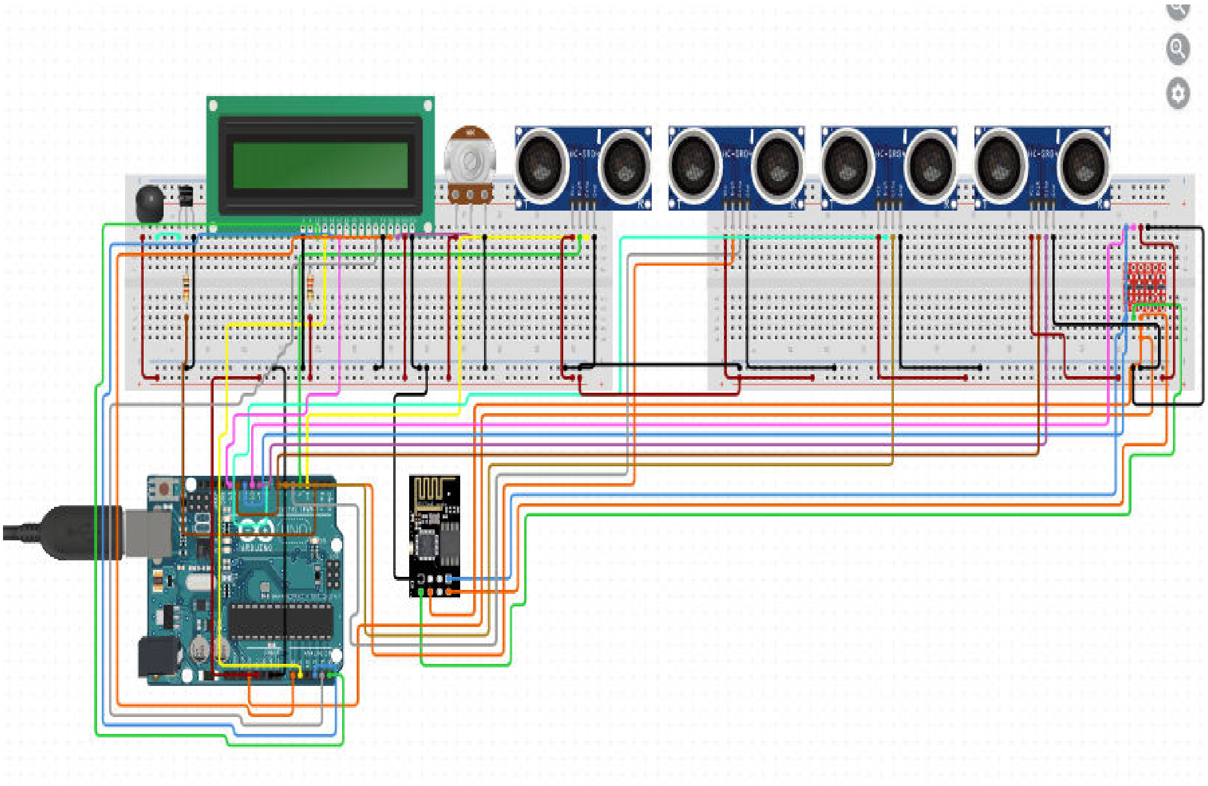


Figure : 2.2 Schematic Diagram

SCHEMATIC DIAGRAM DESCRIPTION

In this project, we are going to make a Liquid level monitoring system which uses the ultrasonic sensor to monitor the level of water. We have used Arduino UNO as the heart of the circuit. All the hardware required such as WIFI Module, ultrasonic sensor, Buzzer, LED, LCD display are connected to the Arduino UNO by using the GPIO pins present on the Arduino UNO board. LCD display shows HIGH level whenever the water is full. The buzzer connected to Arduino also sets “ON”. The microcontroller collects the data and displays on the LCD display. The WIFI Module is an ESP2866 microcontroller which transfers the information to the Think speak server. The project will use Arduino UNO with AtmegaP328 microcontroller as heart of the circuit. ATmega328P has 28 pins. The Power requirement of LPC2148 Microcontroller is 5VDC and VSS ground. In this project, we have microcontroller which collects the data and displays it on to the LCD display as well as on any system that has internet access. Firstly, connect all the components to the Arduino UNO and turn on the UNO board by using 5V supply given from USB or LAN port. Switch on the WIFI in order to transfer the information to server. To check the output switch on the power supply. Now place an object in front of sensor until the ultrasonic sensor senses the level. The level is detected and can be viewed graphically as well as digitally on any system by login to the think speak portal. The buzzer starts buzzing as the level is high and the LED glows indicating the water is full. As the buzzer and LED are set high the output of Arduino Uno also becomes high. This process continues till the water level is controlled.

The hardware modules are interfaced to the micro controller as follows:

LCD is interfaced from 8^{th} to 13^{th} pin of the Arduino.

The ultrasonic sensor1 is given to the 4^{th} pin of the Arduino.

The ultrasonic sensor1 is given to the 5^{th} pin of the Arduino.

The ultrasonic sensor2 is given to the 6^{th} pin of the Arduino.

The ultrasonic sensor2 is given to the 7^{th} pin of the Arduino.

The ultrasonic sensor3 trig pin is given to analog pin A4.

The ultrasonic sensor3 echo pin is given to analog pin A5.

The ultrasonic sensor4 trig pin is given to analog pin A1.

The ultrasonic sensor4 echo pin is given to analog pin A2.

The Buzzer is given to the 7^{th} pin of the Arduino.

CHAPTER 3

HARDWARE DESCRIPTION

HARDWARE MODULE

The hardware and software components used for the implementation of Water Level Monitoring System at Railway Bridges are

1. Arduino UNO.

2. WIFI Module

3. Regulated Power Supply

4. LCD display.

5. Ultrasonic Sensor.

6. Buzzer.

7. LED indicator.

8. Voltage Regulator

3.1 ARDUINO UNO

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message and turn it into an output activating a motor, turning on an led, publishing as it reached a wider community, the Arduino board started something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based Wiring), the Arduino Software (IDE), based on Processing. Arduino was born at the area Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming wearable, 3dprinting, and embedded environments. All Arduino boards are completely open-source, empowering from everyday objects to complex scientific instruments. Professional gathered around open-source plat form, The software too is open-source, it is growing through the contributions of users worldwide, the Arduino board is shown in below figure.

Figure 3.1 Arduino UNO board

WHY ARDUINO?

Arduino UNO is inexpensive. Arduino is cross platform, its IDE can run on multiple Operating Systems such as Windows, Macintosh, Linux and many more. It has simple and clear programming environment. Its software and hardware are open-source and extensible, that is, Arduino programming can be expanded through C++ libraries and can also leap to AVR 'C' language programming. It is simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove the chemistry and physics principles or to get started with programming and robotics.

Designers and the architects build interactive prototypes, musicians, artists use it for installations and to experiment with new musical instruments. Makers of course use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers can start tinkering just following the step by step instructions of a kit or sharing ideas online with other members of the Arduino community.

There many other microcontrollers platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Fidgets, MIT's Handy board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems.

INEXPENSIVE

Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50

SIMPLE, CLEAR PROGRAMMING ENVIRONMENT

The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

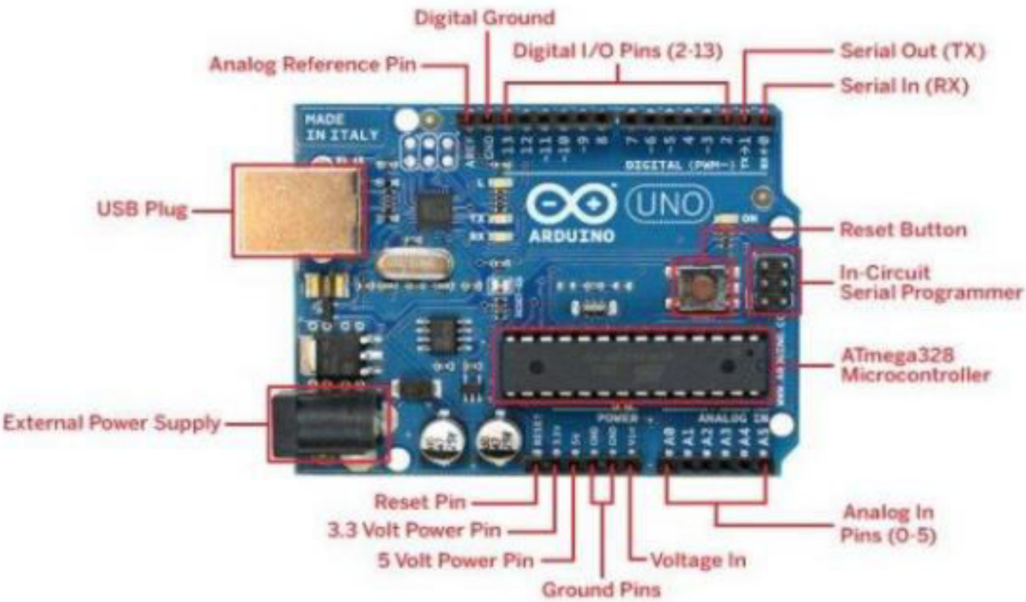


Figure 3.2 Arduino Board Description

OPEN SOURCE AND EXTENSIBLE SOFTWARE

The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

OPEN SOURCE AND EXTENSIBLE HARDWARE

The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money and the parts of Arduino board is shown below.

FEATURES OF ARDUINO UNO

Micro controller	Atmega328
Operating voltage	5V
Input Voltage(recommended)	7-12V
Input Voltage	6-20V
Digital I/O	14(of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 Ma
DC Current for 3.3V Pin	50 Ma
Flash Memory	32 KB(Atmega328 of which 0.5 KB used by bootloader)
SRAM	2 KB (ATmega32)
EEPROM	1 KB(ATmeha328)
Clock Speed	16 MHz

Table 3.1 : Features of Arduino

ADVANTAGES

1. Inexpensive -Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50.
2. Cross-platform - The Arduino software runs on Windows, Macintosh OSX, and the Linux operating systems. Most microcontroller systems are limited to Windows.

3.Simple programming environment- Here the Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino.

4. Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

4. Open source and extensible hardware - The Arduino is based on Atmel's ATMEGA8 and ATMEGA168 microcontrollers. The plans for the modules are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users that can build the breadboard version of the module in order to understand how it works and save money.

APPLICATIONS OF ARDUINO

- It is an easy USB interface. This allows interface with USB as this is like a serial device.

- The chip on the board plugs straight into your USB port and supports on your computer as a virtual serial port. The benefit of this setup is that serial communication is an extremely easy protocol which is time-tested and USB makes connection with modern computers and makes it comfortable.

• It is easy-to-find the microcontroller brain which is the ATmega328 chip. It has more number of hardware features like timers, external and internal interrupts, PWM pins and multiple sleep modes.

• It is an open source design and there is an advantage of being open source is that it has a large community of people using and troubleshooting it. This makes easy to help in debugging projects.

• It is 16MHz clock which is fast enough for most applications and does not speeds up the microcontroller.

• It is very convenient to manage power inside it and it had a feature of built-in voltage regulation. This also be powered directly off a USB port without any external power. You can connect an external power source of up to 12v and this regulates it to both 5v and 3.3v.

• 13 digital pins and 6 analog pins. This sort of pins allows you to connect hardware to your Arduino Uno board externally. These pins are used as a key for extending the computing capability of the Arduino Uno into the real world.

This has an ICSP connector for bypassing the USB port and interfacing the Arduino directly as a serial device. This port is necessary to re-boot load your chip if it corrupts and can no longer used to your computer.

• It has a 32 KB of flash memory for storing your code.

• An on-board led is attached to digital pin 13 to make fast the debugging of code and to make the debug process easy finally it has the button to reset.

POWER

The Arduino Uno can be powered via through USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. the power pins are as follows.

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

- **GND.** Ground pins.

INTRODUCTION TO ATMEGA328

ATMEGA328 is an eight (8) bit Microcontroller. It can handle the data sized of up to eight (8) bits. It is an AVR based micro-controller. Its built in internal memory is around 32KB. It operates ranging from 3.3V to 5V. It has an ability to store the data even when the electrical supply is removed from its biasing terminals. it is the excellent features include the cost efficiency and low power dissipation, programming lock for security purposes, real timer counter with the separate oscillator and It is generally or normally used in Embedded Systems applications. ATmega-328 is shown in the figure given below.

Figure 3.3 ATMega328 Microcontroller

ARDUINO UNO ATMEGA328

The Arduino UNO is a micro controller board based on the ATMega. UNP means one in Italian and is named to mark the upcoming release of the given Arduino 1.0 . It has digital input/output pins, 6 analog inputs, 16MHz ceramic resonator. A USB connection. A power jack, An ICSP header and a reset button.it contains information everything needed to support the microcontroller. simply connect it to a computer with a USB cable or power it with a ac-to-dc adapter or battery to get started. The UNO differs from all preceding boards in that it does not use the FTDI USB-to serial driver chip.

ARCHITECTURE AND WORKING OF ARDUINO UNO

The Arduino UNO can be also powered via the USB connection or with an external power supply.the power source is selected automatically.External (non-USB) power can come either from an AC-to-DC adapter or battery. The adapter can connected by plugging a 2.1mm center positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the power connector. The board can operate on an external supply of 6to 20 volts. If supplied with less than 7v however, the 5v pin may supply less than five volts and the board may be unstable. If using more than 12v.the voltage regulator may overheat and damage the board.the 21recommended range is 7 to 12 volts. The processor of the Arduino board uses the Harvard architecture where the program code and program data have separate memory. It consists of two memories such as program memory and data memory. Where in the data is stored in data memory and the code is stored in the flash program memory. The Atmega328 microcontroller has 32kb of flash memory, 2kb of SRAM 1kb of EPROM and operates with a 16MHz clock speed. Architecture of Atmega328 is shown below.

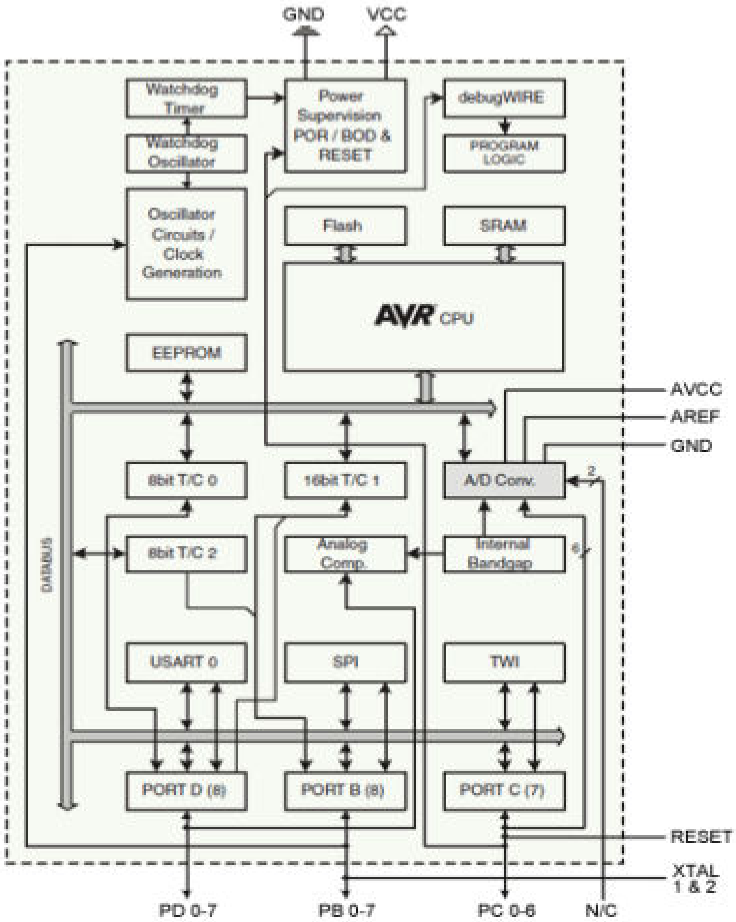


Figure 3.4 ATMega328 Architecture

ATMEGA328 PINS DESCRIPTION

Functions associated with the pins must be known in order to use the device appropriately.ATmega-328 pins are divided into different ports which are given in detail below.

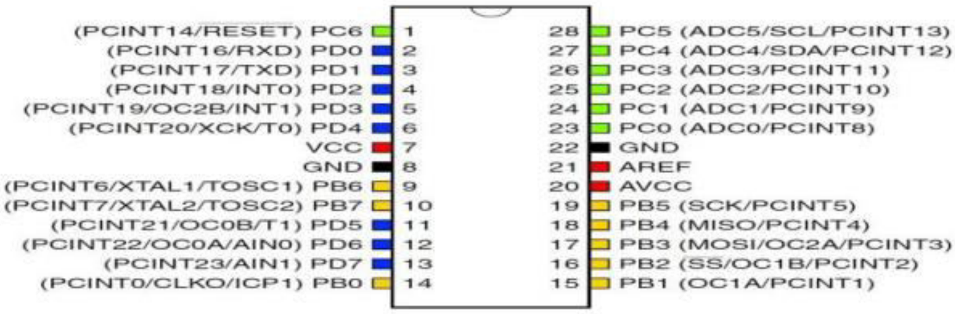


Figure 3.5 ATMega328 ports

AVCC is a supply voltage pin for analog to digital converter.

VCC is a digital voltage supply.

GND denotes Ground and it has a 0V.

Port A consists of the pins from **PA0** to **PA7.** These pins serve as analog input to analog to digital converters. If analog to digital converter is not used, **port A** acts as an eight (8) bit bidirectional input/output port.

Port B consists of the pins from **PB0** to **PB7.** This port is an 8 bit bidirectional port having an internal pull-up resistor.

Port C consists of the pins from **PC0** to **PC7.** The output buffers of **port C**

Port D consists of the pins from **PD0** to **PD7.** It is also an 8 bit input/output port having an internal pull-up resistor.All of the AVR ports are shown in figure given below.

3.2 WIFI MODULE

The ESP8266 is a low-cost Wi-Fi microchip, with a full TCP/IP stack and microcontroller capability, produced by Espressif Systems[1] in Shanghai, China The ESP8285 is an ESP8266 with 1 MiB of built-in flash, allowing the building of single-chip devices capable of connecting to Wi-Fi.

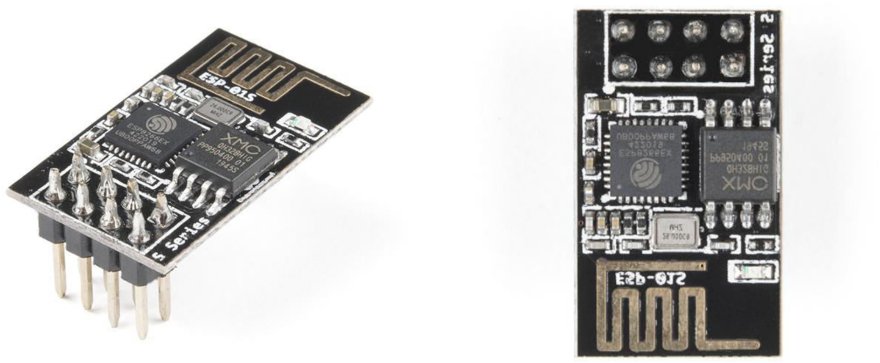


Figure 3.6 ESP8266 WIFI Module

This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections using Hayes-style commands. The very low price and the fact that there were very few external components on the module, which suggested that it could eventually be very inexpensive in volume, attracted many hackers to explore the module, the chip, and the software on it.

These microcontroller chips have been succeeded by the ESP32 family of devices, including the pin-compatible ESP32-C3.

Processor: L106 32-bit RISC microprocessor core based on the Tensilica Xtensa Diamond Standard 106Micro running at 80 MHz[5]

Memory:[*citation needed*]

32 KiB instruction RAM

32 KiB instruction cache RAM

80 KiB user-data RAM

16 KiB ETS system-data RAM

External QSPI flash: up to 16 MiB is supported (512 KiB to 4 MiB typically included)

The pin out is as follows for the common ESP-01 module:

1. GND, Ground (0 V)
2. GPIO 2, General-purpose input/output No. 2
3. GPIO 0, General-purpose input/output No. 0

4. RX, Receive data in, also GPIO3
5. VCC, Voltage (+3.3 V; can handle up to 3.6 V)
6. RST, Reset
7. CH_PD, Chip power-down
8. TX, Transmit data out, also GPIO1

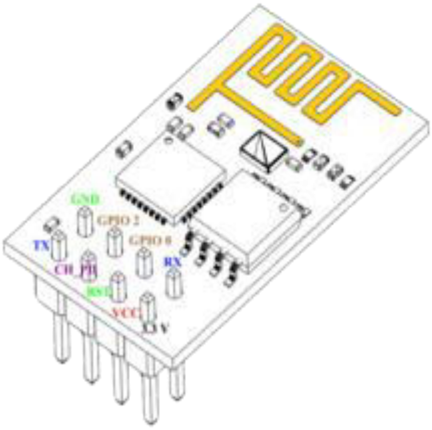


Figure 3.7 ESP8266 pin configuration

The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all WiFi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existence interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support. In the Documents section below you will find many resources to aid you in using the ESP8266, even instructions on how to transform this module into an IoT (Internet of Things) solution!

Note: The ESP8266 Module is not capable of 5-3V logic shifting and will require an external Logic Level Converter. Should not power it directly from 5V developed board.

3.3 REGULATED POWER SUPPLY

A variable regulated power supply, also called a variable bench power supply, is one where you can continuously adjust the output voltage to your requirements. Varying the output of the power supply is the recommended way to test a project after having double checked parts placement against circuit drawings and the parts placement guide.

This type of regulation is ideal for having a simple variable bench power supply. Actually this is quite important because one of the first projects a hobbyist should undertake is the construction of a variable regulated power supply. While a dedicated supply is quite handy, it's much handier to have a variable supply on hand, especially for testing.

Mainly the microcontroller needs 5 volt power supply. To use these parts we need to build a regulated 5 volt source. Usually you start with an unregulated power To make a 5 volt power supply, we use a 7805 voltage regulator IC (Integrated Circuit).

V_{out} range	1.25V – 37V
V_{in} – V_{out} difference	3V – 40V
Operation ambient temperature	0 – 125°C
Output I_{max}	<1.5A
Minimum Load $Current_{max}$	10Ma

Table:3.2 Circuit features

Block Diagram

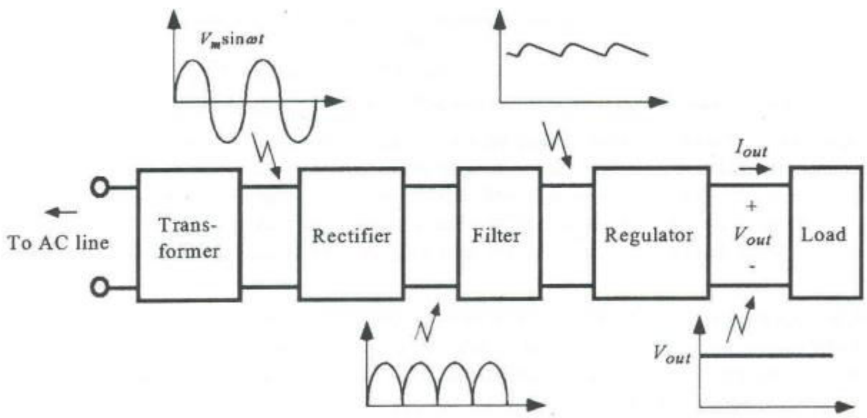


Figure 3.8 Block diagram of RPS

Power supply

Every electrical and electronic device that we use in our day-to-day life will require a power supply. In general, we use an AC supply of 230V 50Hz, but this power has to be changed into the required form with required values or voltage range for providing power supply to different types of devices. There are various types of power electronic converters such as step-down converter, step-up converter, voltage stabilizer, AC to DC converter, DC to DC converter, DC to AC converter, and so on. For example, consider the microcontrollers that are used frequently for developing many embedded systems' based projects and kits used in real-time applications. These microcontrollers require a 5V DC supply, so the AC 230V needs to be converted into 5V DC using the step-down converter in their power supply circuit.

Power supply circuit, the name itself indicates that this circuit is used to supply the power to other electrical and electronic circuits or devices. There are different types of power supply circuits based on the power they are used to provide for devices. For example, the micro-controller based circuits, usually the 5V DC regulated power supply circuits, are used, which can be designed using different techniques for converting the available 230V AC power to 5V DC power. Generally the converters with output voltage less than the input voltage are called as step-down converters.

Step Down the Voltage Level

The step-down converters are used for converting the high voltage into low voltage. The converter with output voltage less than the input voltage is called as a step-down converter, and the converter with output voltage greater than the input voltage is called as step-up converter. There are step-up and step-down transformers which are used to step up or step down the voltage levels. 230V AC is converted into 12V AC using a step-down transformer. 12V output of step down transformer is an RMS value and its peak value is given by the product of square root of two with RMS value, which is approximately 17V.

Figure 3.9 Step-down Transformer

Step-down transformer consists of two winding's, namely primary and secondary winding's where primary can be designed using a less-gauge wire with more number of turns as it is used for carrying low-current high-voltage power, and the secondary winding using a high-gauge wire with less number of turns as it is used for carrying high-current low-voltage power. Transformers works on the principle of Faraday's laws of electromagnetic induction.

Convert AC to DC

230V AC power is converted into 12V AC (12V RMS value wherein the peak value is around 17V), but the required power is 5V DC; for this purpose, 17V AC power must be primarily converted into DC power then it can be stepped down to the 5V DC.

But first and foremost, we must know how to convert AC to DC? AC power can be converted into DC using one of the power electronic converters called as Rectifier. There are different types of rectifiers, such as half-wave rectifier, full-wave rectifier and bridge rectifier. Due to the advantages of the bridge rectifier over the half and full wave rectifier, the bridge rectifier is frequently used for converting AC to DC.

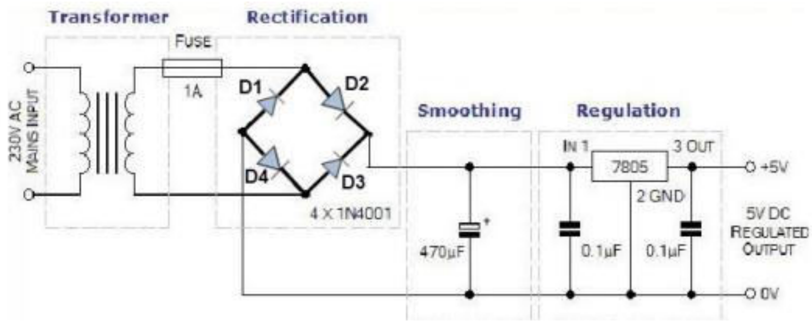


Figure 3.10 AC to DC converter

Bridge rectifier consists of four diodes which are connected in the form a bridge. We know that the diode is an uncontrolled rectifier which will conduct only forward bias and will not conduct during the reverse bias. If the diode anode voltage is greater than the cathode voltage then the diode is said to be in forward bias. During positive half cycle, diodes D2 and D4 will conduct and during negative half cycle diodes D1 and D3 will conduct. Thus, AC is converted into DC; here the obtained is not a pure DC as it consists of pulses. Hence, it is called as pulsating DC power. But voltage drop across the diodes is (2*0.7V) 1.4V; therefore, the peak voltage at the output of this rectifier circuit is 15V (17-1.4) approx.

Smoothing the Ripples using Filter

15V DC can be regulated into 5V DC using a step-down converter, but before this, it is required to obtain pure DC power. The output of the diode bridge is a DC consisting of ripples also called as pulsating DC. This pulsating DC can be filtered using an inductor filter or a capacitor filter or a resistor-capacitor-coupled filter for removing the ripples. Consider a capacitor filter which is frequently used in most cases for smoothing.

Figure 3.11 Filter

We know that a capacitor is an energy storing element. In the circuit, capacitor stores energy while the input increases from zero to a peak value and, while the supply voltage decreases from peak value to zero, capacitor starts discharging. This charging and discharging of the capacitor will make the pulsating DC into pure DC, as shown in figure.

REGULATING 12V DC INTO 5V DC USING VOLTAGE REGULATOR

15V DC voltage can be stepped down to 5V DC voltage using a DC step-down converter called as voltage regulator IC7805. The first two digits ‘78’ of IC7805 voltage regulator represent positive series voltage regulators and the last two digits ‘05’ represents the output voltage of the voltage regulator.

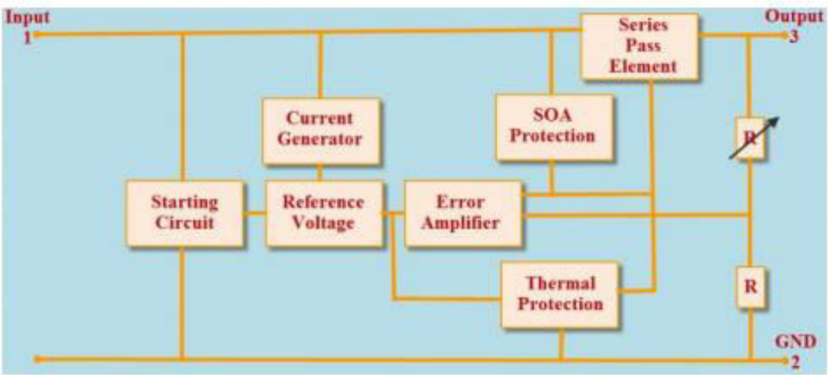


Figure 3.12 Voltage Regulator

The block diagram of IC7805 voltage regulator is shown in the figure consists of an operating amplifier acting as error amplifier, zener diode used for providing voltage reference, as shown in the figure.

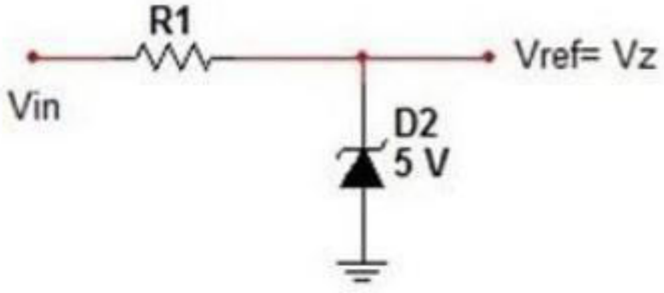


Figure 3.13 Zener Diode as Voltage Reference

Transistor as a series pass element used for dissipating extra energy as heat; SOA protection (Safe Operating Area) and heat sink are used for thermal protection in case of excessive supply voltages. In general, an IC7805 regulator can withstand voltage ranging from 7.2V to 35V and gives maximum efficiency of 7.2V voltage and if the voltage exceeds 7.2V, then there is loss of energy in the form of heat. To protect the regulator from over heat, thermal protection is provided using a heat sink. Thus, a 5V DC is obtained from 230V AC power.

3.4 LCD DISPALY (16x2)

Figure: 3.14 LCD Display

LIQUID CRYSTAL DISPLAY

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCD’s connected to the controllers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines(RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

FEATURES:

(1) Interface with either 4-bit or 8-bit microprocessor.

(2) Display data RAM

(3) 80x□8 bits (80 characters).

(4) Character generator ROM

(5) 160 different 5 x7 dot-matrix character patterns.

(6) Character generator RAM

(7) 8 different user programmed 5x7 dot-matrix patterns.

(8) Display data RAM and character generator RAM may be accessed by the microprocessor.

(9) Numerous instructions

(10) Clear Display, Cursor Home, Display ON/OFF, Cursor ON/OFF, Blink Character, Cursor Shift, Display Shift.

(11) Built-in reset circuit is triggered at power ON.

(12) Built-in oscillator.

DESCRIPTION OF 16x2:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all Parallel Ports. It however doesn't show the use of the Status Port as an input. So what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board.

SCHEMATIC DIAGRAM:

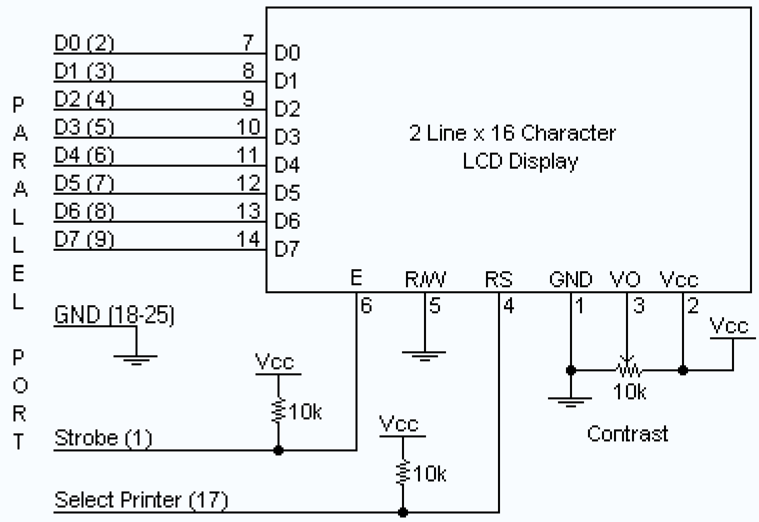


Figure:3.15 Schematic Diagram of LCD

Above is the quite simple schematic. The LCD to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.

The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780. The one I used to test this circuit was a Power tip PC-1602F and an old Philips LTN211F-10 which was extracted from a Poker Machine! The diagram to the right, shows the pin numbers for these devices. When viewed from the front, the left pin is pin 14 and the right pin is pin 1

16 x 2 ALPHANUMERIC LCD MODULE FEATURES:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

16 x 2 ALPHANUMERIC LCD MODULE SPECIFICATIONS:

Pin	Symbol	Level	Function
1	V_{SS}	-	Power, GND
2	V_{DD}	-	Power, 5V
3	Vo	-	Power, for LCD Drive
4	RS	H/L	Register Select Signal H: Data Input L: Instruction Input
5	R/W	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)

6	E	H,H->L	Enable
7-14	DB0-DB7	H/L	Data Bus; Software selectable 4- or 8-bit mode
15	NC	-	**NOT CONNECTED**
16	NC	-	**NOT CONNECTED**

Table 3.3 Specifications of LCD

FEATURES:

- 5 x 8 dots with cursor
- Built-in controller (KS 0066 or Equivalent)
- + 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)
- N.V. optional for + 3V power supply

Data can be placed at any location on the LCD. For 16×1 LCD, the address Locations are:

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

Table: 3.4 Address Locations for a 1x16 line LCD

Even limited to character based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8,16,20,24,32 and 40 characters are all standard, in one, two and four line versions.Several different LC technologies exists. “super-twist” types, for example, offer Improved contrast and viewing angle over the older “twisted pneumatic” types. Some modules are available with back lighting, so that they can be viewed in dimly-lit conditions. The back lighting may be either “electro-luminescent”, requiring a high voltage inverter circuit, or simple LED illumination.

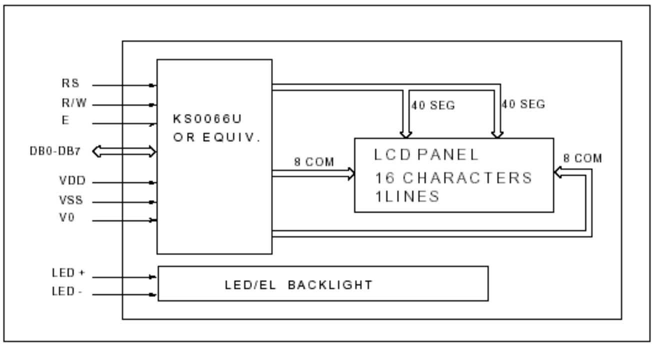


Figure: 3.16 Electrical Block Diagram

POWER SUPPLY FOR LCD DRIVING:

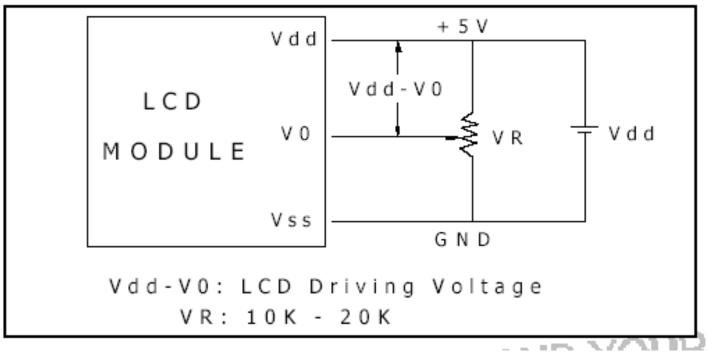


Figure: 3.17 : Power Supply for LCD

PIN DESCRIPTION:

Most LCD's with 1 controller has 14 Pins and LCD's with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections).

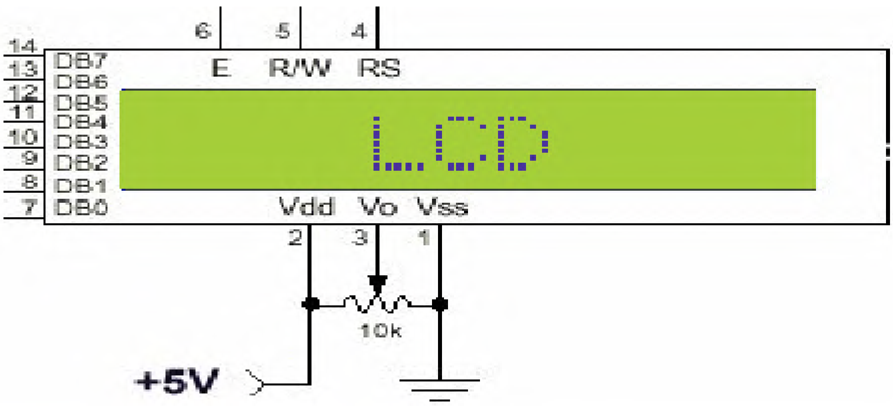


Figure 3.18: Pin Diagram of 1x16 lines LCD

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Table: 3.5 Pin Specifications

CONTROL LINES:

EN: Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD data sheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS: Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW: Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands, so RW will almost always be low.

Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Logic status on control lines:

E - 0 Access to LCD disabled- 1 Access to LCD enabled R/W - 0 Writing data to LCD 1 Reading data from LCD RS - 0 Instructions-1 Character

Writing data to the LCD:

1) Set R/W bit to low
2) Set RS bit to logic 0 or 1 (instruction or character)
3) Set data to data lines (if it is writing)
4) Set E line to high
5) Set E line to low

Read data from data lines (if it is reading) on LCD:

1) Set R/W bit to high
2) Set RS bit to logic 0 or 1 (instruction or character)
3) Set data to data lines (if it is writing)
4) Set E line to high
5) Set E line to low

Entering Text:

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the d.i.l. switch pack with a of re-wiring is necessary.

SWITCHES:

The switches must be the type where On = 0, so that when they are turned to the zero position, all four outputs are shorted to the common pin, and in position “F”, all four outputs are open circuit. All the available characters that are built into the module are shown in Table 3. Studying the table, you will see that codes associated with the characters are quoted in binary and hexadecimal, most significant bits (“left-hand” four bits) across the top, and least significant bits (“right-hand” four bits) down the left. Most of the characters conform to the ASCII standard, although the Japanese and

Greek characters (and a few other things) are obvious exceptions. Since these intelligent modules were designed in the "Land of the Rising Sun," it seems only fair that their Katakana phonetic symbols should also be incorporated. The more extensive Kanji character set, which the Japanese share with the Chinese, consisting of several thousand different characters, is not included! Using the switches, of whatever type, and referring to Table 3, enter a few characters onto the display, both letters and numbers. The RS switch (S10) must be "up" (logic 1) when sending the characters, and switch E (S9) must be pressed for each of them. Thus the operational order is: set RS high, enter character, trigger E, leave RS high, enter another character, trigger E, and so on. The first 16 codes in Table 3, 00000000 to 00001111, ($00 to $0F) refer to the CGRAM. This is the Character Generator RAM (random access memory), which can be used to hold user-defined graphics characters. This is where these modules really start to show their potential, offering such capabilities as bar graphs, flashing symbols, even animated characters. Before the user-defined characters are set up, these codes will just bring up strange looking symbols. Codes 00010000 to 00011111 ($10 to $1F) are not used and just display blank characters. ASCII codes "proper" start at 00100000 ($20) and end with 01111111 ($7F). Codes 10000000 to 10011111 ($80 to $9F) are not used, and 10100000 to 11011111 ($A0 to $DF) are the Japanese characters.

Upper 4 bits / Lower 4 bits	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
0 0000	CG RAM (1)			0	@	P	`	p				ー	タ	ミ	α	p
1 0001	CG RAM (2)		!	1	A	Q	a	q			。	ア	チ	ム	ä	q
2 0010	CG RAM (3)		"	2	B	R	b	r			「	イ	ツ	メ	β	θ
3 0011	CG RAM (4)		#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
4 0100	CG RAM (5)		$	4	D	T	d	t			、	エ	ト	ヤ	μ	Ω
5 0101	CG RAM (6)		%	5	E	U	e	u			・	オ	ナ	ユ	σ	ü
6 0110	CG RAM (7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
7 0111	CG RAM (8)		'	7	G	W	g	w			ァ	キ	ヌ	ラ	g	π
8 1000	CG RAM (1)		(	8	H	X	h	x			ィ	ク	ネ	リ	√	x̄
9 1001	CG RAM (2)		)	9	I	Y	i	y			ゥ	ケ	ノ	ル	⁻¹	y
A 1010	CG RAM (3)		*	:	J	Z	j	z			ェ	コ	ハ	レ	j	千
B 1011	CG RAM (4)		+	;	K	[	k	{			ォ	サ	ヒ	ロ	ˣ	万
C 1100	CG RAM (5)		,	<	L	¥	l	|			ャ	シ	フ	ワ	¢	円
D 1101	CG RAM (6)		-	=	M	]	m	}			ュ	ス	ヘ	ン	£	÷
E 1110	CG RAM (7)		.	>	N	^	n	→			ョ	セ	ホ	゛	ñ	
F 1111	CG RAM (8)		/	?	O	_	o	←			ッ	ソ	マ	゜	ö	█

Table: 3.6 Character details in LCD

Initialization by Instructions:

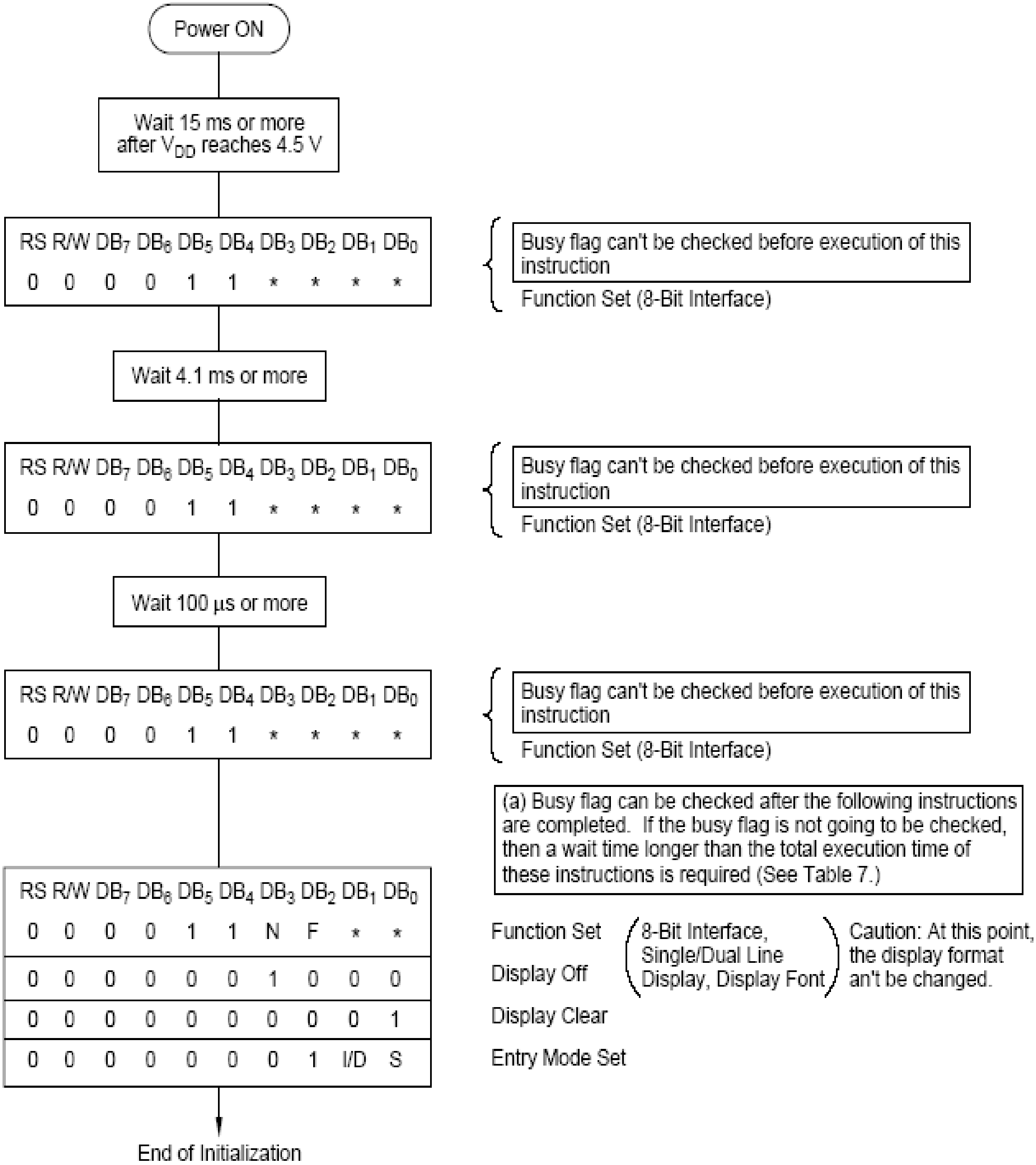


Figure 3.19: Flow Chart of LCD

If the power conditions for the normal operation of the internal reset circuit are not satisfied, then executing a series of instructions must initialize LCD unit. The procedure for this initialization process is as above show.

LCD COMMANDS:

There are some present commands instructions in LCD, which we need to send to LCD through some microcontroller. Some important command instructions are given below:

Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left
0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

Table: 3.7 Commands

3.5 ULTRA SONIC SENSOR

ULTRASONC PRINCIPLE:

Ultrasonic sensors emit short, high-frequency sound pulses at regular intervals. These propagate in the air at the velocity of sound. If they strike an object, then they are reflected back as echo signals to the sensor, which itself computes the distance to the target based on the time-span between emitting the signal and receiving the echo. As the distance to an object is determined by measuring the time of flight and not by the intensity of the sound, ultrasonic sensors are excellent at suppressing background interference. Virtually all materials which reflect sound can be detected, regardless of their color. Even transparent materials or thin foils represent no problem for an ultrasonic sensor.Micro sonic ultrasonic sensors are suitable for target distances from 30 mm to 10 m and as they measure the time of flight they can ascertain a measurement with pinpoint accuracy. Some of our sensors can even resolve the signal to an accuracy of less than 0.18 mm. Ultrasonic sensors can see through dust-laden air and ink mists. Even thin deposits on the sensor membrane do not impair its function. Sensors with a blind zone of just 30 mm and an extremely narrow beam spread are finding totally new applications these days: measuring levels in yogurt pots and test tubes as well as scanning small bottles in the packaging sector - no trouble for our sensors. Even thin wires are reliably detected.

Figure 3.20 Ultrasonic Sensor Module

SPECIFICATION:

The ultrasonic range sensor detects objects in it’s path and can be used to calculate the range to the object. It is sensitive enough to detect a 3cm diameter broom handle at a distance of over 2m.

Voltage - 5v

Current - 30mA Type. 50mA Max.

Frequency - 40 KHz

Max Range - 3 m

Min Range - 3 cm

Sensitivity - Detect 3cm diameter broom handle at > 2 m

Input Trigger - 10uS Min. TTL level pulse

Echo Pulse - Positive TTL level signal, width proportional to range.

Small Size - 43mm x 20mm x 17mm height

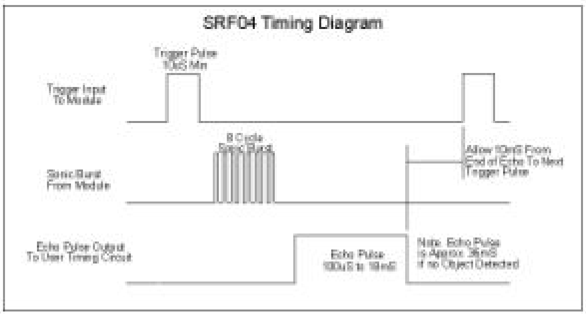


Figure 3.21 Ultrasonic Sensor Timing Diagram

ELECTRIC CONNECTION:

The SRF004 ultrasonic range finder has 5 connections pins. The power supply is connected to the 5V and 0V ground connections on the SRF004. (Note that BOTH the ‘Mode’ (hole 4) and ‘0V Ground’ (hole 5) connections MUST be connected to 0V for correct operation with the PICAXE system). Take care not to overheat, and therefore damage, the solder connection pads whilst making connections.The SRF004 Trigger Input is connected to a PICAXE output pin.The SRF004 Echo Output is connected to a PICAXE input pin.Ultrasonic sensors have set new standards in automation

Figure 3.22 Pin configuration of Ultrasonic Sensor Module

CONNECTION:

The SRF004 must be mounted above the buggy (e.g. by using a small home-made aluminum bracket (not supplied)). The SRF004 has five solder connections which must be connected via wires to the solder joints on the bottom of the buggy PCB.

1. Hole 1 – 5v Supply – to PIC chip leg 14 (V+ Supply)

2. Hole 2 – Echo Output – to PIC chip leg 15 (input 6)

3. Hole 3 – Trigger Input – to PIC chip leg 9 (output 3)

4. Hole 5 – 0V Ground – to PIC chip leg 5 (0V Ground)

3.6 BUZZER

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

Figure 3.23:Buzzer

.

Mechanical

A joy buzzer is an example of a purely mechanical buzzer and they require drivers. Other examples of them are doorbells.

Piezoelectric

Nowadays, it is more popular to use a ceramic-based piezoelectric sounder like a Son alert which makes a high-pitched tone. Usually these were hooked up to "driver" circuits, which varied the pitch of the sound or pulsed the sound on and off.

Figure 3.24 Piezoelectric Buzzer

A piezoelectric element may be driven by an oscillating electronic circuit or other audio signal source, driven with a piezoelectric audio amplifier. Sounds commonly used to indicate that a button has been pressed are a click, a ring or a beep. Piezo buzzer is an electronic device commonly used to produce sound. Light weight, simple construction and low price make it usable in various applications like car/truck reversing indicator, computers, call bells etc. Basically, the sound source of a piezoelectric sound component is a piezoelectric diaphragm as shown in fig(1). A piezoelectric diaphragm consists of a piezoelectric ceramic plate which has electrodes on both sides and a metal plate (brass or stainless steel, etc.). A piezoelectric ceramic plate is attached to a metal plate with adhesives. Applying D.C. voltage between electrodes of a piezoelectric diaphragm causes mechanical distortion due to the piezoelectric effect. For a misshaped piezoelectric element, the distortion of the piezoelectric element expands in a radial direction. And the piezoelectric diaphragm bends toward the direction. The metal plate bonded to the piezoelectric element does not expand. Conversely, when the piezoelectric element shrinks, the piezoelectric diaphragm bends in the direction Thus, when AC voltage is applied across electrodes, the bending is repeated, producing sound waves in the air. Buzzers typically provide a user with an alert when a given condition within the circuit has occurred. Normally man's audible frequency range is 20Hz to 20kHz , but we can hear up to frequency ranges from 2kHz to 4kHz .

To interface a buzzer the standard transistor interfacing circuit is used(Fig 2). Note that if a different power supply is used for the buzzer, the 0V rails of each power supply must be connected to provide a common reference. If a battery is used as the power supply, it is worth remembering that piezo sounders draw much less current than buzzers. Buzzers also just have one 'tone', whereas a piezo sounder is able to create sounds of many different tones. Here transistor acts as a switch.

To switch on buzzer -high 1

To switch off buzzer -low 1

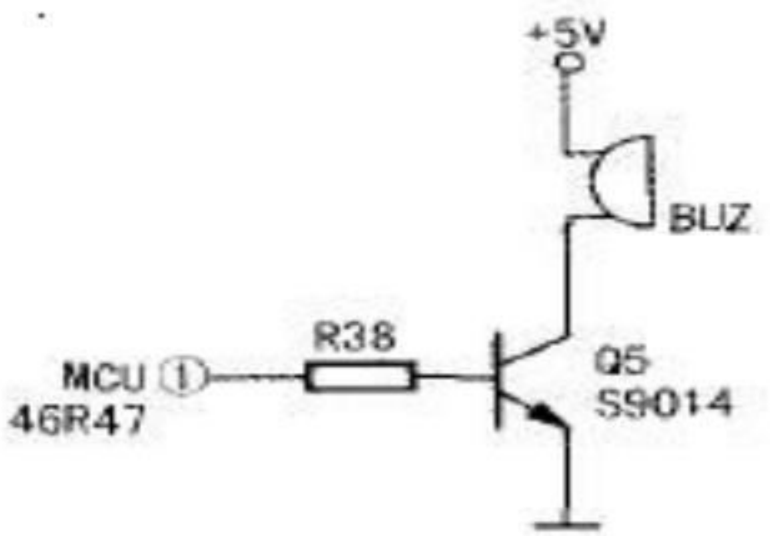


Figure 3.25 Buzzer with Transistor Interfacing Circuit

Notice (Handling) In Using Self Drive Method:

1) When the piezoelectric buzzer is set to produce intermittent sounds, sound may be heard continuously even when the self drive circuit is turned ON / OFF at the "X" point as shown. This is because of the failure of turning off the feedback voltage.

2) Build a circuit of the piezoelectric sounder exactly as per the recommended circuit shown in the catalog. Life of the transistor and circuit constants is designed to ensure stable oscillation of the piezoelectric sounder.

3) Design switching which ensures direct power switching.

4) The self drive circuit is already contained in the piezoelectric buzzer. So there is no need to prepare another circuit to drive the piezoelectric buzzer.

5) Rated voltage (3.0 to 20Vdc) must be maintained. Products which can operate with voltage higher than 20Vdc are also available.

6) Do not place resistors in series with the power source, as this may cause abnormal oscillation. If a resistor is essential to adjust sound pressure, place a capacitor (about 1μF) in parallel with the piezo buzzer.

7) Do not close the sound emitting hole on the front side of casing.

8) Carefully install the piezo buzzer so that no obstacle is placed within 15mm from the sound release hole on the front side of the casing.

MODERN APPLICATIONS

While technological advancements have caused buzzers to be impractical and undesirable, there are still instances in which buzzers and similar circuits may be used. Present day applications include:

- Novelty uses
- Judging panels
- Educational purposes
- Annunciator panels
- Electronic metronomes
- Game show lock-out device
- Microwave ovens and other household appliances

- Sporting events such as basketball games
- Electrical alarms
- Joy buzzer (mechanical buzzer used for pranks)

3.7 VOLTAGE REGULATOR

A **voltage regulator** is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx' indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx' indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consists the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

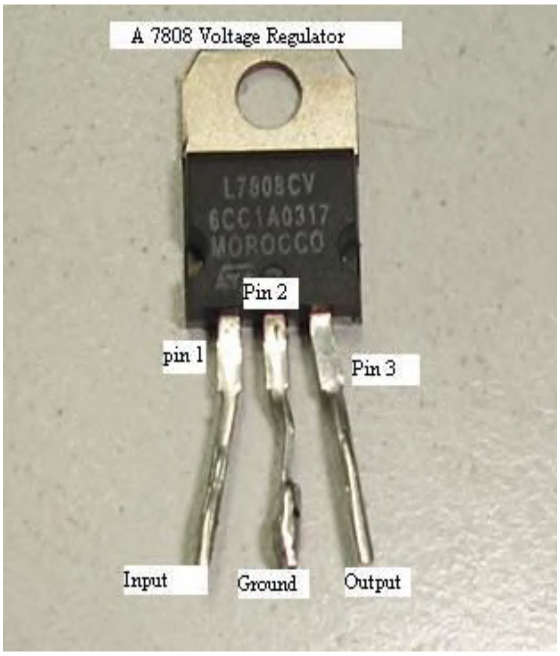


Figure: 3.26 Regulator

3.8 LIGHT EMITTING DIODE (LED)

The Light emitting diode is a two-lead semiconductor light is a special type of diode and they have similar electrical characteristics of a PN junction diode. Hence the led allows the flow of current in the forward direction and blocks the current in the reverse direction. The led occupies the small area which is less than that of the 1mm2 here the led figure is shown below.

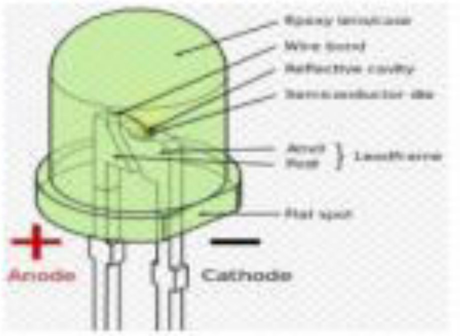


Figure: 3.27 Light Emitting Diode

WORKING OF LED

The light emitting diode simply, we know as a diode. When the diode is forward biased, then the electrons & holes are moving fast across the junction and they are combining constantly, removing one another out. Soon after the electrons are moving from the n-type to the p-type silicon, it combines with the holes, then it disappears. Hence it makes the complete atom & more stable and it gives the little burst of energy in the form of a tiny packet or photon of light.

CHAPTER 4

SOFTWARE DESCRIPTION

4.1 SOFTWARE DESIGN

ARDUINO IDE

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++[2]. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.[3]The source code for the IDE is released under the GNU General Public License, version 2.[4] The Arduino IDE supports the languages C and C++ using special rules of code structuring.[5] The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution.[6] The Arduino IDE employs the program to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.[7]

Arduino Pro IDE	
Developer(s)	Arduino Software
Preview release	v0.0.2 / 28 October 2019; 3 months ago[8]
Repository	github.com/arduino/Arduino
Written in	C, C++
Operating system	Windows, macOS, Linux

Platform	IA-32, x86-64, ARM
Type	Integrated development environment
License	LGPL or GPL license
Website	blog.arduino.cc/2019/10/18/arduino-pro-ide-alpha-preview-with-advanced-features/

Table 4.1 Features of Arduino IDE

PROGRAMMING

The Arduino Uno can be programmed with the Arduino Software (IDE). Select Arduino Uno from the Tools Board After your code is compiled using Arduino IDE, it should be uploaded to the main microcontroller of the Arduino UNO using a USB connection.

Because the main microcontroller doesn't have a USB transceiver, you need a bridge to convert signals between the serial interface (UART interface) of the microcontroller and the host USB signals. The ATmega328 on the Arduino Uno comes pre programmed with a boot loader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol.

You can also bypass the boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino are similar. The ATmega16U2 (or 8U2 in the rev1&rev2 boards) firmware source code is available in the Arduino repository.

The ATmega16U2 or 8U2 is loaded with a DFU boot loader, which can be activated by On Rev1 boards connecting the solder jumper on the back of the board and then resetting the 8U2. On Rev2 or later boards there is a resistor that pulling the 8U2 or 16U2 HWB line to ground, making it easier to put into DFU mode. You can then use Atmel's FLIP software (Windows) or the DFU programmer (MacOS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU boot loader)**.** Arduino is an open-source hardware platform that is being used by people around the globe for building electronics projects. It is an integrated platform which contains both the physical and programmable circuit otherwise known as microcontroller and a software (or IDE) that you can run on your computer to write and upload the code onto the physical board.

Arduino Board is quite popular among many people who want to get started with electronics, and unlike other embedded system boards Arduino does not require any additional hardware to upload the code (generally known as programmer). The Arduino Program can be written and uploaded using the Arduino IDE that needs just an USB cable to connect. Since the interface is simple and are less complicated. Arduino is preferred by most of the aspiring engineers.

Here we will try to understand about the Arduino Architecture and its functionalities. The processor of the Arduino Board uses Harvard Architecture for which the program code and program data have separate memory. The memory of it is divided into two namely program memory and data memory. The data will be stored in the data memory whereas the 17program code will be stored in the flash program memory. For ex: The Atmega328 microcontroller has 32kb of flash memory, 2kb of SRAM, 1kb of EPROM and operates at 16MHz clock speed. Some of the other basic functions of Arduino are.

Digital write pin is used to write the digital value of the given pin.

Pin mode pin is used to set the pin to I/O mode.

Analog read pin reads and returns the value.

Analog write pin writes the value of the pin

Serial. Begins pin sets the beginning of serial communication by setting the rate of bit

Digital read pin reads the digital value of given pin

4.2 PROTEUS:

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

ABOUT PROTEUS:

It is a software suite containing schematic, simulation as well as PCB designing.
ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.The designer can also develop 2D drawings for the product.

FEATURES:

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.
ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

STARTING NEW DESIGN

Step 1: Open ISIS software and select New design in File menu

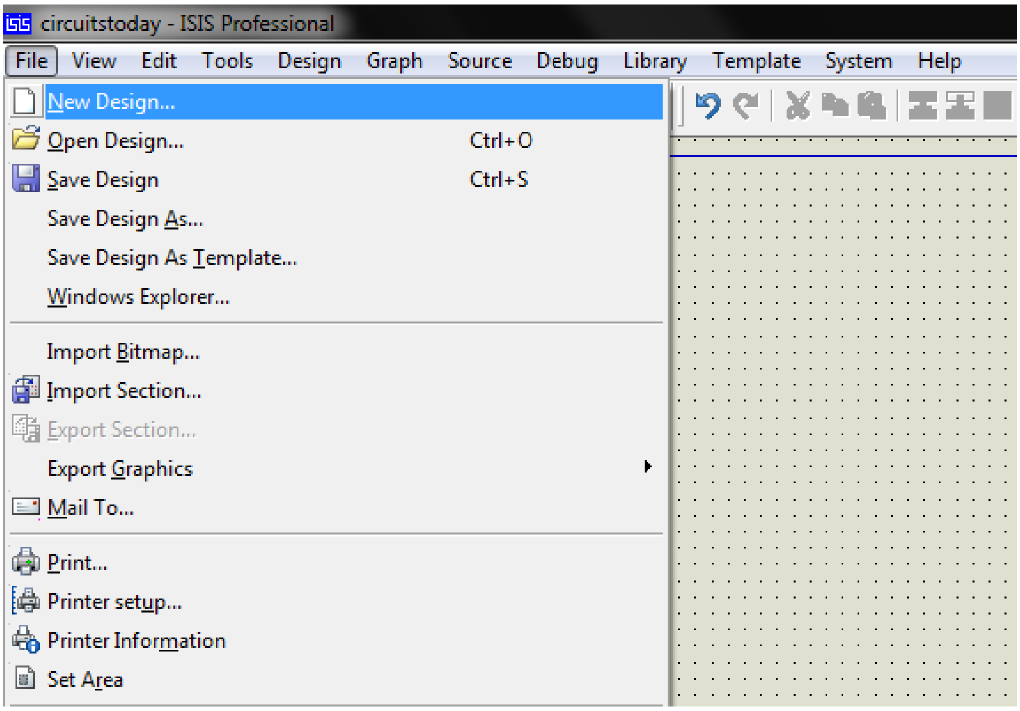


Figure 4.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the project size while printing. For now, select default or according to the layout size of the circuit.

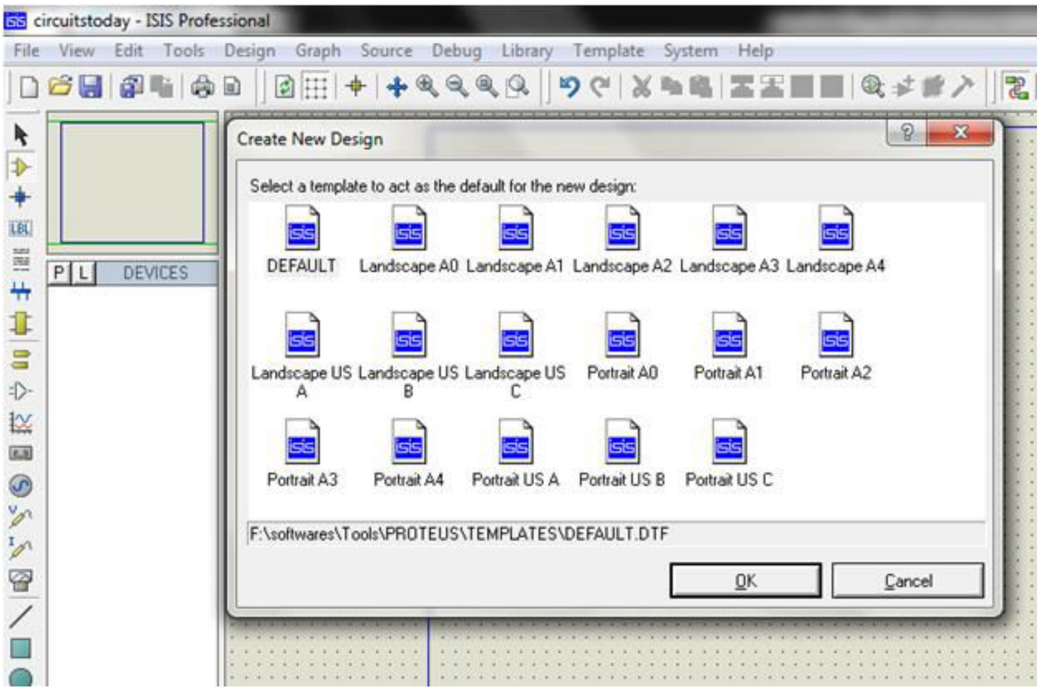


Figure 4.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

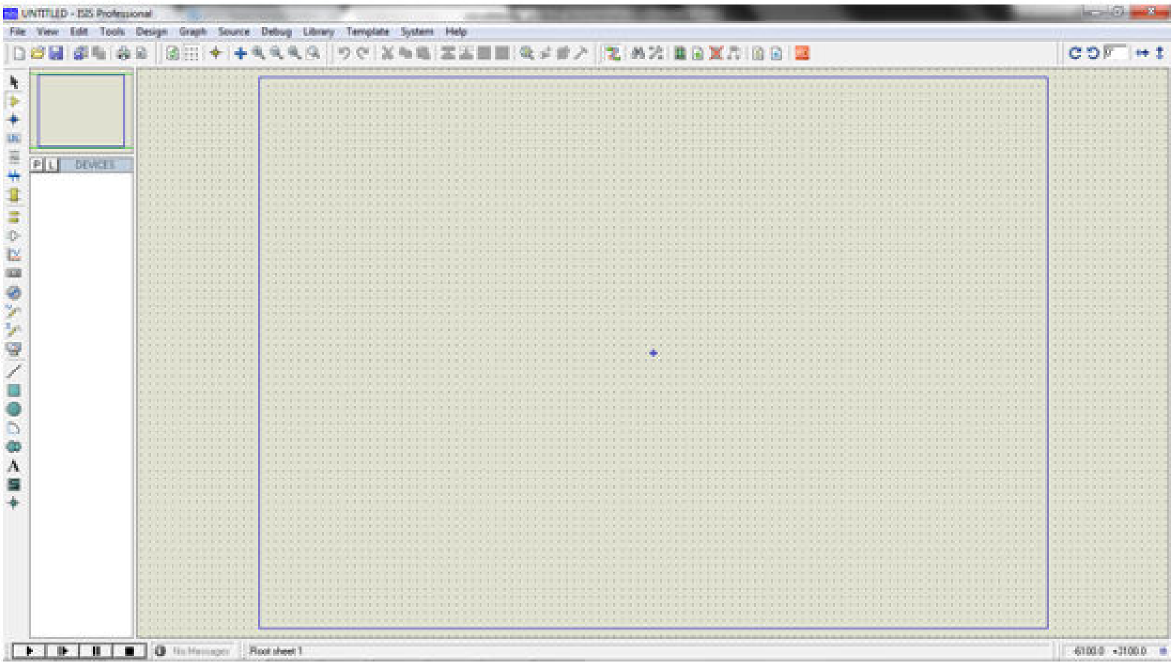


Figure 4.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

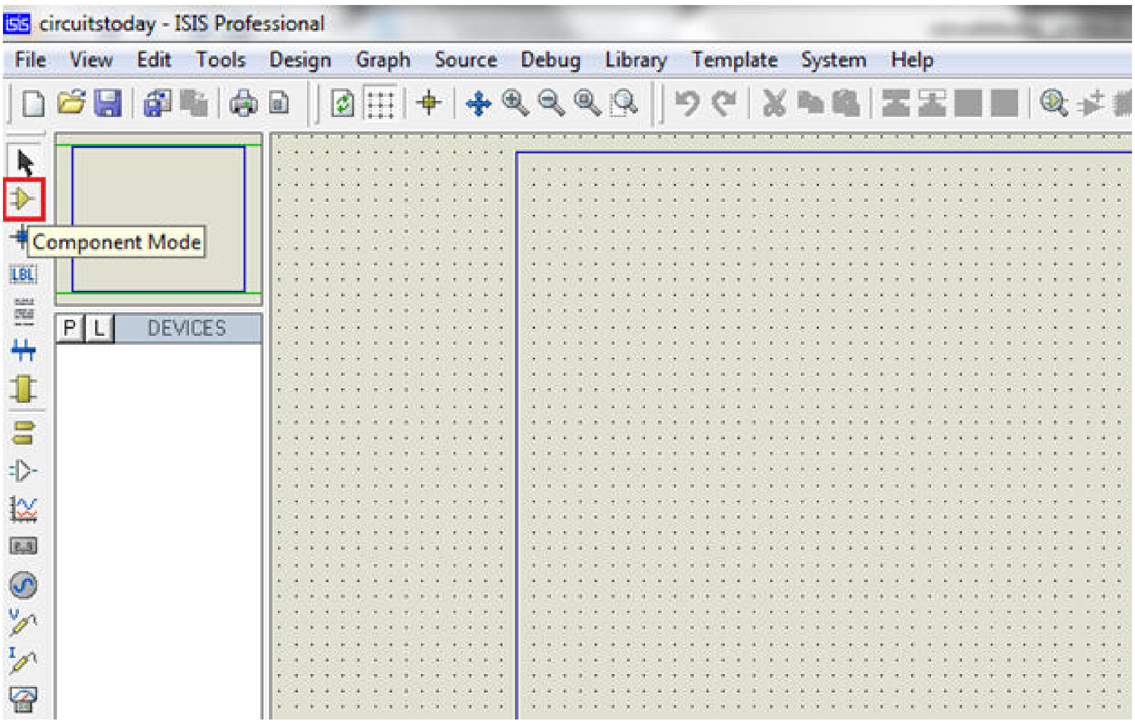


Figure 4.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

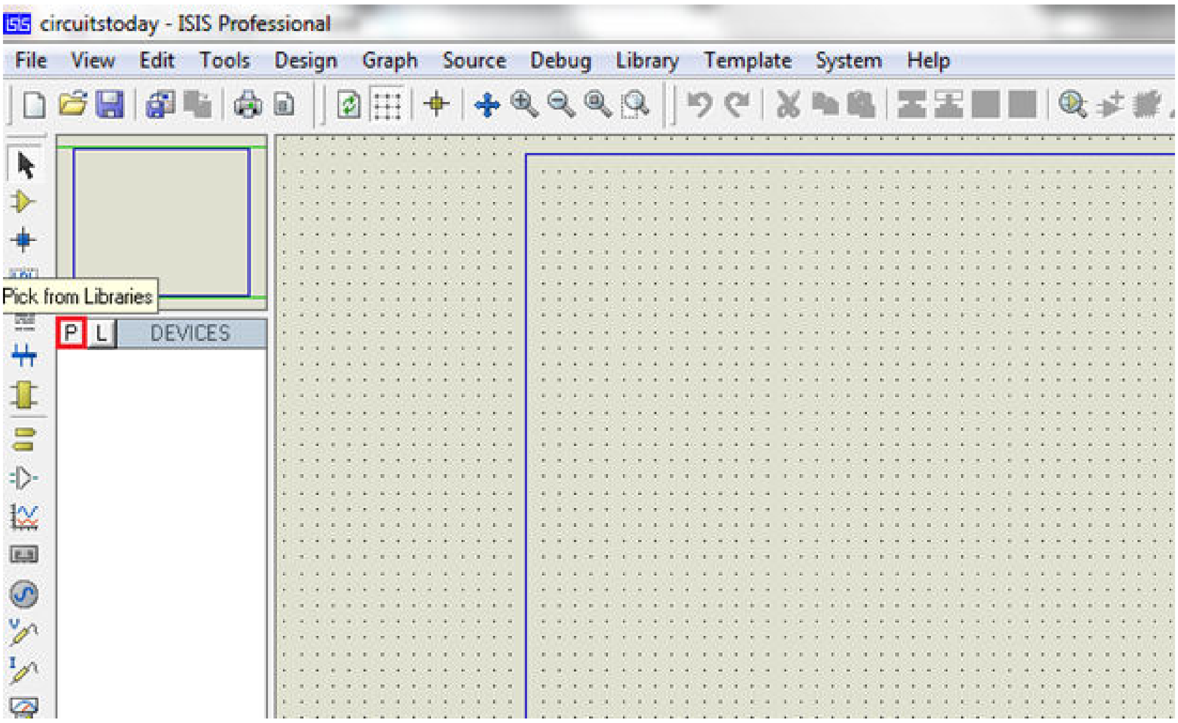


Figure 4.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

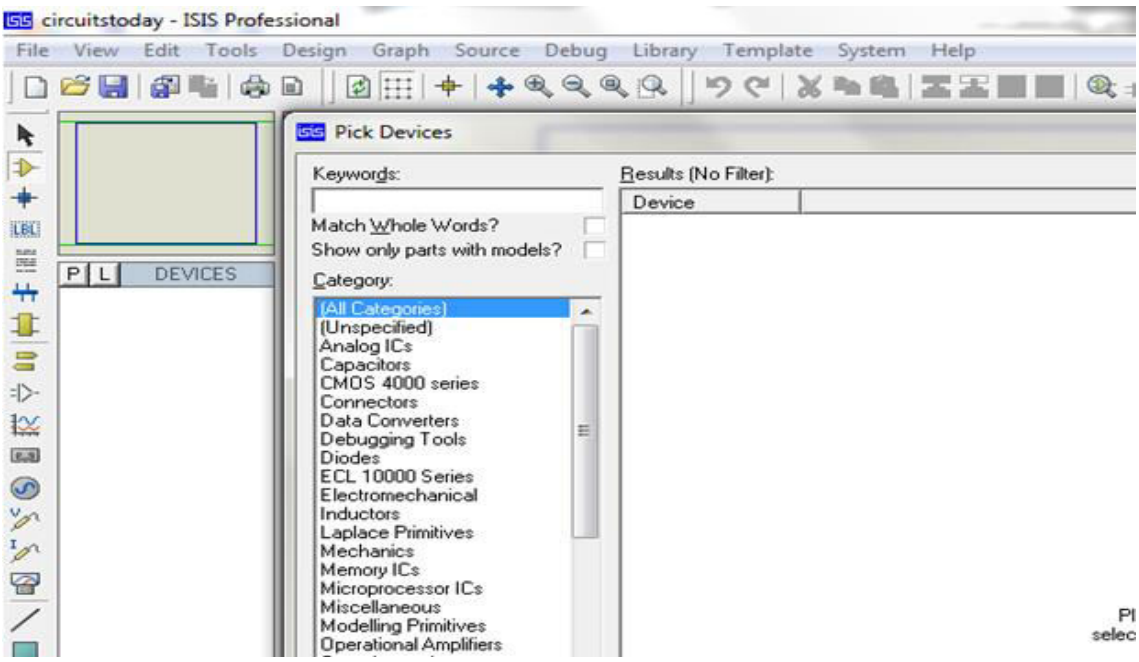


Figure 4.6 Keywords Text box

Example shows selection of push button. Select the components accordingly.

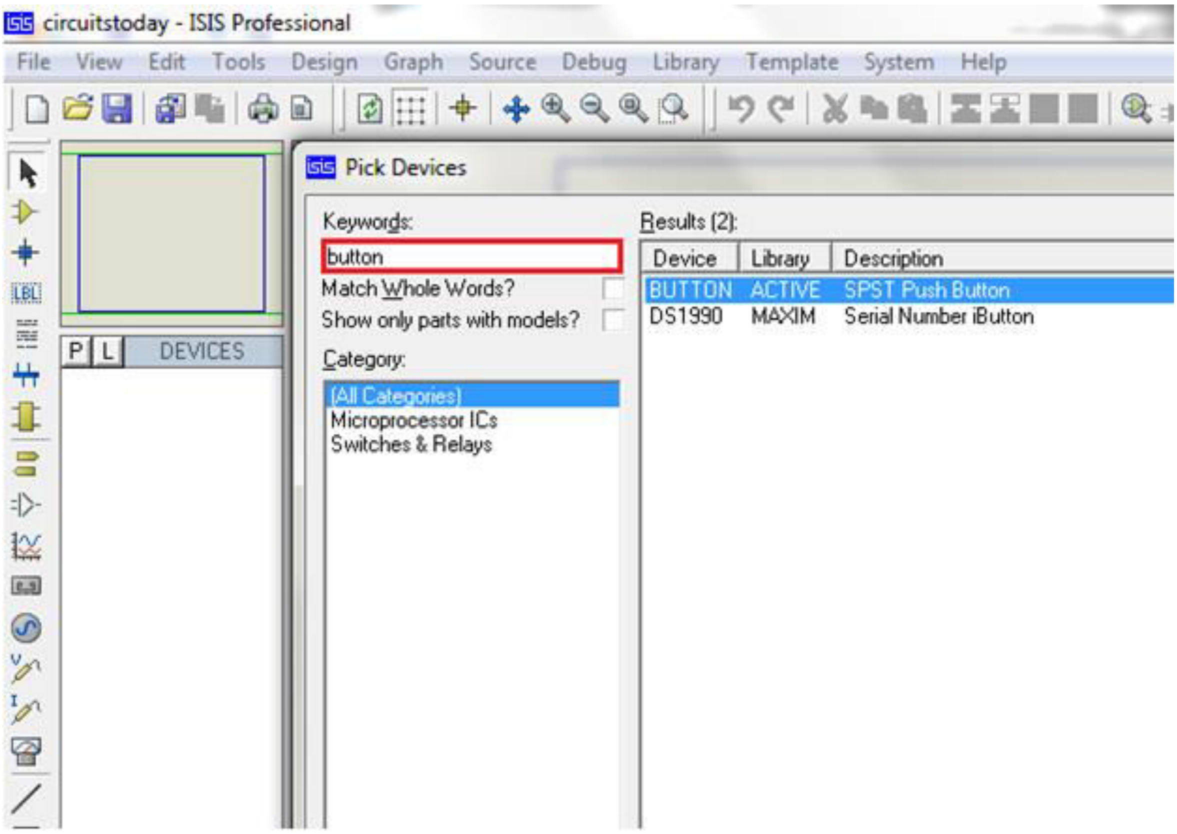


Figure 4.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

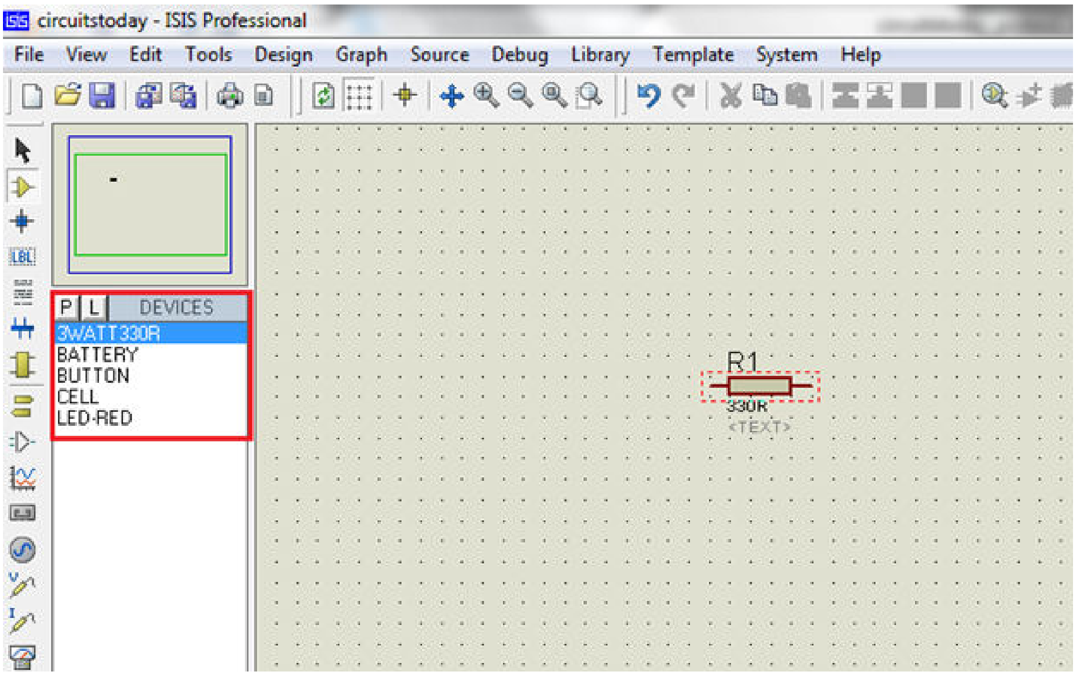


Figure 4.8 Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

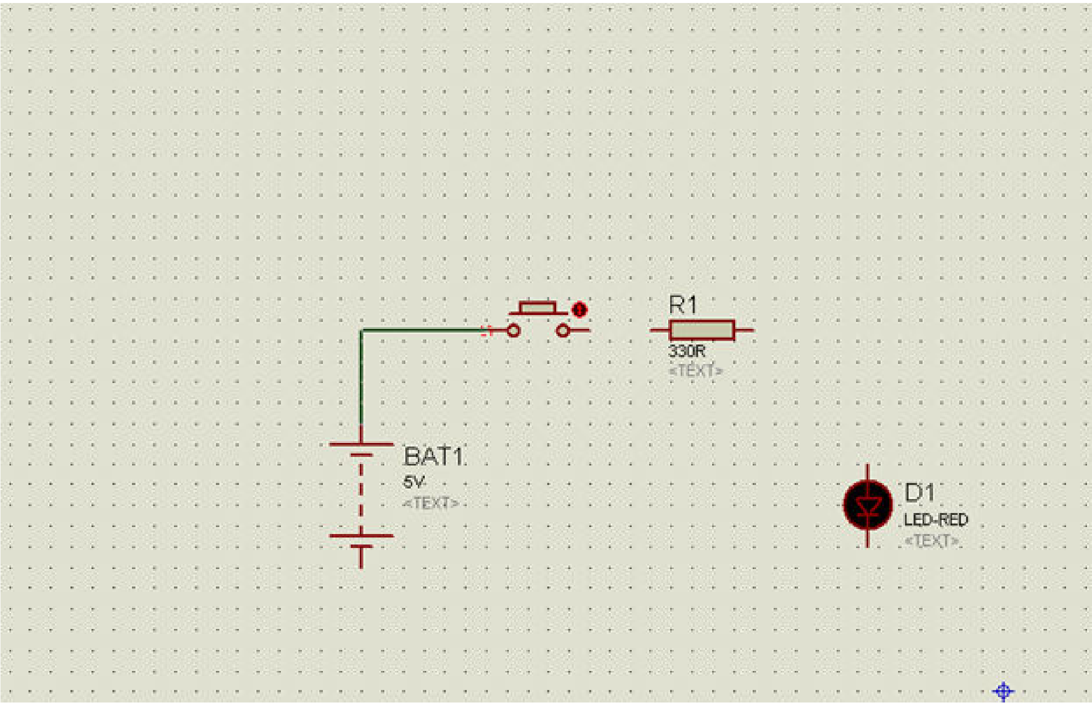


Figure 4.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on OK.

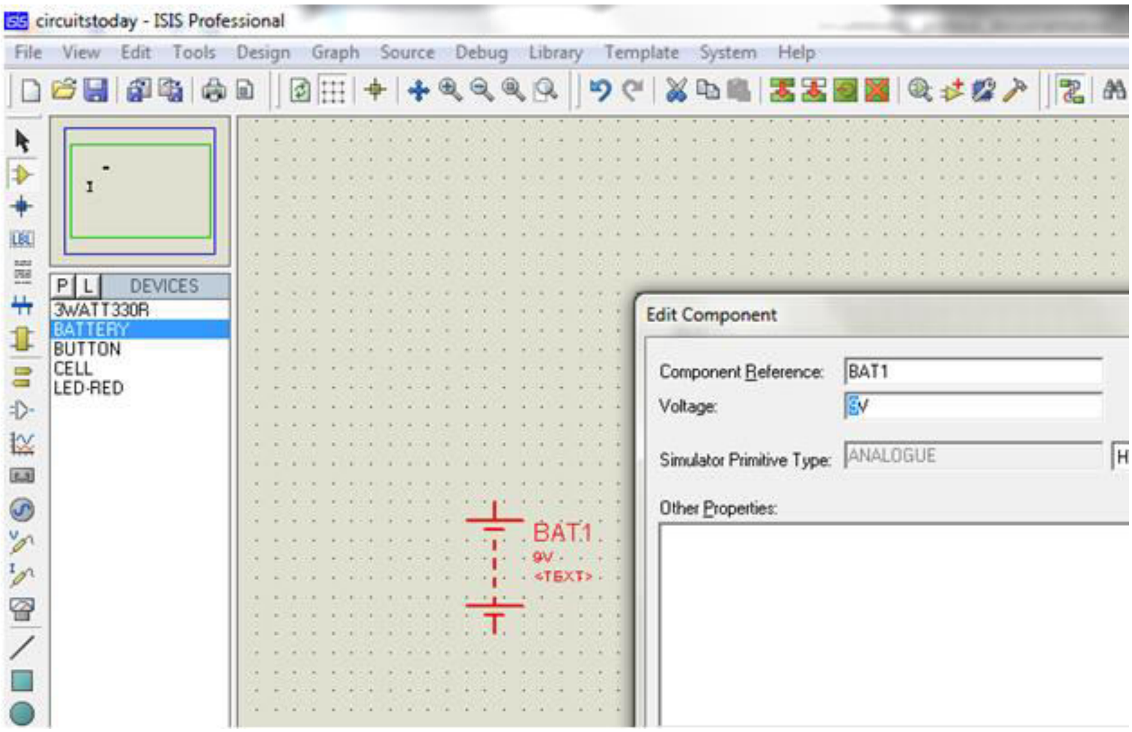


Figure 4.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

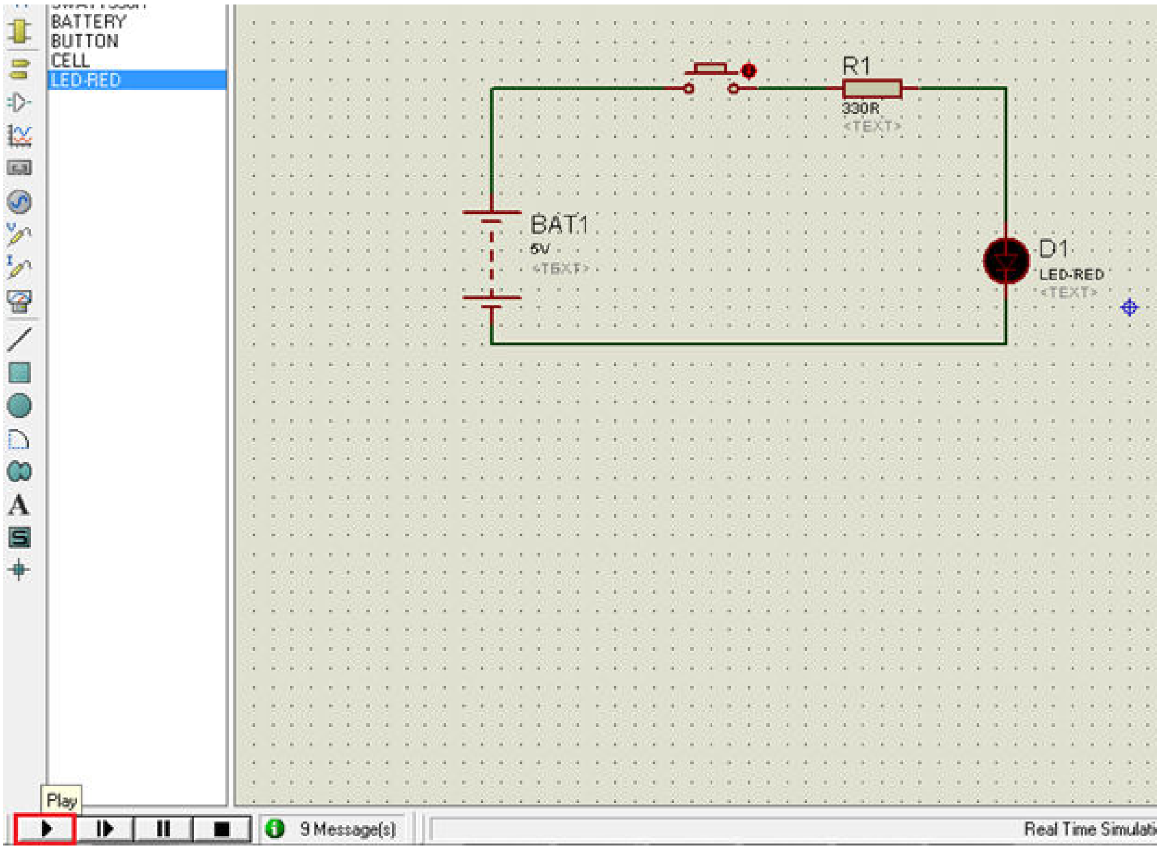


Figure 4.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

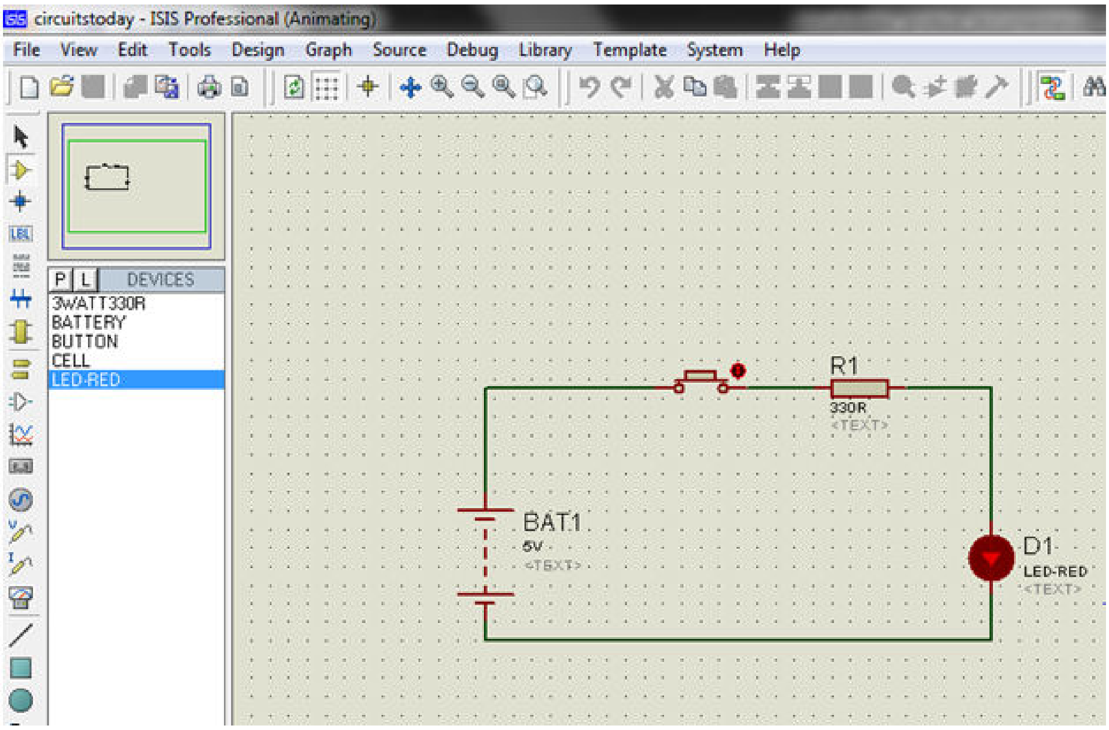


Figure 4.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

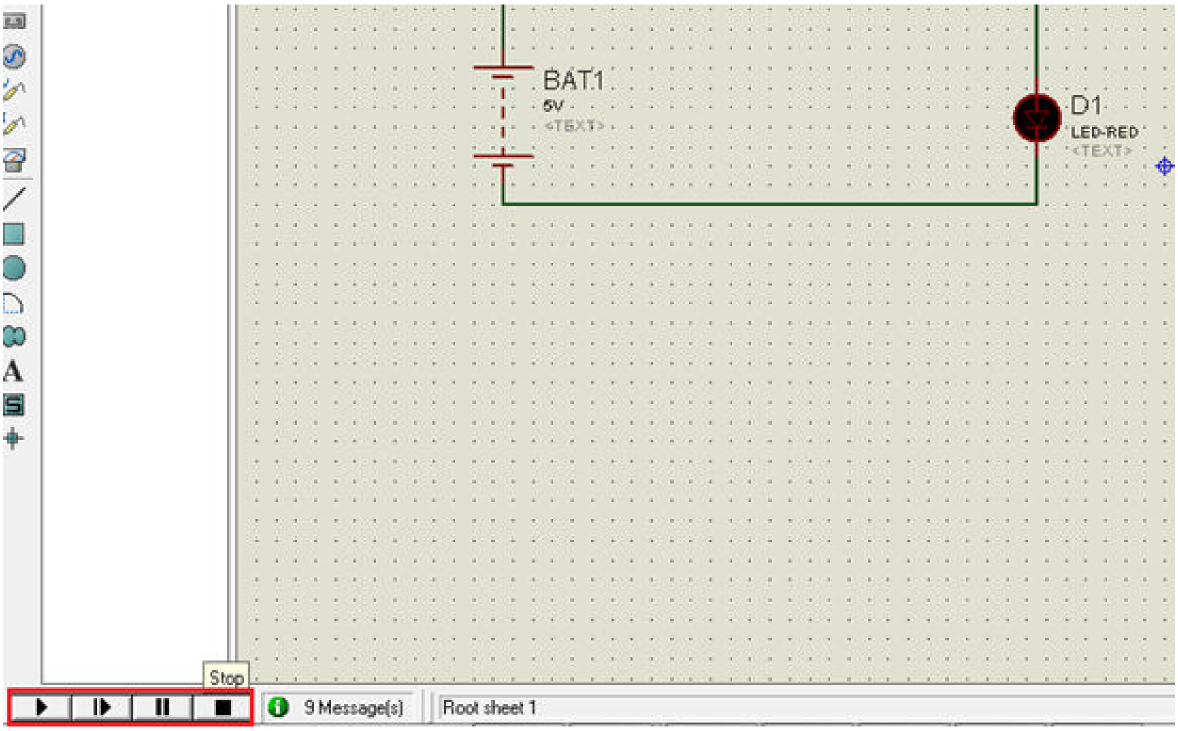


Figure 4.13 Simulation Step-Pause-Stop Buttons

4.3 THINK SPEAK PLATFORM

ThingSpeak is an IoT cloud platform service that allows you to aggregate, visualize and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices to ThingSpeak.You can send data to ThingSpeak from your devices, create instant visualization of live data, and send alerts.ThingSpeak is IoT Cloud platform where you can send sensor data to the cloud. You can also analyze and visualize your data with MATLAB or other software, including making your own applications.

ThingSpeak enables sensors, instruments, and websites to send data to the cloud where it is stored in either a private or a public channel. ThingSpeak stores data in private channels by default, but public channels can be used to share data with others.A channel is where you send your data to store. Each channel includes 8 fields for any type of data, 3 location fields, and 1 status field. Once you have a ThingSpeak Channel you publish data to the channel, have ThingSpeak process the data, then have your application retrieve the data.

In order to retrieve data from Think speak cloud, below steps are to be followed

1. On the ThingSpeak.com website, select Channels > My Channels.
2. Under the channel that you created for this target hardware, click Settings.
3. Click the API Keys tab.
4. Copy the key from Read API Key parameter.
5. Open the ThingSpeak Read block in your model and paste the copied API key.

THING SPEAK ACCOUNT CREATION

1. Visit www.thingspeak.com.

2. Click "Sign Up".

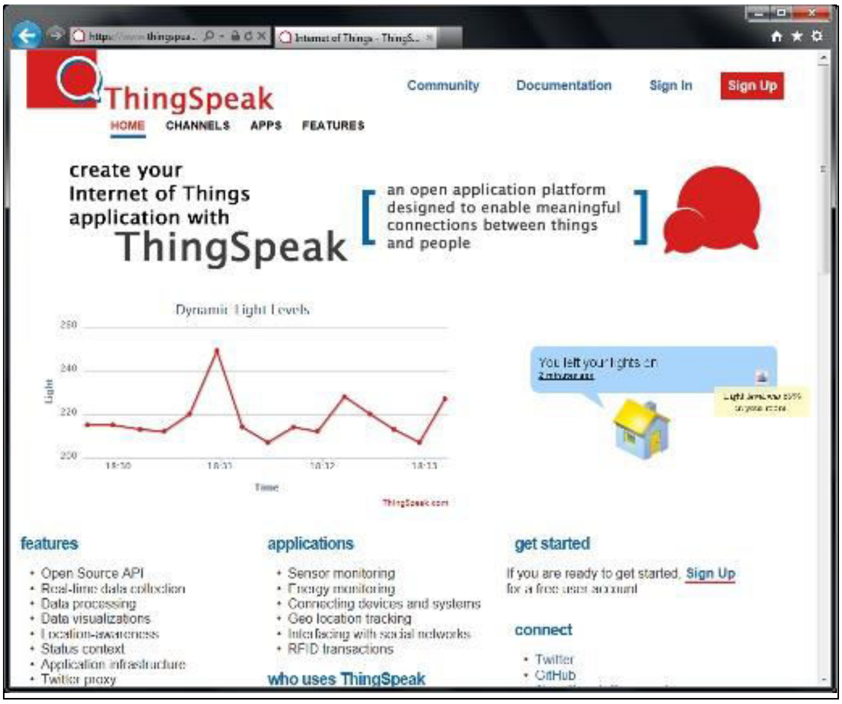


Figure 4.14 Sign up

3. Fill the following mandatory fields: User ID, E-mail, Time Zone, Password and Password Confirmation.

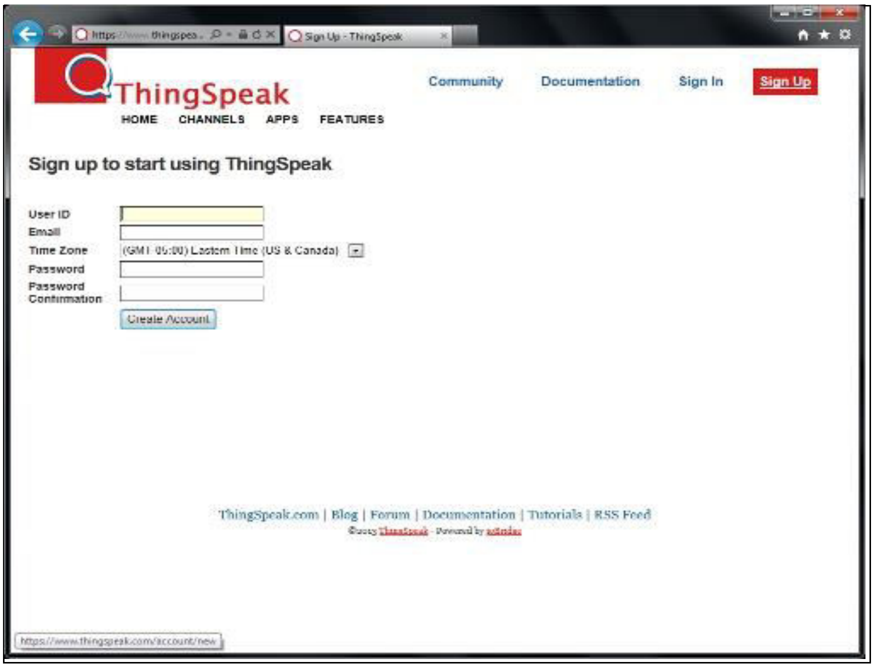


Figure 4.15 Filling the mandatory fields

4. .Click"CreateAccount".

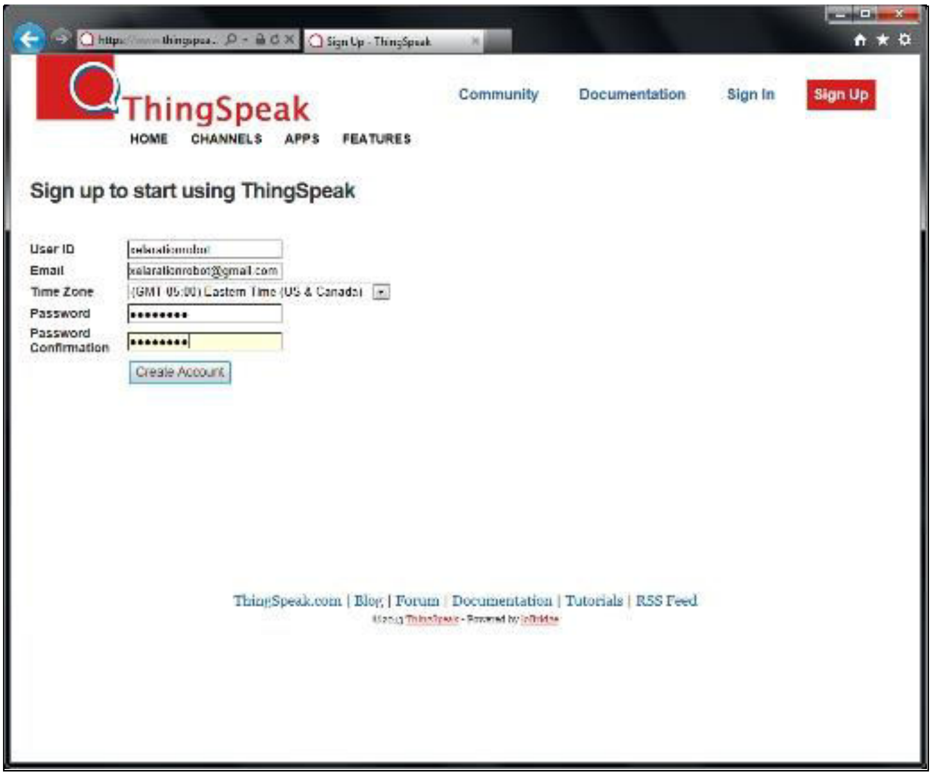


Figure 4.16 Creating account

5. Click "Create New Channel".

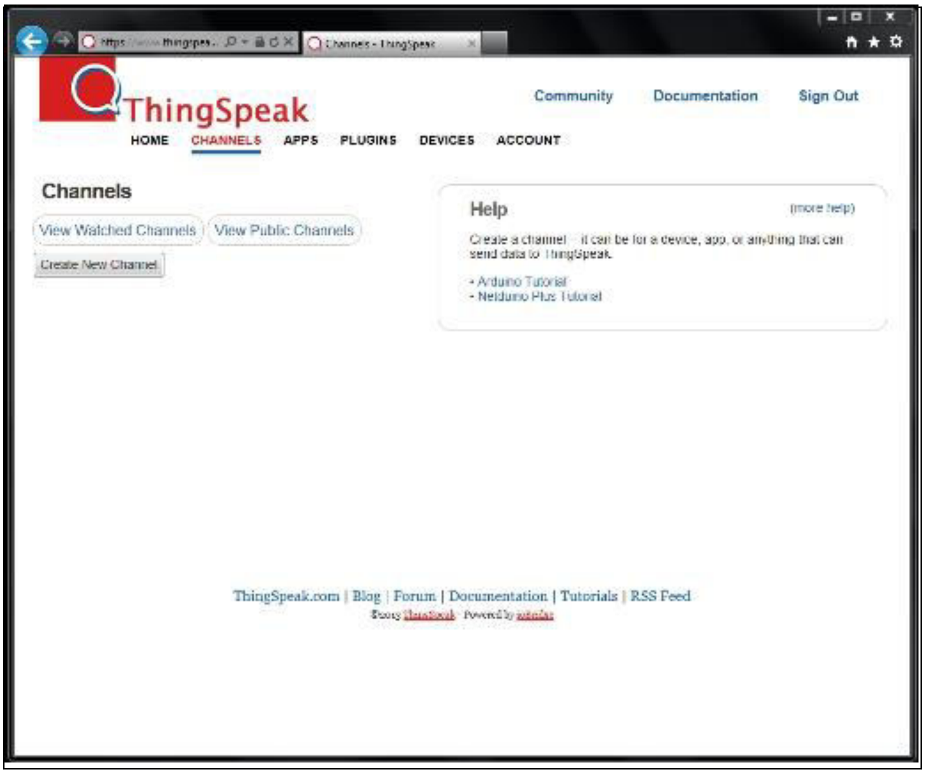


Figure 4.17 Creating new channel

6.

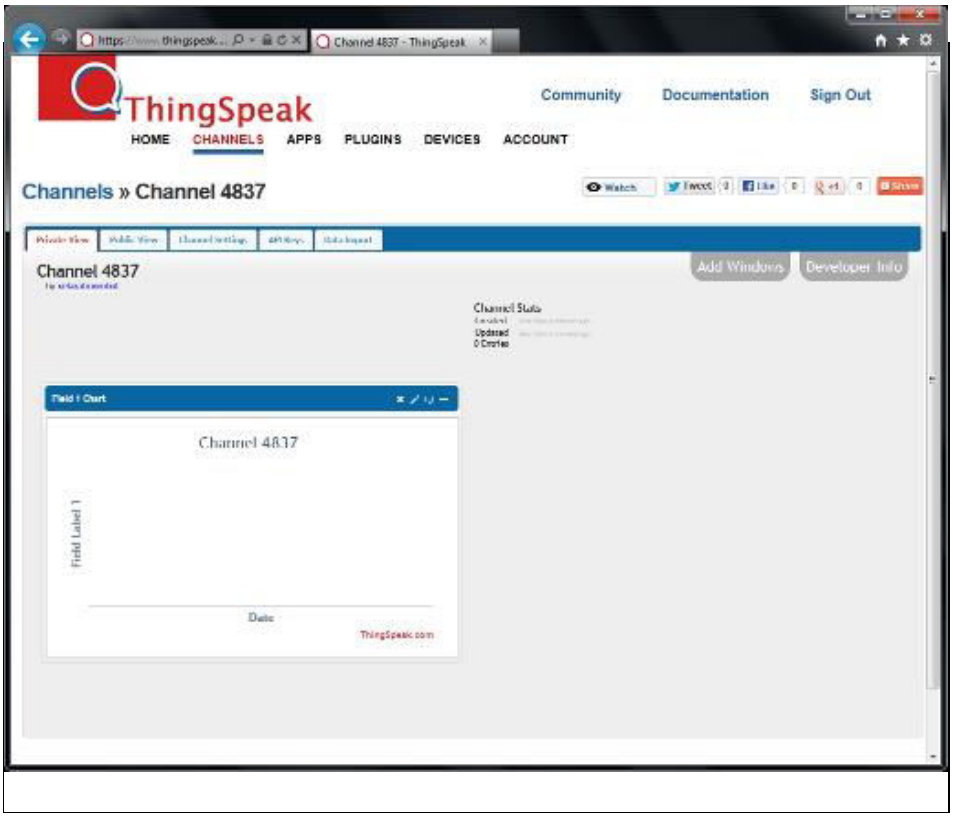


Figure 4.18 Channel settings

7. Fill Fields 1 to 6 with the following values: Left Wheel, Right Wheel, Vacuum, Bumper, Cliff and Battery.

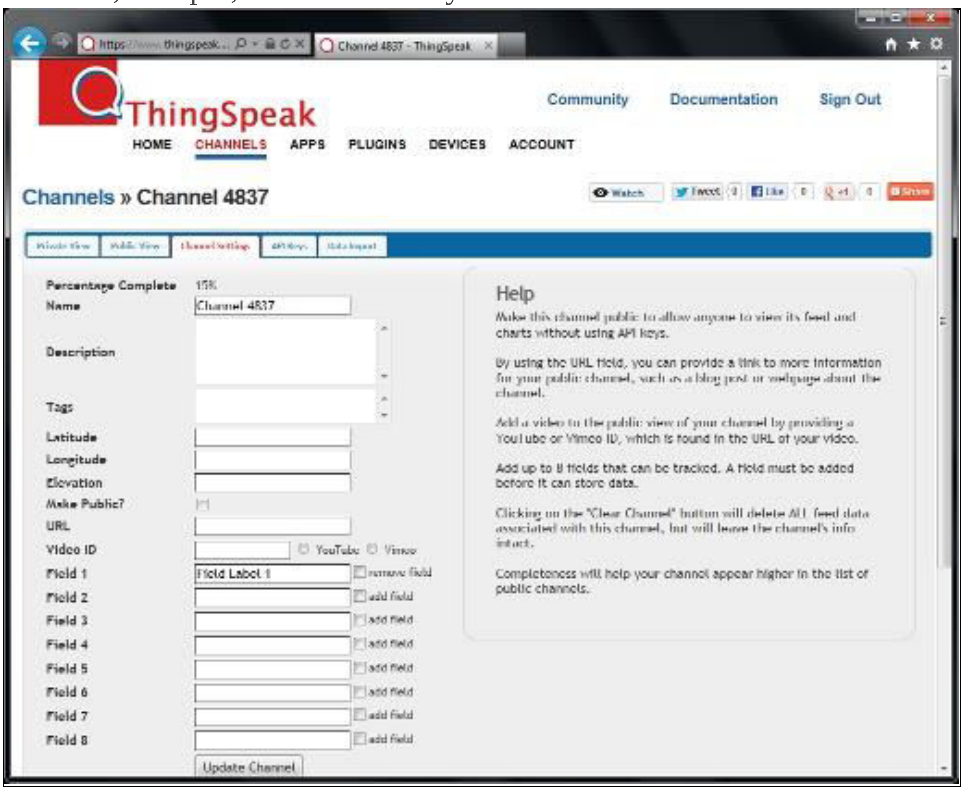


Figure 4.19 Filling the fields related to the project

8.

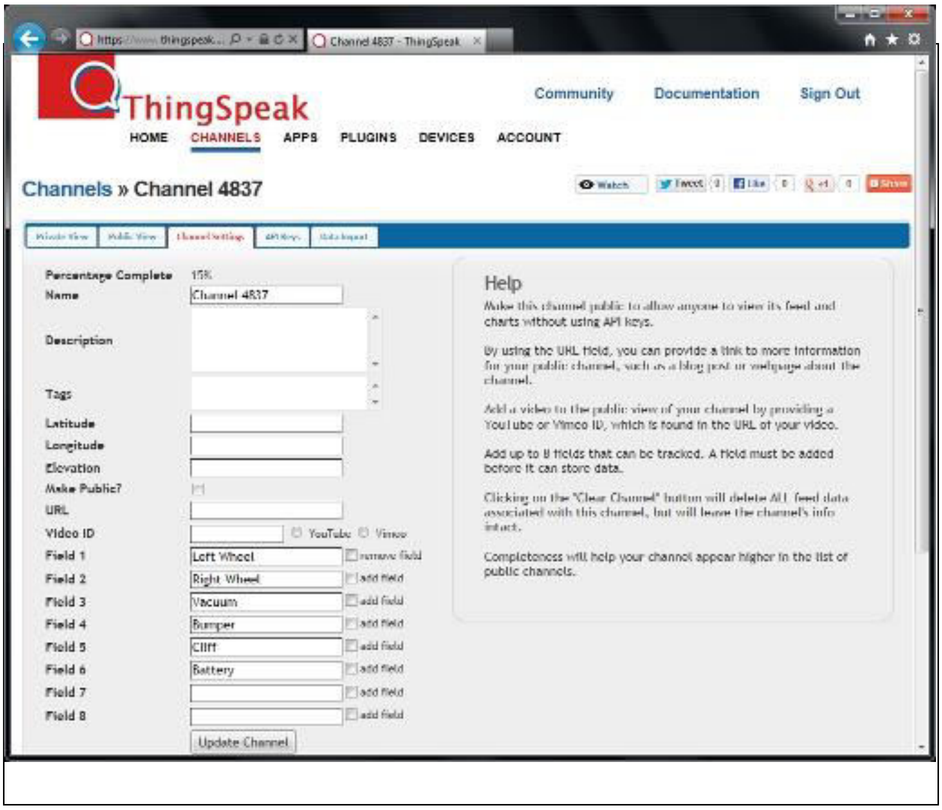


Figure 4.20 Update channel

9. Click the "API Key" tab to get the API key required by the installer.

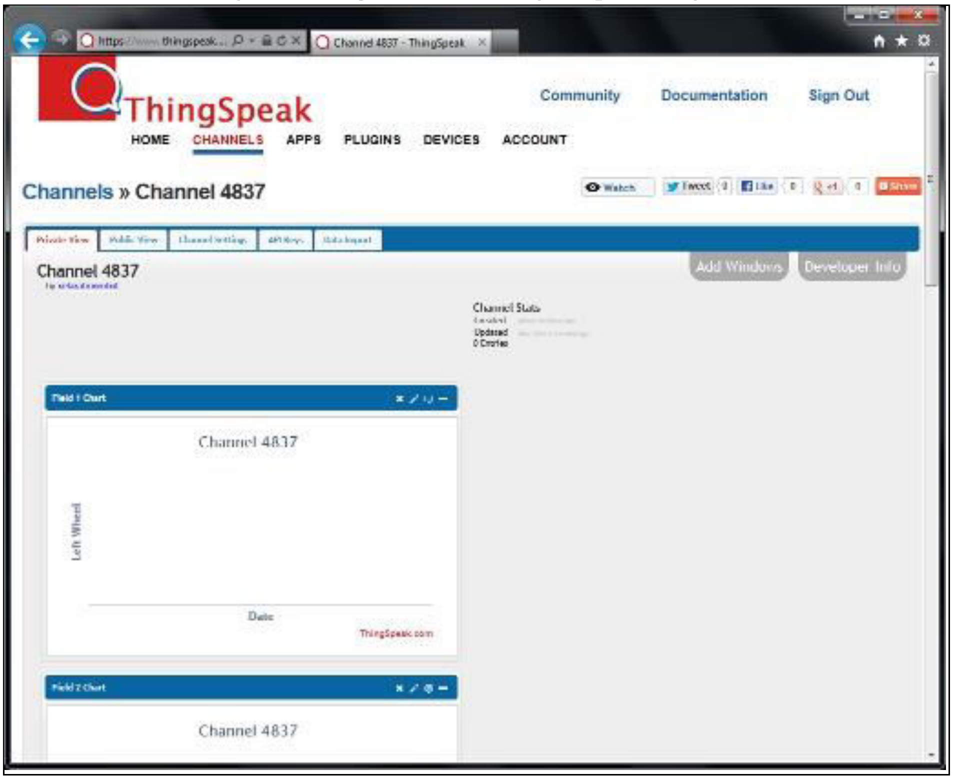


Figure 4.21 API Key generated

CHAPTER 5

RESULT

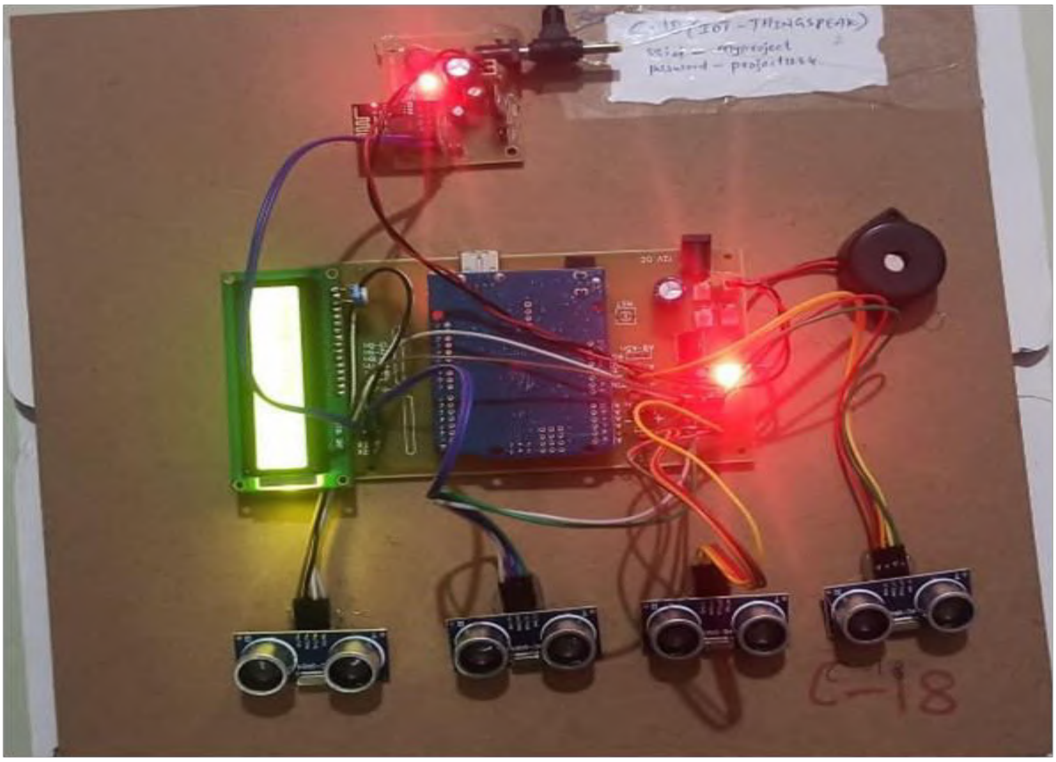
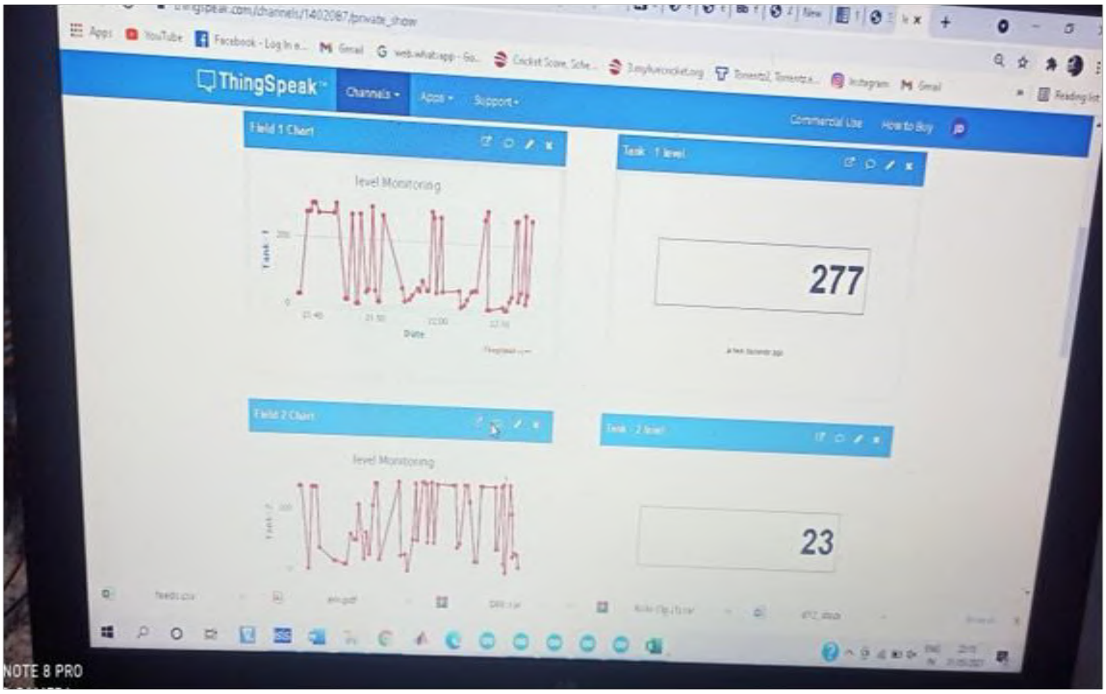


Figure: 5.1 Outlook of the project

Figure: 5.2 Output of the system on LCD screen



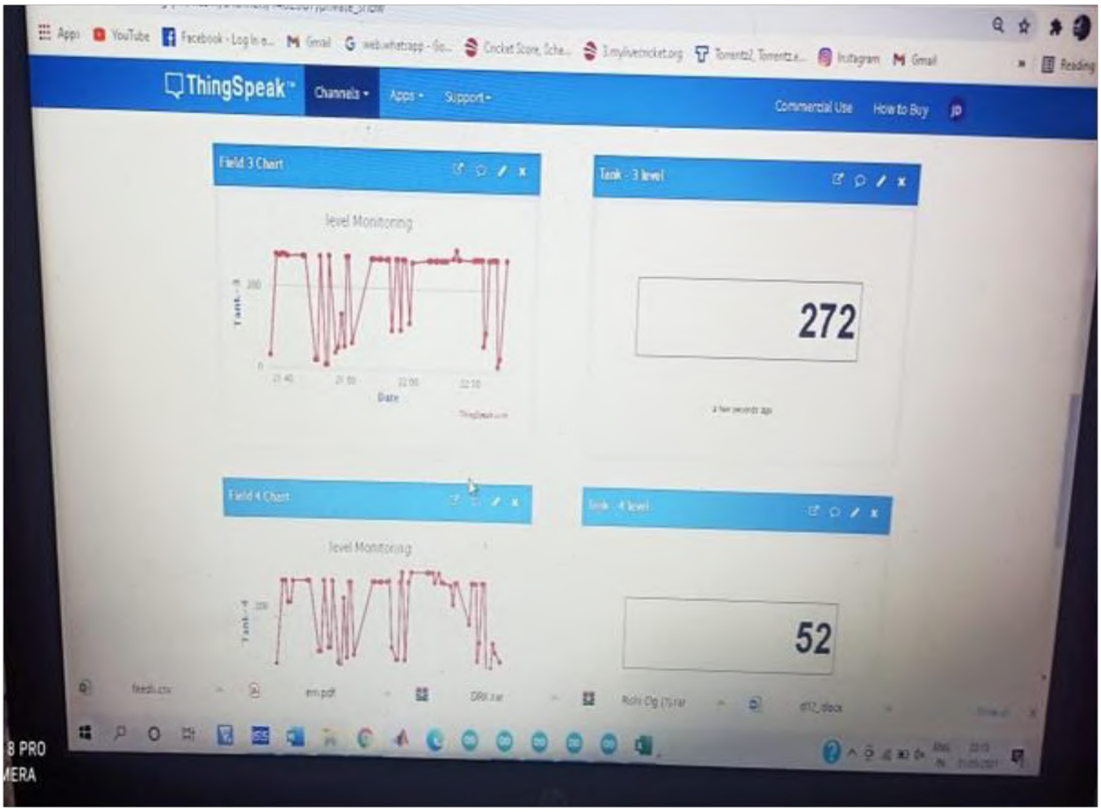


Figure:5.3 graphical and digital view of the 4 tank levels at cloud server

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

Finally the project has been successfully implemented and tested. I am proud to express my delight as the project I embarked upon is successfully finished in such a short span of time.The project proposes a simple liquid level monitoring system with different levels indicated. It also signifies when the water level is below and above then the requirement. Then users can manage the liquid resource and reduce the liquid wastage.

The most important feature of the system is that the users can see the liquid level and its status from anywhere. Thereby especially the industrial firms can manage the liquids effectively. The project gave us more confidence that we will be able to put it in practice, whatever the theoretical knowledge we gained during the course of study till now. It really persuades us to do more and more perhaps better way in future.

6.2 ADVANTAGES

1. This IOT liquid level monitoring system could be installed in that chemical industries which have different chemicals in liquid form and store these chemicals in a separate tank.
2. This IOT based system could be installed in mineral water industries to control the water level in a tank.
3. This system is less costly, more compact, more efficient and more reliable as
4. compared to other liquid level monitoring systems.
5. With IoT, Wireless Water Level Monitoring Becomes Cost-Effective
6. With IoT Human Error Could Be Eliminated
7. IoT Enables Efficient Water Tank Level Monitoring
8. Improved Production Planning with Level Monitoring
9. Leak Detection and Structural Integrity Control
10. Thefts and Tampering Protection

6.3 APPLICATIONS OF LIQUID LEVEL MONITORING SYSTEM

1. Oil Tank Level Monitoring
2. Water Tank Level Monitoring
3. Oil Tanker Monitoring
4. Weather Forecasting & Modelling
5. Diesel Fuel Tank Level Monitoring
6. Petrochemical Plants Monitoring
7. Flood Management
8. Smart Irrigation

6.4 FUTURE SCOPE

Future Work can involve the analysis of water level in a particular area so that the wastage of water is prevented. In future, the proposed system can be used to monitor and analyze water usage of the specific water source thus require developing such logic for the application.

We can also include the GSM-based system where the message will be sent to the particular person to prevent any kind of leakage.The system can also be used to recover the leakage liquid with the help of an electric valve. It opens after getting the information from the controller and the liquid gets stored in a reserved tank.

REFERENCES

•1. Song, Yongxian, Ma, Juanli, Zhang, Xianjin, Feng, Yuan: Design of Wireless Sensor Network-Based Greenhouse Environment Monitoring and Automatic Control System, Journal of Networks, vol. 7(5), 838–844 (2012).

•2. Sakthipriya, N.: An Effective Method for Crop Monitoring Using Wireless Sensor Network, Journal of Scientific Research, 20(9), 1127–1132 (2014).

•3. Dahikar, Snehal S., Rode, Sandeep V.: Agricultural Crop Yield Prediction Using Artificial Neural Network Approach, IJIREEICE, vol. 2(1), 683–685 (2014).

•4. Sonka, Steve: Big Data and the Ag Sector: More than Lots of Numbers, International Food and Agribusiness Management Review, vol. 17(1), 1–20 (2014).

•5. Atzberger, Clement: Advances in Remote Sensing of Agriculture: Context Description, Existing Operational Monitoring Systems and Major Information Needs, Remote Sensing, vol. 5(2), 949–981 (2013).

•6. Satyanarayana, G. V., Mazaruddin, S. D.: Wireless Sensor Based Remote Monitoring System for Agriculture Using ZigBee and GPS, Conference on Advances in Communication and Control Systems, 110–114 (2013).

•7. Patil, V.C., Gaadi, K. A. A., Biradar, D.P., Rangaswamy, M.: Internet of Things (IoT) and Cloud Computing for Agriculture: An Overview, pp. 292–296. Proceedings of AIPA, India (2012).

•8. TongKe, Fan: Smart Agriculture Based on Cloud Computing and IoT, Journal of Convergence Information Technology, vol. 8(2), 210–216 (2013).

•9. Carriots a Platform as a Service for Internet of Things, https://www.carriots.com/.

•10. Freeboard a Dashboard for visualization of Internet of Things Devices, http://freeboard.io/.

CHAPTER 1

INTRODUCTION TO IMAGE PROCESSING

INTRODUCTION

1.1 IMAGE:

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

The image is two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Fig 1 General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

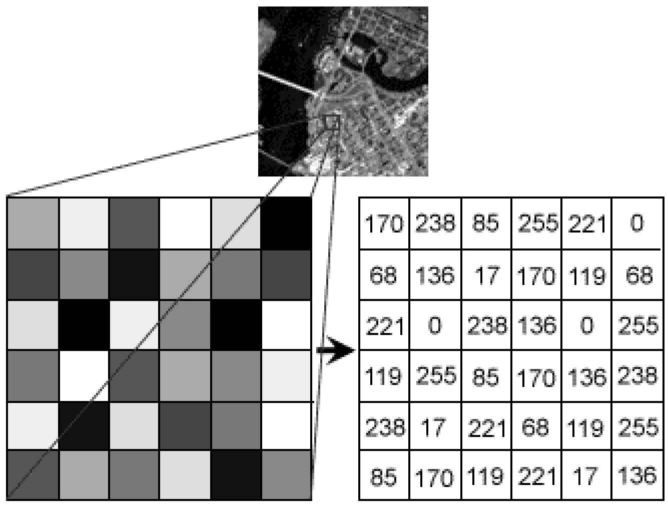


Fig 1.1 Image pixel

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

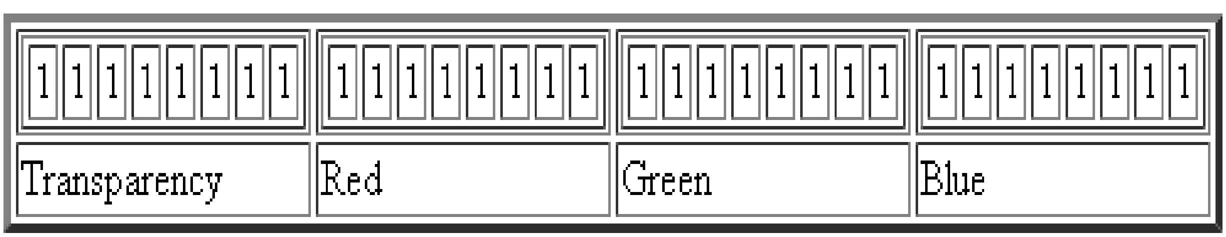


Fig1.2 Transparency image

1.2 IMAGE FILE SIZES:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High-resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 mega-pixels (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

1.3 IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

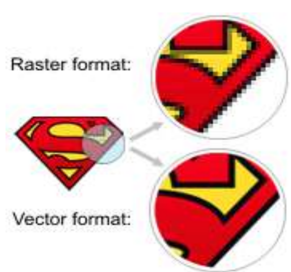


Fig1.3 Resolution image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

1.3.1 RASTER FORMATS:

These formats store images as bitmaps (also known as pix-maps)

- **JPEG/JFIF:**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

- **EXIF:**

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata is recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, the name of the camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

- **TIFF:**

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

- **PNG:**

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

- **GIF:**

GIF (Graphics Interchange Format) is limited to an 8-bit palette or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

- **BMP:**

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

1.3.2 VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM:**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

- **SVG:**

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all-purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

1.4 IMAGE PROCESSING:

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields suffers from myths, miss-connect ions, misunderstandings and misinformation. It is the vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor plowed with imprecise jargon.

Several factors combines to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low-cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

1.5 FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:

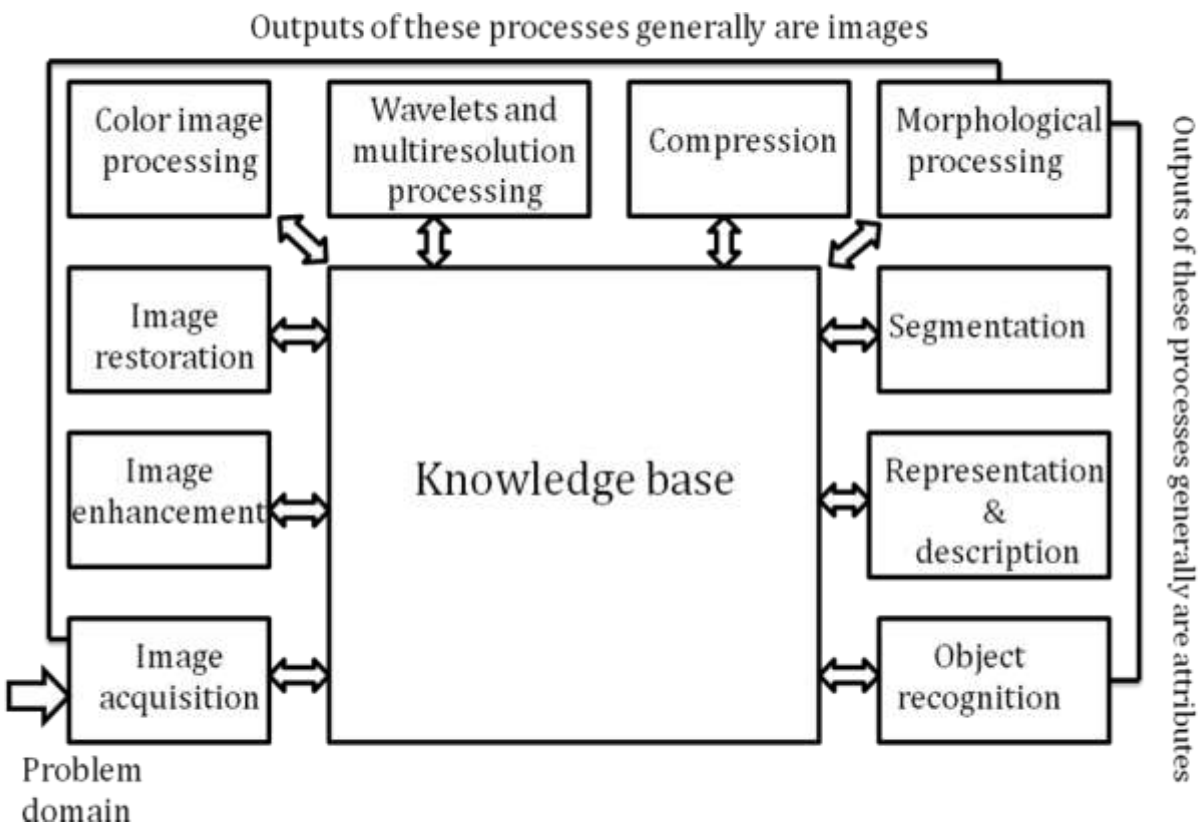


Fig 1.5 Image fundamental

1.5.1 Image Acquisition:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Fig 1.5.1 Digital camera image

The scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Fig 1.5.2 digital camera cell

1.5.2 Image Enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because "it looks better." It is important to keep in mind that enhancement is a very subjective area of image processing.

Fig 1.5.3 Image enhancement

1.5.3 Image restoration:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Fig 1.5.4 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, whereas removal of image blur by applying a deblurring function is considered a restoration technique.

1.5.4 Color image processing:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Fig 1.5.5 Color & Gray scale image

1.5.5 Wavelets and multiresolution processing:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform-based image processing since the late1950's, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

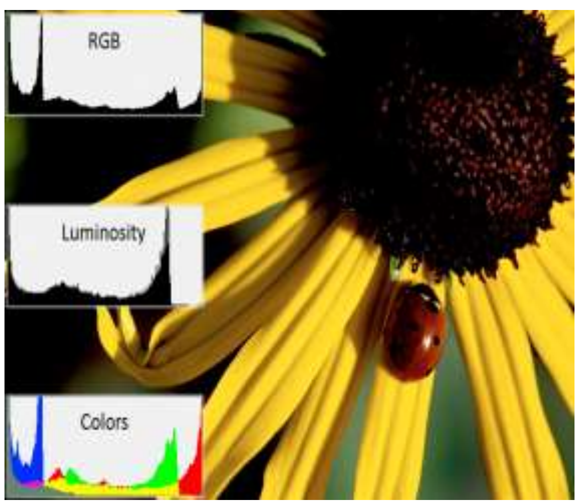


Fig 1.5.6 RGB histogram image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called **Multiresolution** theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub-band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

1.5.6 Compression:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which is characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

1.5.7 Morphological processing:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Fig 1.5.7 Blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

1.5.8 Segmentation:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward a successful solution of imaging problems that require objects to be identified individually.

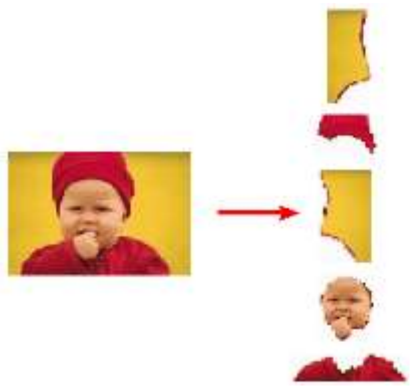


Fig 1.5.8 Image segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

1.5.9 Representation and description:

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

1.5.10 Object recognition:

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

1.5.11 Knowledgebase:

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an interrelated to list of all major possible defects in a materials inspection problem or an image database containing high-resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

1.6 COMPONENTS OF AN IMAGE PROCESSING SYSTEM:

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

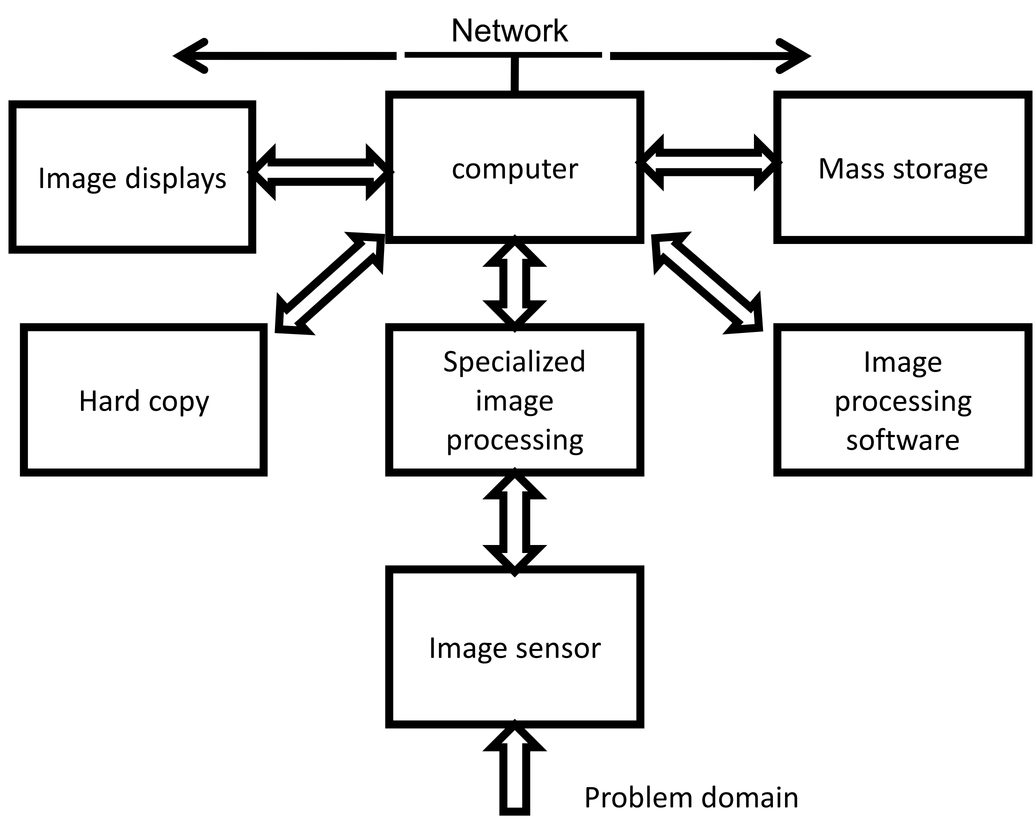


Fig 1.6 Component of image processing

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 1.24 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

- **Image sensors:**

With reference to sense, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

- **Specialized image processing hardware:**

Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

- **Computer:**

The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

- **Image processing software:**

Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

- **Mass storage:**

Mass storage capability is a must in image processing applications. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage for image processing applications falls into three principal categories: (1) short-term storage for use during processing, (2) online storage for relatively fast recall, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), G-bytes (meaning gigs, or one billion, bytes), and T-bytes (meaning term, or one trillion, bytes)

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. The latter method allows virtually instantaneous image zoom, as well as a scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig. 1.24. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing online storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but the infrequent need for access. Magnetic tapes and optical disks housed in “jukeboxes” are the usual media for archival applications.

- **Image displays:**

Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are their requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

- **Hardcopy:**

Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. The film provides the highest possible resolution, but the paper is the obvious medium of choice for the written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

- **Network:**

Networking is almost a default function in any computer system in use today. Because of a large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of the optical fiber and other broadband technologies.

CCM:

In music

- Contemporary Christian music, a genre of popular music which is lyrically focused on matters concerned with the Christian faith
    - *CCM Magazine*, a magazine that covers Contemporary Christian music
- University of Cincinnati College-Conservatory of Music, the performing arts college of the University of Cincinnati

- Cincinnati Conservatory of Music, a conservatory formed in 1867 as part of a girls' finishing school which later became part of the University of Cincinnati College-Conservatory of Music.
- Contemporary Commercial Music
- In the context of MIDI: control change message.

In cryptography

- CCM mode, a mode of operation for cryptographic block ciphers
- Combined Cipher Machine, a common cipher machine system for securing Allied communications during World War II

In medicine

- Cerebral cavernous malformation, a vascular disorder of the central nervous system that may appear either sporadically or exhibit autosomal dominant inheritance
- Classical Chinese medicine, a medicine that developed from germ theory
- Comprehensive Care Management, a member of the Beth Abraham Family of Health Services
- *Critical Care Medicine*, a peer-reviewed medical journal in the field of critical care medicine

In politics

- Chama Cha Mapinduzi, the ruling political party of Tanzania
- Crown Council of Monaco, a seven-member administrative body which meets at least twice annually to advise the Prince of Monaco on various domestic and international affairs
- Convention on Cluster Munitions is an international treaty that prohibits the use of cluster bombs, a type of explosive weapon which scatters submunitions ("bomblets") over an area.
- Coalition for a Conservative Majority, a nonprofit, political advocacy group organized by Tom DeLay and Kenneth Blackwell

In religion

- Catholic Campus Ministry, a Catholic student organization on many college campuses
- Council of Churches of Malaysia, an ecumenical body in Malaysia comprising mainline Protestant churches and Oriental Orthodox Church
- Christ Church Manchester, otherwise known as CCM is a church in Manchester that meets in numerous locations
- Christian Compassion Ministries, a mission organization in the Philippines

In sports

- CCM (The Hockey Company), a manufacturing company of Canada
- CCM (cycle), a manufacturing company of Canada
- Central Coast Mariners FC, an Australian professional football (soccer) team based on the Central Coast of New South Wales, Australia

In technology

- Continuous Controls Monitoring describes techniques of continuously monitoring and auditing an IT system
- Continuous Current Mode, operational mode of DC-DC converters
- Cisco CallManager, a Cisco product
- Cloud Computing Manifesto
- Configuration & Change Management
- CORBA Component Model, a portion of the CORBA standard for software componentry
- A deprecated abbreviation for the cubic centimeter unit of volume measurement

In transportation

- CCM (cycle), a cycle manufacturer
- CCM Airlines, a regional airline based in Ajaccio, Corsica, France

- Clews Competition Motorcycles, a British motorcycle manufacturer based in Blackburn, England
- Cabin Crew Member, another name for flight attendant

In education

- County College of Morris, a two-year, public community college located off of Route 10 on Center Grove Road in Randolph Township, New Jersey
- City College Manchester, a Further Education college in the United Kingdom.
- City College of Manila, in the Philippines.

In military

- Center for Countermeasures, a United States military center based at White Sands Missile Range, New Mexico
- Command Chief Master Sergeant, a position in the United States Air Force

In other fields

- Corn cob mix, a kind of silage consisting of corn cobs and kernels.
- El Centro Cultural de Mexico, an alternative space in Santa Ana, Orange County, California
- Cerberus Capital Management, a large privately owned hedge fund
- Certified Consulting Meteorologist, a person designated by the American Meteorological Society to possess attributes as they pertain to the field of meteorology
- Crime Classification Manual, FBI produced text for a standardized system to investigate and classify violent crimes.
- Cervecería Cuauhtémoc Moctezuma, a major brewery in Mexico that produces brands such as Dos Equis and Tecate.

Color and texture are two low-level features widely used for image classification, indexing, and retrieval. Color is usually represented as a histogram, which is a first order statistical measure that captures the global distribution of color in an image One of the main drawbacks of the histogram- based approaches are that the spatial distribution and local variations in color are ignored. Local spatial variation of pixel intensity is commonly used to capture texture information in an image. Grayscale Co-occurrence Matrix (GCM) is a well-known method for texture extraction in the spatial domain. A GCM stores the number of pixel neighborhoods in an image that has a particular grayscale combination. Let I am an image and let p and Np respectively denote any arbitrary pixel and its neighbor in a given direction. If GL denotes the total number of quantized gray levels and GL denotes the individual gray levels, where, GL $\in$ {0,., GL _ 1}, then each component of GCM can be written as follows:

$$\mathrm{gcm}(i, j) = \mathbf{Pr}((\mathrm{gl}_{\mathrm{p}}, \mathrm{gl}_{N_{\mathrm{p}}}) = (i, j))$$

Jim(i, j) is the number of times the gray level of a pixel p denoted by glp equals I, and the gray level of its neighbor Np denoted by glop equals j, as a fraction of the total number of pixels in the image. Thus, it estimates the probability that the gray level of an arbitrary pixel in an image is I, and that of its neighbor is j. One GCM matrix is generated for each possible neighborhood direction, namely, 0, 45, 90 and 135.Average and range of 14 features like Angular Second Moment, Contrast, Correlation, etc., are generated by combining all the four matrices to get a total of 28 features. In the GCM approach for texture extraction, color information is completely lost since only pixel gray levels are considered.

To incorporate spatial information along with the color of image pixels, a feature called color correlogram has recently been proposed. It is a three-dimensional matrix that represents the probability of finding pixels of any two given colors at a distance ‘d’ apart Auto correlogram is a variation of correlogram, which represents the probability of finding two pixels with the same color at a distance ‘d’ apart. This approach can effectively represent color distribution in an image. However, correlogram features do not capture intensity variation. Many image databases often contain both colors as well as gray scale images. The color correlogram method does not constitute a good descriptor in such databases.

Another method called Color Co-occurrence Matrix (CCM) has been proposed to capture color variation in an image. CCM is represented as a three-dimensional matrix, where a color pair of the pixels p and Np are captured in the first two dimensions of the matrix and the spatial distance' between these two pixels is captured in the third dimension. This approach is a generalization of the color correlogram and reduces to the pure color correlogram for d = 1. CCM is generated using only the Hue plane of the HSV (Hue, Saturation, and Intensity Value) color space. The Hue axis is quantized into HL number of levels. If individual hue values are denoted by hl, where $\mathrm{hl} \in \{0, \ldots, \mathrm{HL} - 1\}$, then each component of CCM can be written as follows:

$$\mathrm{ccm}(i, j) = \Pr((\mathrm{hl}_{\mathrm{p}}, \mathrm{hl}_{N_{\mathrm{p}}}) = (i, j))$$

Four matrices representing neighbors at angles 0, 90, 180 and 270 are considered. This approach was further extended by separating the diagonal and the non-diagonal components of CCM to generate a Modified Color Co-occurrence Matrix (MCCM). MCCM, thus, may be written as follows: MCCM = (CCMD;CCMND)

Here, CCMD and CCMND correspond to the diagonal and off-diagonal components of CCM. The main drawback of this approach is that, like correlogram, it also captures only color information and intensity information is completely ignored.

An alternative approach is to capture intensity variation as a texture feature from an image and combine it with color features like histograms using suitable weights. One of the challenges of this approach is to determine suitable weights since these are highly application-dependent. In certain applications like Content-based Image Retrieval (CBIR), weights are often estimated from relevance feedback given by users.

While relevance feedback is sometimes effective, it makes the process of image retrieval user-dependent and iterative. There is also no guarantee on the convergence of the weight-learning algorithms. In order to overcome these problems, researchers have tried to combine color and texture features together during extraction.

proposed two approaches for capturing color and intensity variations from an image using the LUV color space. In the Single-channel Co-occurrence Matrix (SCM), variations for each color channel, namely, L, U, and V are considered independently. In the Multichannel Co-occurrence Matrix (MCM), variations are captured taking two channels at a time – UV, LU, and LV. Since the LUV color space separates out chrominance (L and U) from luminance (V), SCM in effect generates one GCM and two CCMs from each image independently. As a result, correlation between the color channels is lost

However, in MCM, the count of pair wise occurrences of the values of different channels of the color space is captured. Thus, each component of MCM can be written as follows:

$$mcmUV(i; j) = Pr((up; vNp) = (i; j))$$

$$mcmLU(i; j) = Pr((lp; uNp) = (i; j))$$

$$mcmLV(i; j) = Pr((lp; vNp) = (i; j))$$

Here, mcmUV(i, j) is the number of times the U chromaticity value of a pixel p denoted by up equals I, and the V chromaticity value of its neighbor Np denoted by VPN equals j, as a fraction of the total number of pixels in the image. Similarly, mcm(i, j) and mcmLV(i, j) are defined. One MCM matrix is generated for each of the four neighborhood directions, namely, 0, 45, 90 and 135.

Deng and Manjunath (2001) proposed a two-stage method called JPEG, which combines color and texture after image segmentation. In the first stage, colors are quantized to the required levels for differentiating between various regions of an image. Pixel values of the regions are then replaced by their quantized color levels to form a color map. Spatial variation of color levels between different regions in the map is viewed as a type of texture composition of the image.

Yu et al. (2002) suggested the use of color texture moments to represent both color and texture of an image. This approach is based on the calculation of

Local Fourier Transformation (LFT) coefficients. Eight templates equivalent to LFT are operated over an image to generate a characteristic map of the image. Each template is a 3 · 3 filter that considers eight neighbors of the current pixel for LFT calculation. First and second-order moments of the characteristic map are then used to generate a set of features.

In this paper, we propose an integrated approach for capturing spatial variation of both color and intensity levels in the neighborhood of each pixel using the HSV color space. In contrast to the other methods, for each pixel and its neighbor, the amount of color and intensity variation between them is estimated using a weight function. Suitable constraints are satisfied while choosing the weight function for effectively relating visual perception of color and the HSV color space properties. The color and intensity variations are represented in a single composite feature known as Integrated Color and Intensity Co-occurrence Matrix (ICICM). While the existing schemes generally treat color and intensity separately, the proposed method provides a composite view of both color and intensity variations in the same feature. The main advantage of using ICICM is that it avoids the use of weights to combine individual color and texture features. We use the ICICM feature in an image retrieval application from large image databases.

The Early result of this work was reported in (Vadivel et al., 2004a). In the next section, we describe the proposed feature extraction technique after introducing some of the properties of the HSV color space. Choice of quantization levels for color and intensity axes, selection of parameter values and a brief overview of the image retrieval application

Integrated color and intensity co-occurrence matrix:

We propose to capture color and intensity variation around each pixel in a two-dimensional matrix called Integrated Color and Intensity Co-occurrence Matrix (ICICM). This is a generalization of the Grayscale Co-occurrence Matrix and the Color Co-occurrence Matrix techniques. For each pair of neighboring pixels, we consider

their contribution to both color perception as well as gray level perception to the human eye. Some of the useful properties of the HSV color space and their relationship to human color perception are utilized for extracting this feature. In the next sub-section, we briefly explain relevant properties of the HSV color space. In the subsequent subsection, we describe how the properties can be effectively used for generating ICICM.

HSV color space:

HSV Color space: Basically, there are three properties or three dimensions of color that being hue, saturation and value HSV means Hue, Saturation and Value. It is important to look at because it describes the color based on three properties. It can create the full spectrum of colors by editing the HSV values. The first dimension is the Hue. Hue is the other name for the color or the complicated variation in the color. The quality of color as determined by its dominant wavelength. This Hue is broadly classified into three categories. They are primary Hue, Secondary Hue and Tertiary Hue. The first and the foremost is the primary Hue it consists of three colors they are red, yellow and blue. The secondary Hue is formed by the combination of the equal number of colors of the primary Hue and the colors of the secondary Hue which was formed by the primary Hue are Orange, Green and violet. The remaining one is the tertiary Hue is formed by the combination of the primary Hue and the secondary Hue. The limitless number of colors are produced by mixing the colors of the primary Hue in different amounts. Saturation is the degree or the purity of color. Then the second dimension is the saturation. Saturation just gives the intensity to the colors. The saturation and intensity drop just by mixing the colors or by adding black to the color. By adding the white to the color in spite of more intense the color becomes lighter. Then finally the third dimension is the Value. The value is the brightness of the color. When the value is zero the color space is totally black with the increase in the color there is also increase in the brightness and shows the various colors. The value describes the contrast of the color. That means it describes the lightness and darkness of the color. As similar to the saturation this value consists of the tints and shades. Tints are the colors with the added white and shades are the colors with the added black.

Properties of the HSV color space:

Sensing of light from an image in the layers of human retina is a complex process with rod cells contributing to scotopic or dim-light vision and cone cells to photopic or bright-light vision (Gonzalez and Woods, 2002). At low levels of illumination, only the rod cells are excited so that only gray shades are perceived. As the illumination level increases, more and more cone cells are excited, resulting in increased color perception. Various color spaces have been introduced to represent and specify colors in a way suitable for storage, processing or transmission of color information in images. Out of these, HSV is one of the models that separate out the luminance component (Intensity) of a pixel color from its chrominance components (Hue and Saturation). Hue represents pure color, which is perceived when incident light is of sufficient illumination and contains a single wavelength. Saturation gives a measure of the degree by which a pure color is diluted by white light. For light with low illumination, corresponding intensity value in the HSV color space is also low.

The HSV color space can be represented as a Hexa cone, with the central vertical axis denoting the luminance component, I (often denoted by V for Intensity Value). Hue, is a chrominance component defined as an angle in the range [0,2p] relative to the red axis with red at angle 0, green at 2p/3, blue at 4p/3 and red again at 2p. Saturation, S, is the other chrominance component, measured as a radial distance from the central axis of the hexa-cone with value between 0 at the center to 1 at the outer surface. For zero saturation, as the intensity is increased, we move from black to white through various shades of gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the color represented by its hue. When saturation is near 0, all the pixels in an image look alike even though their hue values are different.

As we increase saturation towards 1, the colors get separated out and are visually perceived as the true colors represented by their hues. Low saturation implies presence of a large number of spectral components in the incident light, causing loss of color information even though the illumination level is sufficiently high. Thus, for low values of saturation or intensity, we can approximate a pixel color by a gray level while for higher saturation and intensity, the pixel color can be approximated by its hue. For low intensities, even for a high saturation, a pixel color is close to its gray value. Similarly, for low saturation even for a high value of intensity, a pixel is perceived as gray. We use these properties to estimate the degree by which a pixel contributes to color perception and gray level perception.

One possible way of capturing color perception of a pixel is to choose suitable thresholds on the intensity and saturation. If the saturation and the intensity are above their respective thresholds, we may consider the pixel to have color dominance; else, it has gray level dominance. However, such a hard thresholding does not properly capture color perception near the threshold values. This is due to the fact that there is no fixed level of illumination above which the cone cells get excited. Instead, there is a gradual transition from scotopic to photopic vision. Similarly, there is no fixed threshold for the saturation of cone cells that leads to loss of chromatic information at higher levels of illumination caused by color dilution. We, therefore, use suitable weights that vary smoothly with saturation and intensity to represent both color and gray scale perception for each pixel.

NON-INTERVAL QUANTIZATION:

Due to the large range for each component by directly calculating the characteristics for the retrieval then the computation will be very difficult to ensure rapid retrieval. It is essential to quantify HSV space component to reduce computation and improve efficiency. At the same time, because the human eye to distinguish colors is limited, do not need to calculate all segments. Unequal interval quantization according the human color perception has been applied on H, S, V components.

Based on the color model of substantial analysis, we divide color into eight parts. Saturation and intensity are divided into three parts separately in accordance with the human eyes to distinguish. In accordance with the different colors and subjective color perception quantification, quantified hue(H), saturation(S) and value(V)

In accordance with the quantization level above, the H, S, V three-dimensional feature vector for different values of with different weights to form one dimensional feature vector and is given by the following equation:

G = Qs*Qv*H+Qv*s+V

Where Qs is the quantized series of S and Qv is the quantized series of V. And now by setting Qs = Qv = 3, Then G = 9H+3S+V

$$H = \begin{cases} 0 & if \quad h \in [316,20] \\ 1 & if \quad h \in [21,40] \\ 2 & if \quad h \in [41,75] \\ 3 & if \quad h \in [76,155] \\ 4 & if \quad h \in [156,190] \\ 5 & if \quad h \in [191,270] \\ 6 & if \quad h \in [271,295] \\ 7 & if \quad h \in [296,315] \end{cases} \qquad S = \begin{cases} 0 & if \quad s \in [0,0.2) \\ 1 & if \quad s \in [0.2,0.7) \\ 2 & if \quad s \in [0.7,1) \end{cases} \qquad V = \begin{cases} 0 & if \quad v \in [0,0.2) \\ 1 & if \quad v \in [0.2,0.7) \\ 2 & if \quad v \in [0.7,1) \end{cases}$$

In this way three component vector of the HSV from one dimensional vector, which quantize the whole color space for the 72 kinds of the main colors. We can handle 72 bins of one-dimensional histogram. This qualification is effective in reducing the images by the effect of the light intensity, but also reducing the computational time and complexity.

IMAGE RETRIEVAL:

Image retrieval is nothing but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords or the descriptions to the images so that the retrieval can be performed. Manual image annotation is time consuming, expensive and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc.

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others. one can today generate volumes of images with content as diverse as family get-togethers and national park visits. Low-cost storage and easy Web hosting fueled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchable image data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real-world image search system designers.

As far as technological advances are concerned, growth in Content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real-world implications, applications, and constraints of the technology. Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future.

An image retrieval system designed to serve a personal collection should focus on features such as personalization, flexibility of browsing, and display methodology. For example, Google's Picasa system [Picasa 2004] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods. The notable few applications include an FPGA implementation of a color-histogram-based image retrieval system [Kotoulas and Andreadis 2003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 2003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 2002].

Discussion. Regardless of the nature of the collection, as the expected user-base grows, factors such as concurrent query support, efficient caching, and parallel and distributed processing of requests become critical. For future real-world image retrieval systems, both software and hardware approaches to address these issues are essential. More realistically, dedicated specialized servers, optimized memory and storage support, and highly parallelizable image search algorithms to exploit cluster computing powers are where the future of large-scale image search hardware support lies.

OVERVIEW OF TEXTURE:

We all know about the term Texture but for defining it is a hard time. One can differentiate the two different Textures by recognizing the similarities and differences. Commonly there are three ways for the usage of the Textures:

Based on the Textures the images can be segmented to differentiate between already segmented regions or to classify them. We can reproduce Textures by producing the descriptions. The texture can be analyzed in three different ways. They are Spectral, Structural and Statistical:

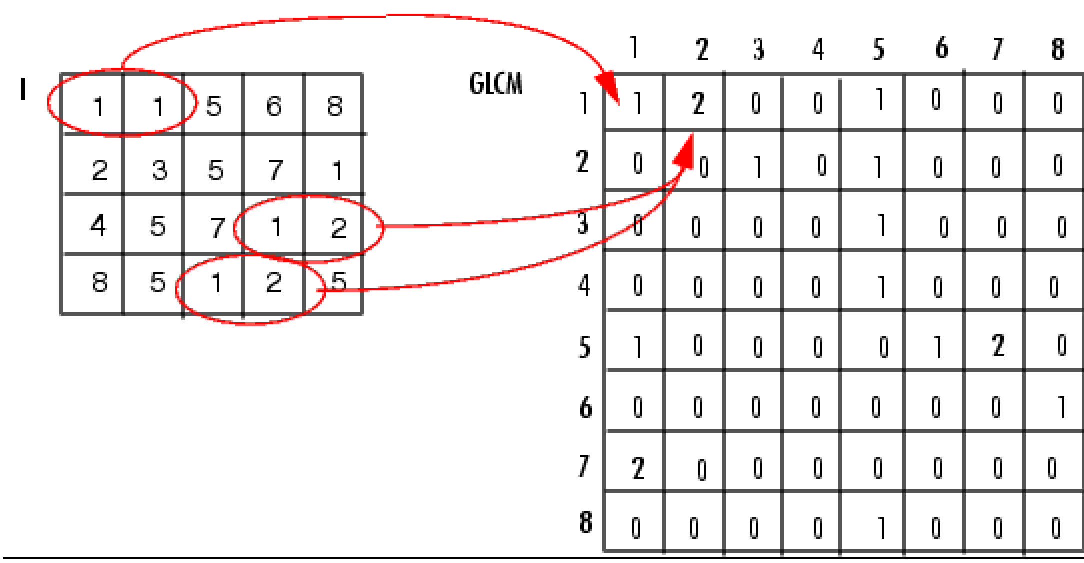


Fig 1.7 Over-view of texture

CHAPTER 2

REVIEW OF RELATED LITERATURE

2.1 Literature Review

This part summarizes some recent methods related to image forgery detection as an overview and also references from which the researchers have new ideas and solutions. For copy move detection, the searching of similar regions is the main purpose in almost all methods while the searching of inconsistencies of features is considered the solution in splicing detection. Although there are numerous related methods published, most of them solve problems of copy-move or splicing separately and only few papers can solve problems of both copy-move and splicing in the same image. Therefore, developing an algorithm to detect any forgery regions, not limited to copy-move or forgery, is still a challenge for scientists in the field of image forensics.

A. Copy-Move Forgery Detection For copy-move detection, a survey in [3] covers and evaluates methods published until 2012 in which the duplicated regions are confirmed based on feature vectors comparisons. Feature vectors can be extracted directly from tested images or after applying a transformation such as DWT and DCT. The difference on feature extractions and the way to compare feature vectors comprise the variety in methods. After that, a new method to extract the image features by describing the spatial structure of the gray image texture called Local Binary Pattern (LBP) was introduced by Leida Li et al. [4] in 2013. In the case of color image, it should be first converted to gray image by using $I=0.299R+0.587G+0.114B$ and low pass filter should be applied to obtain the low-frequency features which is more stable than the high-frequency ones. As the previous methods, the feature matching is defined based on the threshold. Moreover, the post-processing including a special designed filter and morphological operations is also considered in the process of detection. The method is robust to JPEG compression, noise contamination, blurring, rotation and flipping. However, it is difficult to detect the rotated regions with general angles. Investigation of invariant block features and appropriate selection of the dimension of features are suggested to improve the random rotation. Using Undecimated Dyadic Wavelet Transform (UDWT) and Zernike moments is proposed as a new method to detect the forgery in copy-move images by Jiyun Yang [5] in 2013. In this paper, the applying UDWT is firstly used to collect the low frequencies (LL) components. Traditional ZMs is then computed to produce feature vectors of overlapping blocks on LL and find the copied regions from these vectors. Lexicographical sorting, correlation coefficients with a threshold value are used to find the similar vectors and limit the exact forged blocks from the groups of similar vectors obtained in lexicographical sorting step, respectively. Blur invariants are also used to produce feature vectors in copy-move image forgery detection [6]. Based on this idea, the LL sub-band from DWT of an image using Harr basis is divided into small overlapping blocks whose features are then represented by blur moment invariants. Each block feature vector consists of 24 blur invariants in case of grayscale images and 72 ones in the RGB and is reduced dimension by applying PCA. The block similarity analysis will detect the duplicated regions by considering the Euclidean distances and a user defined threshold. This is applied to image with noise, blur and contrast changes. The applying other basis or DCT is suggested in the coming research. In [7], an image is decomposed into four sub-bands using DWT in which the LL sub-band is considered for the coming steps. The proposed algorithm uses SIFT on each small overlapping block divided from LL sub-band to extract feature vector. These feature vectors are used to create a descriptor vector and compared to detect if there is a copy-move manipulation in the image. This method is checked with MICC-F200 database with high accuracy, less time, robust to scale and rotation. The authors of this algorithm developed SUFT (Speed-Up Robust Feature) to extract the features of image block instead of using SIFT as in [7]. The combination of SUFT and DWT, DyWT are also presented. With the results obtained from the proposed method, SUFT is proved faster than SIFT while SIFT is mostly used to select the invariant features [8]. B. Splicing Forgery Detection Splicing is more complex than copy-move, not only in the forgery manipulation but also in detection. The key idea of many splicing detection methods is searching regions being inconsistent with camera characteristics or image features. Regions which are resampled, double compressed, and those with blur discrepancies or sharpness differences can be considered traces of splicing. However, because of the variety of splicing, more and more algorithms have been developed in recent years. Conditional Co-occurrence Probability Matrix (CCPM) is used to detect the splicing in image based on the third order statistical features [9]. CCPM contains the discriminative information which are included in higher order statistical features and independent to the image features. However, the higher dimensionality of features is, the more complex computation is. Therefore, principal Component Analysis (PCA) is also used to improve the computational complexity of the proposed method which is robust and better than Markov features both in spatial domain and block discrete cosine transform (BDCT) domain. Rescaling and its factor are used to detect the forgery caused by splicing [10]. A region copied from an image will be resized or scaled before pasting to the destination image. Scaling makes the pasted portion resampled and inconsistent. In addition, properties of the zero-crossing of the second difference are considered to calculate the scaling factor with different interpolation schemes. The algorithm of rescale detection and estimation proposed clearly in five steps including pre-processing to convert the RGB to grayscale and extract Y component from YCbCr conversion; calculate the second difference, their zero-crossing and Discrete Fourier Transform (DFT) before searching for the periodicity and peak detection. Differences of JPEG compression in an image can be caused by the splicing [11]. JPEG forgery detection based on 8x8 block Discrete Cosine Transform (DCT) transform to detect the shift of DCT block alignment. The splicing detection was proposed by analyzing and suggesting solutions for cases making the differences in compression history including detections of Aligned Double JPEG, non-aligned Double JPEG, Primary Quantization Table, JPEG ghost. Illumination inconsistencies and intrinsic resampling properties are also parameters to detect the splicing [12]. The first requires an input image and a database for training. The algorithm begins with 30x30 blocks which will be transformed into an opponent color space HSV before extracting features of contrast and mean. The contrast is calculated from the standard deviation while the mean is obtained by computing the average grey level. These features will decide suitable algorithm. Illumination color estimation, illumination map creation, Wavelet-based features extraction and classifier are the following steps of the proposed method by illumination differences detection. The second solution in this paper proposes a resampling detection scheme to detect forgery in which second difference in horizontal or vertical, Radon transform, FFT of covariance, high-pass filtering, feature extraction and classifier are included. C. Forgery Detection for both Copy-Move and Splicing An integrated technique, which combines DCT and Speeded Up Robust Features (SURF), to detect the image forgery in term of copy-move or splicing was proposed in 2011 [13]. This means the tested images can be optional, not classified in copy-move or splicing in advance. The paper finds new traces based on recompression to detect the counterfeit of recompressed images. Periodicity analysis with double compression effect in both spatial and DCT domain is applied before using SURF descriptor to against the variation of rotation and scaling. The proposed method located the forgery regions efficiently for both copy-move and splicing image, especially, discriminated the positions of original and forged regions. At the EUROCON 2013 in Croatia, a method can detect both copy-move and splicing in image using a multi-resolution Web Law Descriptor (WLD) was presented [14]. The algorithm firstly converts a RGB image into YCbCr so that the WLD can extract the features from chrominance components which are less sensitive than luminance. To extract the features effectively, WDM are expressed by two components of differential excitation and orientation, which based on Weber' law. The multi-resolution WLD histogram is comprised of the histogram of three neighbors of (8,1), (16,2), and (24,3) where the first argument is the number of neighboring pixels and the second is the radius of the neighbors from the center pixel. A support vector machine, which involves to training and testing image, is used for classification purpose.

CHAPTER 3

SOFTWARE INTRODUCTION

3.1. Introduction to MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

3.2 The MATLAB system:

The MATLAB system consists of five main parts

- **Development Environment**:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library**:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language**:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

- **Graphics:**

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

3.3 GRAPHICAL USER INTERFACE (GUI):

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.) and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension.fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.

- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type

guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted,

GUIDE opens a new (i.e., blank) window.

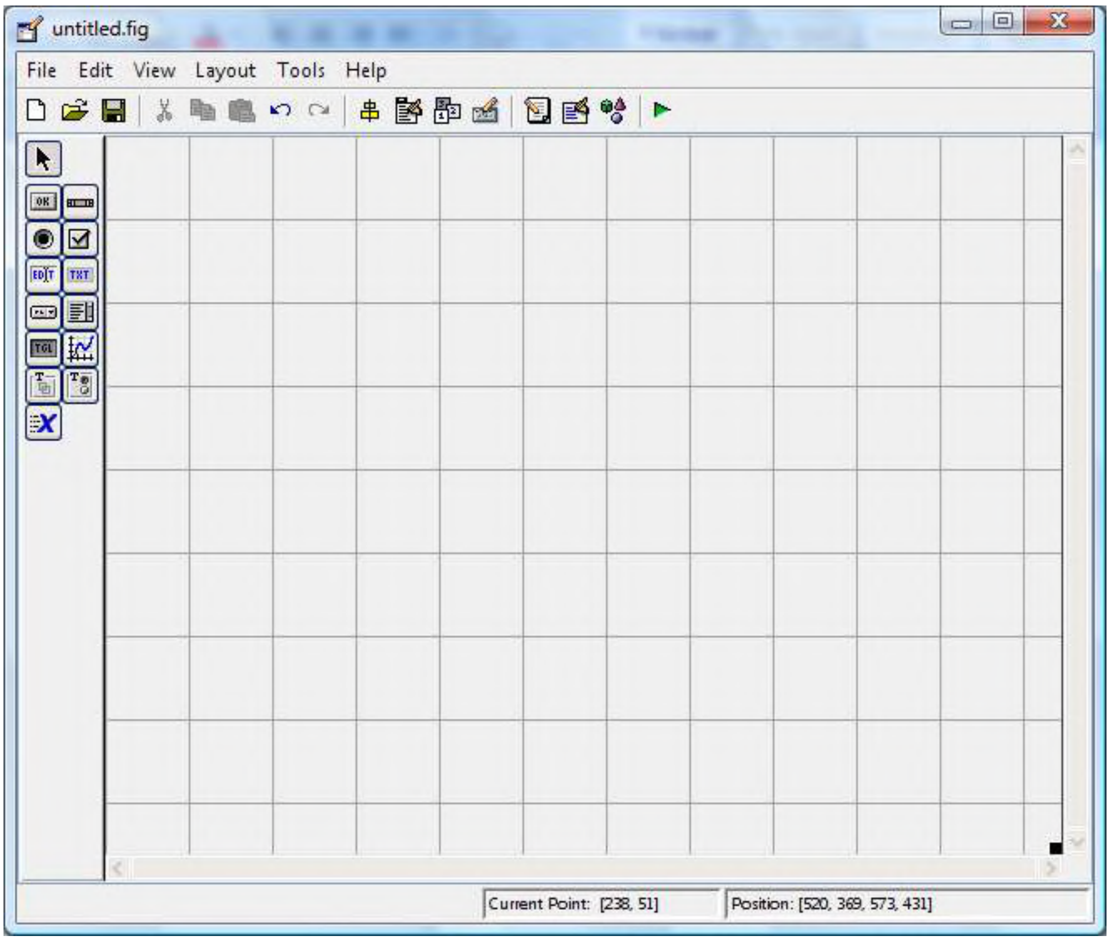


Fig 3.3 MATLAB window

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

3.4 Getting Started

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

3.4.1 Introduction - describes the components of the MATLAB system.

3.4.2 Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

3.4.3 Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

3.4.4 Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

3.4.5 Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

3.5 DEVELOPMENT ENVIRONMENT

3.5.1Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

3.5.2 Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.on a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

3.5.3 Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

3.5.4 MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB. The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

3.5.5 Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type helpbrowser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Content tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorite's tab - View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to run MATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

3.6 MANIPULATING MATRICES

3.6.1 **Entering Matrices**

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Dürer's matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

MATLAB displays the matrix you just entered.

```
A =

    16     3     2    13

     5    10    11     8

     9     6     7    12

     4    15    14     1
```

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

3.6.2 Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

3.6.3 Operators

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

Table 1: Arithmetic Operators

3.6.4 Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, for a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.14159265...
I	Imaginary unit, $\sqrt{-1}$
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

Table 2: MATLAB Functions

3.7 GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

3.7.1 Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

3.7.2 GUI Development Environment

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of its

children (ui-controls and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the

callbacks, which are defined as subfunctions. This M-file is referred to as the

application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

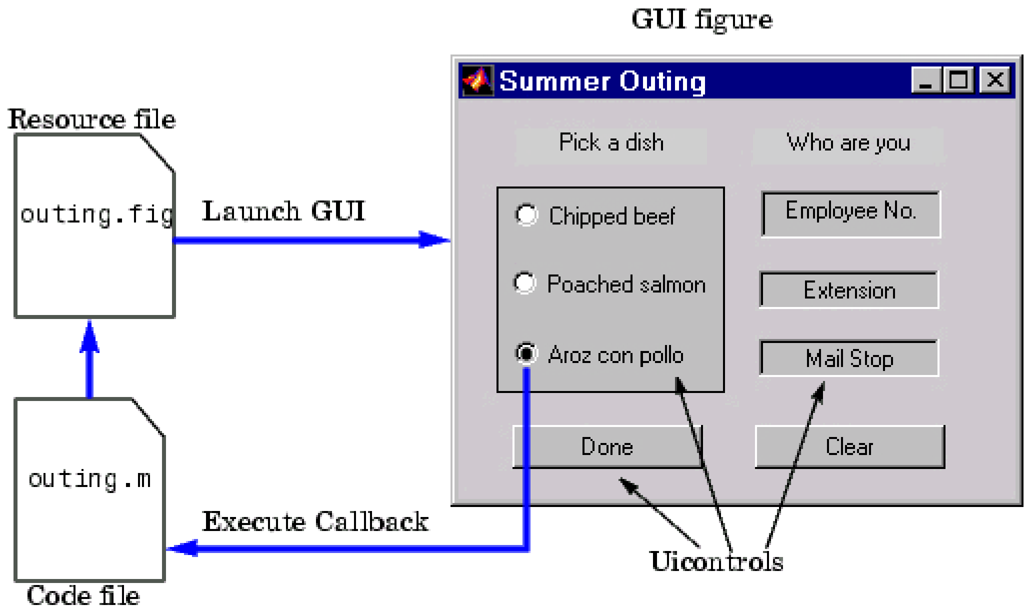


FIG 3.7.2 graphical user blocks

3.7.3 Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

3.7.4 Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques

used in the application M-file.

Application Examples - a collection of examples that illustrate techniques

which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

3.7.5 User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- List boxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)

button_state = get(h,'Value');

if button_state == get(h,'Max')

   % toggle button is pressed

elseif button_state == get(h,'Min')

   % toggle button is not pressed

end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a true-color image. For example, the array a defines 16-by-128 true-color image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);

a(:,:,2) = rand(16,128);

a(:,:,3) = rand(16,128);

set(h,'CData',a)
```

Radio Buttons

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)

set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)

off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];

mutual_exclude(off)

% Continue with callback

 .

 .

 .
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)
if (get(h,'Value') == get(h,'Max'))
   % then checkbox is checked-take approriate action
else
   % checkbox is not checked-take approriate action
end
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)
user_string = get(h,'string');
% proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)

user_entry = str2double(get(h,'string'));

if isnan(user_entry)

   errordlg('You must enter a numeric value','Bad Input','modal')

end

% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menu-bar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menu-bar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text

Static text controls display lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Non-integer values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

Listboxes differentiate between single and double clicks on an item and set the figure SelectionType property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure SelectionType property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item. This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

switch val

case 1

% The user selected the first item

case 2

% The user selected the second item

% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

string_list = get(h,'String');

selected_string = string_list{val}; % convert from cell array to string

% etc.
```

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is

grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes ButtonDownFcn property to define the callback.

3.7.6 Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

```
axes(handles.axes1)
```

makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axes. See GUI with Multiple Axes for an example that uses two axes.

Figure

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control.

CHAPTER 4

PROJECT IMPLEMENTATION

4.1 Implementation stages

4.1.1 Extraction of low high pixels:

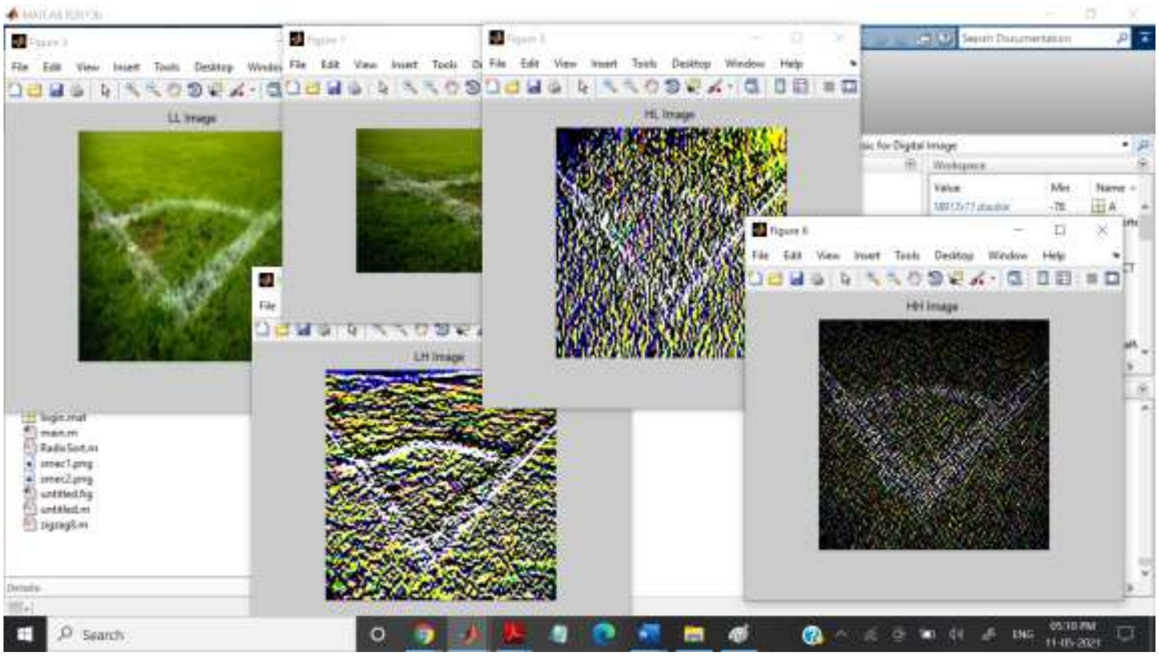


Fig: 4.1 Low high pixel extraction

4.2 Results

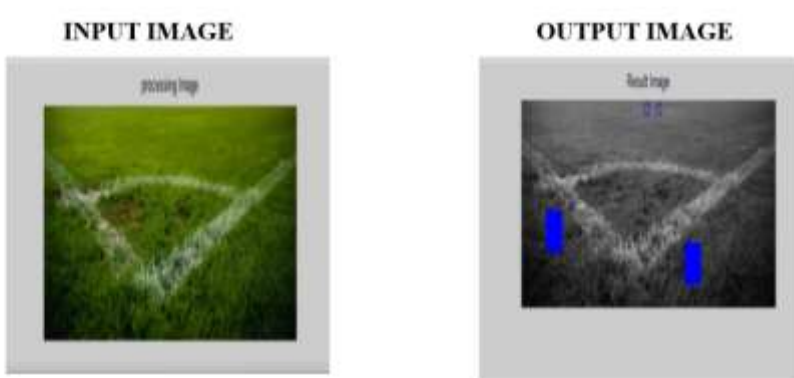


Fig 4.2 Simulation results

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

5.1 CONCLUSION

In this work, a CMFD technique consisting of oriented FAST and rotated BRIEF (ORB) as the feature extraction method and 2NN with HAC as the feature matching method is proposed. The ORB parameters, namely the number of features to retain and patch size are optimized using PSO. The optimization is essential in obtaining a balance between performance and runtime. Evaluation of the proposed CMFD technique is performed on images which suffers from various geometrical attacks. An overall accuracy rate of 84.33% and 82.79% was obtained for evaluation carried out with images from the MICCF600 and MICC-F2000 database. When evaluation is carried out on tampered images with different geometric attacks, the proposed CMFD technique performs accurately for tampered images with object translation, different degree of rotation and enlargement.

5.2 FUTURE ENHANCEMENT

Digital image forgery is a growing field of research. Although limitations are there in the existing methods, still it promises the improvement in detection methods.

The major concern of this field is that the method should achieve better performance than existing methods in terms of true positive rate and false alarms.

The other challenge is the robustness of existing methods to various malicious operations. Every method should be designed only after consulting with the image forensic experts and after receiving the feedback from them to implement the changes. Another future scope is that till date there is no unified algorithm for image forgery detection which can identify every type of forgery.

5.3 REFERENCES

[1] H. Farid, “Exposing digital forgeries from JPEG ghosts,” IEEE Trans. Inf. Forensics Secur., vol. 4, no. 1, pp. 154–160, 2009. [2] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, “A SIFT-based forensic method for copymove attack detection and transformation recovery,” IEEE Trans. Inf. Forensics Secur., vol. 6, no. 3 Part 2, pp. 1099–1110, 2011. [3] J. Fridrich, D. Soukal, and J. Lukáš, “Detection of Copy-Move Forgery in Digital Images,” Int. J., vol. 3, no. 2, pp. 652–663, 2003. [4] A. C. Popescu and H. Farid, “Exposing digital forgeries by detecting duplicated image regions,” Dept. Comput. Sci., Dartmouth Coll. Tech. Rep. TR2004-515, no. 2000, pp. 1–11, 2004. [5] H. He, X. Huang, and K. Jun, “Exposing copy-move forgeries based on a dimension-reduced SIFT method,” Information Technology Journal, vol. 12, no. 14. pp. 2975–2979, 2013. [6] Z. Mohamadian and A. A. Pouyan, “Detection of duplication forgery in digital images in uniform and non-uniform regions,” Proc. - UKSim 15th Int. Conf. Comput. Model. Simulation, UKSim 2013, vol. 1, pp. 455–460, 2013. [7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-Up Robust Features (SURF),” Comput. Vis. Image Underst., vol. 110, no. 3, pp. 346–359, 2008. [8] B. Xu, J. Wang, G. Liu, and Y. Dai, “Image copy-move forgery detection based on SURF,” Proc. - 2010 2nd Int. Conf. Multimed. Inf. Netw. Secur. MINES 2010, pp. 889–892, 2010. [9] P. Mishra, N. Mishra, S. Sharma, and R. Patel, “Based on SURF and HAC,” vol. 2013, no. July 2008, 2013. [10] M. F. Hashmi, V. Anand, and A. G. Keskar, “A copy move image forgery detection based on speeded up robust feature transform and wavelet transforms,” Proc. - 5th IEEE Int. Conf. Comput. Commun. Technol. ICCCT 2014, pp. 147–152, 2015. [11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” Proc. IEEE Int. Conf. Comput. Vis., pp. 2564–2571, 2011. [12] P. L. Rosin, “Measuring Corner Properties,” Comput. Vis. Image Underst., vol. 73, no. 2, pp. 291–307, 1999. [13] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: Binary robust independent elementary features,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6314 LNCS, no. PART 4, pp. 778–792, 2010. [14] R. Kaur and A. Kaur, “Copy-Move Forgery Detection Using ORB and SIFT Detector,” vol. 4, no. 4, pp. 804– 813, 2016.

PROGRAM CODE

MATLAB Code:

```
clc
clear all;
close all;
[filename, pathname] = uigetfile({'*.bmp';'*.jpg'},'pick an image');

if isequal(filename,0) || isequal(pathname,0)
   warndlg('File is not selected');
else
input_image1=imread(filename);
input_image=imresize(input_image1,[512 512]);

figure,imshow(input_image);title('Original image');
end
%% block processing

fun = @(block_struct) imresize(block_struct.data,0.15);
block_img = blockproc(input_image,[2 2],fun);
block_img=imresize(block_img,[512 512]);

figure;
imshow(block_img);title('block procee image')
blocksize=8;
overlapp=1;
Nd=16;
Th=0.9999;
s_threshold=1;

%%%%%%%%%%%%%%%% Apply Discrete wavelet transform
%%%%%%%%%%%%%%%

[cA_img,cH,cV,cD]=dwt2(block_img ,'haar');
figure,imshow(mat2gray(cA_img));title('LL Image');
figure,imshow(cH);title('LH Image');
figure,imshow(cV);title('HL Image');
figure,imshow(cD);title('HH Image');
[r,c,n] = size(cA_img);
if n > 1
   im=rgb2gray(cA_img);
else
   input_image1=cA_img; % It's already gray.
end
```

```
tic,
input_image=1;
figure,imshow(mat2gray(im)), title(' DWT  Image')
for j=1:overlapp:(c-blocksize)+1
   for i=1:overlapp:(r-blocksize)+1

      sondos(input_image).block=input_image1(i:i+blocksize-1,j:j+blocksize-1);
      sondos(input_image).position=[i j];
      sondos(input_image).index=input_image;
      input_image=input_image+1;
   end
end

%% dct trnasformation for gray level checking
sz=size(sondos,2);
DC=[];
QZ=4;
for input_image=1:sz
   FDCT=dct2(sondos(input_image).block);
   FDCT=round(FDCT./QZ);
   DC(input_image,1)=FDCT(1,1);
end

%% applying k-means algoritham
G=[];
numclass=4;
[centers,mincenter,mindist,q2,quality] = FastKmean(DC,numclass,1);
for n=1:numclass
   ind= find(mincenter==n);
   for i=1:length(ind)

      G(n,i)= ind(i);

   end
end

%% detect copy and move part
figure,imshow(mat2gray(input_image1)),title('Result Image');
Th=0.99;
TotalMatch=0;
%parfor nG=1:size(G,1)
   for nG=1:size(G,1)
   emp=find( G(nG,:)==0);
   if isempty(emp)==0
      if emp(1)==1
```

```
        time=toc
        disp('time')
        disp(time)
      end
      total_mtx=[];
      for input_image=1:emp(1)-1
        [f,vec]=featureExtraction(sondos(G(nG,input_image)).block);
        total_mtx(input_image,1:9)=f;
        total_mtx(input_image,10)=sondos(G(nG,input_image)).position(1);
        total_mtx(input_image,11)=sondos(G(nG,input_image)).position(2);
      end
    else
      total_mtx=[];
      for input_image=1:size(G,2)
        [f,vec]=featureExtraction(sondos(G(nG,input_image)).block);
        total_mtx(input_image,1:9)=f;
        total_mtx(input_image,10)=sondos(G(nG,input_image)).position(1);
        total_mtx(input_image,11)=sondos(G(nG,input_image)).position(2);
      end
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Asorted=RadixSort(total_mtx,9);

    for i=1:size(Asorted,1)-1

      similar=[];
      for l=1:9

        s=abs(Asorted(i+1,l)-Asorted(i,l));

        if s<s_threshold
          %similar%
          similar(l)=1;

        else
          %not similar%
          similar(l)=0;
        end


      end
      if isempty(find(similar==0))

        x1=Asorted(i,10);
```

```
            x2=Asorted(i+1,10);
            y1=Asorted(i,11);
            y2=Asorted(i+1,11);
            D= sqrt((x1-x2)^2+(y1-y2)^2);
            if D>Nd

                TotalMatch=TotalMatch+1;
                rectangle('Position',[y1,x1,blocksize,blocksize],'Edgecolor','b');
                rectangle('Position',[y2,x2,blocksize,blocksize],'Edgecolor','b');
%                   line([y1,y2],[x1,x2],'Color','r','LineWidth',1)
                %end


            end
        end
    end

  toc;
end

% end
```

A

PROJECT REPORT

On

PLANT SOIL MOISTURE & pH SENSING ALRAM

Submitted by

1)Mr.A.Naveen Kumar (18K85A0433)
2) Mr.G.Durga Mahesh (18K85A0428)

3) Mr.A.Ramesh Kumar (17K81A04G8)

in partial fulfillment for the award of the degreeof

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

K.Nishakar

(Associate Professor)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

An Autonomous Institute

Dhulapally, Secunderabad – 500 100

JUNE 2021

BONAFIDE CERTIFICATE

This is to certify that the project entitled **Plant soil moisture and pH sensing alarm**, is being submitted by 1**. Mr. A.Naveen Kumar (18K85A0433), 2. Mr.G. Durga Mahesh (18K85A0428), 3. Mr.A. Ramesh Kumar (17K81A04G8)** in partial fulfilment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Internal Guide	Head of the Department
K. Nishakar	B. Hari Krishna
Department of ECE	Department of ECE

Internal Examiner	External Examiner

Place:

Date:

TUESDAY, 15

JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **DURGA MAHESH** WITH ROLL NO.**18K85A0428, NAVEEN KUMAR** WITH ROLL NO.**18K85A0433**, **RAMESH KUMAR** WITH ROLL NO.**17K81A04G8,** OF B. TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**PLANT SOIL MOISTURE AND PH SENSING ALARM**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT

DIRECTOR

TRAININGS & PLACEMENTS LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014 Email: contact@lasyainfotech.com, ov@lasyainfotech.com
Website: www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of **Bachelor of Technology** in Department of Electronics and communication engineering', session: <2017 – 2021>, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled **PLANT SOIL AND pH SENSING ALRAM** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

A. Naveen Kumar (18K85A0433)
G. Durga Mahesh (18K85A0428)
A. Ramesh kumar (17K81A04G8)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

We extended our deep sense of gratitude to Principal**, Dr. P. SANTOSH KUMAR PATRA**, St. Martin's Engineering College, Dhulapally, for permitting us to undertake this project.

We are also thankful to **Dr.B.HARI KRISHNA**, Head of the Department, ECE , St. Martin's Engineering College, Dhulapally, for his support and guidance throughout our project and as well as our project coordinator **Dr.A.ANAND** professor in ECE, for his valuable support.

We would like to express our sincere gratitude and indebtedness to our project supervisor **Mr,K.NISHAKAR** Associate professor in ECE, St. Martin's Engineering College, Dhulapally, for his support and guidance throughout our project.

Finally, we express thanks to all those who have helped us successfully to completing this project. Furthermore, we would like to thank our family and friends for their moral support and encouragement.

We express thanks to all those who have helped us in successfully completing the project.

A. Naveen Kumar (18K85A0433)

G. Durga Mahesh (18K85A0428)

A. Ramesh kumar (17K81A04G8)

ABSTRACT

Plants need water as well as good soil (pH rich) to ensure proper plant growth. It becomes difficult to monitor these things manually each time to ensure proper growth and any ignorance may lead to bad plant health or plant decay/death. Here we propose an Arduino microcontroller base system that makes use of soil moisture sensor along with pH value sensor to constantly check for these values. The system microcontroller ensures the plant gets proper moisture and pH by continuously monitoring for it. It also displays this on a display screen for user to monitor.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1.1. Project Overview:

India is a developing nation with a very large population. Due to increasing population, the basic need such as food and water is increasing day by day. Thus, there is a need of saving these resources and utilize them in an efficient manner. Since water is one of the most important elements in our daily life, thus we must use efficient ways to utilize water and save it for future generations. One of method is efficient irrigation management practices for fields. Irrigation water management practices could greatly benefit by the knowledge of moisture in the soil. To determine the soil moisture, we have designed and developed a nickel probes based soil moisture sensor and a response monitoring system. By knowing the moisture value, we can estimate when to water and how much to water the fields so that there is no over- watering or wilting of crops. These practices will increase crop yield, improve quality of crops, conserve water resources, save energy, and decrease fertilizer supplies.

1.2 Objective of The Project:

The objective of this project is to provide a combination of manual supervision and partial automation. It is similar to manual set up in most respects but it reduces the labour involved in terms of irrigation design is simple. Easy to install microcontroller-based circuit to monitor and record the values of temperature and soil moisture.

CHAPTER-2

INTRODUCTION ABOUT EMBEDDED SYSTEMS

2.1 Introduction:

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.
The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes.

The applications software on such processors is sometimes referred to as firmware.
The simplest devices consist of a single microprocessor (often called a "chip"), which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine.

As the embedded system is the combination of both software and hardware.

2.2 Block Diagram of Embedded System:

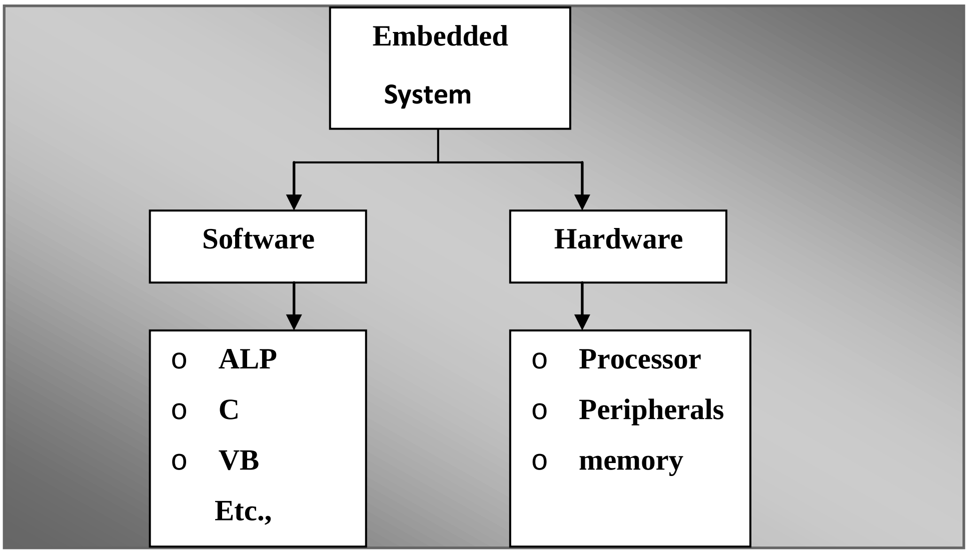


Fig.2.1: Block diagram of embedded system

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory

2.3 Applications of Embedded Systems:

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

2.4 Micro Processor (μp):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Processors are classified into four types like:

- Micro Processor (µp)
- Micro controller (µc)
- Digital Signal Processo1r (DSP)
- Application Specific Integrated Circuits (ASIC)

2.5 Three Basic Elements of a Microprocessor:

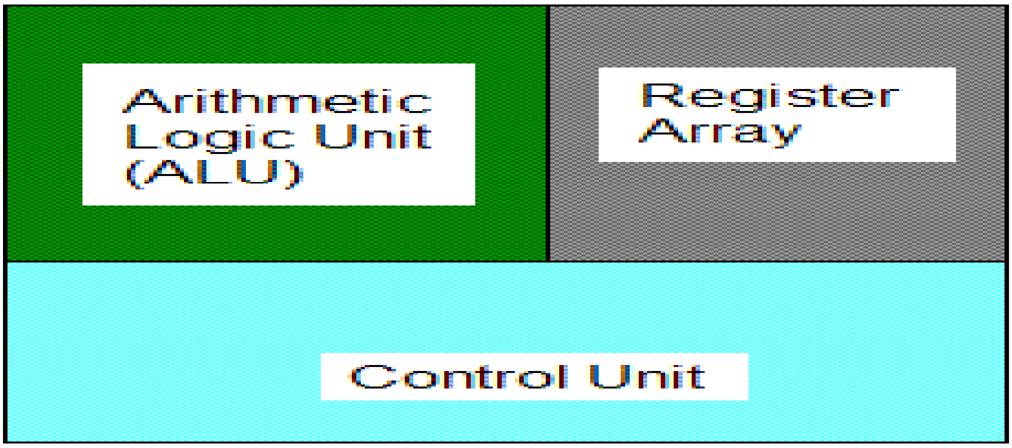


Fig.2.2: Three basic elements of a microprocessor

2.6 Harvard Architecture:

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

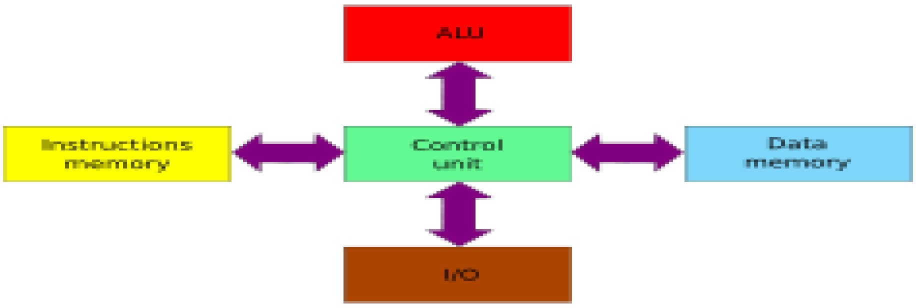


Fig.2.3: Harvard Architecture

2.6.1 Uses of the Harvard architecture*:*

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

- Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behaviour must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.
- Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

2.6.2 Von-Neumann Architecture

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. They cannot be performed at the same time. The von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A stored-program digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

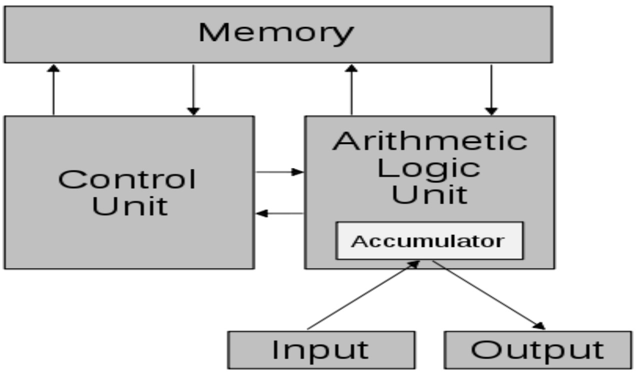


Fig.2.3.1: Schematic of the Von-Neumann Architecture.

CHAPTER-3

MICRO CONTROLLER UNIT

3.1 ATMEGA328

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The ATmega8 microcontroller contains 32 general purpose working registers. As shown in the below figure these registers are directly connected to ALU. Two registers can carry one single instruction consequently in one clock cycle.

Specifications:

Table 3.1: Atmega328 specifications

Microcontroller	ATmega328P – 8-bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 Ma
DC Current on 3.3V Pin	50 Ma
Flash Memory	32 KB (0.5 KB is used for Boot loader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

Arduino:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

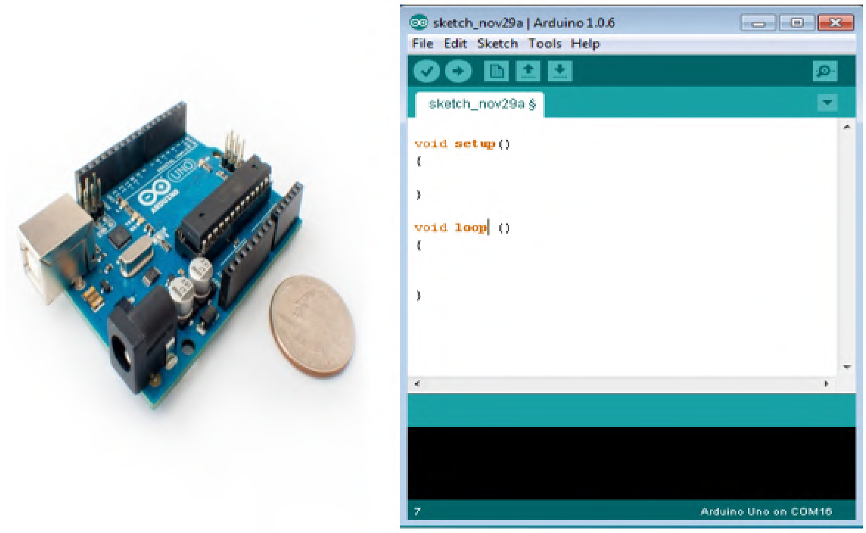


Fig 3.1 Arduino Uno

3.2 Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Table 3.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16 U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16 U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini-3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fios	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Lilypad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible

Table 3.2 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 3.3 Arduino boards based on ATMEGA2560 microcontroller

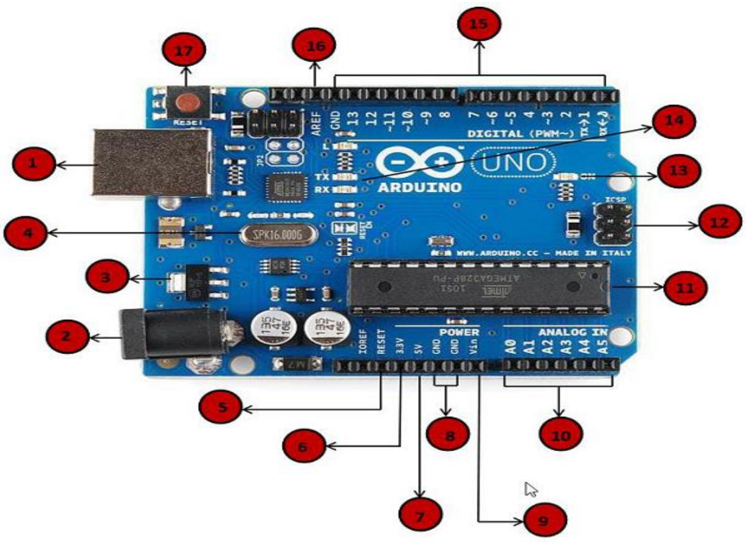
Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 3.4 Arduino boards based on AT91SAM3X8E microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

3.2.1 Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.



1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).

3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volts. • GND (8) (Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different

	from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labelled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

3.3 Arduino Family

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project , check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

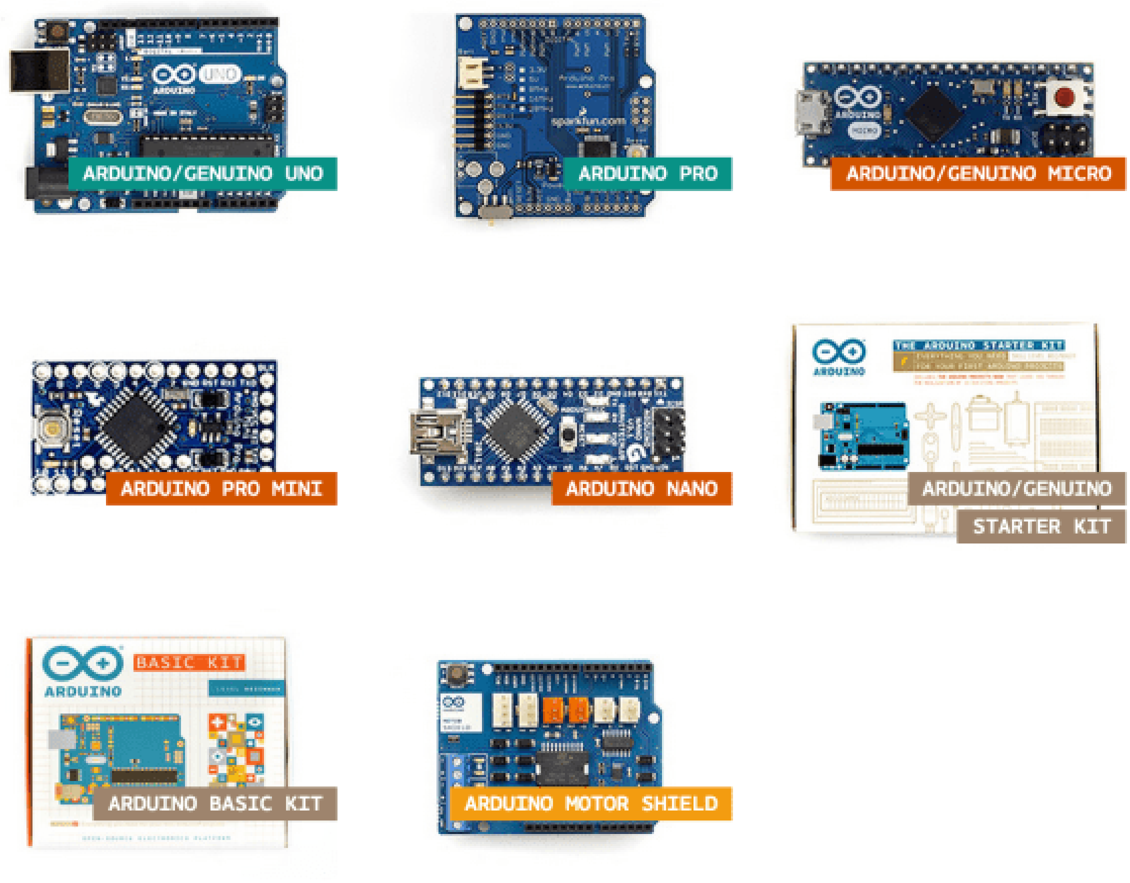


Fig.3.3: Arduino Family

3.4 Shields

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

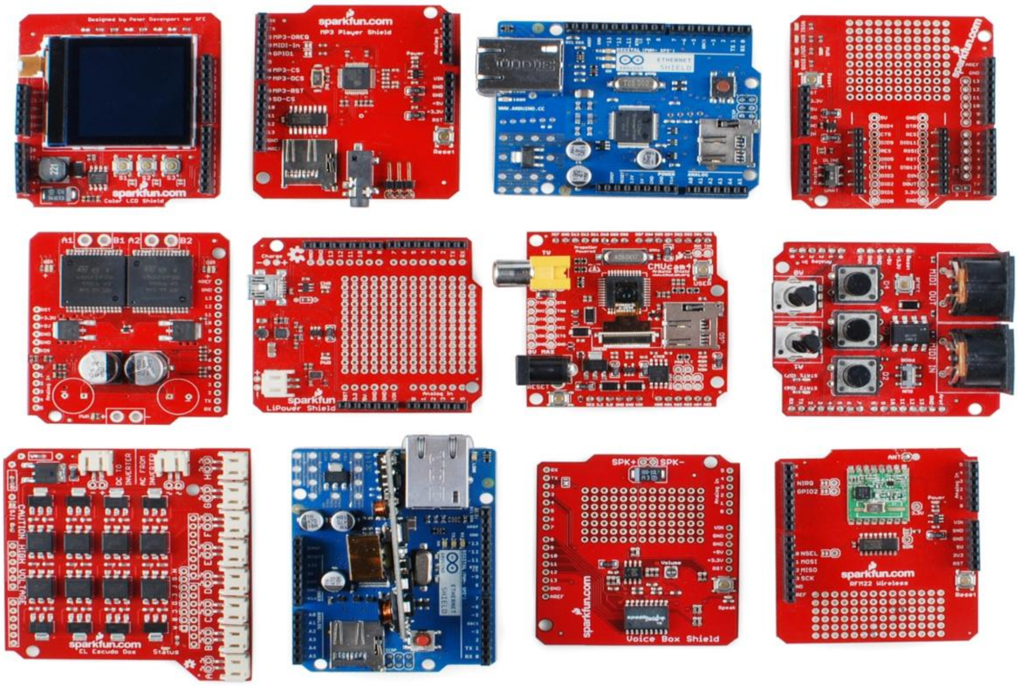


Fig.3.4: Arduino Shields

3.5 PIN DESCRIPTION OF ATMEGA328

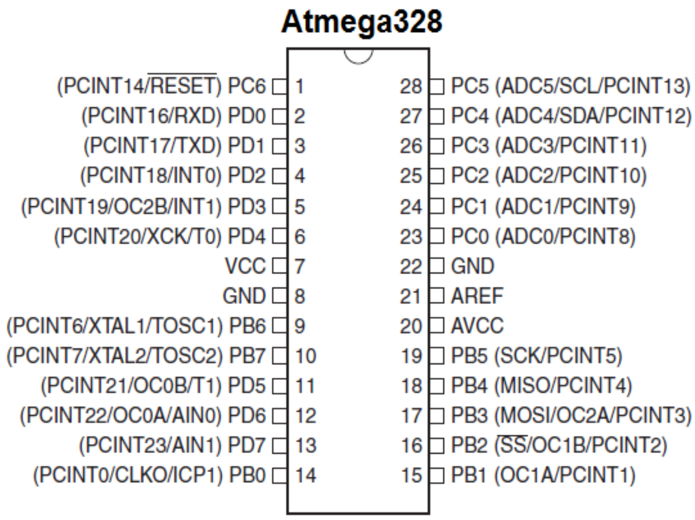


Fig.3.5: Pin description of ATMEGA328

3.6 Advantages Of Arduino:

- It is cheap
- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
- For beginners, it is very simple to use.

3.7 Applications:

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

CHAPTER-4

POWER SUPPLY UNIT

4.1 Introduction:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as "Regulated D.C Power Supply".

4.1.1 Block Diagram of Power Supply:

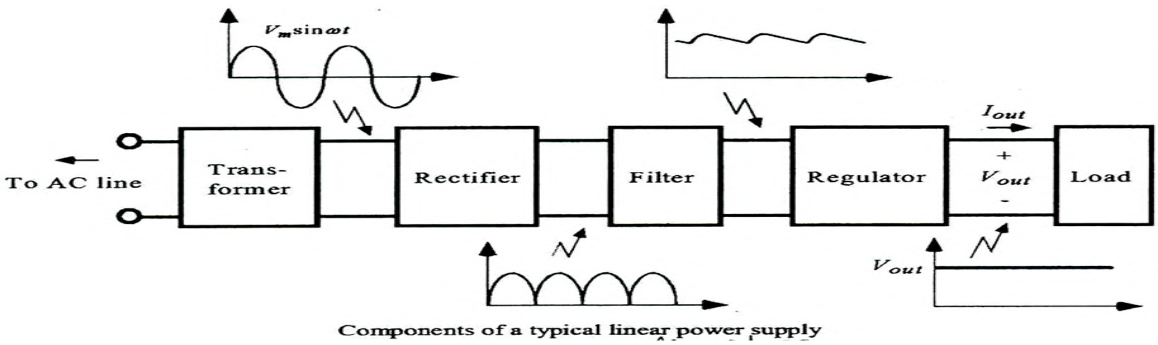


Fig.4.1.1: Block Diagram of Power Supply

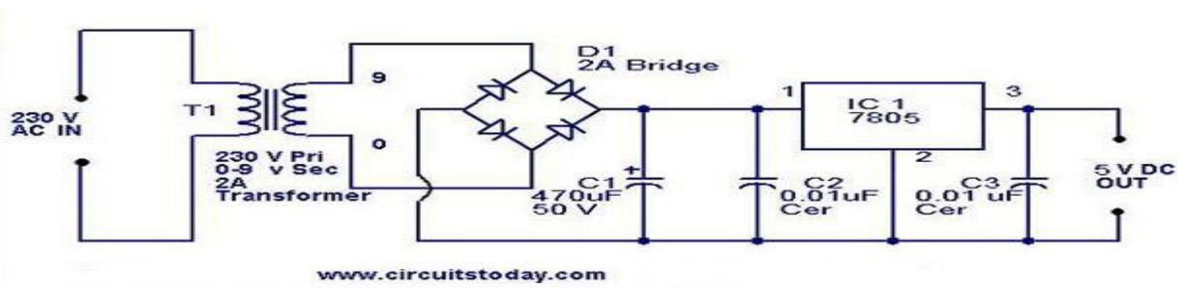


Fig.4.1.1(A): Schematic Diagram of Power Supply

4.1.2 Description of Power Supply:

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a a different form of energy – such as solar, mechanical, or chemical - into electrical energy.

A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered. For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

4.2 Transformer:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

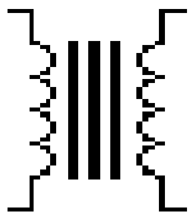


Fig.4.2: Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

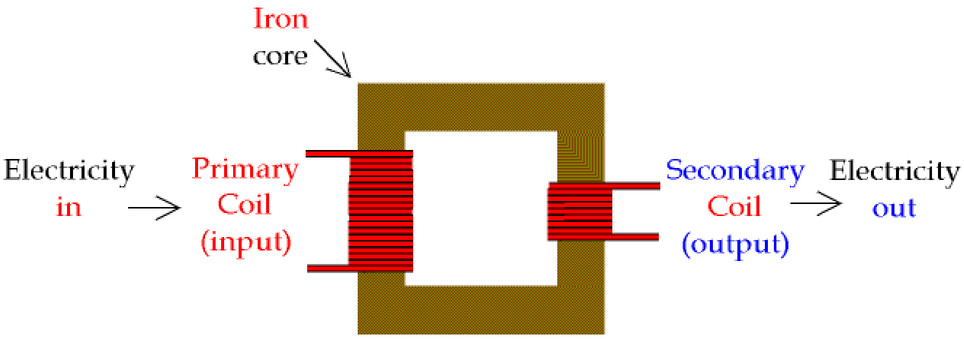


Fig.4.2(A): Transformer

4.2.1 Basic Principle of Transformer:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

From Faraday's Law

$$\frac{V_S}{V_P} = \frac{N_S}{N_P}$$

For ideal transformer

The voltage ratio is equal to the turns ratio, and power in equals power out.

From conservation of energy

$$P_P = V_P I_P = V_S I_S = P_S$$

4.2.2 Working of Transformer:

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

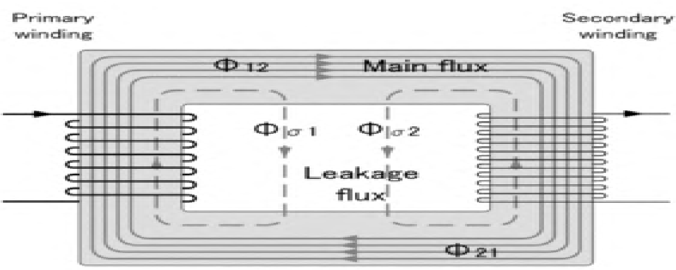


Fig.4.2.2: Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f. The iron core is laminated to prevent 'eddy currents' flowing in the core.

These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

4.2.3 Classification of Transformer:

- Step-Up Transformer
- Step-Down Transformer

Step-Down Transformer:

Step down transformers is designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step-down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn's ratio of the two sets of windings determines the amount of voltage transformation.

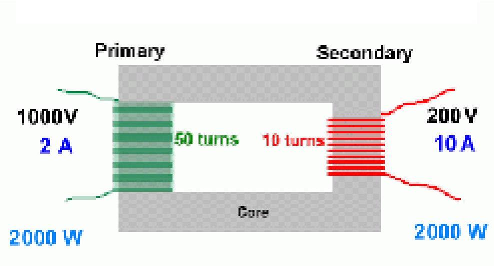


Fig.4.2.2: Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts, then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single phase steps down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase steps up or step-down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

4.2.2 Step-Up Transformer:

A step-up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step-up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step-up transformer is needed to use a 220v product in a country with a 110v supply. A step-up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So, a step-up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step-up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, Nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

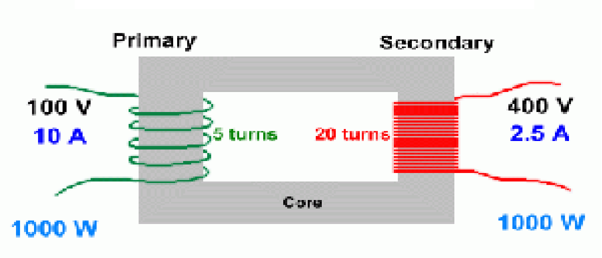


Fig.4.2.2(a): Step-Up Transformer

4.3 Diodes:

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

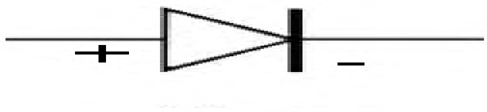


Fig.4.3: Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

4.4 Rectifier:

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

The Half-Wave Rectifier:

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure.

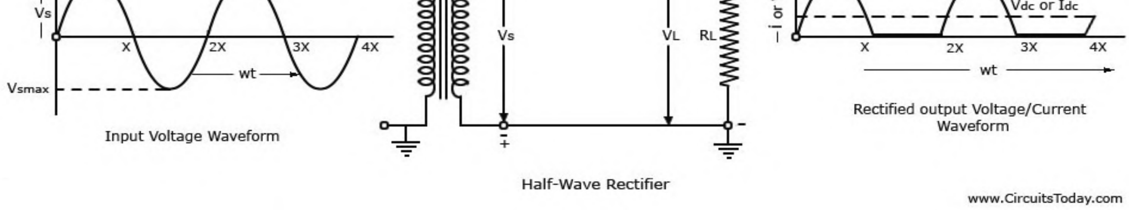


Fig.4.4(a): Half Wave Rectifier

Figure 2 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

The Full-wave Rectifier:

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC.

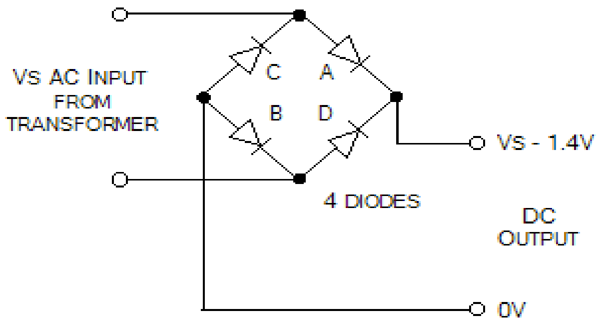


Fig.4.4(b): Full-Wave Rectifier

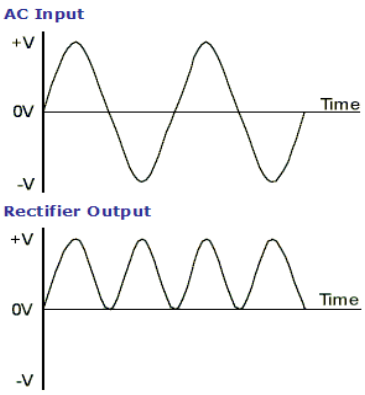


Fig.4.4(c): Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased. While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12AC in, you will 10.6V DC out.

4.5 Capacitor Filter:

The capacitor-input filter, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

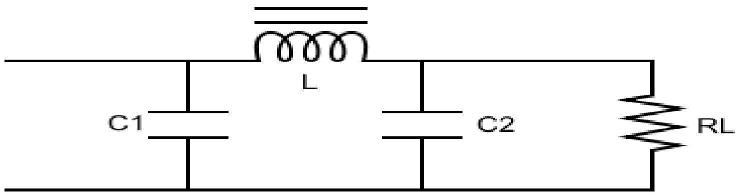


Fig.4.5: Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result, the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result, the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result, only the DC component appears across the load RL.

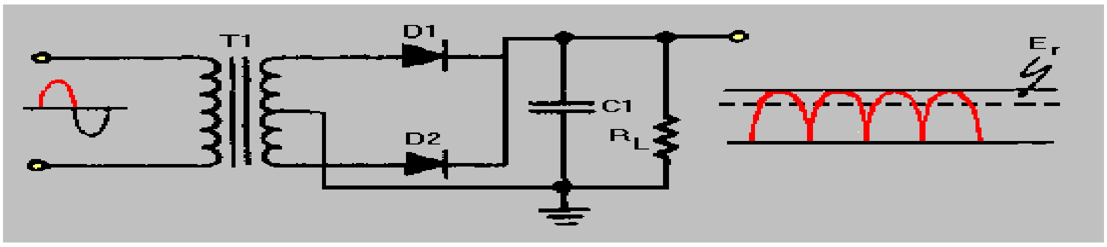


Fig.4.5(A): Centered Tapped Full-Wave Rectifier with a Capacitor Filter

4.6 Voltage Regulator:

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulators are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:’78’ indicate the positive series and ‘xx’ indicates the voltage rating. Suppose 7805 produces the maximum 5V.’05’indicates the regulator output is 5V.

79xx:’78’ indicate the negative series and ‘xx’ indicates the voltage rating. Suppose 7905 produces the maximum -5V.’05’indicates the regulator output is -5V.

These regulators consist the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

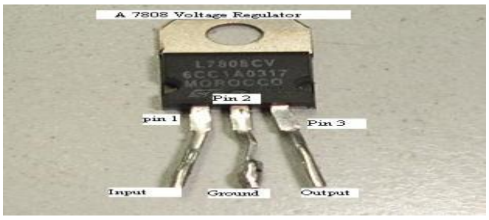


Fig.4.6: Regulator

CHAPTER-5

PROJECT DISCRIPTION

5.1 Hardware Requirement:

- Arduino uno
- LCD
- Power Supply
- Soil Moisture Sensor
- PH Sensor
- ESP-01
- Relay
- AC Pump

Software Components:

- Arduino IDE
- Proteous

5.2 Block Diagram of Project:

Block diagram

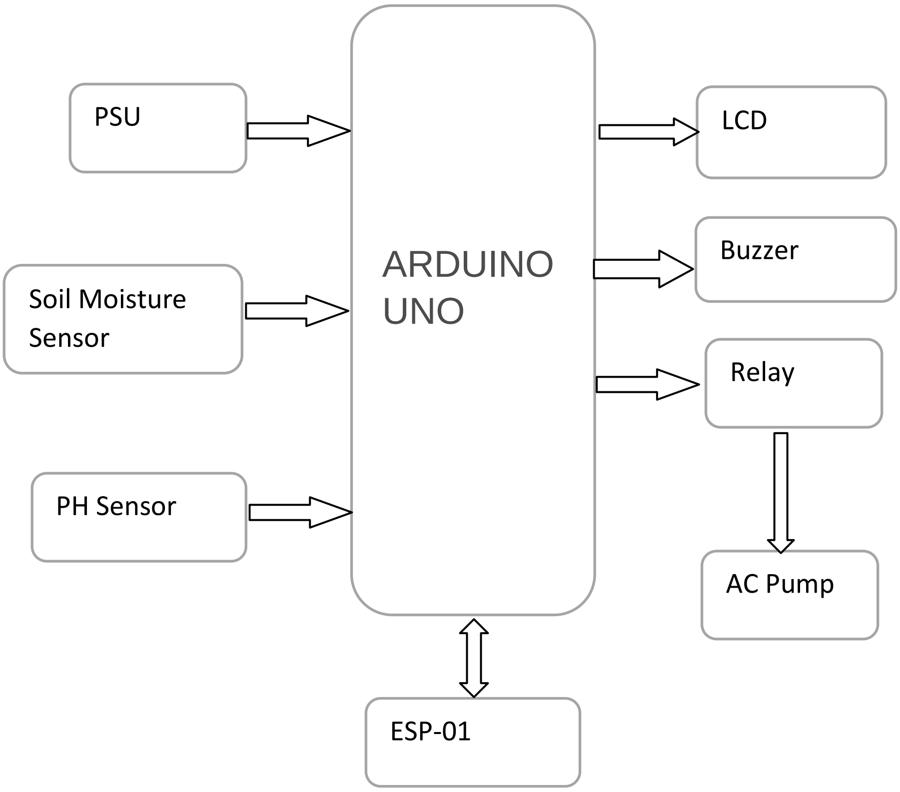


Fig.5.2: Block diagram of the project

Wi-Fi Module:

The ESP8266 is a low-cost Wi-Fi module that can be integrated easily into IoT devices. We've featured several projects using this module, such as **How To Make Smart Home Electronics: A Smart Mailbox** and **How To Read Your Arduino's Mind: Building A Childproof Lock**. This tutorial will walk you through setting up ESP8266 Wi-Fi module which can be used with Arduino. The ESP8266 comes in many models with different functionalities. We'll be focusing on the ESP8266 ESP-01 module, the most common and basic one available.

What is ESP8266?

The ESP8266 is a small Wi-Fi module built around the ESP8266 chip that can connect your microcontroller to the internet wirelessly for a very small cost. It can be a great option for Internet of Things (IoT) projects, but can be difficult to work with for beginner hobbyists who do not have prior experience with the module. In this tutorial, we hope to show you how to interface the ESP8266 with an Arduino and perform some basic functions like connecting it to a Wi-Fi network.

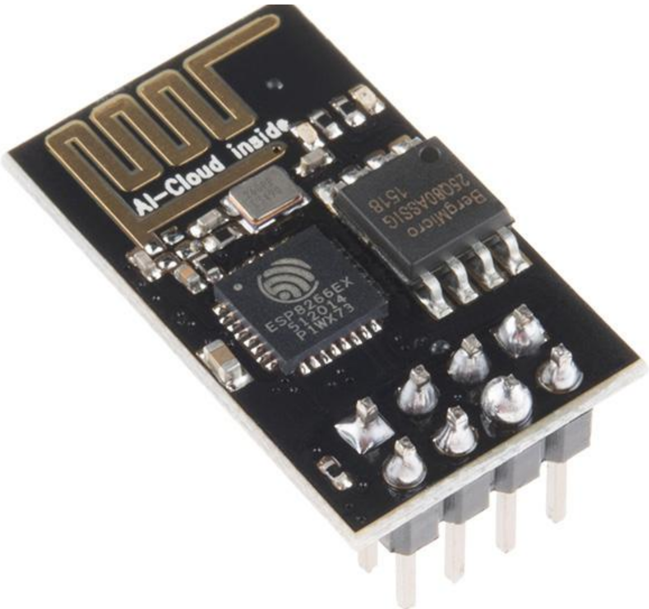


Figure 1. ESP8266 ESP-01 module / ©Sparkfun

ESP-01 Features – Sparkfun**:**

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 1MB Flash Memory
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms

- Standby power consumption of < 1.0mW (DTIM3)

The first feature to notice about the ESP8266 is its awkwardly spaced header pins. The module has 8 pins that serve different functions, but they are packed in a 4×2 arrangement that makes plugging the module into a breadboard impossible. This means that to prototype projects on a breadboard, you'll need male-female jumper wires to connect the pins on the ESP8266 to rows on the breadboard. If you'd like to make your prototyping more compact, you can also purchase breadboard breakouts for the ESP8266 such as this one. For prototyping, I chose to just use jumper wires.

The pinout for the ESP8266's pins is according to the following diagram:

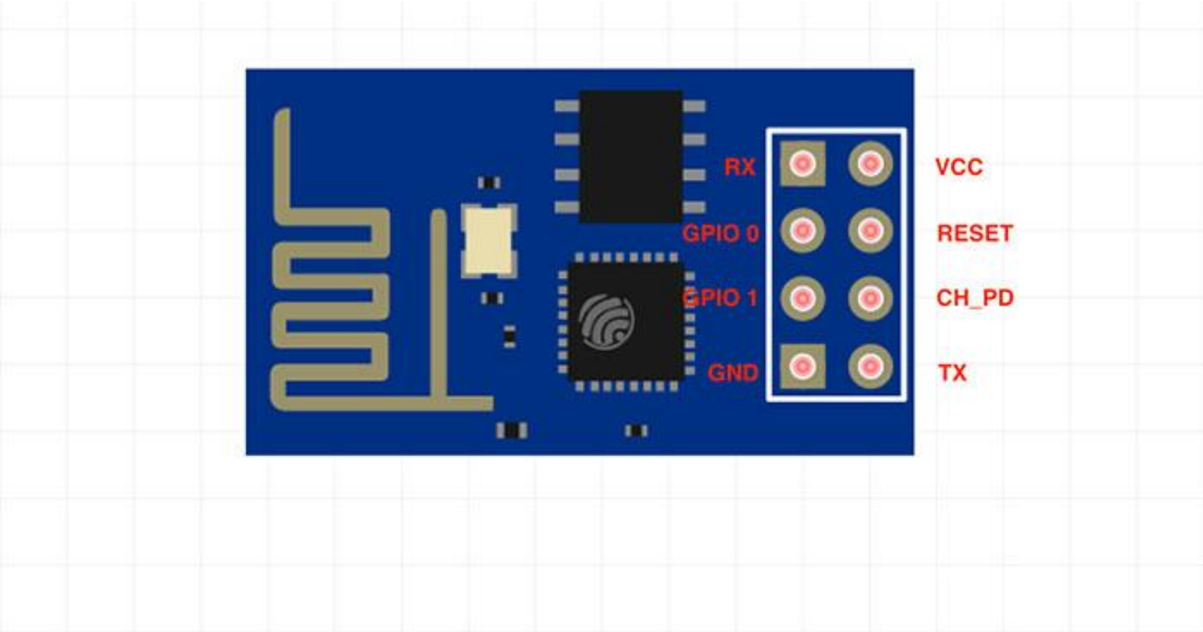


Figure 2. ESP8266 Pinout / ©Github.com/ydonnelly

RX: UART serial communication receive pin

GPIO 0: GPIO pin (unused in this project)

GPIO 1: GPIO pin (unused in this project)

GND: Connection to Ground

VCC: Connection to 3.3V Vcc (Vcc cannot exceed 3.3V!)

RESET: Reset pin (pull down to reset)

CH_PD: Chip enable and power down pin

TX: UART serial communication transmit pin

Note that the maximum voltage input for the ESP8266 is 3.3V. Any input voltage greater than 3.3V will damage the module! To program settings on the ESP8266, we'll first need to connect it to a serial terminal on a computer through which we can send it special commands. Settings that we'll have to program include, for example, the SSID and password for the Wi-

Fi network the module will be connected to. To connect the ESP8266 to a computer and configure its settings, we'll need a USB to serial adapter with 3.3V logic, along with a serial terminal program. Fortunately for us, we have the Arduino and the Arduino IDE's serial monitor! This means that we'll just have to connect the ESP8266 module to the Arduino and upload a custom sketch to the Arduino.

Use the following diagram to connect the ESP8266 module to the Arduino:

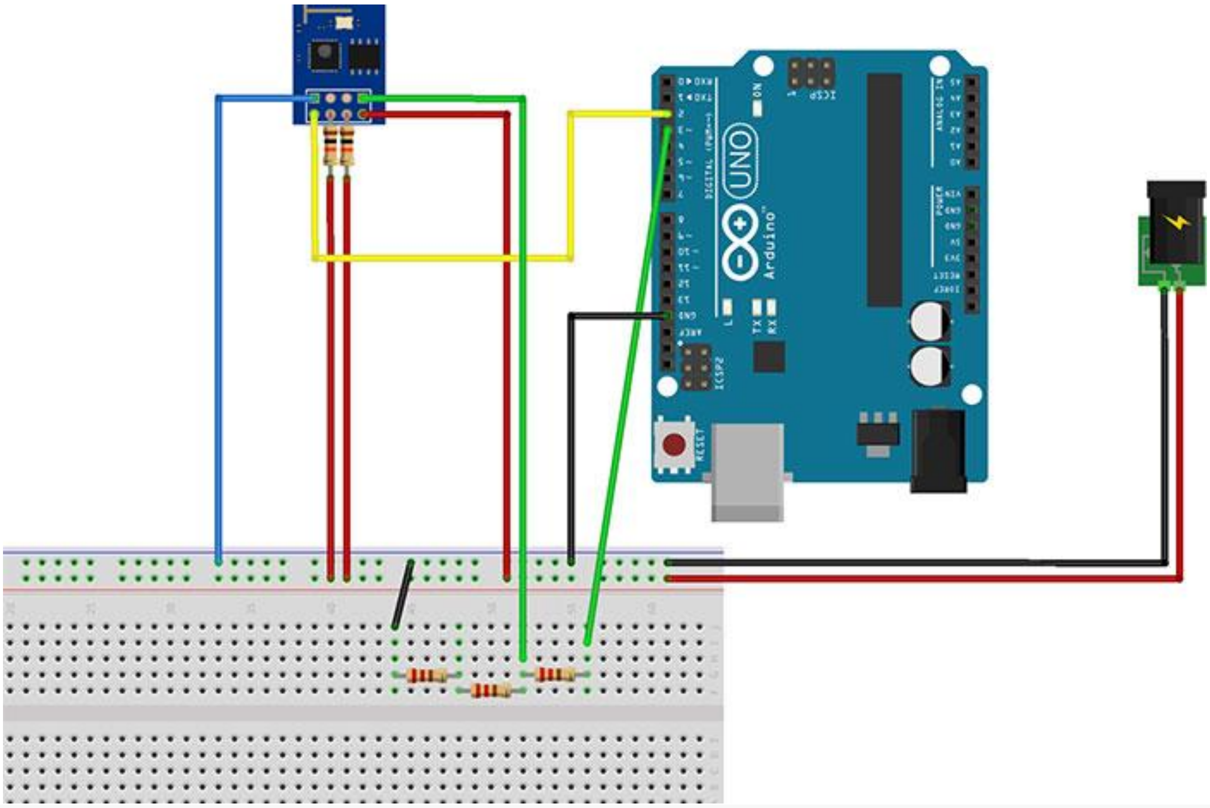


Figure 3. Connection between ESP8266 and Arduino Uno

You will need the following parts:

- Arduino Uno or similar module
- ESP8266 ESP-01 module
- 2 x 1kΩ resistors for CH_PD and RESET pull-up
- 3 x 220Ω resistors for serial line voltage divider
- 3.3V regulated power supply for ESP8266 (you cannot plug the power line into the Arduino's 3.3V out pin because the ESP8266 draws a lot of current!)
    - You can use something like this: https://www.sparkfun.com/products/114
- Breadboard and jumper wires

A couple of features of this circuit stand out immediately.

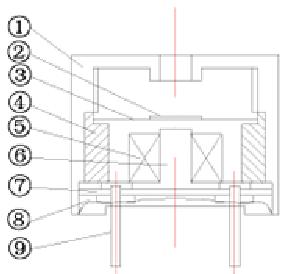
First, the ESP8266 module is powered by a 3.3V regulated power supply. As described earlier, the ESP8266 has a maximum voltage input rating of 3.3 volts. Using a power supply rated higher than this recommended 3.3 volts could fry the module! When connecting the ESP8266 to this power supply, you should not only connect the Vcc and Ground pins to the power supply lines, but also connect the power supply's ground line to the Arduino's ground pin. Since the Arduino will be powered by the USB connection to the laptop, creating a common ground essentially creates a common reference (you can think of it as a baseline) to compare voltages and thereby interpret digital high and low signals.

Second, the receive (RX) line for the ESP8266 module is connected to the output of a resistor voltage-divider circuit. We do this to shift the serial communication logic level (the highs and lows of the digital signals that make up the serial communications) from a logic high of 5 volts on the Arduino to a logic high of 3.3 volts on the ESP8266 module. Again, the ESP8266 is specified with 3.3V logic, so connecting the module's receive line directly to the Arduino's transmit line could damage the device. We do not need to shift levels on the ESP8266's transmit line because the module's 3.3V logic high is a high enough voltage to also register as a logic high on the Arduino.

Finally, there are two pull-up resistors on the ESP8266's CH_PD and RESET pins. These are specified by the module's creators to ensure that the device functions properly while it is being used. To reset the ESP8266 or disable it, these pins must be pulled down, but since we do not want those processes to occur, we will instead pull these pins high through some 10kΩ pull-up resistors.

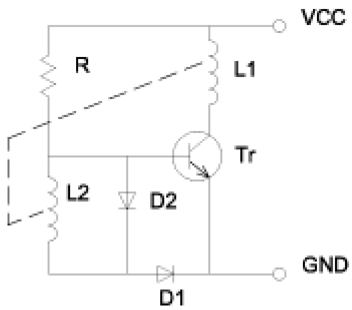
Buzzer:

1. Magnetic Transducer



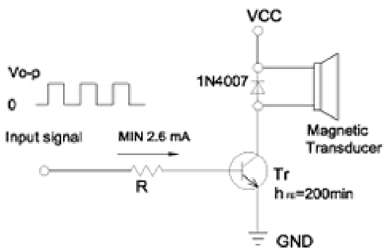
Magnetic transducers contain a magnetic circuit consisting of a iron core with a wound coil and a yoke plate, a permanent magnet and a vibrating diaphragm with a movable iron piece. The diaphragm is slightly pulled towards the top of the core by the magnet's magnetic field.

When a positive AC signal is applied, the current flowing through the excitation coil produces a fluctuating magnetic field, which causes the diaphragm to vibrate up and down, thus vibrating air. Resonance amplifies vibration through resonator consisting of sound hole(s) and cavity and produces a loud sound.



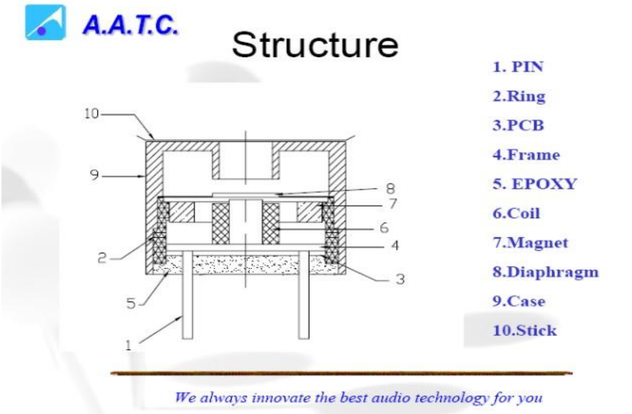
2. Magnetic Buzzer (Sounder)

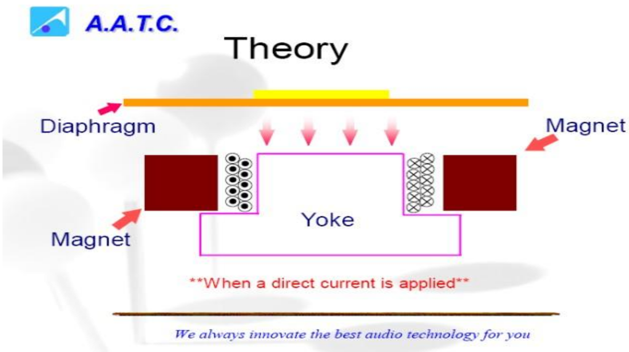
Buzzers like the TMB-series are magnetic audible signal devices with built-in oscillating circuits. The construction combines an oscillation circuit unit with a detection coil, a drive coil and a magnetic transducer. Transistors, resistors, diodes and other small devices act as circuit devices for driving sound generators. With the application of voltage, current flows to the drive coil on primary side and to the detection coil on the secondary side. The amplification circuit, including the transistor and the feedback circuit, causes vibration. The oscillation current excites the coil and the unit generates an AC magnetic field corresponding to an oscillation frequency. This AC magnetic field magnetizes the yoke comprising the magnetic circuit. The oscillation from the intermittent magnetization prompts the vibration diaphragm to vibrate up and down, generating buzzer sounds through the resonator.

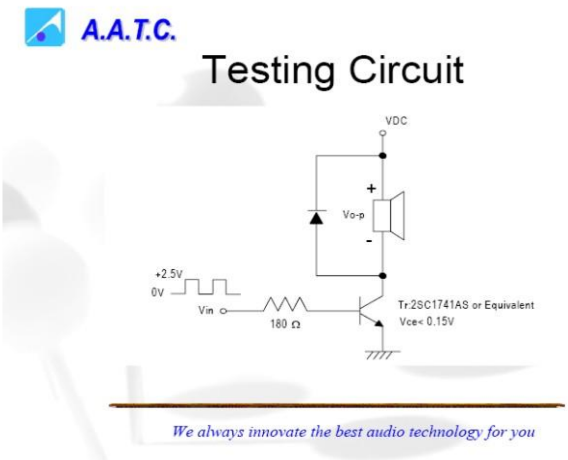


Recommended Driving Circuit for Magnetic Transducer

Introduction of Magnetic Buzzer (Transducer)







Specifications:

Rated Voltage: A magnetic buzzer is driven by 1/2 square waves (V o-p).

Operating Voltage: For normal operating. But it is not guaranteed to make the minimum Sound Pressure Level (SPL) under the rated voltage.

Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.

Direct Current Resistance: The direct current resistance is measured by ammeter directly.

Sound Output: The sound output is measured by decibel meter. Applying rated voltage and

1/2 square waves, and the distance of 10 cm.

Rated Frequency: A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.

Operating Temp.: Keep working well between -30℃ and +70℃.

How to choose:

Driving methods: AX series with built drive circuit will be the best choice when we cannot provide frequency signal to a buzzer, it only needs direct current.

Dimension: Dimension affects frequency, small size result in high frequency.

Voltage: Depend on V o-p (1/2 square waves)

Fixed methods: From the highest cost to the lowest- DIP, wires/ connector, SMD.

Soldering methods: AS series is soldered by hand, the frequency is lower because of the holes on the bottom. On the other hand, we suggest AC series for the reflow soldering, the reliability is better.

How to choose a buzzer

There are many different kinds of buzzer to choose, first we need to know a few parameters, such as voltage, current, drive method, dimension, mounting type, and the most important thing is how much SPL and frequency we want.

Operating voltage: Normally, the operating voltage for a magnetic buzzer is from 1.5V to 24V, for a piezo buzzer is from 3V to 220V. However, in order to get enough SPL, we suggest giving at least 9V to drive a piezo buzzer.

Consumption current: According to the different voltage, the consumption current of a magnetic buzzer is from dozens to hundreds of mill amperes; oppositely, the piezo type saves much more electricity, only needs a few mill amperes, and consumes three times current when the buzzer start to work.

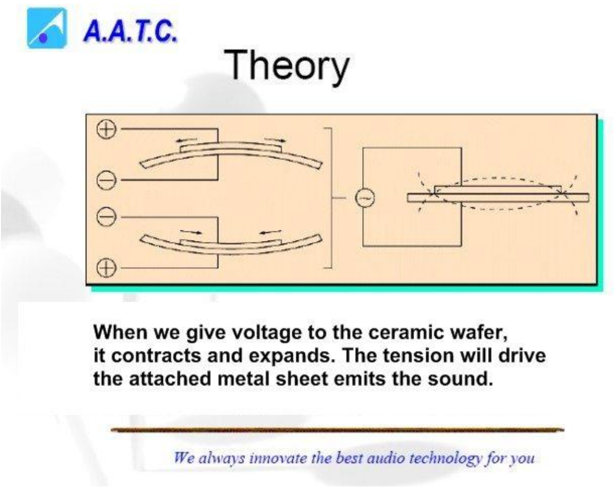
Driving method: Both magnetic and piezo buzzer have self-drive type to choose. Because of the internal set drive circuit, the self-drive buzzer can emit sound as long as connecting with the direct current. Due to the different work principle, the magnetic buzzer need to be driven by 1/2 square waves, and the piezo buzzer need square waves to get better sound output.

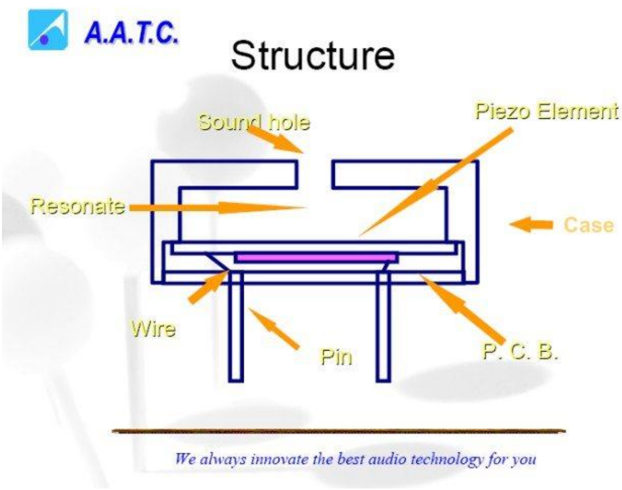
Dimension: The dimension of the buzzer affects its SPL and the frequency, the dimension of the magnetic buzzer is from 7 mm to 25 mm; the piezo buzzer is from 12 mm to 50 mm, or even bigger.

Connecting way: Dip type, Wire type, SMD type, and screwed type for big piezo buzzer are usually seen.

Sound Pressure Level (SPL): Buzzer is usually tested the SPL at the distance of 10 cm, if distance double, the SPL will decay about 6 dB; oppositely, the SPL will increase 6 dB when the distance is shortened by one time. The SPL of the magnetic buzzer can reach to around 85 dB/ 10 cm; the piezo buzzer can be designed to emit very loud sound, for example, the common siren, are mostly made of piezo buzzer.

Introduction of Piezo Buzzer





Specifications:
Rated Voltage: A piezo buzzer is driven by square waves (V p-p).
Operating Voltage: For normal operating. But it is not guaranteed to make the minimum SPL under the rated voltage.
Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.
Capacitance: A piezo buzzer can make higher SPL with higher capacitance, but it consumes

more electricity.

Sound Output: The sound output is measured by decibel meter. Applying rated voltage and square waves, and the distance of 10 cm.

Rated Frequency: A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.

Operating Temp.: Keep working well between -30℃ and +70℃.

How to choose:

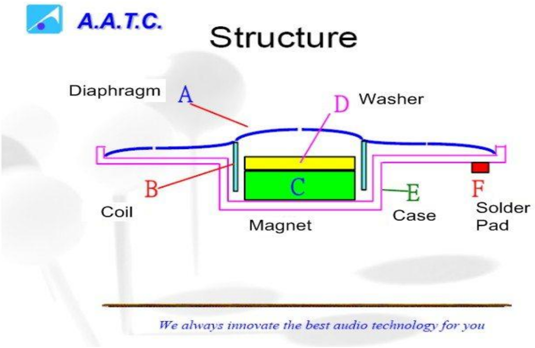
Driving methods: AZ-XX's-x series with built drive circuit will be the best choice when we cannot provide frequency signal to a buzzer, it only needs direct current. Besides, there are different tone nature for you to choose, such as continuous, fast pulse, and slow pulse.

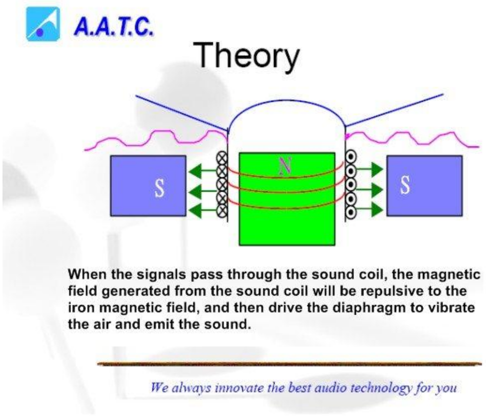
Dimension: Dimension affects frequency, small size result in high frequency.

Voltage: Driven by square waves (V p-p), the higher voltage results in the higher SPL.

Pin Pitch: The numerous spec. for the piezo buzzers lead to the difficulty in finding a spec. in facsimile, therefore we suggest that you can firstly choose a spec. with the same pitch and similar frequency.

Introduction of Micro Speaker:





How to choose:

The factors which affect the SPL: the square measure of diaphragm, the amplitude of vibration, magnetic field intensity, power, impedance, resonant chamber, the pattern and the thickness of diaphragm, and the holes.

Power vs. SPL: Suppose all the conditions are the same, increasing the power does not mean the SPL will increase as well. We need to revise the diaphragm and the sound coil to load the higher power, but it leads to lower SPL instead.

Dimension vs. SPL: A larger speaker can vibrate more air; therefore, it provides higher SPL. In addition, the thicker speaker can give wider amplitude of vibration which also leads to higher SPL.

Acoustics: What we request most is how much SPL a micro speaker can output.

Matching: It will be better to provide the power slightly higher than the rated power for the enlarged circuit.

Question for mechanism:

The volume of the resonant chamber: The general problem of the consuming products is that the resonant chambers are not big enough. We can only try to find space to enlarge the volume of the resonant chamber.

Sound Hole: Must be more than 1/8 of the diaphragm's area at least.

Airtight: The front and back sound fields of the speaker should be separated to avoid neutralization.

Shock absorber: When a speaker works the vibration will also happen at the same time. In order to reduce interference, it will do good to have some material between speaker and case to absorb the shock.

Mounting: The speakers are usually fixed on the case. Firmly fixed is important especially for the iron housing or the large size to avoid separating in the drop test.

How to choose the speaker

Dimension: To the micro speaker, size has decisive influence on its volume. 5mm difference of diameter might result in double or half area of diaphragm, therefore the SPL is quite different. Besides, the thicker speaker has more space to vibrate the air, and usually has bigger magnet, so it will be more powerful to push the air and emit louder sound.

Power: Mainly refer to how much power can a speaker bear, there is no direct relation to the SPL. The speaker with larger power needs to use thicker diaphragm and sound coil to bear larger power, which will lead to lower efficiency (SPL). Therefore, according to the mechanical design, try to select a larger speaker which matches the outputting power from the amplified circuit, then the best SPL would be emitted.

Impedance: Higher impedance can save more electricity; however, the SPL and the loaded power will go down. The reason is that we have to use thinner wire or to coil more, the front makes the power lower, and the after leads to heaviness and low efficiency.

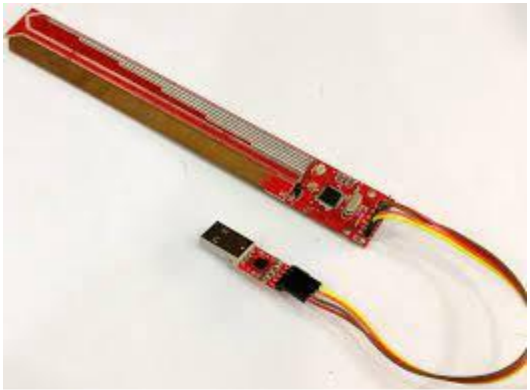
The material of diaphragm: Most speakers (diameter less than 50mm) use mylar diaphragms, which are easily finished, cheaper and waterproof. However, mylar diaphragm is not good at heatproof and the sound is stiff.

The patterns of diaphragm: The speaker with concentric circles diaphragm is good for the speech sounds. Generally, the SPL is good at the frequency before 5-6 KHz, but will dramatically decrease after 6 KHz. On the other hand, the speaker with radiate diaphragm has average frequency response. Supposing other conditions are all the same, the SPL of radiate diaphragm will lower than the concentric circles one at the frequency before 6 KHz.

PH SENSOR:

In nearly every industrial and scientific application, pH is determined by measuring the voltage of an electrochemical cell. Figure 2-1 shows a simplified diagram of a pH cell. The cell consists of a measuring electrode, a reference electrode, a temperature sensing element, and the liquid being measured. The voltage of the cell is directly proportional to the pH of the liquid. The pH meter measures the voltage and uses a temperature-dependent factor to convert the voltage to pH. Because the cell has high internal resistance, the pH meter must have a very high input impedance. Figure 2-1 shows separate measuring and reference electrodes. In most process sensors, the electrodes and the temperature element are combined into a single body. Such sensors are often called combination electrodes. The cell voltage is the algebraic sum of the potentials of the measuring electrode, the reference electrode, and the liquid junction. The potential of the measuring electrode depends only on the pH of the solution. The potential of the reference electrode is unaffected by pH, so it provides a stable reference voltage. The liquid junction potential depends in a complex way on the identity and concentration of the ions in the sample. It is always present, but if the sensor is properly

designed, the liquid junction potential is usually small and relatively constant. All three potentials depend on temperature. The construction of the electrodes and the electrical potentials associated with them are discussed in Sections 2.2 and 2.3. THEORY AND PRACTICE OF pH MEASUREMENT SECTION 2.0 CELLS FOR MEASURING pH SECTION 2.0 CELLS FOR MEASURING pH 2.1 GENERAL 2.2 MEASURING ELECTRODE 2.3 REFERENCE ELECTRODE 2.4 LIQUID JUNCTION POTENTIAL 2.5 DOUBLE JUNCTION REFERENCE ELECTRODES FIGURE
2-1. pH Measurement Cell. The cell consists of a measuring and reference electrode. The voltage between the electrodes is directly proportional to the pH of the test solution. The proportionality constant depends on temperature, so a temperature sensor is also necessary. A 100 Ω platinum RTD is commonly used, although 1000 Ω platinum RTDs, 3 kΩ Balco RTDs, and thermistors are also used.



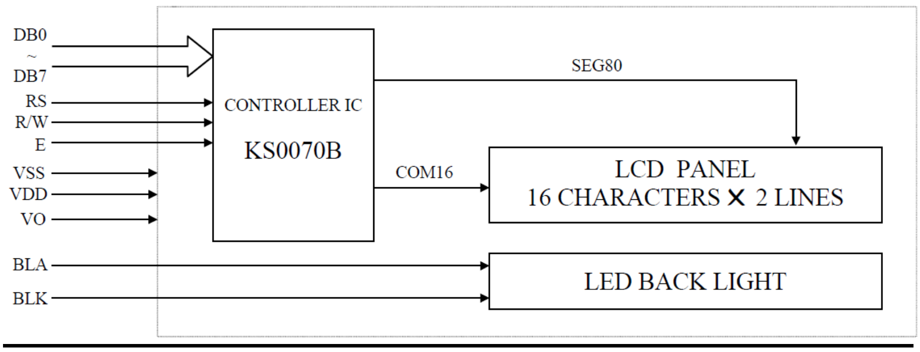
MEASURING ELECTRODE Figure 2-2 shows the internals of the measuring electrode. The heart of the electrode is a thin piece of pH-sensitive glass, which is blown onto the end of a length of glass tubing. The pH-sensitive glass, usually called a glass membrane, gives the electrode its common name: glass electrode. Sealed inside the electrode is a solution of potassium chloride buffered at pH 7. A piece of silver wire plated with silver chloride contacts the solution. The silver wire-silver chloride combination in contact with the filling solution constitutes an internal reference electrode. Its potential depends solely on the chloride concentration in the filling solution. Because the chloride concentration is fixed, the electrode potential is constant. See Appendix A for a more detailed discussion of how the chloride concentration determines the electrode potential. As Figure 2-2 shows, the outside surface of the glass membrane contacts the liquid being measured, and the inside surface contacts the filling solution. Through a complex mechanism, an electrical potential directly proportional to pH develops at each glass-liquid interface. Because the pH of the filling solution is fixed, the potential at the inside surface is constant. The potential at the outside surface, however, depends on the pH of the test solution. The overall potential of the measuring electrode equals the potential of the internal reference electrode plus the potentials at the glass membrane surfaces. Because the potentials inside the electrode are constant, the overall electrode potential depends solely on the pH of the test solution. The potential of the measuring electrode also depends on temperature. If the pH of the sample remains constant but the temperature changes, the electrode potential will change. Compensating for changes in glass electrode potential with temperature is an important part of the pH measurement. Figure 2-3 shows a cross-section through the pH glass. pH sensitive glasses absorb water. Although the water does not penetrate more than about 50 nanometers (5 x 10-8 m) into the glass, the hydrated layer must be present for the glass to respond to pH changes. An ion exchange mechanism involving alkali metals and hydrogen ions in the hydrated layer is responsible for the pH response of the glass. The layer of glass between the two hydrated layers remains dry. The dry layer makes the glass a poor conductor of electricity and causes the high internal resistance (several hundred megohms) typical of glass electrodes.

16 * 2 Alphanumeric LCD

Description

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

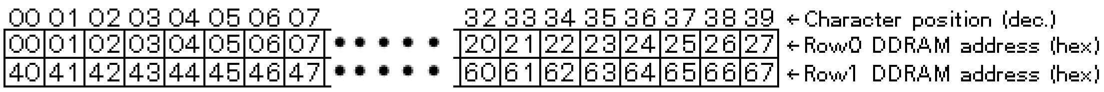
Block Diagram



Display Data RAM

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So, whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.



DDRAM Address for 2 Line LCD

Character Generator ROM

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM.

Busy Flag

Busy Flag is a status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e. BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

Instruction Register (IR) and Data Register (DR)

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g., LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD. Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

16 x 2 Alphanumeric LCD Module Features

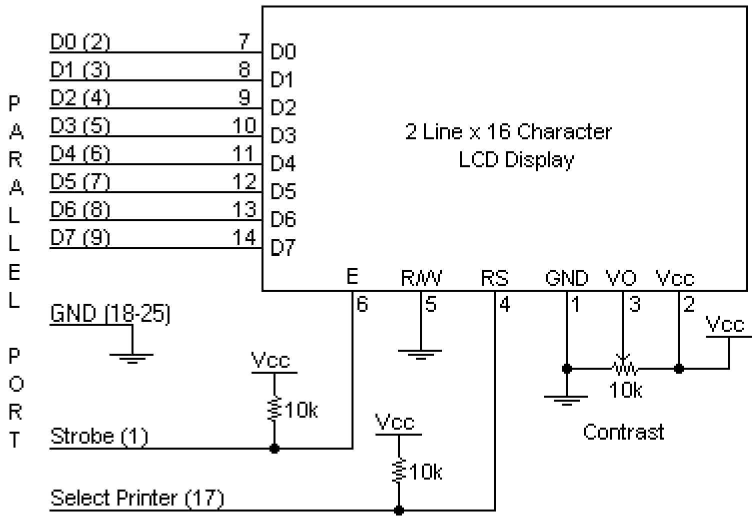
- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

Schematic

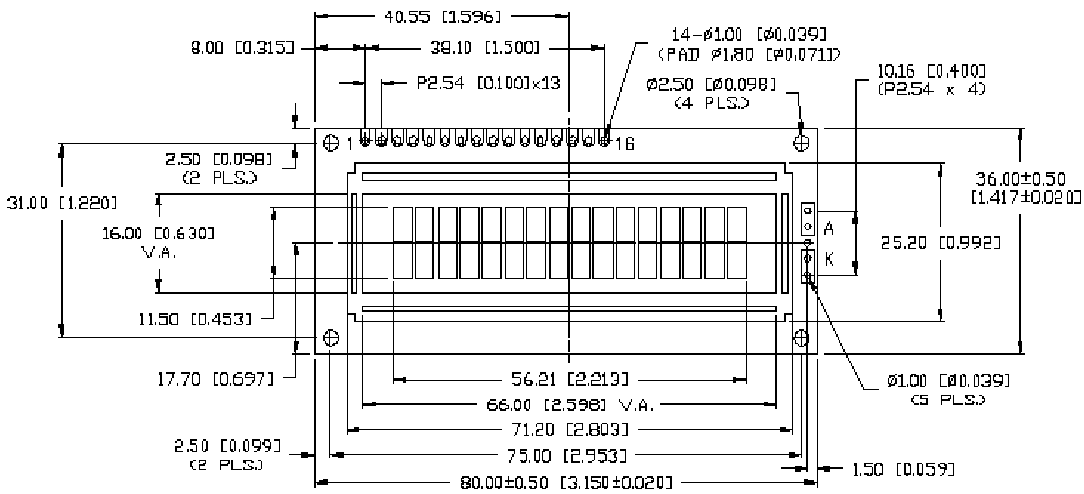
Connector Pin Assignment:

Pin	*Symbol*	*Function*	*Pin*	*Symbol*	*Function*
14	DB7	Data Bus Line	6	E	Enable
13	DB6	Data Bus Line	5	R/W	Read/Write
12	DB5	Data Bus Line	4	RS	Register select input
11	DB4	Data Bus Line	3	Vcontrast	Contrast (0V for max contrast)
10	DB3	Data Bus Line	2	**GND (0V)**	Power supply
9	DB2	Data Bus Line	1	**Vcc +5V**	Power supply
8	DB1	Data Bus Line			
7	DB0	Data Bus Line			

Note1: Pin 1 is +5V and pin 2 is GND! This is different from most other HD44780 compatible models.
Note2: Pin 14 is marked with a “1” on the back of the PCB. Do not get confused by this.



Specifications



Circuit Description

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore, by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore, we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result, we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly. The 2-line x 16-character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

Soil Moisture Sensor:

Soil moisture sensors measure the water content in soil. A soil moisture probe is made up of multiple soil moisture sensors. One common type of soil moisture sensors in commercial use is a Frequency domain sensor such as a capacitance sensor. Another sensor, the neutron moisture gauge, utilize the moderator properties of water for neutrons. Cheaper sensors - often for home use- are based on two electrodes measuring the resistance of the soil. Sometimes this simply consists of two bare (galvanized) wires, but there are also probes with wires embedded in gypsum

Agriculture

Measuring soil moisture is important in agriculture to help farmers manage their irrigation systems more efficiently. Not only are farmers able to generally use less water to grow a crop, they are able to increase yields and the quality of the crop by better management of soil moisture during critical plant growth stages.

Besides agriculture, there are many other disciplines using soil moisture sensors. Golf courses are now using sensors to increase the efficiencies of their irrigation systems to prevent over watering and leaching of fertilizers and other chemicals offsite.

Landscape irrigation

In urban and suburban areas, landscapes and residential lawns are using soil moisture sensors to interface with an irrigation controller. Connecting a soil moisture sensor to a simple

irrigation clock will convert it into a "smart" irrigation controller that prevents an irrigation cycle when the soil is wet.

Submersive Motor:

A submersible motor is an electric motor which can operate while being submerged in water. This sealed motor is typically found in use on pumps. Both sump pumps and water wells use a submersible motor to power the pumping mechanism used in the system. Due to the submersion in water, it is critical that all electrical connections be made water-tight to prevent damage to the motor as well as to anyone working on the unit.

The typical water well relies on a submersible motor to pump the water out of the well. The motor is attached to a pump and lowered down through the well casing. Many well contractors attach a small cable to the pump unit to lower it down, while many simply lower the pump unit down by the electrical cord. In either case, the pump and the submersible motor rest at the bottom of the casing suspended in the water. A hose fitted to the nipple on the pump sends the water up and out of the well when the submersible motor turns on.

This design is critical with an electric pump due to the advantage of pushing rather than pulling liquid. A pump is able to push a much greater amount of liquid than it is able to pull, thereby requiring the submersible motor to place the pump at the source of the liquid. Used in this manner, a pump is able to work effectively with a much smaller motor. This results in less electrical power used, a cooler running pump motor as well as a pump that is able to move a great quantity of liquid in a short time.

The typical submersible motor is contained within a rubber sock which maintains the pump's dryness. The sock is removed when servicing the pump or motor and replaced once the service is complete. This allows the pump to operate for a greater length of time without becoming damaged by the water. Many submersible motors offer years of uninterrupted use.

Relay

A relay is used to isolate one electrical circuit from another. It allows a low current control circuit to make or break an electrically isolated high current circuit path. The basic relay consists of a coil and a set of contacts. The most common relay coil is a length of magnet wire wrapped around a metal core. When voltage is applied to the coil, current passes through the wire and creates a magnetic field. This magnetic field pulls the contacts together and holds them there until the current flow in the coil has stopped. The diagram below shows the parts of a simple relay.

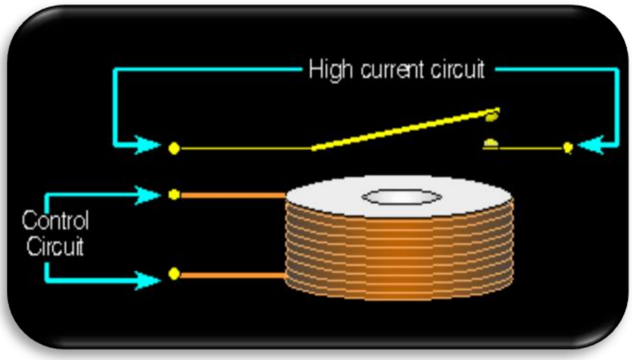


Figure: Relay

Operation:

When a current flow through the coil, the resulting magnetic field attracts an armature that is mechanically linked to a moving contact. The movement either makes or breaks a connection with a fixed contact. When the current is switched off, the armature is usually returned by a spring to its resting position shown in figure 6.6(b). Latching relays exist that require operation of a second coil to reset the contact position.

By analogy with the functions of the original electromagnetic device, a solid-state relay operates a thyristor or other solid-state switching device with a transformer or light-emitting diode to trigger it.

dwq

Pole and throw

SPST

SPST relay stands for Single Pole Single Throw relay. Current will only flow through the contacts when the relay coil is energized.

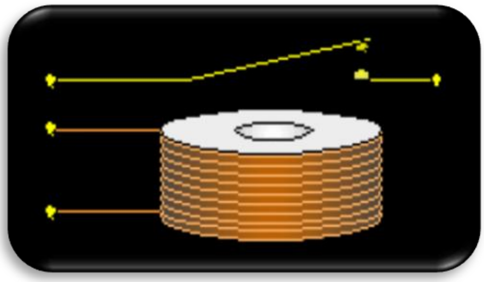


Figure: SPST Relay

SPDT Relay

SPDT Relay stands for Single Pole Double Throw relay. Current will flow between the movable contact and one fixed contact when the coil is De-energized and between the movable contact and the alternate fixed contact when the relay coil is energized. The most commonly used relay in car audio, the Bosch relay, is a SPDT relay.

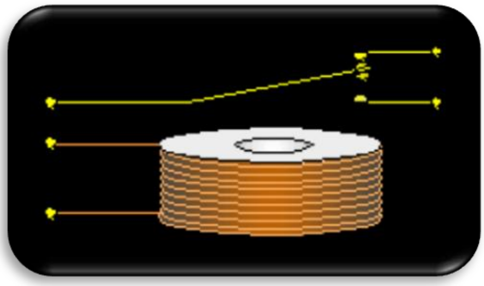


Figure: SPDT Relay

DPST Relay

DPST relay stands for Double Pole Single Throw relay. When the relay coil is energized, two separate and electrically isolated sets of contacts are pulled down to make contact with their stationary counterparts. There is no complete circuit path when the relay is De-energized.

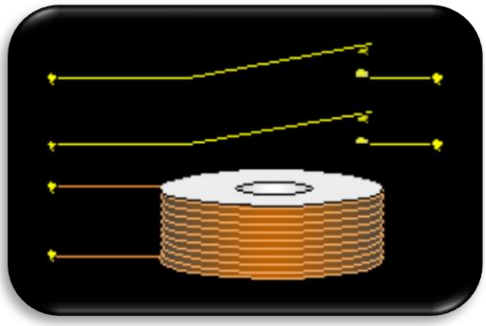


Figure: DPST Relay

DPDT Relay:

DPDT relay stands for Double Pole Double Throw relay. It operates like the SPDT relay but has twice as many contacts. There are two completely isolated sets of contacts.

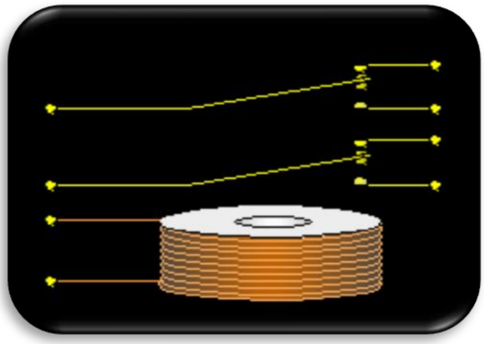


Figure: DPDT Relay

This is a 4 Pole Double Throw relay. It operates like the SPDT relay but it has 4 sets of isolated contacts.

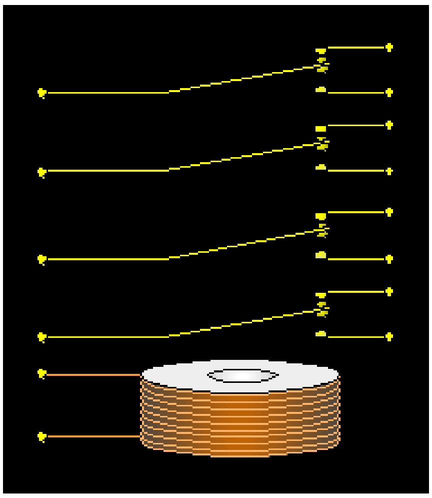


Figure: 4 Pole Double Throw relay

Types of relays:

1. Latching Relay
2. Reed Relay
3. Mercury Wetted Relay
4. Machine Tool Relay
5. Solid State Relay (SSR)

Latching relay

Latching relay, dust cover removed, showing pawl and ratchet mechanism. The ratchet operates a cam, which raises and lowers the moving contact arm, seen edge-on just below it. The moving and fixed contacts are visible at the left side of the image.

A **latching relay** has two relaxed states (bi-stable). These are also called "impulse", "keep", or "stay" relays. When the current is switched off, the relay remains in its last state. This is achieved with a solenoid operating a ratchet and cam mechanism, or by having two opposing coils with an over-centre spring or permanent magnet to hold the armature and contacts in position while the coil is relaxed, or with a remanent core. In the ratchet and cam example, the first pulse to the coil turns the relay on and the second pulse turns it off. In the two-coil example, a pulse to one coil turns the relay on and a pulse to the opposite coil turns the relay off. This type of relay has the advantage that it consumes power only for an instant, while it is being switched, and it retains its last setting across a power outage. A remanent core latching relay requires a current pulse of opposite polarity to make it change state.

Figure: Latching relay

Reed relay

A reed relay has a set of contacts inside a vacuum or inert gas filled glass tube, which protects the contacts against atmospheric corrosion. The contacts are closed by a magnetic field generated when current passes through a coil around the glass tube. Reed relays are capable of faster switching speeds than larger types of relays, but have low switch current and voltage ratings.

Mercury-wetted relay

A **mercury-wetted reed relay** is a form of reed relay in which the contacts are wetted with mercury. Such relays are used to switch low-voltage signals (one volt or less) because of their low contact resistance, or for high-speed counting and timing applications where the mercury eliminates contact bounce. Mercury wetted relays are position-sensitive and must be mounted vertically to work properly. Because of the toxicity and expense of liquid mercury, these relays are rarely specified for new equipment. See also mercury switch.

Machine tool relay

A **machine tool relay** is a type standardized for industrial control of machine tools, transfer machines, and other sequential control. They are characterized by a large number of contacts (sometimes extendable in the field) which are easily converted from normally-open to normally-closed status, easily replaceable coils, and a form factor that allows compactly installing many relays in a control panel. Although such relays once were the backbone of automation in such industries as automobile assembly, the programmable logic controller (PLC) mostly displaced the machine tool relay from sequential control applications.

Solid-state relay

A solid state relay (SSR) is a solid state electronic component that provides a similar function to an electromechanical relay but does not have any moving components, increasing long-term reliability. With early SSR's, the tradeoff came from the fact that every transistor has a small voltage drop across it. This voltage drop limited the amount of current a given SSR could handle. As transistors improved, higher current SSR's, able to handle 100 to 1,200 Amperes, have become commercially available. Compared to electromagnetic relays, they may be falsely triggered by transients.

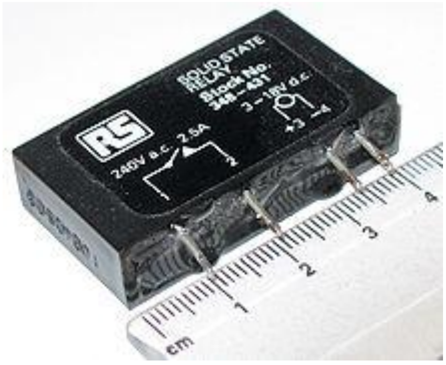


Figure: Solid relay, which has no moving parts

Specification

- Number and type of contacts – normally open, normally closed, (double-throw)
- Contact sequence – "Make before Break" or "Break before Make". For example, the old-style telephone exchanges required Make-before-break so that the connection didn't get dropped while dialling the number.
- Rating of contacts – small relays switch a few amperes, large contactors are rated for up to 3000 amperes, alternating or direct current
- Voltage rating of contacts – typical control relays rated 300 VAC or 600 VAC, automotive types to 50 VDC, special high-voltage relays to about 15 000 V
- Coil voltage – machine-tool relays usually 24 VAC, 120 or 250 VAC, relays for switchgear may have 125 V or 250 VDC coils, "sensitive" relays operate on a few milli-amperes

Applications

Relays are used: To control a high-voltage circuit with a low-voltage signal, as in some types of modems,

- To control a high-current circuit with a low-current signal, as in the starter solenoid of an automobile,
- To detect and isolate faults on transmission and distribution lines by opening and closing circuit breakers (protection relays),
- To isolate the controlling circuit from the controlled circuit when the two are at different potentials, for example when controlling a mains-powered device from a low-voltage switch. The latter is often applied to control office lighting as the low

voltage wires are easily installed in partitions, which may be often moved as needs change. They may also be controlled by room occupancy detectors in an effort to conserve energy,

- To perform logic functions. For example, the Boolean AND function is realized by connecting relay contacts in series, the OR function by connecting contacts in parallel. Due to the failure modes of a relay compared with a semiconductor, they are widely used in safety critical logic, such as the control panels of radioactive waste handling machinery.
- As oscillators, also called vibrators. The coil is wired in series with the normally closed contacts. When a current is passed through the relay coil, the relay operates and opens the contacts that carry the supply current. This stops the current and causes the contacts to close again. The cycle repeats continuously, causing the relay to open and close rapidly. Vibrators are used to generate pulsed current.
- To generate sound. A vibrator, described above, creates a buzzing sound because of the rapid oscillation of the armature. This is the basis of the electric bell, which consists of a vibrator with a hammer attached to the armature so it can repeatedly strike a bell.
- To perform time delay functions. Relays can be used to act as an mechanical time delay device by controlling the release time by using the effect of residual magnetism by means of a inserting copper disk between the armature and moving blade assembly.

CHAPTER -6

SOFTWARE EXPLINATION

6.1 Software Requirements

- Proteus simulation
- Arduino software
- Programming language

6.1.1 Arduino software:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals

- has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux.

Teachers and students use it to build low-cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step-by-step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handy board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open-source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version
- Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money
- Getting Started with Arduino and Genuino products: -

Install the Arduino Software (IDE) on Windows PCs-

…This explains how to install the Arduino Software (IDE) on Windows machines.

- ✓ Download the Arduino Software (IDE)
- ✓ Proceed with board specific instructions.

How to Download the Arduino Software (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

Installation:

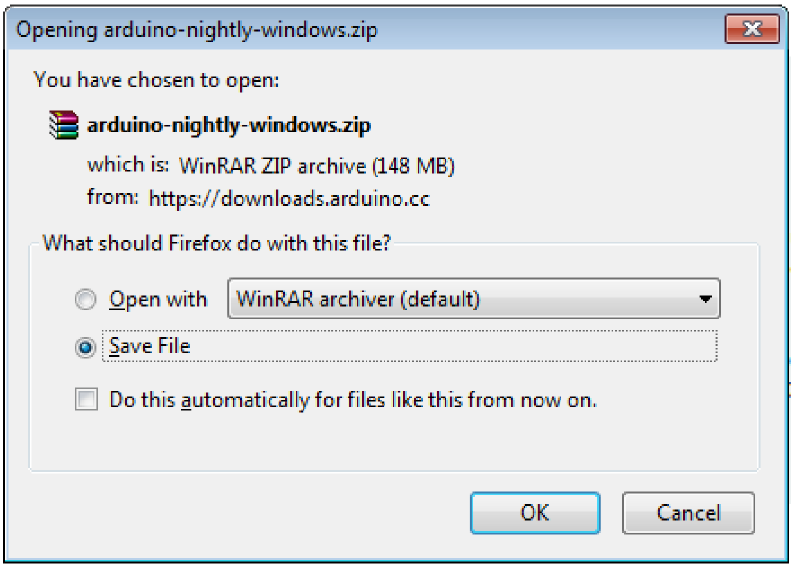
In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



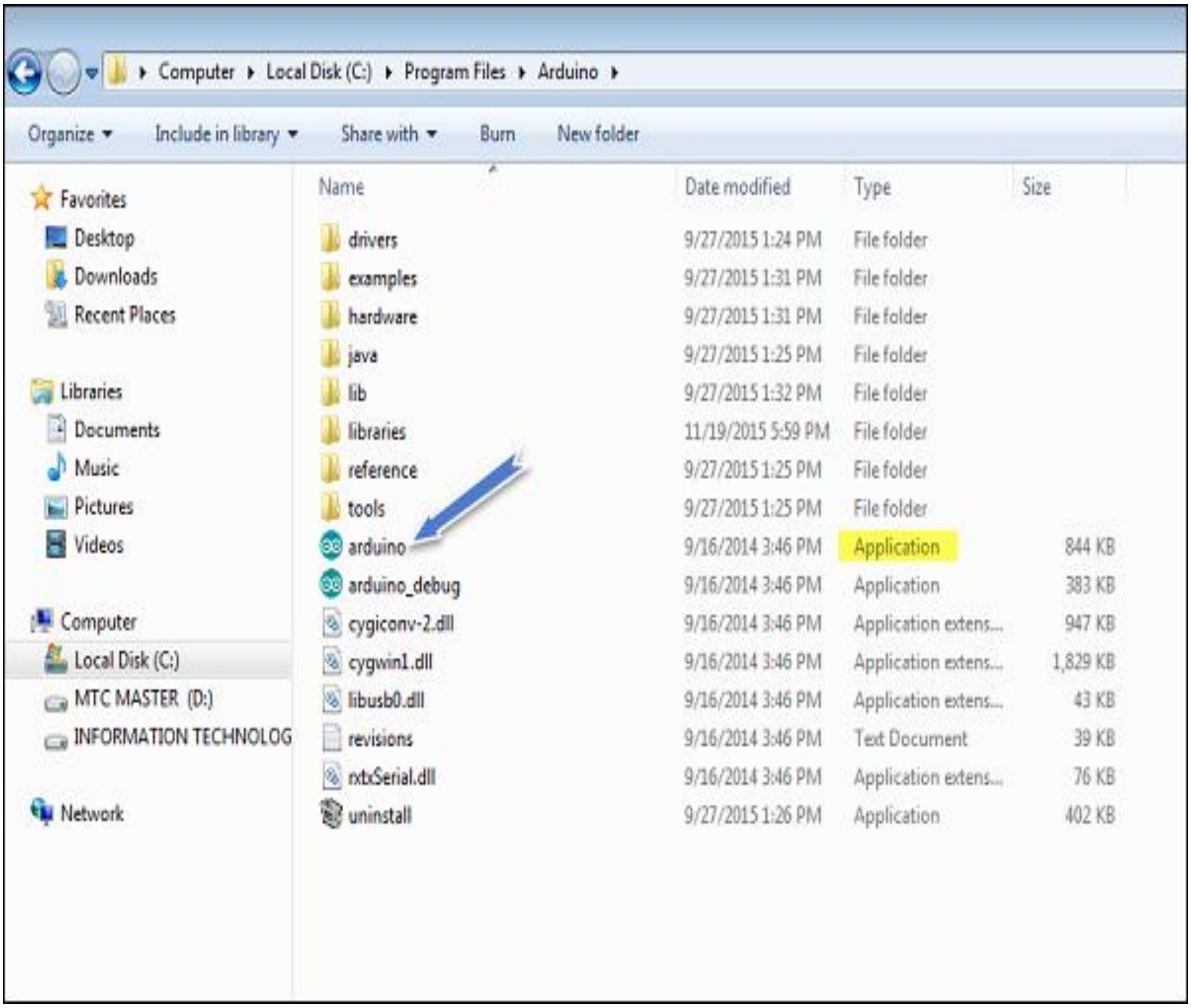
Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

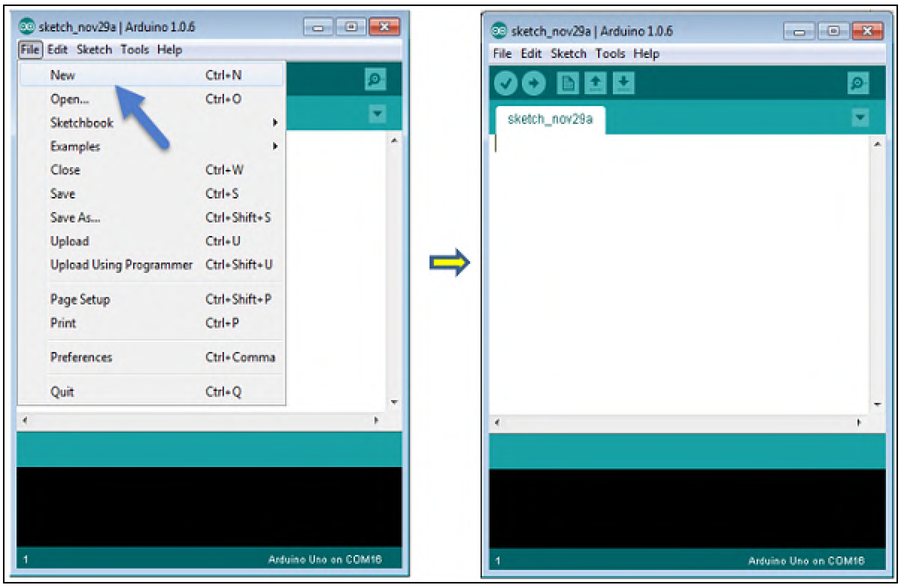


Step 5 − Open your first project.

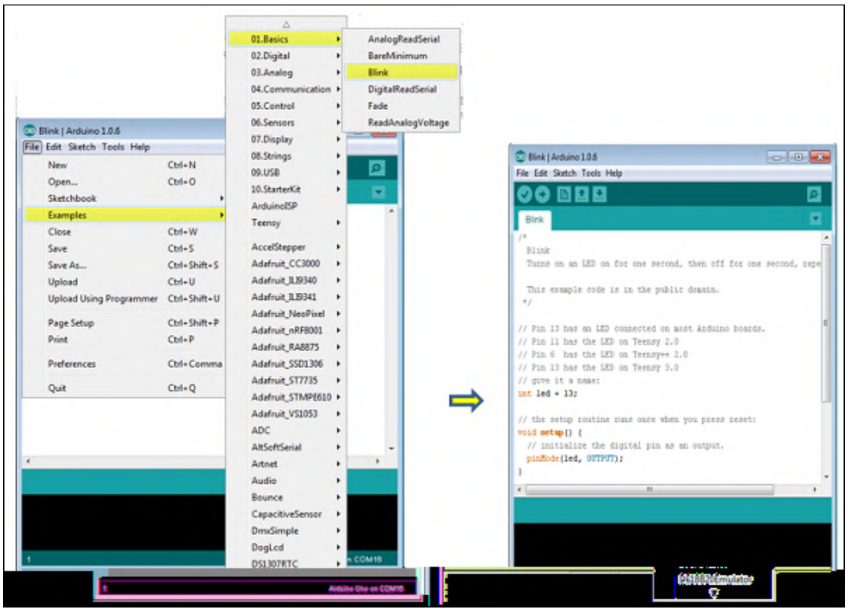
Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.



To open an existing project example, select File → Example → Basics → Blink.

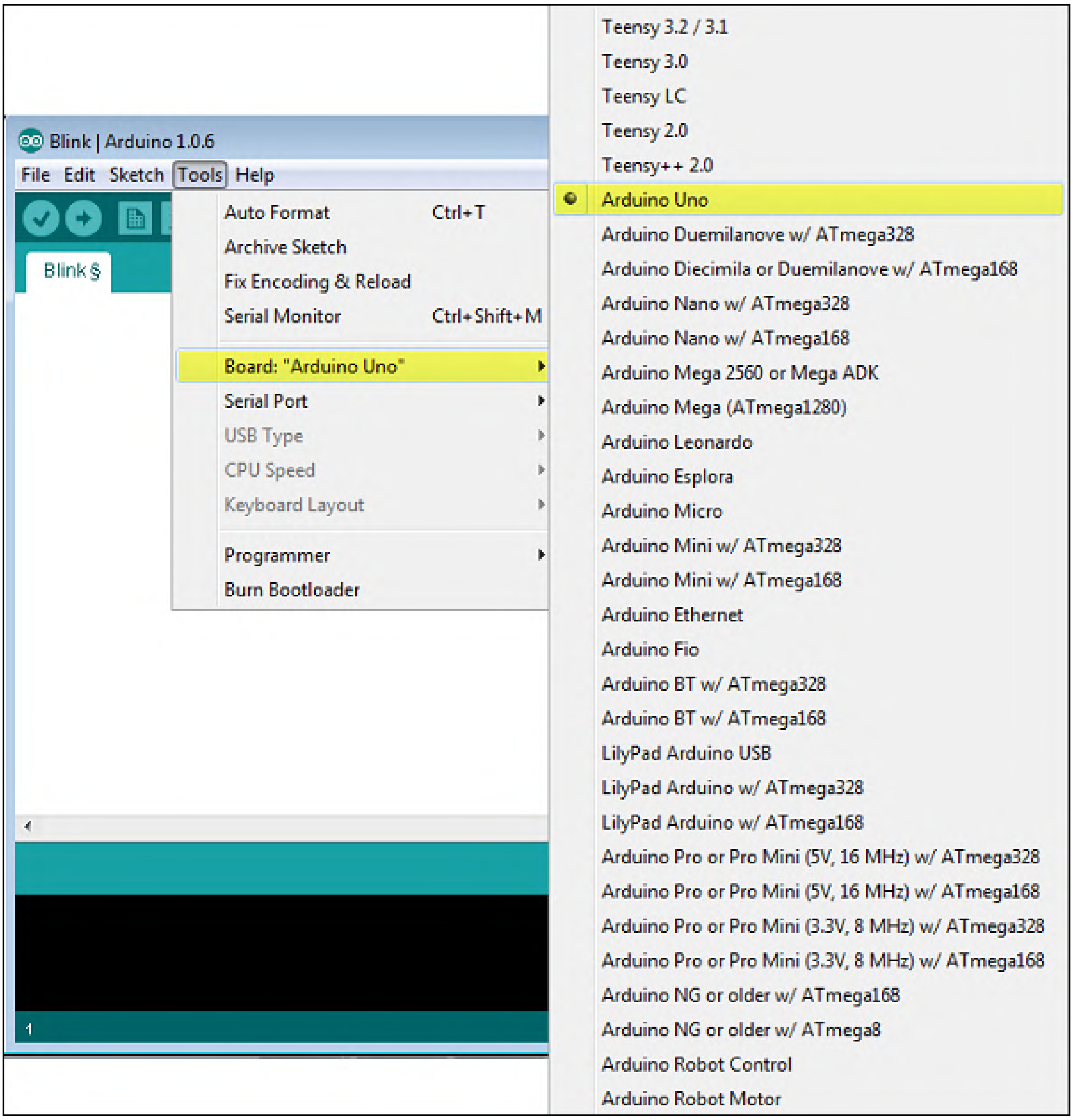


Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

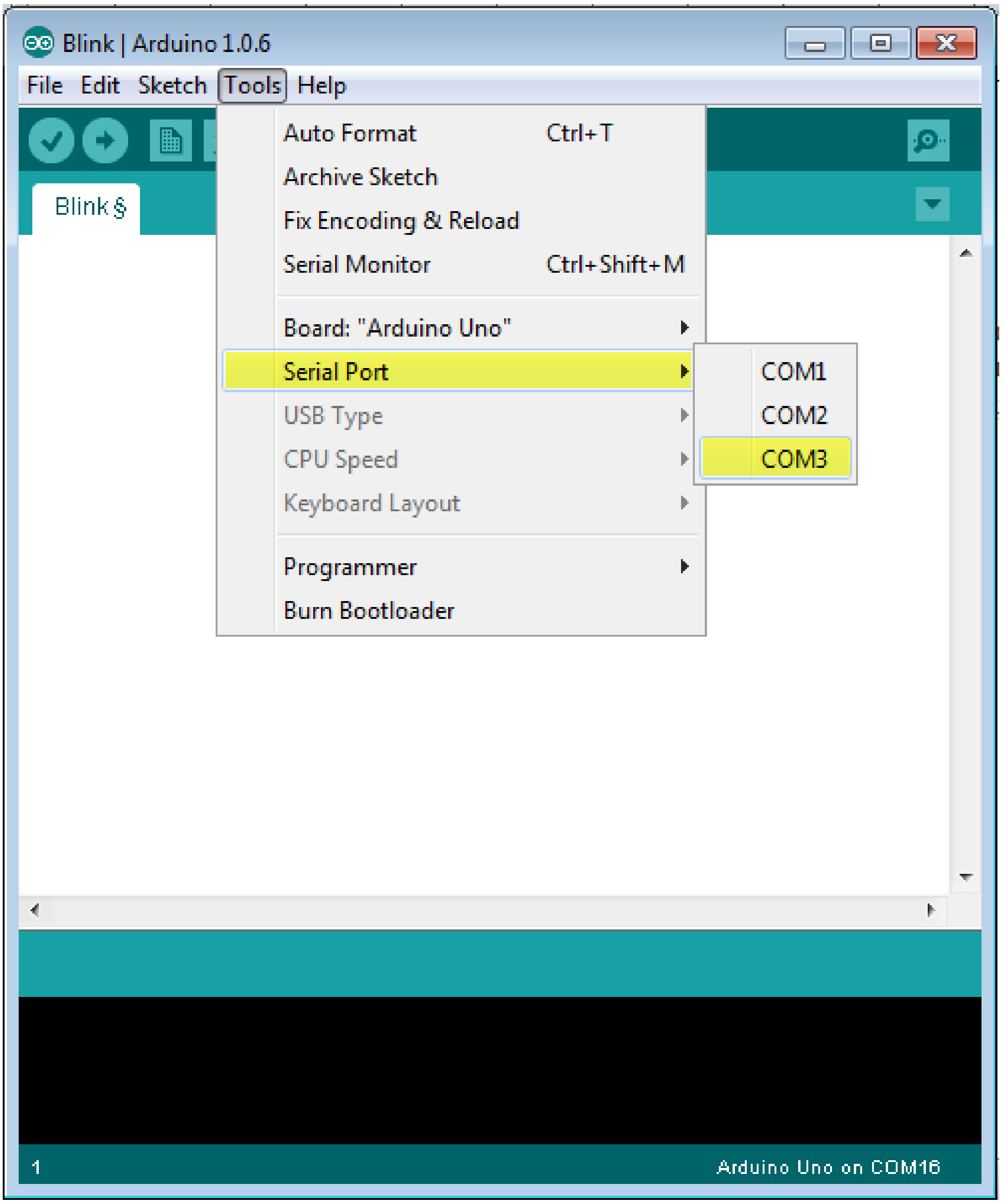
Go to Tools → Board and select your board.



Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

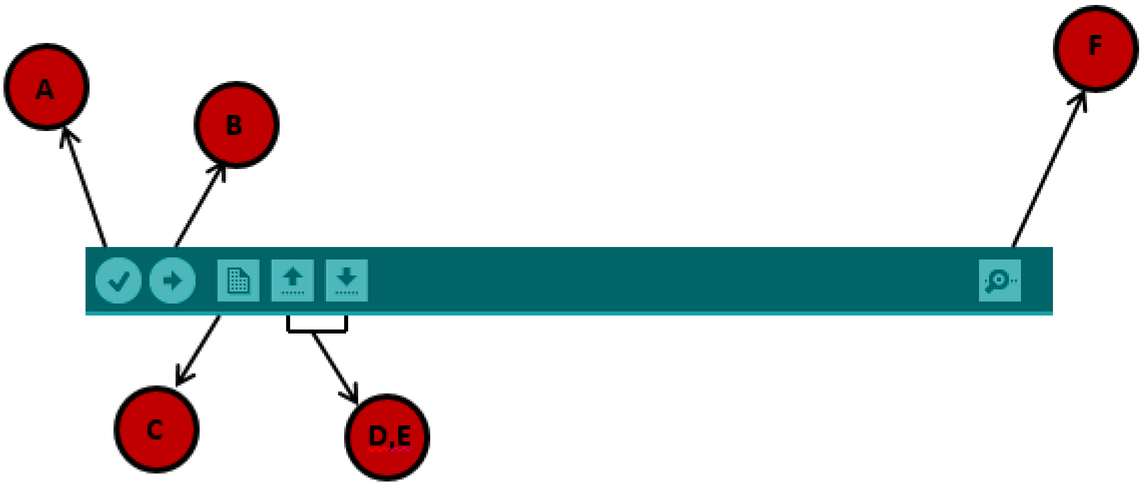
Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.



Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.



A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketches.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24 V (for example, you could use a car battery), a standard 9 V battery is convenient.

While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

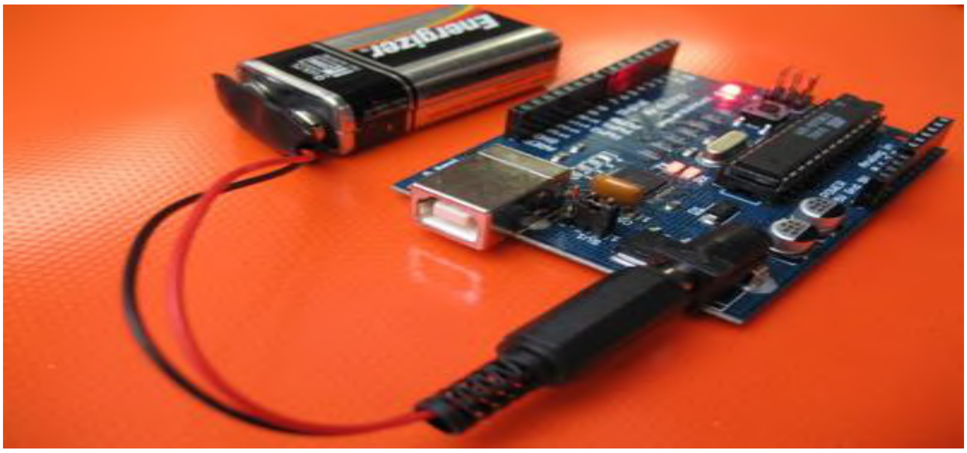


Fig.6.1: Arduino with battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called “sketches”, but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
```

}

void loop() {}

6.2 MC Programming Language: Embedded C

This is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements. Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

6.3 Proteus:

Proteus:

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possesses 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing.

ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.

The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design

Step 1: Open ISIS software and select new design in File menu

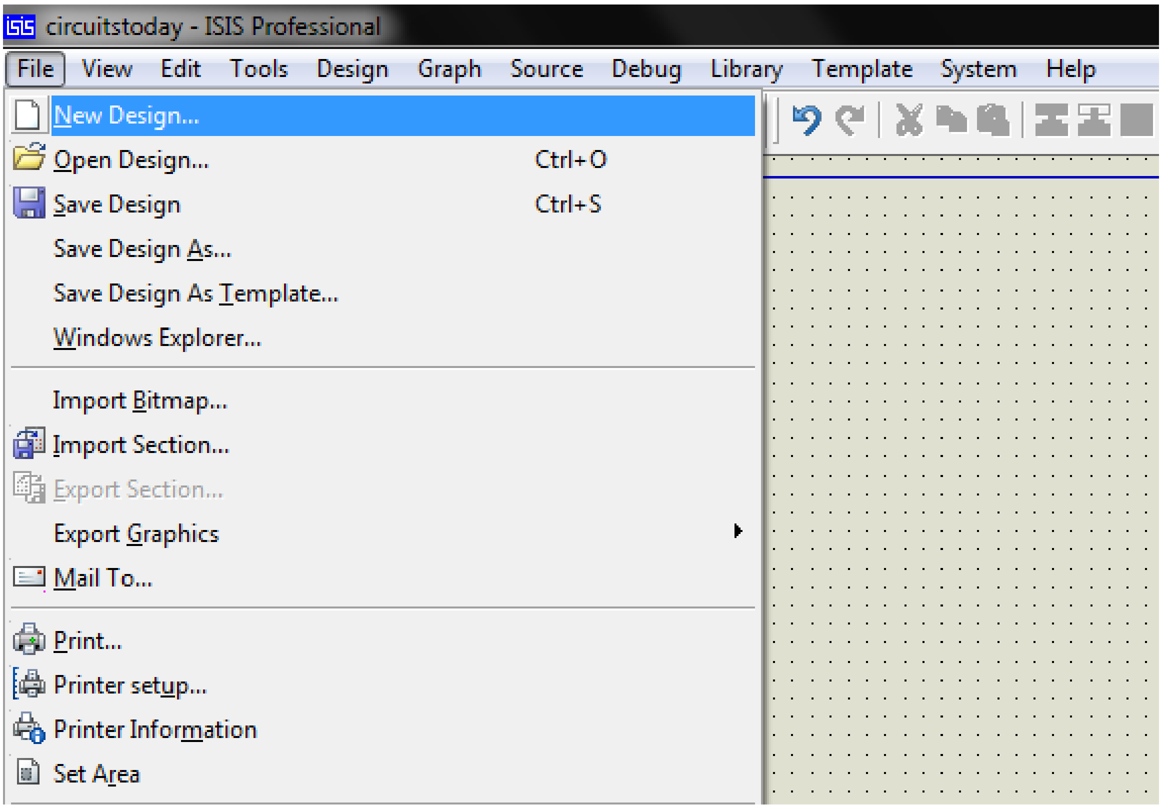


Fig Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

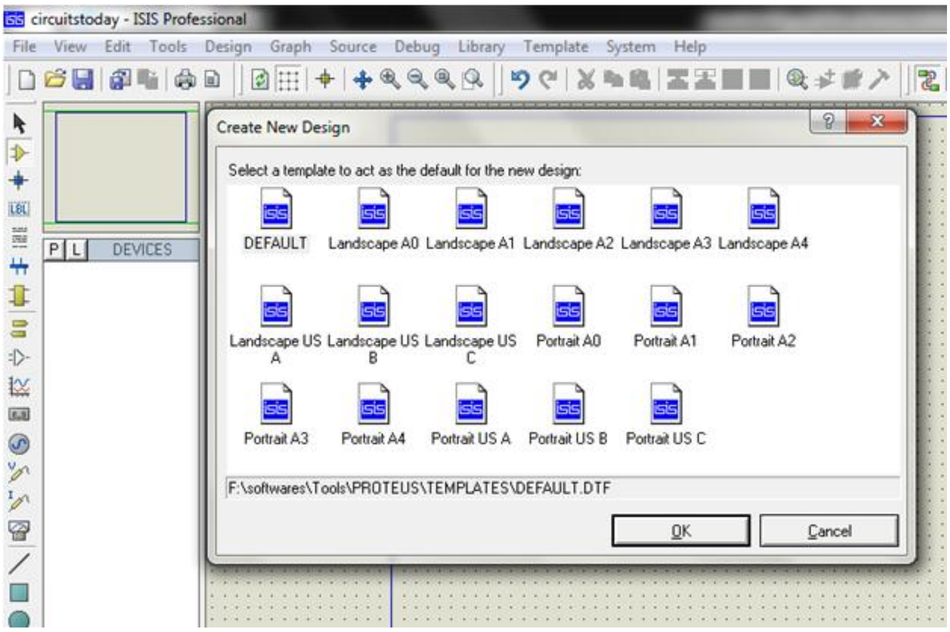


Fig Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

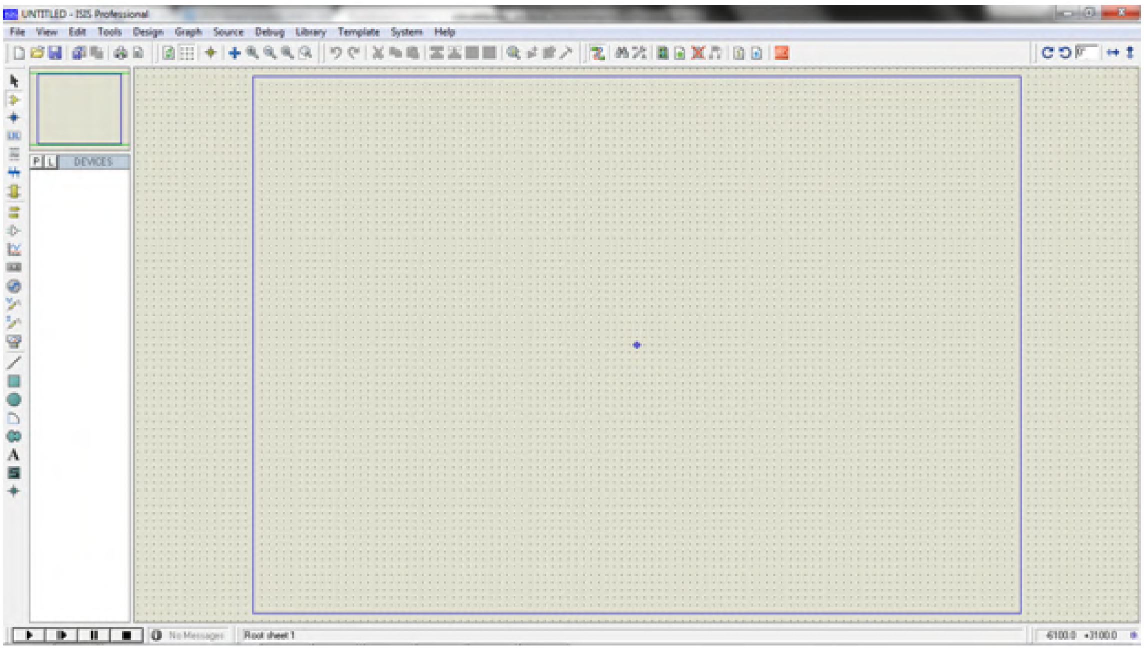


Fig Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

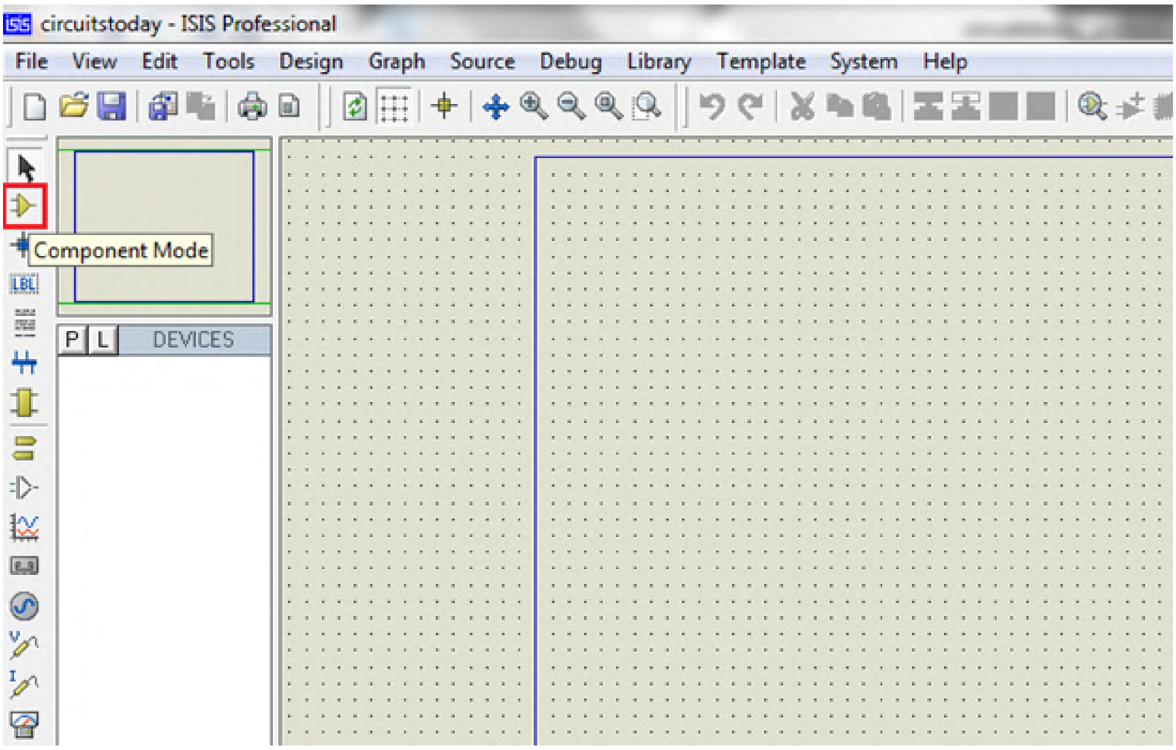


Fig Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

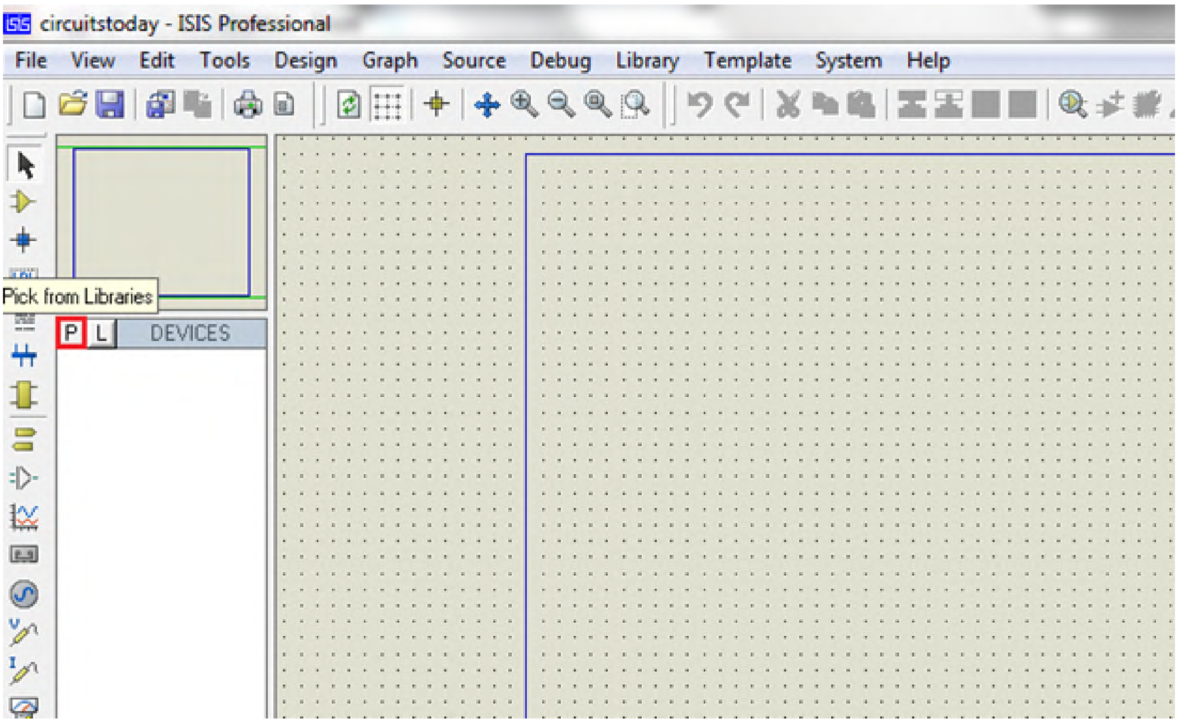


Fig Pick from Libraries

Step 6: Select the components from categories or type the part name in Keyword's text box.

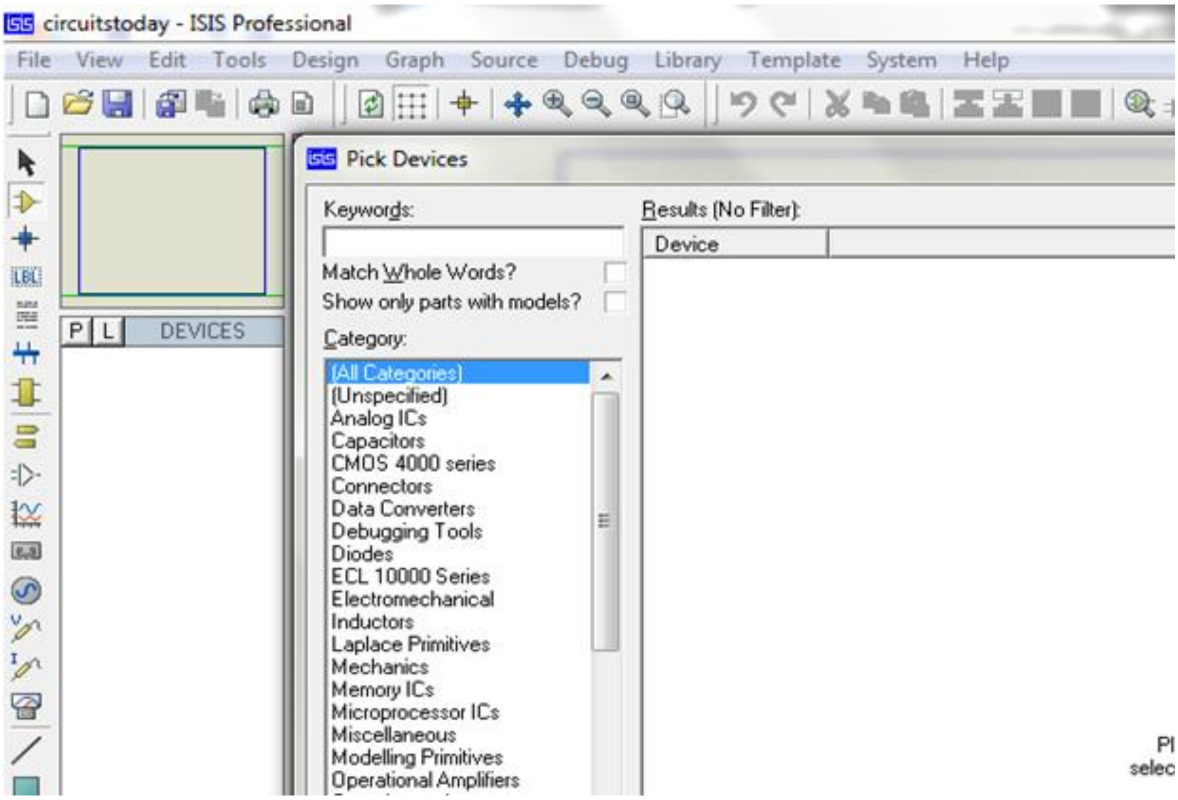


Fig Keywords Textbox

Example shows selection of push button. Select the components accordingly.

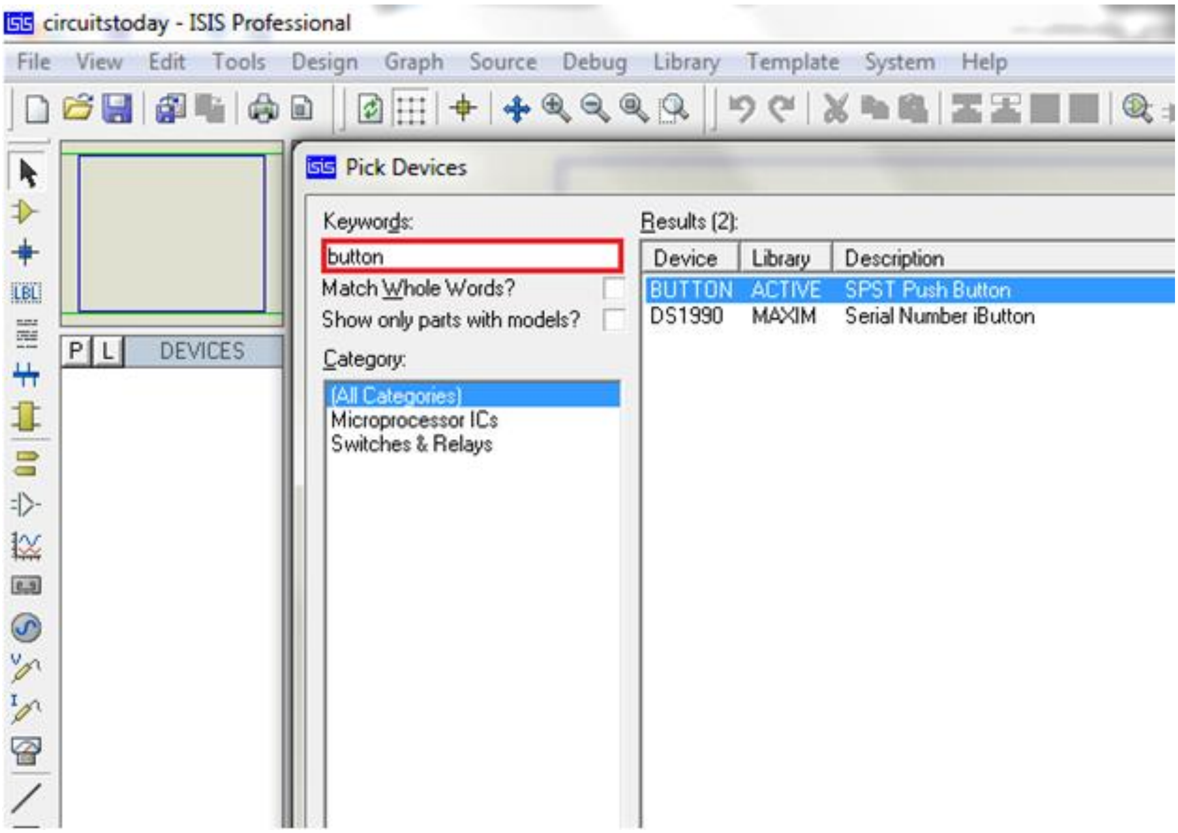


Fig Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

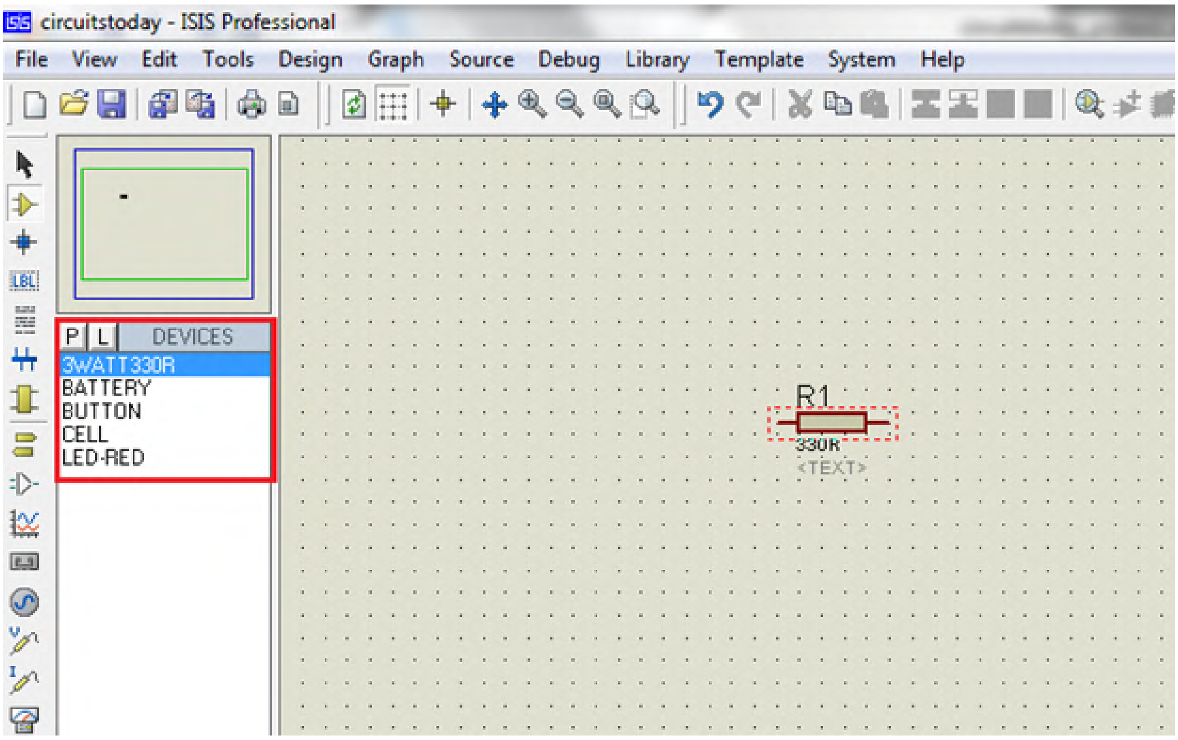


Fig Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

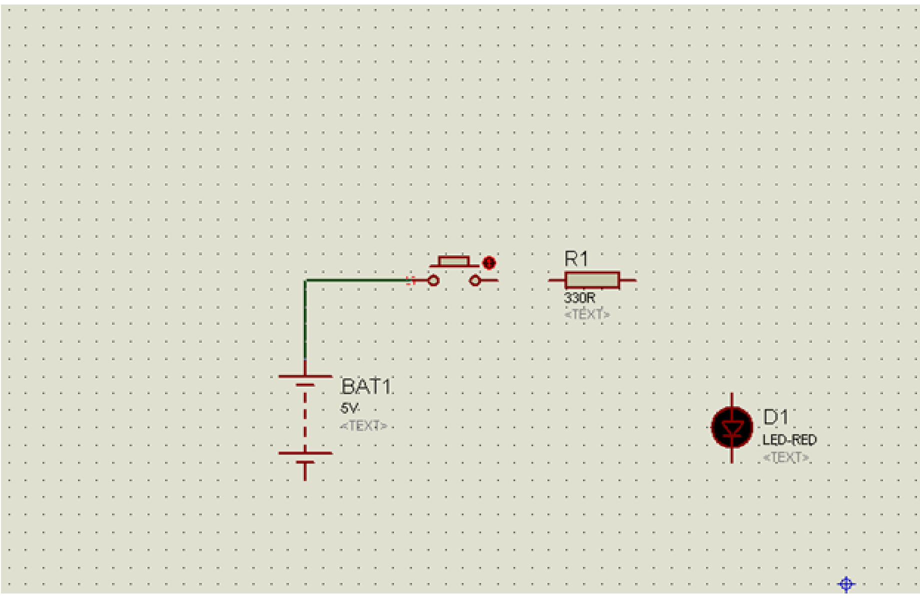


Fig Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

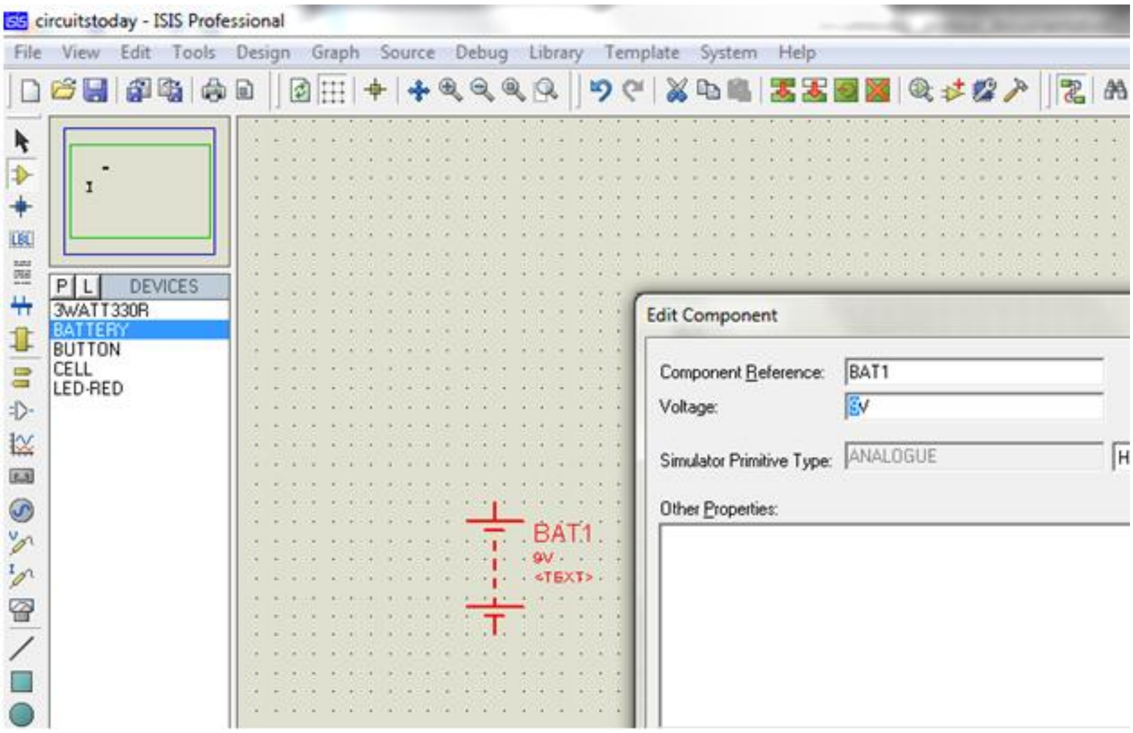


Fig Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

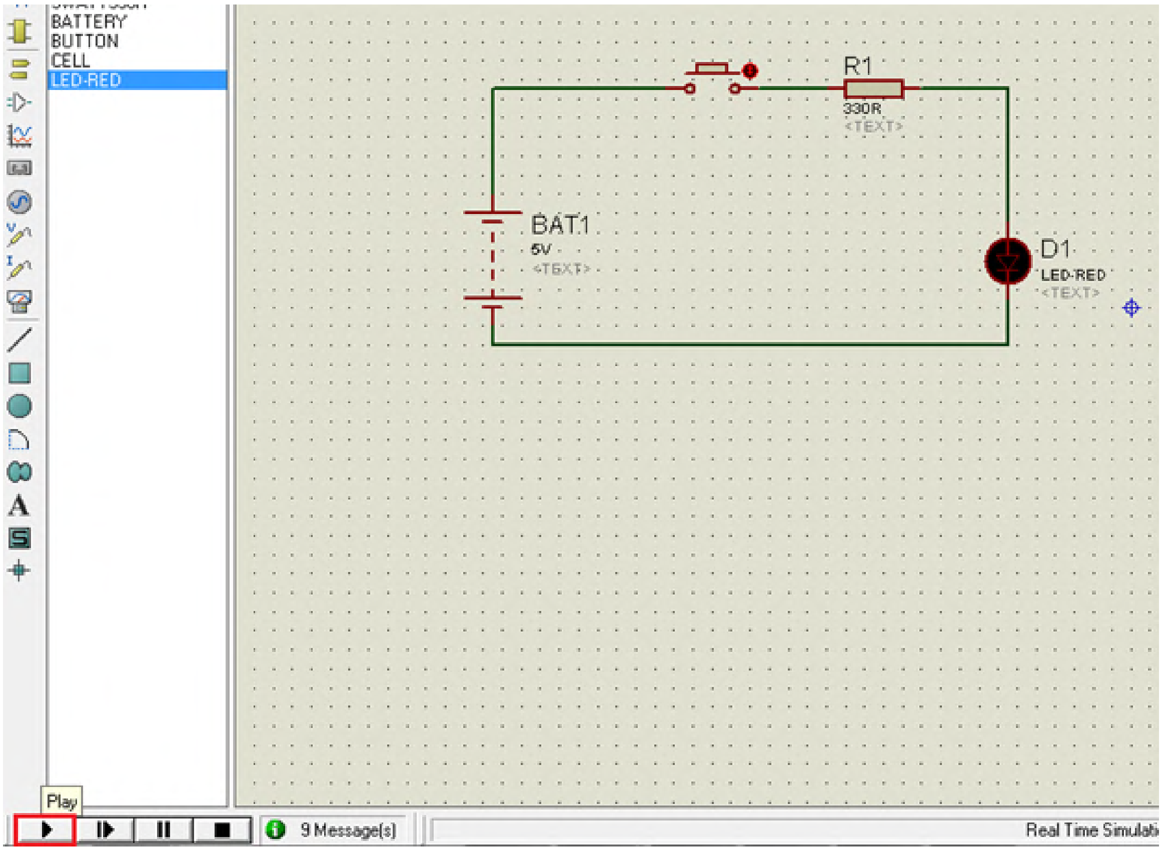


Fig Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

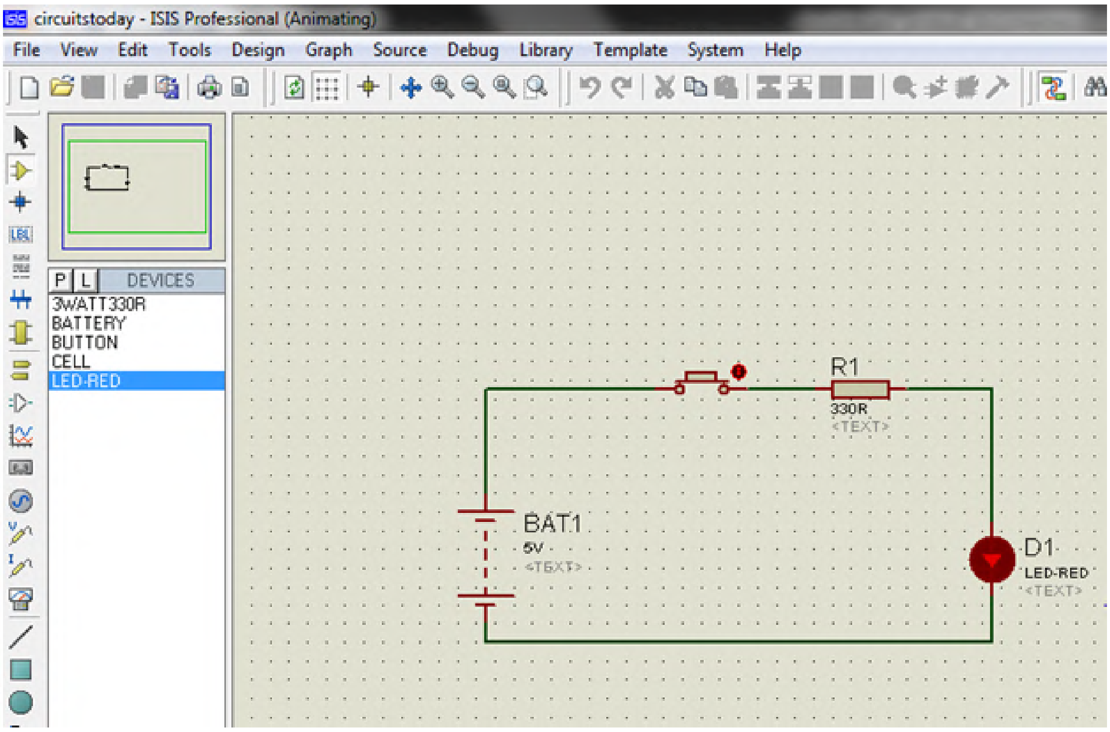


Fig Simulation Animating

Simulation can be stepped, paused or stopped at any time.

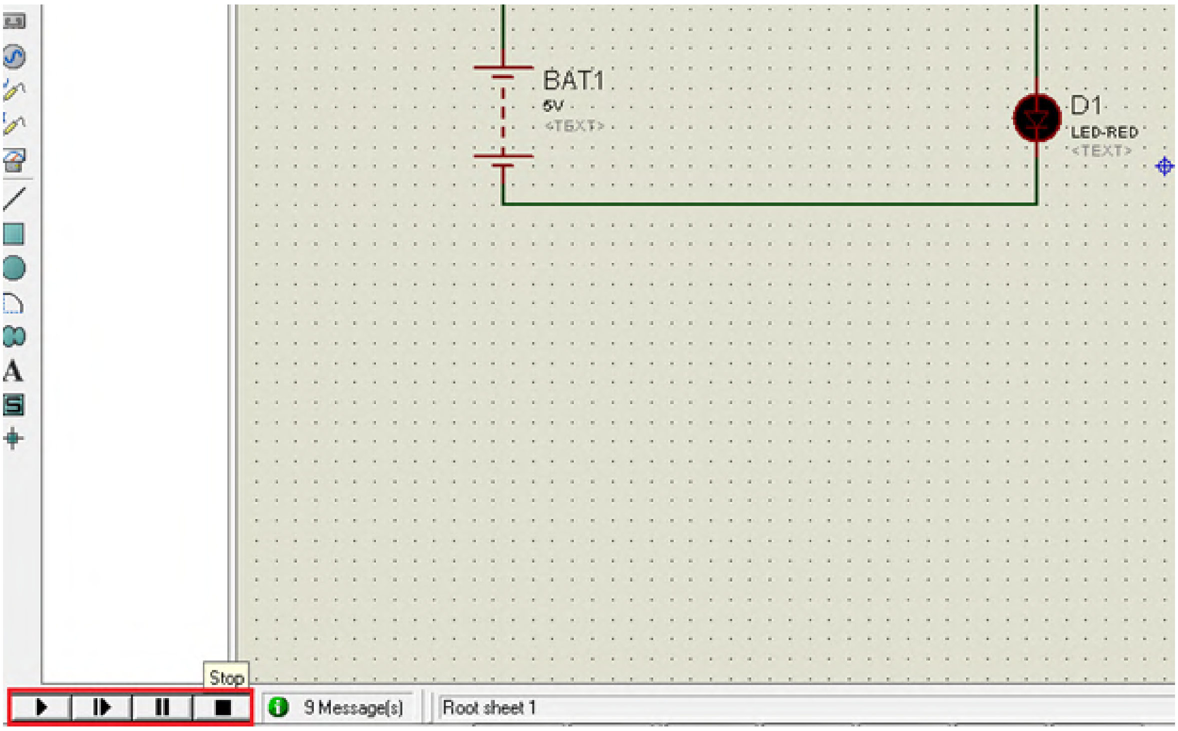
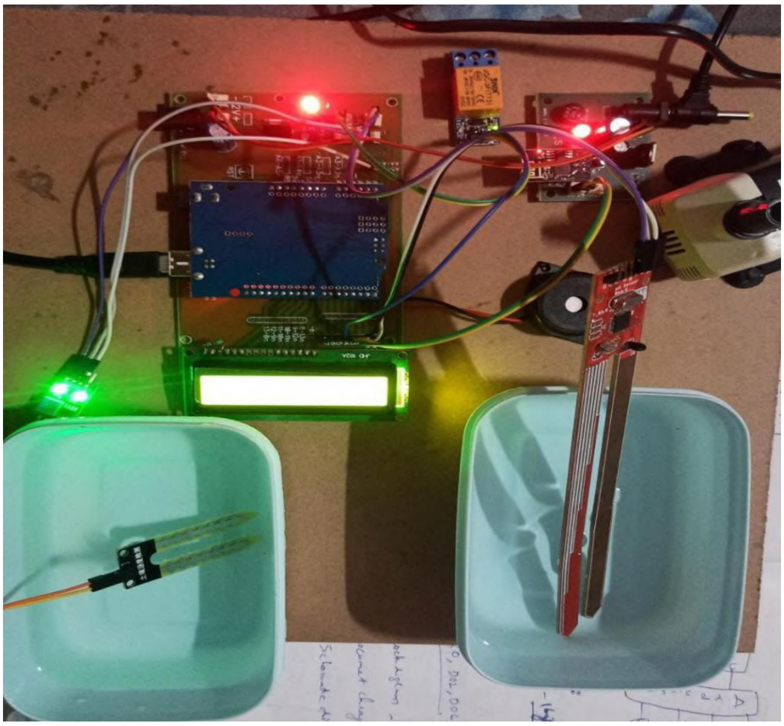


Fig Simulation Step-Pause-Stop Buttons

CHAPTER-7

RESULT & CONCLUSION

7.1 Experimental Result:



Project Code:

```
#include <LiquidCrystal.h>
 LiquidCrystal lcd (13,12,11,10,9,8);
 #include <SoftwareSerial.h>
 SoftwareSerial mySerial (2,3);
  //adc
 int temp = A0, soil = A5, water = A2, relay=4, buz=7;
char res [130];
 //pins

////////////////////
void serialFlush () {
  while (Serial.available() > 0) {
    char t = Serial.read();
  }
```

```
}

void myserialFlush () {
  while (mySerial. Available () > 0) {
    char t = mySerial. Read ();
  }
}

char check (char* ex, int timeout)
{
  int i=0;
  int j = 0, k=0;
  while (1)
  {
    sl:
    if (mySerial. Available () > 0)
    {
      res[i] = mySerial.read();
      if(res[i] == 0x0a || res[i]=='>' || i == 100)
      {
        i++;
        res[i] = 0; break;
      }
      i++;
    }
    j++;
    if (j == 30000)
    {
      k++;
      //Serial.println("kk");
      j = 0;
    }
    if (k > timeout)
    {
```

```
     // Serial.println("timeout");
      return 1;
     }
  }//while 1
  if (! strncmp (ex, res, strlen(ex)))
  {
    //Serial.println("ok...");
    return 0;
   }
  else
  {
   // Serial.print("Wrong ");
   // Serial.println(res);
    i=0;
    go to sl;
   }
}
char buff [200], k=0;
int phvalue=0;
void upload1();
```

Project Code:

```
#include <LiquidCrystal.h>
 LiquidCrystal lcd(13,12,11,10,9,8);
 #include <SoftwareSerial.h>
 SoftwareSerial mySerial(2,3);
  //adc
 int temp = A0,soil = A5,water = A2,relay=4,buz=7;
char res[130];
 //pins

///////////////////
void serialFlush(){
```

```
  while(Serial.available() > 0) {
    char t = Serial.read();
  }
}

void myserialFlush(){
  while(mySerial.available() > 0) {
    char t = mySerial.read();
  }
}

char check(char* ex,int timeout)
{
  int i=0;
  int j = 0,k=0;
  while (1)
  {
    sl:
    if(mySerial.available() > 0)
    {
      res[i] = mySerial.read();
      if(res[i] == 0x0a || res[i]=='>' || i == 100)
      {
        i++;
        res[i] = 0;break;
      }
      i++;
    }
    j++;
    if(j == 30000)
    {
      k++;
      //Serial.println("kk");
      j = 0;
```

```
    }
    if(k > timeout)
    {
     // Serial.println("timeout");
      return 1;
     }
  }//while 1
  if(!strncmp(ex,res,strlen(ex)))
  {
    //Serial.println("ok..");
    return 0;
   }
  else
  {
   // Serial.print("Wrong  ");
   // Serial.println(res);
    i=0;
    goto sl;
   }
}
char buff[200],k=0;
int phvalue=0;
void upload1();


const char* ssid = "project";
const char* password = "project1235";
int T;
int tt;
void setup() {
    int i=0;
  char ret;
    pinMode(temp,INPUT);
    pinMode(soil,INPUT);
```

```
   pinMode(water,INPUT);
   pinMode(buz,OUTPUT);
   pinMode(relay,OUTPUT);
   digitalWrite(relay,LOW);
   digitalWrite(buz,HIGH);



  Serial.begin(9600);
  mySerial.begin(115200);
  lcd.begin(16,2);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("WELCOME");
  delay(3000);
//  serialFlush();
//gsm
st:
  mySerial.println("ATE0");
  Serial.println("ATE0");
  ret  = check((char*)"OK",50);
  mySerial.println("AT");
  Serial.println("AT");
  ret  = check((char*)"OK",50);
  if(ret != 0)
  {
   delay(1000);
   goto st;
  }

  lcd.clear();lcd.setCursor(0, 0);lcd.print("CONNECTING");
  mySerial.println("AT+CWMODE=1");
  Serial.println("AT+CWMODE=1");
   ret  = check((char*)"OK",50);
 cagain:
```

```
  myserialFlush();
  Serial.print("AT+CWJAP=\"");
  mySerial.print("AT+CWJAP=\"");
  mySerial.print(ssid);
  Serial.print(ssid);
  mySerial.print("\",\"");
  Serial.print("\",\"");
  mySerial.print(password);
  Serial.print(password);
  mySerial.println("\"");
  Serial.println("\"");
  if(check((char*)"OK",300))goto cagain;
  mySerial.println("AT+CIPMUX=1");
  Serial.println("AT+CIPMUX=1");
  delay(1000);

 lcd.clear();lcd.setCursor(0, 0);lcd.print("WAITING");
}
void loop() {
  for(int i=1; i<=10; i++) {
        String phdata = Serial.readStringUntil(':');
        Serial.println(phdata);
        if(phdata != ""){
            String ph = Serial.readStringUntil('$');

            Serial.println(ph);
            phvalue=ph.toFloat();
             Serial.println();
            Serial.println("PH Value");
            Serial.println(phvalue);
            delay(1000);}
   }
  int td = analogRead(temp)/4;
  int sd = digitalRead(soil);
```

```
  int wd = analogRead(water);

  Serial.print(td);
  Serial.print("\r\n");
  Serial.print(wd);
  Serial.print("\r\n");
  lcd.setCursor(0, 0);lcd.print("T:");lcd.print(td);lcd.print("    ");
  lcd.setCursor(5, 0);lcd.print("S:");lcd.print(sd);lcd.print("    ");
  lcd.setCursor(10, 0);lcd.print("W:");lcd.print(wd);lcd.print("    ");
  lcd.setCursor(0, 1);lcd.print("PH:");lcd.print(phvalue);lcd.print("    ");
  delay(1000);
//////////////// ldr1 start /////////
/*if(td > 35 )
{
  lcd.clear();
  lcd.setCursor(0, 0);lcd.print("HIGH TEMPRATURE");
  delay(3000);
   digitalWrite(buz,LOW);
  upload1(td,sd,wd,phvalue);
  delay(500);
}*/

/*if(wd > 100 )
{
  lcd.clear();
  lcd.setCursor(0, 0);lcd.print("WATER LEVEL FULL");
  delay(3000);
     digitalWrite(buz,HIGH );

 upload1(td,sd,wd,phvalue);
  delay(500);

}
if(wd < 10 )
```

```
{
  lcd.clear();
  lcd.setCursor(0, 0);lcd.print("WATER LEVEL LOW");
  delay(3000);
  digitalWrite(buz,LOW);
 upload1(td,sd,wd,phvalue);
  delay(500);

}*/
/*if(sd == 0 )
{
  lcd.clear();
  lcd.setCursor(0, 0);lcd.print("WET DETECTED");
  delay(3000);
 upload1(td,sd,wd,phvalue);
  delay(500);

}*/
if(sd == 1 )
{
  lcd.clear();
  lcd.setCursor(0, 0);lcd.print("DRY DETECTED");
  delay(3000);
 digitalWrite(relay,HIGH);
    digitalWrite(buz,LOW);

 upload1(td,sd,wd,phvalue);
  delay(500);

}
else
{
 digitalWrite(relay,LOW);
 delay(4000);
```

```
 digitalWrite(buz,HIGH);

upload1(td,sd,wd,phvalue);

}

}//loop


char bf2[100];
void upload1(unsigned char *chr ,unsigned char *chr1,unsigned char *chr2,unsigned char
*chr3)
{
  delay(2000);
  lcd.clear();lcd.setCursor(0, 1);lcd.print("UPLOADING");
  myserialFlush();
  mySerial.println("AT+CIPSTART=4,\"TCP\",\"api.thingspeak.com\",80");
 // Serial.println("AT+CIPSTART=4,\"TCP\",\"embeddedspot.top\",80");
      delay(8000);
      sprintf(buff,"GET
https://api.thingspeak.com/update?api_key=ZHR6DIPXM6G1SBYN&field1=0=%u&field2=
%u\r\n\r\n",chr,chr1,chr2,chr3);
      //sprintf(buff,"GET
http://embeddedspot.top/iot/storedata.php?name=sensors010&s1=%u\r\n\r\n",chr);
      myserialFlush();
      sprintf(bf2,"AT+CIPSEND=4,%u",strlen(buff));
      mySerial.println(bf2);

         delay(5000);


          myserialFlush();
          mySerial.print(buff);
          Serial.print(buff);
```

```
        delay(2000);

        mySerial.println("AT+CIPCLOSE");
        Serial.println("AT+CIPCLOSE");
        lcd.setCursor(0, 1);lcd.print("UPLOADDED");   lcd.clear();

}
        lcd.setCursor(0, 1);lcd.print("UPLOADDED");   lcd.clear();

}
```

7.2 Conclusion:

In present days especially farmers are facing major problems in watering their agriculture fields, it’s because they have no proper idea about when the power is available so that they can pump water. Even after then they need to wait until the field is properly watered, which makes them to stop doing other activities. Here is an idea which helps not only formers even for watering the gardens also, which senses the soil moisture and switches the pump automatically when the power is ON.

7.3 Further Enhancement:

The working of project is basically dependent on the output of the humidity sensor. Whenever there is need of excess water in the desired field (RICE crops) then it will not be possible by using sensor technology. By using this we will be able to irrigate the desired fields and in desired amount.

REFERENCE

1. Nagare Brushalim, M, M. Agarkar, B.S. and. Kapale, N.B. (2015). Automated irrigation system using WSN and GPRS/GSM, International Journal of Advance Research and Innovative Ideas in Education, Vol (1), 83-92.

2. Devika, S.V, Khamuruddeen, S.K. Jayanth Thota and Khalesha, Shaik. (2014). Arduino based automatic plant watering system. International Journal of Advanced Research in Computer Science and Software Engineering

3. Sanjay Kumawat, Mayur Bhamare, Apurva Nagare and Ashwini Kapadnis. (2017). Sensor based automatic irrigation system and soil pH. detection using image processing. International Research Journal of Engineering and Technology

4. Babanna, Kumbar, Basavaraj Galagi, Bheema Shankar and Naveen Honnalli. (2016). Smart irrigation system using internet of things. Bonfring International Journal of Research and Communication Engineering, Vol. (6), 4-9.

5. Abhinav Rajpal, Sumit Jain, Nistha Khare and Anil Kumar Shukla "Micro controlled -based Automatic Irrigation System with Moisture Sensor", ICSE, 2011.

6. S R Kumbar, Arjun P Ghatule "Microcontroller based Controlled Irrigation System for Plantation", International Multiconference of Engineers and Computer Scientists volume II, March 2013.

7. Pavithra D S, M S Srinath, "GSM based Automatic Irrigation control System for Efficient use of Resources and Crop Planning by Using an Android Mobile", IOSTR Journal of mechanical and civil engineering, volume II, July 2014

8. Sonali D Gainwar, Dinesh V Rojatkar, "Soil Parameter Monitoring with Auto- -matic Irrigation System", International Journal of Science, Engineering and technology Research, volume IV, Issue 11, November 2015.

A Major Project report on

Performance Evaluation For 5g Nr Based Uplink Millimeter Wave MIMO System Under Urban MicroCell

Submitted By

Kevin Martin	**(17K81A04F0)**
Shaik Nazar Basha	**(17K81A04H1)**
Suhas Sunil Shelar	**(17K81A04H3)**

In partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Mr. M Govind Raj MTech

Assistant Professor

Department of Electronics and Communications

ST.MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

Certified that this project reported title" **"Performance Evaluation for 5G NR based Uplink Millimeter wave MIMO Systems under Urban Micro Cell"** is being submitted by **1. Kevin Martin (17K81A04F0), 2. Shaik Nazar Basha (17K81A04H1), 3. Suhas Sunil Shelar (17K81A04H3).** Who carried out the work under my supervision, for the partial fulfilment of the requirements for the award of the degree of *Bachelor of Technology, Electronics and Communication Engineering.* Certified further that to the best of my knowledge and belief, the work reported here in does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion.

INTERNAL GUIDE	HEAD OF DEPARTMENT
Mr. M. Govind Raj	Dr. B. Hari Krishna, Professor
Assistant Professor (M.Tech)	Department of ECE
Department of ECE	

EXTERNAL EXAMINER

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT, KEVIN MARTIN (17K81A04F0), SHAIK NAZAR BASHA (17K81A04H1), SUHAS SUNIL SHELAR (17K81A04H3) OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "Performance Evaluation for 5G NR based Uplink Millimeter wave MIMO Systems under Urban Micro Cell" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. ★ Kompally-500014 ★

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

DECLARATION

We declare that this project report titled Performance Evaluation for 5G NR based Uplink Millimeter wave MIMO Systems under Urban Micro Cell submitted in partial fulfillment of the degree of B. Tech in Electronics And Communication Engineering record of original work carried out by us under the guidance and supervision of Mr. M. Govind Raj and has not formed the name basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

1) Mr. Kevin Martin	2) Mr. Shaik Nazar Basha	3) Mr. Suhas Sunil Shelar
(17K81A04F0)	(17K81A04H1)	(17K85A04H3)

Place:

Date:

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P. Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna**, **Head &Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come

We would like to express our gratitude to **our dept. project coordinator** Mr.M.Govind Raj for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

We dedicate this thesis to almighty

1. Mr. Kevin Martin (17K81A04F0)
2. Mr. Shaik Nazar Basha (17K81A04H1)
3. Mr. Suhas Sunil Shelar (17K85A04H3)

ABSTRACT

In next generation mobile communication environment, a huge number of smart devices will require a large amount of data traffic. To accommodate these, 5G mobile communication systems using millimeter-wave bands are being studied, and the new radio access is standardized by 3rd Generation Partnership Project (3GPP). This paper investigates a 5G NR based uplink millimeter-wave massive multiple-input multiple-output (MIMO) system. The spectral efficiency is analyzed in a 3GPP spatial channel model scenario, urban micro cell.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

With the advent of the Internet of Things, the number of smart devices requiring wireless communication has exploded. They require a large amount of data traffic. To accommodate this, 5G mobile communication systems are being studied in many countries around the world. 5G mobile telecommunication businesses aim to increase frequency efficiency, increase density of user's connection, decrease delay speed, and increase network capacity more than 4G. To meet the requirements of 5G mobile communication, 3rd Generation Partnership Project (3GPP) standardized the New Radio Access (NR) that supports millimetre-wave bands above 6 GHz, in December 2017 [9]. Millimetre based mobile telecommunication technology is being studied as a key 5G technology to accommodate the growing traffic of mobile telecommunication.

Design a 5G NR transmission frame and the uplink millimetre-wave massive multiple-input multiple output (MIMO) system based on 5G NR considering multi users in the urban micro cell scenario. In addition, a spatial multiplexing technique is applied to support streams with high spectral efficiency, and the performance of the system is evaluated through spectral efficiency analysis.. we describe the millimeter-wave channel model. The 5G NR based uplink millimeter-wave massive MIMO system is described .

Multiple-Input Multiple-Output (MIMO) technology is a wireless technology that uses multiple transmitters and receivers to transfer more data at the same time. MIMO technology takes advantage of a radio-wave phenomenon called multipath where transmitted information bounces off walls, ceilings, and other objects, reaching the receiving antenna multiple times via different angles and at slightly different times

Figure 1.1 : MIMO Technology uses multiple radios to transfer more data at the same time

MIMO, is a radio communications technology or RF technology that is being mentioned and used in many new technologies these days. Wi-Fi, LTE; Long Term Evolution, and many other radio, wireless and RF technologies are using the new MIMO wireless technology to provide increased link capacity and spectral efficiency combined with improved link reliability using what were previously seen as interference paths.

Even now many there are many MIMO wireless routers on the market, and as this RF technology is becoming more widespread, more MIMO routers and other items of wireless MIMO equipment will be seen.

MIMO technology leverages multipath behavior by using multiple, “smart” transmitters and receivers with an added “spatial” dimension to dramatically increase performance and range. MIMO allows multiple antennas to send and receive multiple spatial streams at the same time.

MIMO makes antennas work smarter by enabling them to combine data streams arriving from different paths and at different times to effectively increase receiver signal-capturing power. Smart antennas use spatial diversity technology, which puts surplus antennas to good use. If there are more antennas than spatial streams, the additional antennas can add receiver diversity and increase range.

MIMO -Multiple Input Multiple Output basics:

As a result of the use of multiple antennas, MIMO wireless technology is able to considerably increase the capacity of a given channel. By increasing the number of receive and transmit antennas it is possible to linearly increase the throughput of the channel with every pair of antennas added to the system. This makes MIMO wireless technology one of the most important wireless techniques to be employed in recent years. As spectral bandwidth is becoming an ever more valuable commodity for radio communications systems, techniques are needed to use the available bandwidth more effectively. MIMO wireless technology is one of these techniques. MIMO is effectively a radio antenna technology as it uses multiple antennas at the transmitter and receiver to enable a variety of signal paths to carry the data, choosing separate paths for each antenna to enable multiple signal paths to be used.

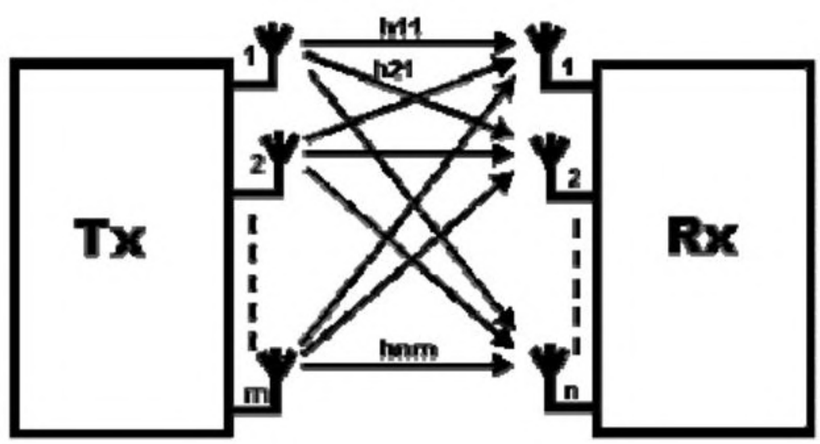


Figure 1.2 : MIMO Block Diagram

One of the core ideas behind MIMO wireless systems space-time signal processing in which time is complemented with the spatial dimension inherent in the use of multiple spatially distributed antennas, i.e. the use of multiple antennas located at different points. Accordingly MIMO wireless systems can be viewed as a logical extension to the smart antennas that have been used for many years to improve wireless. It is found between a transmitter and a receiver, the signal can take many paths. Additionally by moving the antennas even a small distance the paths used will change. The variety of paths available occurs as a result of the number of objects that appear to the side or even in the direct path between the transmitter and receiver. Previously these multiple paths only served to introduce interference. By using MIMO, these additional paths can be used to advantage. They can be used to provide additional robustness to the radio link by improving the signal to noise ratio, or by increasing the link data capacity.

One of the key advantages of MIMO spatial multiplexing is the fact that it is able to provide additional data capacity. MIMO spatial multiplexing achieves this by utilising the multiple paths and effectively using them as additional "channels" to carry data. The maximum amount of data that can be carried by a radio channel is limited by the physical boundaries defined under Shannon's Law. Multiple-input, multiple-output (MIMO) antenna systems are used in modern wireless standards, including in IEEE 802.11n, 3GPP LTE, and mobile WiMAX systems. The technique supports enhanced data throughput even under conditions of interference, signal fading, and multipath. The demand for higher data rates over longer distances has been one of the primary motivations behind the development of MIMO orthogonal- frequency-division-multiplexing (OFDM) communications systems. Shannon's law defines the maximum rate at which error free data can be transmitted over a given bandwidth in the presence of noise. It is usually expressed in the form:

$$\text{Capacity} = BW \log2(1 + SNR) \text{ ---- Eq. 1}$$

Where C is the channel capacity in bits per second, BW is the bandwidth in Hertz, and SNR is Signal to Noise Ratio.

The above Eq.1 shows, an increase in a channel's SNR results in marginal gains in channel throughput. As a result, the traditional way to achieve higher data rates is by increasing the signal bandwidth. Unfortunately, increasing the signal bandwidth of a communications channel by increasing the symbol rate of a modulated carrier increases its susceptibility to multipath fading. For wide bandwidth channels, one partial solution to solving the multipath challenge is to use a series of narrowband overlapping subcarriers.

Not only does the use of overlapping OFDM subcarriers improve spectral efficiency, but the lower symbol rates used by narrowband subcarriers reduces the impact of multipath signal products. MIMO communications channels provide an interesting solution to the multipath challenge by requiring multiple signal paths. In effect, MIMO systems use a combination of multiple antennas and multiple signal paths to gain knowledge of the communications channel. By using the spatial dimension of a communications link, MIMO systems can achieve significantly higher data rates than traditional single-input, single-output (SISO) channels.

In a 2 x 2 MIMO system, signals propagate along multiple paths from the transmitter to the receiver antennas. Using this channel knowledge, a receiver can recover independent streams from each of the transmitter's antennas. A 2 x 2 MIMO system produces two spatial streams to effectively double the maximum data rate of what might be achieved in a traditional 1 x 1 SISO communications channel. The maximum channel capacity of a MIMO system, the channel capacity can be estimated as a function of N spatial streams. A basic approximation of MIMO channel capacity is a function of spatial streams, bandwidth, and signal-to-noise ratio (SNR) and is shown in the following Eq. :

$$\text{Capacity} = N\ BW \log_2 (1 + SNR) \text{ --- Eq. 2}$$

Given the equation for MIMO channel capacity, it is possible to investigate the relationship between the number of spatial streams and the throughput of various implementations of SISO and MIMO configurations. As an example, the IEEE 802.11g specs prescribe that a wireless-local-area network (WLAN) channel uses a SISO configuration. With this standard, the maximum coded data rate of 54 Mb/s requires use of a 64-QAM modulation scheme and a code rate of 3/4. As a result, the un-coded bit rate is 72 Mb/s (4/3 x 54 Mb/s). With minimum transmitter error vector magnitude (EVM) at -25 dB, an SNR of 25 dB can be estimated as the requirement for a 64-state quadrature amplitude- modulation (64QAM) scheme. While EVM and SNR are not equivalent in all cases, we can assume that the magnitude error of a symbol will dominate the signal error as the SNR approaches its lower limit. The maximum data rate of IEEE 802.11g maps closely with the maximum channel capacity dictated by the Shannon- Hartley theorem. According to this theorem, a Gaussian channel with an SNR of 25 dB should produce an un coded data rate of 94 Mb/s in a 20-MHz channel bandwidth. By contrast, Eq. 2 would suggest that a MIMO channel with four spatial streams should be capable of four times the capacity of the SISO channel. 20-MHz channel with a signal-to-noise ratio (SNR) of 25 dB and four spatial streams should have an un coded bit rate of 4 x 94 Mb/s = 376 Mb/s.

This estimation maps closely with the expected data rates of the draft IEEE 802.11n physical layer specs. IEEE 802.11n is designed to support MIMO configurations with as many as four spatial streams. At the highest data rate, bursts using a 64QAM modulation scheme with a 5/6 channel code rate produce a data rate of 288.9Mb/s and an uncoded bit rate of 346.68 Mb/s. At the highest data rate, the IEEE 802.11n channel with four spatial streams produces a data rate that is comparable to the theoretical limit of 376 Mb/s. It can be observed that the bit rate of a 4 x 4 (four spatial stream) MIMO configuration exceeds that of the Shannon- Hartley limit at all data rates, making MIMO systems attractive for higher data throughput. While MIMO systems provide users with clear benefits at the application level, the design and test of MIMO devices is not without significant challenges. MIMO systems are a natural extension of developments in antenna array communication. While the advantages of multiple receive antennas, such as gain and spatial diversity, have been known and exploited for some time the use of transmit diversity has only been investigated recently .

The advantages of MIMO communication, which exploits the physical channel between many transmit and receive antennas, are currently receiving significant attention. While the channel can be so nonstationary that it cannot be estimated in any useful sense , in this article we assume the channel is quasi static. MIMO systems provide a number of advantages over single-antenna-to-single-antenna communication. Sensitivity to fading is reduced by the spatial diversity provided by multiple spatial paths. Under certain environmental conditions, the power requirements associated with high spectral-efficiency communication can be significantly reduced by avoiding the compressive region of the information-theoretic capacity bound. Here, spectral efficiency is defined as the total number of information bits per second per Hertz transmitted from one array to the other.

After an introductory section, we describe the concept of MIMO information-theoretic capacity bounds. Because the phenomenology of the channel is important for capacity, we discuss this phenomenology and associated parameterization techniques, followed by examples of space-time codes and their respective receivers and decoders. We performed experiments to investigate channel phenomenology and to test coding and receiver techniques. Capacity We discuss MIMO information-theoretic performance bounds in more detail in the next section. Capacity increases linearly with signal-to-noise ratio (SNR) at low SNR, but increases logarithmically with SNR at high SNR. In a MIMO system, a given total transmit power can be divided among multiple spatial paths (or modes), driving the capacity closer to the linear regime for each mode, thus increasing the aggregate spectral efficiency. As seen in Figure 1, which assumes an optimal high spectral-efficiency MIMO channel (a channel matrix with a flat singular-value distribution), MIMO systems enable high spectral efficiency at much lower required energy per information bit

A channel may be affected by fading and this will impact the signal to noise ratio. In turn this will impact the error rate, assuming digital data is being transmitted. The principle of diversity is to provide the receiver with multiple versions of the same signal. If these can be made to be affected in different ways by the signal path, the probability that they will all be affected at the same time is consid erably reduced. Accordingly, diversity helps to stabilise a link and improves performance, reducing error rate.

Several different diversity modes are available and provide a number of advantages:

- **Time diversity:** Using time diversity, a message may be transmitted at different times, e.g. using different timeslots and channel coding.
- **Frequency diversity:** This form of diversity uses different frequencies. It may be in the form of using different channels, or technologies such as spread spectrum / OFDM.
- **Space diversity:** Space diversity used in the broadest sense of the definition is used as the basis for MIMO. It uses antennas located in different positions to take advantage of the different radio paths that exist in a typical terrestrial environment.

MIMO is effectively a radio antenna technology as it uses multiple antennas at the transmitter and receiver to enable a variety of signal paths to carry the data, choosing separate paths for each antenna to enable multiple signal paths to be used.

One of the core ideas behind MIMO wireless systems space-time signal processing in which time (the natural dimension of digital communication data) is complemented with the spatial dimension inherent in the use of multiple spatially distributed antennas, i.e. the use of multiple antennas located at different points. Accordingly MIMO wireless systems can be viewed as a logical extension to the smart antennas that have been used for many years to improve wireless.

It is found between a transmitter and a receiver, the signal can take many paths. Additionally by moving the antennas even a small distance the paths used will change. The variety of paths available occurs as a result of the number of objects that appear to the side or even in the direct path between the transmitter and receiver. Previously these multiple paths only served to introduce interference. By using MIMO, these additional paths can be used to advantage. They can be used to provide additional robustness to the radio link by improving the signal to noise ratio, or by increasing the link data capacity.

The two main formats for MIMO are given below:

- **Spatial diversity:** Spatial diversity used in this narrower sense often refers to transmit and receive diversity. These two methodologies are used to provide improvements in the signal to noise ratio and they are characterized by improving the reliability of the system with respect to the various forms of fading.
- **Spatial multiplexing:** This form of MIMO is used to provide additional data capacity by utilizing the different paths to carry additional traffic, i.e. increasing the data throughput capability.

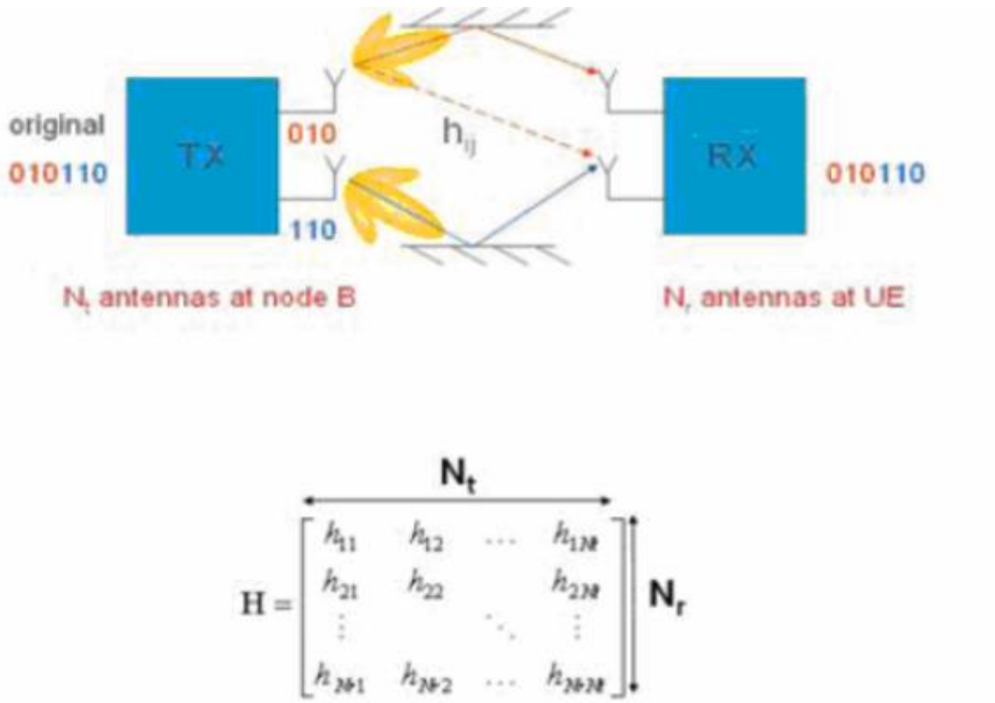


Figure 1.3 : Spatial multiplexing

Each transmit antenna transmits a different data stream. Each receive antenna may receive the data streams from all transmit antennas. The channel (for a specific delay) can thus be described by the following channel matrix H: As above figure. In this general description, Nt is the number of transmit antennas, Nr is the number of receive antennas, resulting in a 2x2 matrix for the baseline LTE scenario. The coefficients hij of this matrix are called channel coefficients from transmit antenna i to receive antenna j, thus describing all possible paths between transmitter and receiver side. The number of data streams that can be transmitted in parallel over the MIMO channel is given by min {Nt, Nr} and is limited by the rank of the matrix H. The transmission quality degrades significantly in case the singular values of matrix H are not sufficiently strong.

This can happen in case the 2 antennas are not sufficiently de-correlated, for example in an environment with little scattering or when antennas are too closely spaced. In LTE, up to 2 code words can be mapped onto different so-called layers. The number of layers for transmission is equal to the rank of the matrix H. There is a fixed mapping between code words to layers.

Figure 1.4 below describes how precoding on transmitter side is used to support spatial multiplexing. This is achieved by applying a precoding matrix W to the signal before transmission.

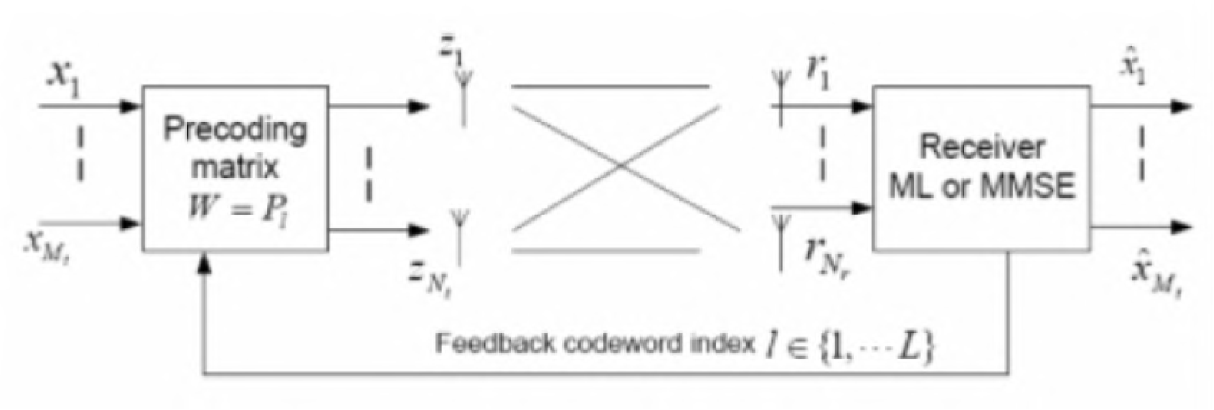


Figure 1.4: Pre-coding principle

As a result of the use multiple antennas, MIMO wireless technology is able to considerably increase the capacity of a given channel while still obeying Shannon's law. By increasing the number of receive and transmit antennas it is possible to linearly increase the throughput of the channel with every pair of antennas added to the system. This makes MIMO wireless technology one of the most important wireless techniques to be employed in recent years. As spectral bandwidth is becoming an ever more valuable commodity for radio communications systems, techniques are needed to use the available bandwidth more effectively. MIMO wireless technology is one of these techniques.

1.2 Benefits Of MIMO Technology

(1) Multiple antenna configurations can be used to overcome the detrimental effects of multi-path and fading when trying to achieve high data throughput in limited-bandwidth channels.

Multiple-input, multiple-output (MIMO) antenna systems are used in modern wireless standards, including in IEEE 802.11n, 3GPP LTE, and mobile WiMAX systems. The technique supports enhanced data throughput even under conditions of interference, multi-path and fading. The demand for higher data rates over longer distances has been one of the primary motivations behind the development of MIMO orthogonal- frequency-division-multiplexing (OFDM) communications systems.

(2) Superior Data Rates, Range and Reliability

Systems with multiple antennas at the transmitter and receiver – also referred to as Multiple Input Multiple Output (MIMO) systems – offer superior data rates, range and reliability without requiring additional bandwidth or transmit power. By using several antennas at both the transmitter and receiver, MIMO systems create multiple independent channels for sending multiple data streams.

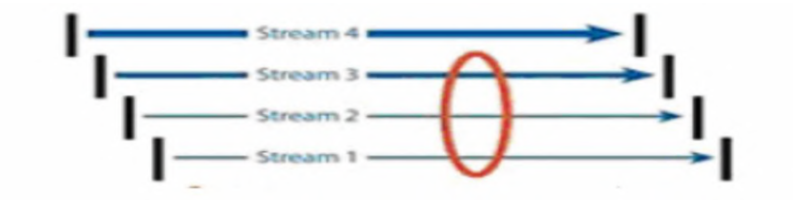


Figure 1.5: Stream combining for enhanced reliability

4x4 MIMO system supports up to four independent data streams. These streams can be combined through dynamic digital beamforming and MIMO receiver processing (in the red oval) to increase reliability and range. The number of independent channels and associated data streams that can be supported over a MIMO channel is equivalent to the minimum number of

antennas at the transmitter or receiver. Thus, a 2x2 system can support at most two streams, a 3x3 system can support three streams and a 4x4 system can support four streams. Some of the independent streams can be combined through dynamic digital beamforming and MIMO receiver processing, as shown in the red oval, which results in increased reliability and range.

LTE MIMO Concepts

MIMO systems form an essential part of LTE in order to achieve the ambitious requirements for throughput and spectral efficiency. MIMO refers to the use of multiple antennas at transmitter and receiver side.

Downlink MIMO

For the LTE downlink, a 2x2 configuration for MIMO is assumed as baseline configuration, i.e. 2 transmit antennas at the base station and 2 receive antennas at the terminal side. Configurations with 4 antennas are also being considered. Different MIMO modes are envisaged. It has to be differentiated between spatial multiplexing and transmit diversity, and it depends on the channel condition which scheme to select. The optimum pre-coding matrix W is selected from a predefined "codebook" which is known at eNodeB and UE side. Unitary pre-coding is used, i.e. the precoding matrices are unitary: WHW= I. The UE estimates the radio channel and selects the optimum pre-coding matrix. The optimum pre-coding matrix is the one which offers maximum capacity. The UE provides feedback on the uplink control channel regarding the preferred pre-coding matrix (pre-coding vector as a special case). Ideally, this information is made available per resource block or at least group of resource blocks, since the optimum pre-coding matrix varies between resource blocks.

Transmit Diversity:

Instead of increasing data rate or capacity, MIMO can be used to exploit diversity. Transmit diversity schemes are already known from WCDMA release 99 and will also form part of LTE as one MIMO mode. In case the channel conditions do not allow spatial multiplexing, a transmit diversity scheme will be used instead, so switching between these two MIMO modes is possible depending on channel conditions. Transmit diversity is used when the selected number of streams (rank) is one.

Uplink MIMO:

Uplink MIMO schemes for LTE will differ from downlink MIMO schemes to take into account terminal complexity issues. For the uplink, MU-MIMO can be used. Multiple user terminals may transmit simultaneously on the same resource block. This is also referred to as spatial domain multiple access (SDMA). The scheme requires only one transmit antenna at UE side which is a big advantage. The UEs sharing the same resource block have to apply mutually orthogonal pilot patterns.

To exploit the benefit of two or more transmit antennas but still keep the UE cost low, antenna subset selection can be used. In the beginning, this technique will be used, e.g. a UE will have two transmit antennas but only one transmit chain and amplifier. A switch will then choose the antenna that provides the best channel to the eNodeB. Multiple-input multiple-output, or MIMO, is a radio communications technology or RF technology that is being mentioned and used in many new technologies these days. Wi-Fi, LTE (3G long term evolution) and many other radio, wireless and RF technologies are using the new MIMO wireless technology to provide increased link capacity and spectral efficiency combined with improved link reliability using what were previously seen as interference paths.

The multiple input multiple output channel technology is aimed to increase the capacity in the wireless communication network. With the invention of MIMO, the technology seems to gain popularity as it is being implemented in the current commercial wireless products and networks such as broadband wireless access systems, wireless local area networks (WLAN), 3G networks, etc. The main idea behind MIMO is that, the sampled signals in spatial domain at both the transmitter and receiver end are combined so that they form effective multiple parallel spatial data streams which increase the data rate. The occurrence of diversity also improves the quality that is the bit-error rate (BER) of the communication.

Single Input Single Output versus MIMO channel capacity In communication systems, input discrete source symbols are mapped into a sequence of channel symbols which are then transmitted through the wireless channel. The transmission of channel symbols through the wireless channel is by nature random and random noise is added to the channel symbols. The measure of how much information that can be transmitted and received with minimum probability of error is called the channel capacity. A Single Input Single Output system involves the use of one antenna both at the transmitter and receiver end. To a telecommunications engineer, there exit a limit at which reliable transmission of information is not possible for a given transmission bandwidth and power. These limits where discovered by Claude Shannon in 1948 when he established the principles of information and communication theory on his various publications. Shannon also established the conditions that enable the transmission of information over a noisy channel at a given rate, for a given power of the signal and noise. These limiting factors are the finite bandwidth and the S/N of the channel.

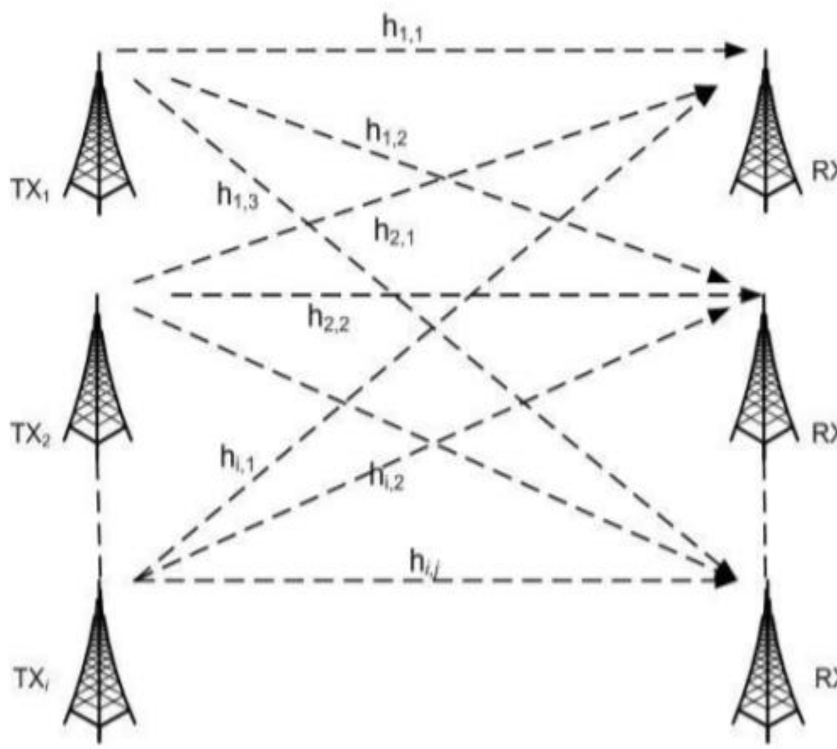


Figure 1.6 : A generalized MIMO wireless communication system.

This is because for a communication channel to accommodate the signal spectrum, enough transmission bandwidth is needed otherwise there will be distortion. ”The higher data rate is to be transmitted, the shorter digital pulses must be used and the shorter digital pulses are used for transmission, the wider bandwidth is required”. For a deterministic channel with a bandwidth (B) with additive noise, Shannon proved that information with a rate of r bits per second (bps) can be transmitted with a small error probability provided that the bit rate is less than the capacity of the channel $r < C$ where S/N, the signal-to-noise ratio and B the bandwidth of the transmission channel. Assuming we have a channel with additive noise N and that we have some freedom of choosing the average transmission power S, to set up a reliable transmission link to send r bits per second. From the Shannon theorem, the data rate r cannot exceed capacity C, $r < C$, but we still have one degree of freedom in the choice of bandwidth B and power S. It can be realized that, for a given signal-to-noise ratio S/N, if we wish to double C, we have to double the bandwidth B. On the other hand, if we double C, for a given B we have to evaluate the S/N.

Representation of MIMO channel :

A 2 x 2 MIMO channel model The first channel model to be considered in this project will be a 2 x 2 MIMO system that is a system with 2 transmits (TX) and 2 receive (RX) antennas where different independent data streams are transmitted from multiple antennas to multiple receive antennas. This channel model will be extended to a 3 x 3 MIMO system and even more to illustrate the channel characteristics in relation to the increase in the number of antennas. The signals considered in the MIMO systems of this project are baseband signals ignoring modulation processes and concentrating on the up and down frequency conversion. Therefore the signals on the i-th transmit antenna will be denoted x□ while the received signal on the j-th receive antenna denoted as y. Figure 2 shows the antenna set-up and the various unknown channel coefficients.

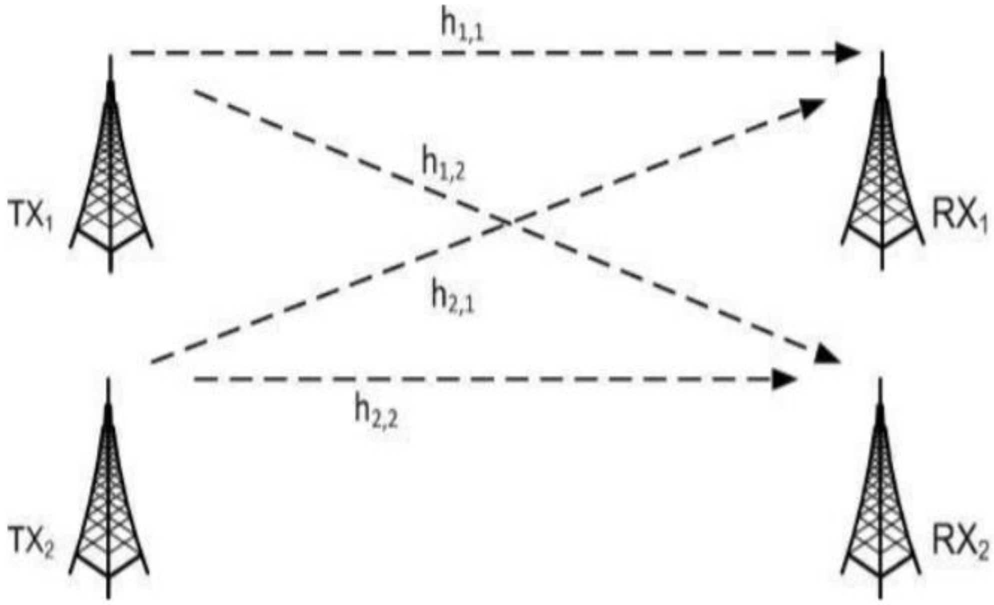


Figure 1.7 : Channel characteristic of a 2 x 2 MIMO wireless communication system.

Since the coefficient of the unknown in the channel matrix Wc and the number of transmitted signal X is equal to the number of received signal Y, the equation can be solved if the channel Wc is inversed which in this case a 2 x 2 matrix inversion.

Channel estimation procedure

Channel characteristics estimation In order to estimate the channel characteristics, we expand each transmitted and received signals in time and write into signal matrix columns K discrete samples in time. The signals matrixes get K rows and as many columns as we have antennas, N or M. In general a MIMO system involves multiple antennas at the transmitting end and multiple receivers at the receiving end. Figure 4 show a general representation of a 3 x 3 general MIMO system.

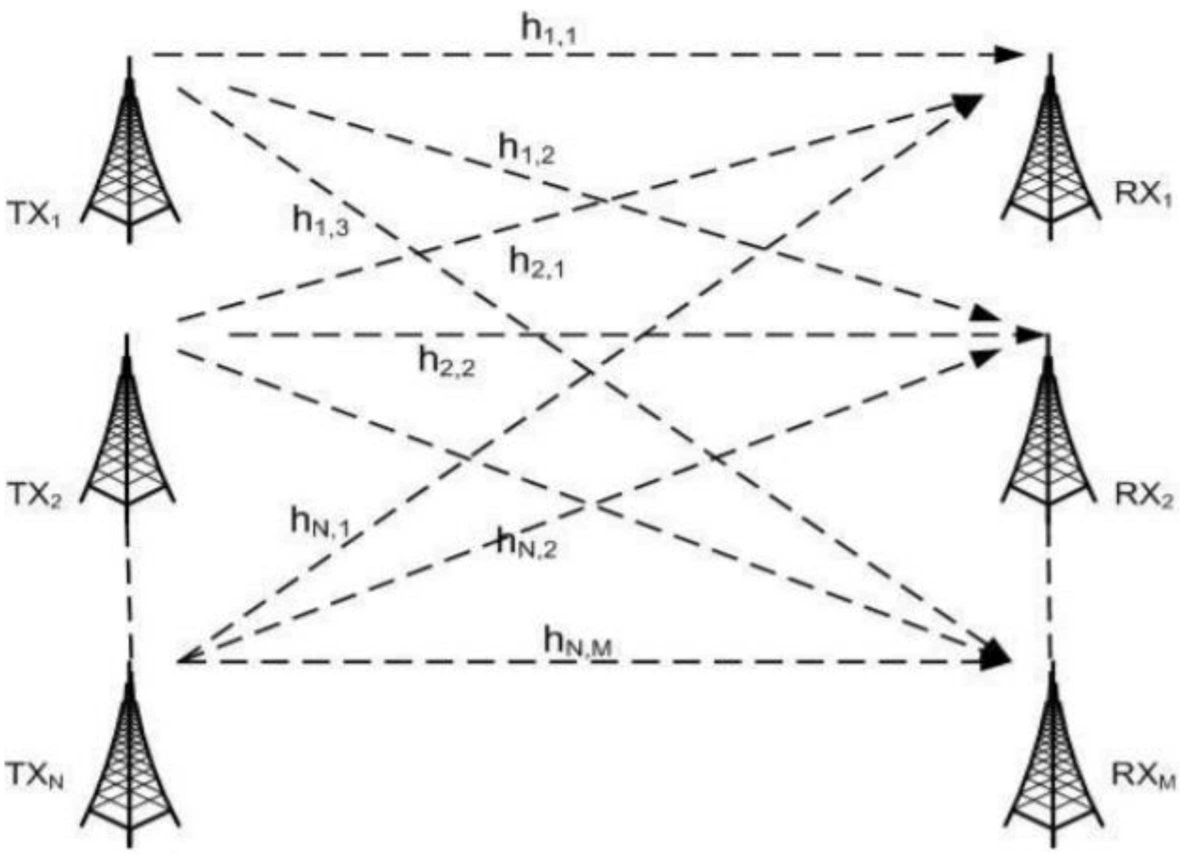


Figure1.8 : General 3 x 3 MIMO system with unknown channel characteristics.

In order to determine the characteristic of the channel, both the transmitted signal X and the received signal Y have to be known. If the transmitted and received signals are of the form

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_N] = \begin{pmatrix} x_{1,1} & \cdots & x_{1,N} \\ \vdots & \ddots & \vdots \\ x_{K,1} & \cdots & x_{K,N} \end{pmatrix}$$

$$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots \mathbf{y}_M] = \begin{pmatrix} y_{1,1} & \cdots & y_{1,M} \\ \vdots & \ddots & \vdots \\ y_{K,1} & \cdots & y_{K,M} \end{pmatrix}$$

If the channel transfer matrix Wc can be determine, then it means the transmitted signals can also be determine because the received signal Y is known. We may write the expression between these vector signals as

$$Y = X\ Wc$$

Orthonormal space concept

Orthonormal basis of transmitted signal:

Orthonormal basis is a coordination system where we can present as many dimensions as is the maximum number of antennas of the transmitted and received signals. The main reason why we need the orthonormal base is to have a common coordinated system in order to combine the transmitted signal X and the received signal Y. The number of dimension of the orthonormal base matrix depends on the maximum number of antennas at both the transmitted and the receiver end of the MIMO system. It does not actually matter what kind of signal (X or Y) used in generating the orthonormal space but it is most convenient to use the transmitted training signal X which is defined to contain linearly independent column signals. Suppose K = 3 time samples per transmitted signal X and per received signal Y respectively of a 3 x 3 channel transmission system were observed. Figure 6 shows a Line of Sight (LOS) MIMO system made up of three transmitters and three receivers.

In figure, N = 3 transmitted signals and M = 3 received signals, the format of the unknown channel transfer matrix Ŏc would be 3 x 3. Assume that all signals are real valued and the transmitted signals and received signals are observed as signal matrices. Then we need the orthonormal basis matrix Ux to solve for an estimated channel transfer matrix Ŏc. It is easier to calculate the orthonormal base matrix Ux of a square matrix; therefore in the next section we estimate the channel model by calculating first the orthonormal basis matrix Ux of a simple 2 x 2 MIMO system and then proceed to a more complex 3 x 3 MIMO system.

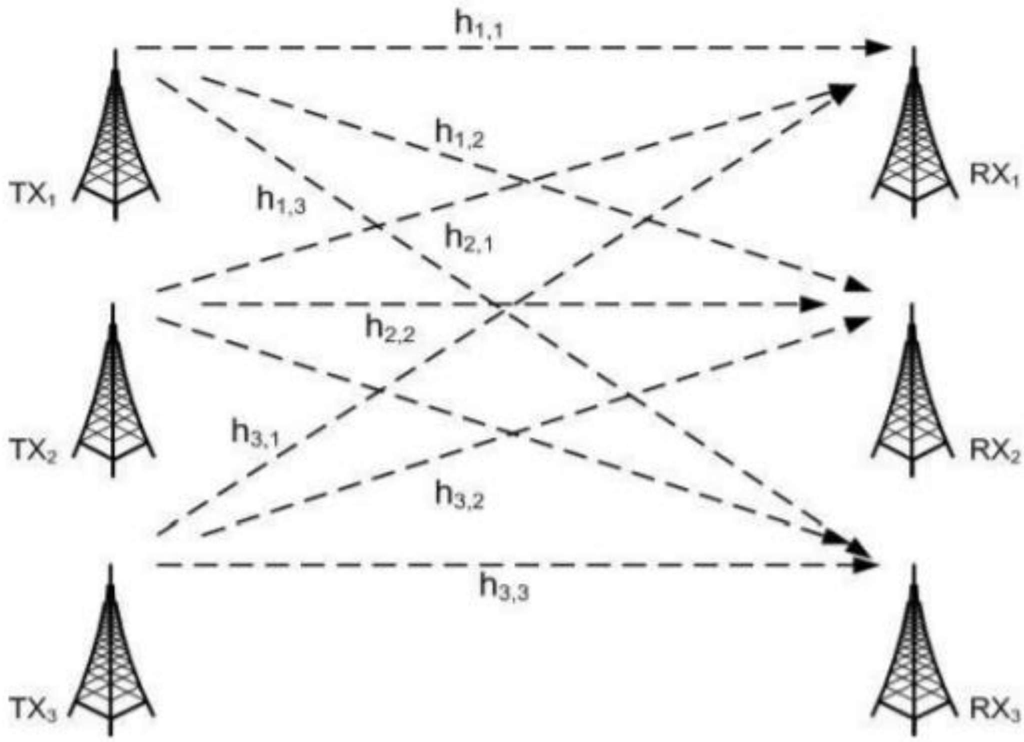


Figure 1.9 : 3 x 3 MIMO wireless communication system.

This is a form of antenna diversity, which uses multiple antennas to improve signal quality and strength of an RF link. The data is split into multiple data streams at the transmission point and recombined on the receive side by another MIMO radio configured with the same number of antennas. The receiver is designed to take into account the slight time difference between receptions of each signal, any additional noise or interference, and even lost signals. By transmitting the same data on multiple streams, the MIMO radios introduce redundancy into data transmission that classic single antenna setups (SISO: Single In, Single Out) can’t provide. This gives MIMO systems several advantages over typical SISO configurations:

- MIMO radios can utilize the bounced and reflected RF transmissions (known as multipath propagation) to actually improve signal strength even without clear line-of-site, since MIMO radios receive and combine multiple streams of the same data that are received at slightly different time intervals. This is particularly useful in urban environments, where signal degradation between single antennas without clear line-of-site is a major issue. Urban environments provide plenty of reflection paths for MIMO signals to take between the transmit and receive radios.
- Overall throughput can be improved, allowing for greater quality and quantity of video or other data to be sent over the network.
- By utilizing multiple data streams, issues such as fading caused lost or dropped data packets can be reduced, resulting in better video or audio quality.

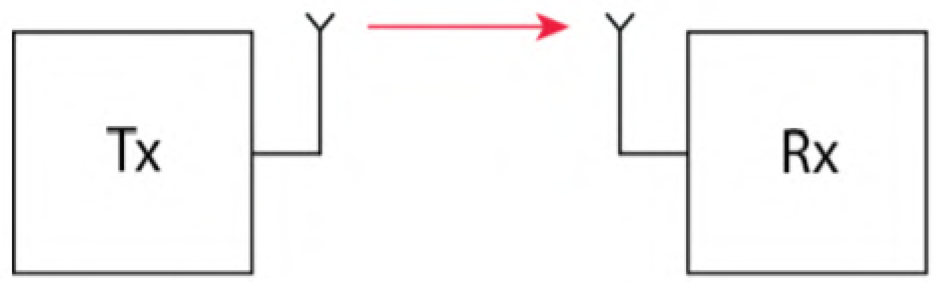


Figure 1.10 :Example of SISO system, where one antenna is used on each side of the RF

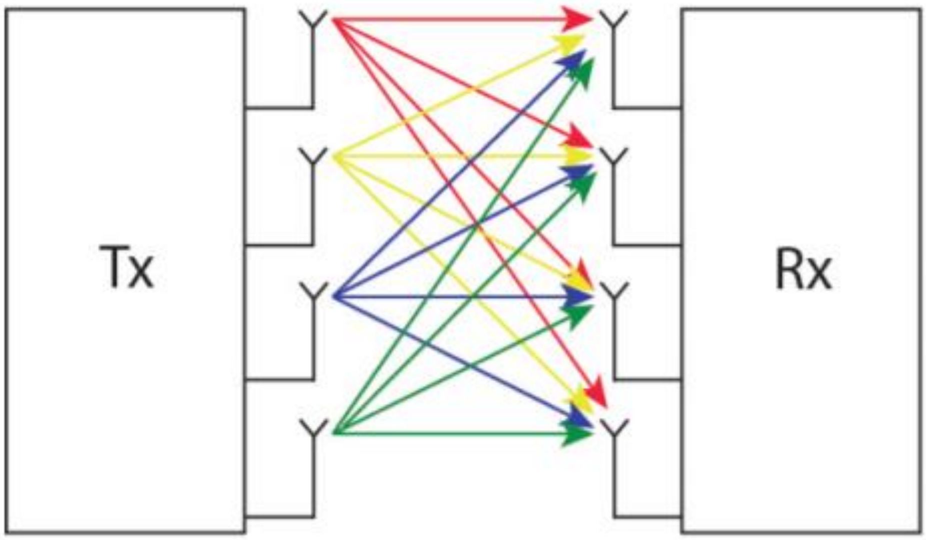


Figure 1.11 : Example of a 4X4 MIMO system

where four antennas from the transmit radio communicate with four antennas on the receive radio to improve link connection strength and bandwidth. Some MIMO radio systems also offer Ad-Hoc networking capability, where individual user nodes (such as a person with hand-held radio or vehicle with on-board MIMO radio and antennas installed) can enter or leave the MIMO network at any time, and automatically forward data from other node users through the network, creating a self-forming, self-healing mesh network that doesn't rely on a central architecture to operate.

This is similar to how scalable MANET radio systems work. The ability to take advantage of this type of networking opens up MIMO radio systems to new groups of users who need reliable, scalable network coverage. These systems are more practical than ever as MIMO radio systems continue to decrease in size and can be packaged into familiar hand-held radio form factors. Base stations and network repeaters can also be quickly deployed for short-term events that need a larger coverage area, without the hassle of permanent installation. A few example of users groups who are taking advantage of MIMO systems:

• First responders who often operate in chaotic, changing situations and can't rely on cellular networks or other existing fixed infrastructure to be operational when needed due to natural disasters, power outages, overloaded networks, or other issues.

• Broadcast television production, such as live sports or news broadcasts, where the story may change during broadcast and video transmission locations have to move without notice, or where the shoot might involve multiple, simultaneous areas of interest. Eliminating long, expensive cable runs is another major advantage for this market.

• Law enforcement or military users who need to operate their own separate communication networks on dedicated radio bands. This includes intra-team communication among small groups, as well as larger networks that include ground vehicles, UAV / UGV systems, and more

.

Figure 1.12 : Example of a Silvus MIMO radio with two gooseneck antennas being used to transmit HD video at the Indianapolis 500.

Standard MIMO Configurations MIMO radio systems utilize multiple antennas in order to send and receive multiple data streams at once. The number of antennas needed is defined by the radio manufacturer based on what they determine will work for optimal transmission and reception with their particular hardware and software. Typical configurations are:

• 2X2 MIMO (two transmit antennas, two receive antennas)

• 3X3 MIMO (three transmit antennas, three receive antennas)

•4X4 MIMO (four transmit antennas, four receive antennas)

• 8X8 MIMO (eight transmit antennas, eight receive antennas) Understanding the Basics of MIMO Communication Technology Generally speaking, the more antennas a system has the more simultaneous data streams can be transmitted at once, improving the radio link. However individual system setups, current physical and RF environmental conditions, and advances in radio technology means that more antennas doesn’t always equal better system performance.

Antenna Choices for MIMO Radios:

There are many antenna choices for MIMO radios, depending on how the user wishes to configure their radio system based on the unique needs of their application. Broadly speaking, antennas for MIMO use can be broken down the following categories:

1. **Traditional vertically polarized dipole antennas.** These are the same antennas used on SISO systems, the only difference being that the MIMO system will utilize two or more of the same antenna. In practice almost any omni directional antenna can be used as long as it meets the frequency band requirements, gain, RF power handling, and other parameters for the system being specified.

Figure 1.13 : Examples of two vertically polarized omni-directional antennas that can utilized in matched pairs on MIMO systems.

2. Multi-antenna products contained within a single radome. These antennas contain two or more RF connectors, which each connect to a different antenna within the radome. These are often designed with a slant left and slant right polarization, with each element offset 90 degrees (orthogonally) from the other, providing additional antenna polarization diversity for busy RF environments without increasing system installation complexity for the end user.

Each product features multiple RF connectors, each connected to a discrete antenna within the product’s radome. These are often cross-polarized for additional polarization and spatial diversity. Just as with SISO systems, MIMO systems can utilize omni-directional antennas or directional panels and sector antennas for different communication scenarios and coverage needs.

CHAPTER 2

CHANNEL MODEL

In order to accommodate the recent surge in mobile traffic, millimeter-wave frequencies above 6 GHz are being considered for the new mobile communication system . The millimeter-wave supports wide bandwidth, and the short wavelength of it enables the miniaturization of antennas. Therefore, millimeter-wave based mobile communication systems can be equipped with more antennas in the same space as long-term evolution (LTE) base stations. However, short wavelengths can cause high path loss and low signalto-noise radio (SNR).

3GPP TR 38.901, a millimeter-wave channel model of 3GPP technical report [10], is considered. This millimeter-wave channel model covers the frequency range from 0.5 GHz to 100 GHz. Thisfollows the frame work of WINNERII/WINNER+ and was modeled considering the radio propagation characteristics of the horizontal and vertical dimension. There are three scenarios, urban micro cell (UMi), urban macro cell (UMa), and indoor-office cell, in 3GPP TR 38.901, and this paper considers an UMi scenario. In UMi scenario, user equipment (UE) can be located randomly between 1.5 m and 22.5 m. The transmit antenna of the base station is located below the roof of the building where the UE can be located. The scenario has the environment with characteristics the line-ofsight (LOS) and the non-line-of-sight (NLOS) for outdoor and indoor areas between a base station (BS) and UEs. Detailedscenario parameters are described in Table 1.

TABLE I. MILLIMETER-WAVE CHANNEL MODEL: UMI SCENARIO

Parameters		Value
ISD		200m
BS antenna height		10m
UE location	Outdoor/Indoor	Outdoor and indoor
	LOS/NLOS	LOS and NLOS
UE height (h_{UE})	general equation	$h_{UE}=3(n_{fl}-1)+1.5$
	n_{fl} for outdoor UEs	1
	n_{fl} for indoor UEs	$n_{fl} \sim$ uniform$(1, N_{fl})$ where $N_{fl} \sim$ uniform$(4,8)$
Indoor UE ratio		80%
UE distribution (horizontal)		Uniform

CHAPTER 3

ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING

Orthogonal frequency division multiplexing (OFDM) is a technique, method or scheme for digital multi-carrier modulation using many closely spaced subcarriers a previously modulated signal modulated into another signal of higher frequency and bandwidth. Each of these subcarriers contains numbers of parallel data streams or channels and is modulated conventionally at a low symbol rate; these are groups of bits of data related to (but not the same as) gross bitrate, which is expressed in bits/second.This term is also known as coded OFDM (COFDM) and discrete multi-tone modulation (DMT), used for both wireless and physical communication mediums.

The term "orthogonal” is actually an adjective describing two things acting independently or in an uncorrelated manner; in this case, any two signals of an OFDM-based product operating without dependence on, or interference with, one another.OFDM is used for wideband digital communication, which is commonly used for digital television and audio broadcasting (radio) as well as broadband Internet access and wireless networking. OFDM is very similar to FDM (frequency division multiplexing) but with technology purposely emphasizing the minimization of crosstalk or signal interference from other nearby signal carrying communication mediums. OFDM uses many narrow band signals as opposed to a signal modulated at a high symbol rate and a large bandwidth.

OFDM-based products include:

- Certain types of broadband access through POTS (plain old telephone service) copper wiring
- Power line communication (PLC)
- Multimedia over Coax Alliance (MoCA) home networking
- Some digital radio systems
- Some digital TV systems
- Some mobile TV systems

The idea of multi-carrier transmission has surfaced recently to be used forcombating the hostility of wireless channel and providing high data rate communications. OFDM is a special form of multi-carrier transmission where all the subcarriersare orthogonal to each other. OFDM promises a high user data rate transmissioncapability at a reasonable complexity and precision.

At high data rates, the channel distortion to the data is very significant, and it issomewhat impossible to recover the transmitted data with a simple receiver. A verycomplex receiver structure is needed which makes use of computationally expensiveequalization and channel estimation algorithms to correctly estimate the channel, sothat the estimations can be used with the received data to recover the originallytransmitted data. OFDM can drastically simplify the equalization problem by turningthe frequency-selective channel into a flat channel. A simple one-tap equalizeris needed to estimate the channel and recover the data. Future telecommunication systems must be spectrally efficient to support a numberof high data rate users. OFDM uses the available spectrum very efficiently whichis very useful for multimedia communications. For all of the above reasons, OFDM has already been accepted by many of the future generation systems

OFDM Overview:

- The Orthogonal Frequency Division Multiplexing (OFDM) transmission scheme is the optimum version of the multicarrier transmission scheme. In the past, as well as in thepresent, the OFDM is referred in the literature Multi-carrier, Multi-tone and FourierTransform.
- The concept of using parallel data transmission and frequency multiplexing was published in the mid 1960s. After more than thirty years of research and development,OFDM has been widely implemented in high speed digital communications. Due torecent advances of digital signal Processing (DSP) and Very Large Scale Integratedcircuit (VLSI) technologies, the initial obstacles of OFDM implementation such asmassive complex computation, and high speed memory do not exist anymore.
- The use of Fast Fourier Transform (FFT) algorithms eliminates arrays of sinusoidal generators and coherent demodulation required in parallel data systems and makes the implementation of the technology cost effective .
- The OFDM concept is based on spreading the data to be transmitted over a large number

of carriers, each being modulated at a low rate.

- In contrast to conventional Frequency Division Multiplexing, the spectral overlapping among sub-carriers are allowed in OFDM since orthogonality will ensure the subcarrierseparation at the receiver, providing better spectral efficiency and the use of steep band pass filter was eliminated.
- OFDM transmission system offers possibilities for alleviating many of the problems encountered with single carrier systems. It has the advantage of spreading out a frequencyselective fade over many symbols. This effectively randomizes burst errors caused byfading or impulse interference so that instead of several adjacent symbols beingcompletely destroyed, many symbols are only slightly distorted.
- This allows successful reconstruction of majority of them even without forward error correction. Because of dividing an entire signal bandwidth into many narrow sub bands, the frequency response over individual sub bands is relatively flat due to sub band are smaller than coherence bandwidth of the channel. Thus, equalization is potentiallysimpler than in a single carrier system and even equalization may be avoided altogether ifdifferential encoding is implemented.
- The orthoganality of sub channles in OFDM can be maintained and individual sub channels can be completely separated by the FFT at the receiver when there are no inter symbol interference (ISI) and intercarrier interference (ICI) introduced by the transmission channel distortion.
- Since the spectra of an OFDM signal is not strictly band limited, linear distortions such asmultipath propagation causes each subchannel to spread energy into the adjacentchannels and consequently cause ISI.
- One way to prevent ISI is to create a cyclically extended guard interval, where eachOFDM symbol is preceded by a periodic extension of the signal itself. When the guardinterval is longer than the channel impulse response or multipath delay, the ISI can be eliminated.
- By using time and frequency diversity, OFDM provides a means to transmit data in a frequency selective channel. However, it does not suppress fading itself. Depending on their position in the frequency domain, individual subchannels could be affected by fading.

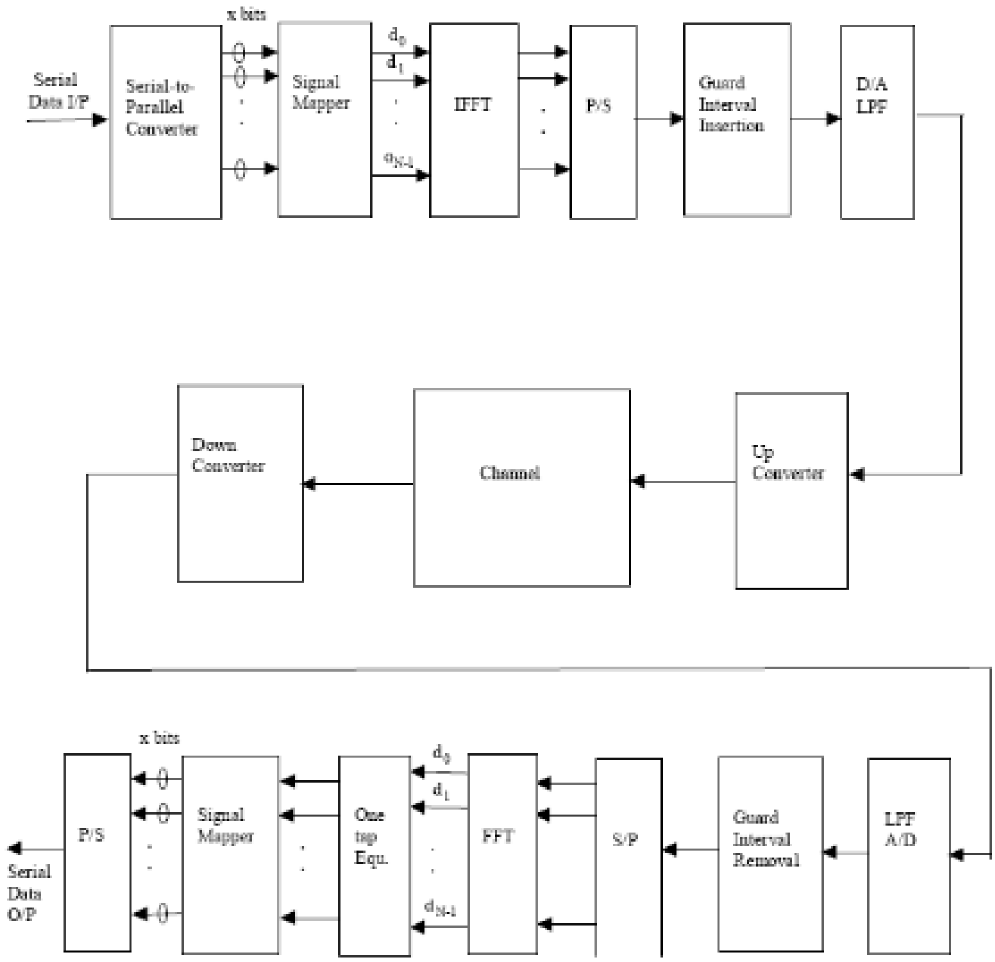


Figure 3.1: General OFDM System

OFDM Transceiver Systems:

A complete OFDM transceiver system is described in Fig. 2 In this model, forwarderror control/correction(FEC) coding and interleaving are added in the systemto obtain the robustness needed to protect against burst errors. An OFDM systemwith addition of channel coding and interleaving is referred to as coded OFDM(COFDM).

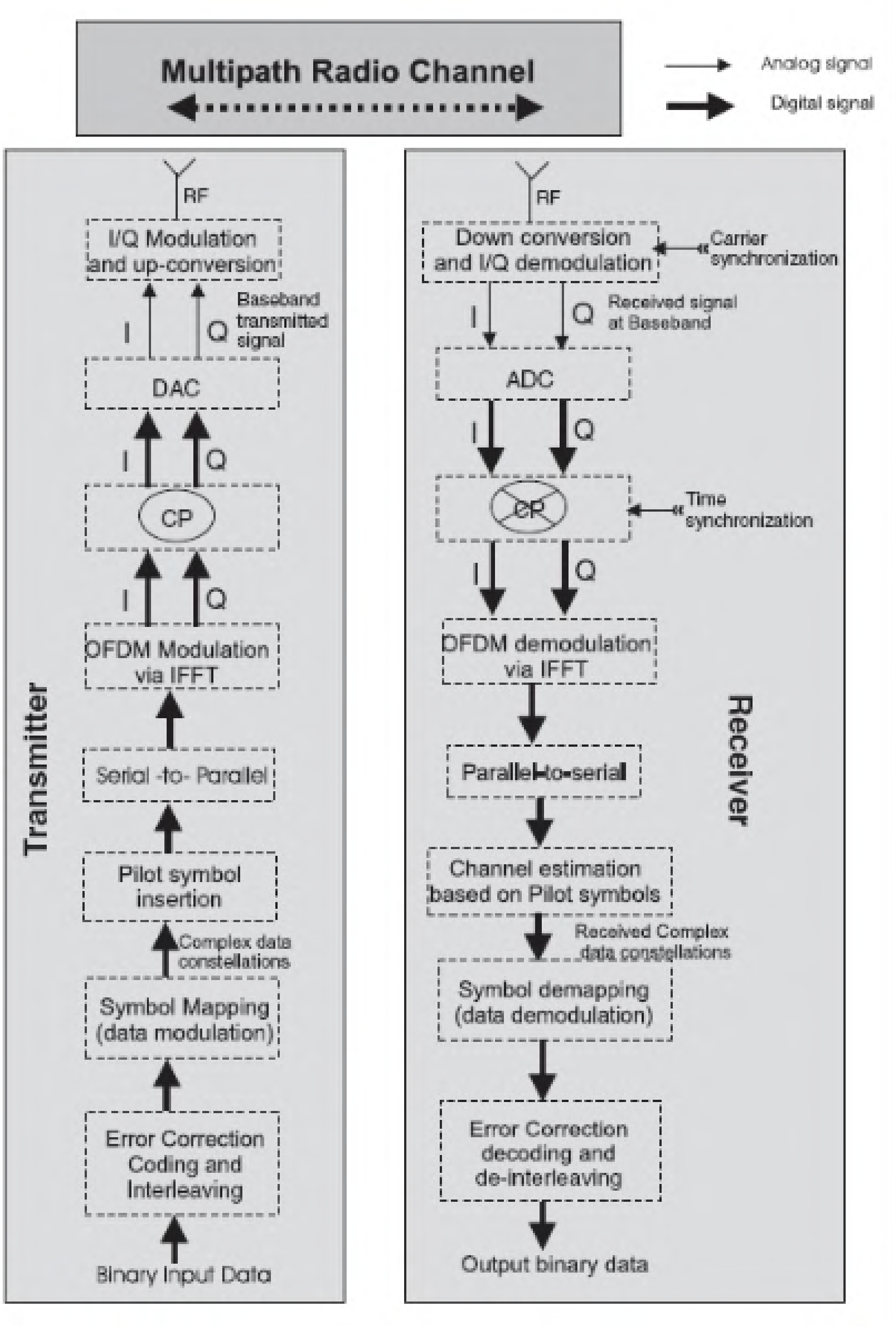


Figure 3.2: OFDM transceiver model

OFDM in Cellular Mobile System:

Severe multipath propagation, arising from multiple scattering by buildings and otherstructures in the vicinity of a mobile unit, makes the design of a mobile communicationchannel very challenging.· The scattering produces rapid random amplitude and phase variations in the receivedsignal as the vehicle moves in the multipath field. In addition, the whole motionintroduces a Doppler shift, which causes a broadening of the signal spectrum.· Since OFDM has the property of robustness to multipath propagation and other favorableproperties, it is likely to be considered for mobile communication systems. Cimini(1985) proposed a cellular mobile radio system based on OFDM.

OFDM combined with CDMA and TDMA is being investigated by many researchers. Various Multicarrier (MC) transmission schemes have been recently introduced intoDirect Sequence- Code Division Multiple Access(DS-CDMA) systems to acquirebenefits such as higher rate data transmission, bandwidth efficiency, frequency diversityand interference reduction.

Synchronization:

Synchronization is a big hurdle in OFDM. Synchronization usually consists of three parts:

1. Frame detection
2. Carrier frequency offset estimation and correction
3. Sampling error correction.

Frame detection is used to determine the symbol boundary so that correct samples for a symbol frame can be taken. Due to the carrier frequency difference of the transmitter and receiver, each signal sample at time t contains an unknown phase factor

e j2pDf ct

where c Df is the unknown carrier frequency offset. This unknown phase factor must be estimated and compensated for each sample before FFT at the receiver since otherwise the orthogonality between subcarriers are lost. For example, when the carrier is at 5GHz, an 100ppm crystal offset corresponding to a frequency offset of 50kHz. For a symbol period of T = 3.2 ms, DfcT = 1.6 . Because the sampling clock difference between the transmitter and receiver, each signal sample is off from its correct sampling time by a small amount which is linearly increasing with the index of the sample. For example, for 100ppm crystal offset, it will be off by 1 sample after 10000 samples. If a symbol contains 100 samples, then within each symbol the maximum offset will be 1% of a sample. Although this may cause the orthogonality degration between the sub-carriers, it can usually be ignored. If sampling error must be corrected, then interpolation filter must be used to construct the signal at correct sampling time

Channel estimation:

For burst communication system, training symbols are used at the beginning of each burst. Since theburst is short, the channel is assumed static over a whole burst so that once the channel is estimated, theinverse of the estimated channel response will be used to compensate the signal for the whole burst.Assuming the received signal after FFT isY(k) = C(k)X (k) + Z(k)where k is sub-carrier index, C is the channel, X is the pilot data, and Z is the noise. The simplest way toestimate the channel is then by

$$\hat{C}(k) = \frac{Y(k)}{X(k)}$$

i.e. dividing the received signal by the known pilot. Without noise, this gives the correct estimation. Whennoise is present, there could be error.

Analytical Model of OFDM System:

In this section, an analytical time-domain model of an OFDM transmitter andreceiver, as well as a channel model, is derived.

Transmitter:

The sth OFDM symbol is found using the sth subcarrier block, Xs[k]. In practice,the OFDM signal is generated using an inverse DFT. In the following model, thetransmitter is assumed to be ideal, i.e., sampling or filtering do not affect the signalat the transmitter side. Therefore, a continuous transmitter output signal may beconstructed directly using a Fourier series representation within each OFDM symbolinterval.

Receiver:

The signal at the receiver side consists of multiple echoes of the transmitted signal,as well as thermal (white gaussian) noise and interference. The RF signal received by the uth user is written as

$$r(t) = \Re e\left\{(\tilde{s}(t) * \tilde{h}_{u,s}(t))e^{j2\pi f_c[s]t}\right\} + \nu(t), \qquad sT_s \le t < (s+1)T_s,$$

where ν(t) is a real-valued, passband signal combining additive noise and interference.The receiver now has to recreate the transmitted signal. Aside from noise andmultipath effects, other imperfections in the receiver may also affect this process:

Timing error:

In order to demodulate the signal, the receiver must establish thecorrect timing. This means that the receiver must estimate which time instantcorresponds to t = 0 in the received signal (as seen from the transmitted signal point of view). As there are different uncertainties involved, a timing error of δtis assumed.

• Frequency error:

Similarly, the local oscillator of the receiver may oscillate atan angular frequency that is different from the angular frequency of the incoming signal. This difference is denoted as δω= 2πδf.The shifted timescale in the receiver is denoted as t_ = t − δt. Furthermore, dueto the angular frequency error δω, the down-converted signal spectrum is shifted infrequency. The down-converted signal is therefore written as

$$\tilde{r}(t) = (\tilde{s}(t') * \tilde{h}_{u,s}(t))e^{j\delta\omega t} + \tilde{\nu}(t'), \qquad sT_s \le t < (s+1)T_s,$$

Sampling:

Although the receiver may be modeled in the continuous time domain, an OFDMreceiver uses discrete signal processing to obtain the estimate of the transmittedsubcarriers.When the received signal is modeled as a Dirac impulse train, i.e., an ideallysampled signal, (2.17) is instead written as

$$\tilde{y}_{s,d}(t) = \sum_{n=0}^{N-1} \tilde{y}_s[n]\delta_c(t - nT),$$

ADVANTAGES OF OFDM:

1. The beauty of OFDM lies in its simplicity. One trick of the trade that makesOFDM transmitters low cost is the ability to implement the mapping of bits tounique carriers via the use of IFFT.
2. Unlike CDMA, OFDM receiver collects signal energy in frequency domain,thus it is able to protect energy loss at frequency domain.
3. In a relatively slow time-varying channel, it is possible to significantly enhancethe capacity by adapting the data rate per subcarrier according to SNR of thatparticular subcarrier.
4. OFDM is more resistant to frequency-selective fading than single-carrier systems.
5. The OFDM transmitter simplifies the channel effect, thus a simpler receiverstructure is enough for recovering transmitted data. If we use coherent modulationschemes, then very simple channel estimation (and/or equalization) isneeded, on the other hand, we need no channel estimator if differential modulationschemes are used.

6. The orthogonality preservation procedures in OFDM are much simpler comparedto CDMA or TDMA techniques even in very severe multipath conditions.
7. It is possible to use maximum likelihood detection with reasonable complexity.
8. OFDM can be used for high-speed multimedia applications with low servicecost.
9. OFDM can support dynamic packet access.
10. Single-frequency networks are possible in OFDM, which is especially attractivefor broadcast applications.
11. Smart antennas can be integrated with OFDM. MIMO systems and space–timecoding can be realized on OFDM and all the benefits of MIMO systems can beobtained easily. Adaptive modulation and tone/power allocation are also realizableon OFDM.

Disadvantages of OFDM System:

a) **Strict synchronization requirement:**

OFDM is highly sensitive to time and frequency synchronization errors, and especiallyat frequency synchronization errors, everything can go wrong . Indeed,demodulation of an OFDM signal with an offset in the frequency can lead to a highbit error rate.The source of frequency synchronization errors is two: first one being the differencebetween local oscillator frequencies in transmitter and receiver, second beingrelative motion between the transmitter and receiver that gives Doppler spread. Localoscillator frequencies at both transmitter and receiver must match as closelyas they can. For higher number of subchannels, the matching should be even better.Motion of transmitter and receiver causes the other frequency error. So, OFDM mayshow significant performance degradation at high-speed moving vehicles.To optimize the performance of an OFDM link, accurate synchronization is ofprime importance. Synchronization needs to be done in three factors: symbol, carrierfrequency, and sampling frequency synchronization. A good description of synchronizationprocedures is given.

b) Peak-to-Average Power Ratio (PAPR):

Peak-to-average power ratio (PAPR) is proportional to the number of subcarriersused for OFDM systems. An OFDM system with large number of subcarriers willthus have a very large PAPR when the subcarriers add up coherently. Large PAPRof a system makes the implementation of digital-to-analog converter (DAC) andanalog-to-digital converter (ADC) extremely difficult. The design of RF amplifieralso becomes increasingly difficult as the PAPR increases.

The clipping and windowingtechnique reduces PAPR by non-linear distortionof the OFDM signal. It thus introduces self-interference as the maximum amplitudelevel is limited to a fixed level. It also increases the out-of-band radiation, but this isthe simplest method to reduce the PAPR. To reduce the error rate, additional forwarderror correcting codes can be used in conjunction with the clipping and windowingmethod.Another technique called linear peak cancelationcan also be used to reduce thePAPR. In this method, time-shifted and time-scaled reference function is subtractedfrom the signal, such that each subtracted reference function reduces the peak powerof at least one signal sample. By selecting an appropriate reference function withapproximately the same bandwidth as the transmitted function, it can be assured thatthe peak power reduction does not cause out-of-band interference. One example of
a suitable reference function is a raised cosine window. Detailed discussion aboutcoding methods to reduce PAPR can be found.

C) Co-channel Interference in Cellular OFDM:

In cellular communication systems, CCI is combated by combining adaptive antennatechniques, such as sectorization, directive antenna, antenna arrays. UsingOFDM in cellular systems will give rise to CCI. Similarly with the traditional techniques,with the aid of beam steering, it is possible to focus the base station's antennabeam on the served user, while attenuating the co-channel interferers.

- **OFDM System Design Issues:**

System design always needs a complete and comprehensive understanding and considerationof critical parameters. OFDM system design is of no exception, as it dealswith some critical, and often conflicting parameters. Basic OFDM philosophy is todecrease data rate at the subcarriers, so that the symbol duration increases, thus themultipaths are effectively removed. This poses a challenging problem, as highervalue for CP interval will give better result, but it will increase the loss of energydue to insertion of CP. Thus, a trade-off must be obtained for a reasonable design.

- **OFDM System Design Requirements**

• **Available bandwidth**: Bandwidth is always the scarce resource, so the mother of the system design should be the available bandwidth for operation. The amount of bandwidth will play a significant role in determining number of subcarriers, because with a large bandwidth, we can easily fit in a large number of subcarriers with reasonable guard space.

• **Required bit rate**: The overall system should be able to support the data rate required by the users. For example, to support broadband wireless multimedia communication, the system should operate at more than 10 Mbps at least.

• Tolerable delay spread: Tolerable delay spread will depend on the user environment. Measurements show that indoor environment experiences maximum delayspread of few hundreds of ns at most, whereas outdoor environment can experienceup to 10 µs. So the length of CP should be determined according to thetolerable delay spread.

• **Doppler values**: Users on a high-speed vehicle will experience higher Dopplershift, whereas pedestrians will experience smaller Doppler shift. These considerationsmust be taken into account.

OFDM System Design Parameters:

The design parameters are derived according to the system requirements. Followingare the design parameters for an OFDM system:

• Number of subcarriers: Increasing number of subcarriers will reduce the data rate via each subcarrier, which will make sure that the relative amount of dispersion in time caused by multipath delay will be decreased . But when there are large numbers of subcarriers, the synchronization at the receiver side will be extremely difficult.

• Guard time (CP interval) and symbol duration: A good ratio between the CPinterval and symbol duration should be found, so that all multipaths are resolvedand not significant amount of energy is lost due to CP. As a thumbrule, the CP interval must be two to four times larger than the root mean square (RMS) delay spread. Symbol duration should be much larger than the guard timeto minimize the loss of SNR, but within reasonable amount. It cannot be arbitrarilylarge, because larger symbol time means that more subcarriers can fit withinthe symbol time. More subcarriers increase the signal processing load at boththe transmitter and receiver, increasing the cost and complexity of the resultingdevice.

• Subcarrier spacing: Subcarrier spacing must be kept at a level so that synchronization is achievable. This parameter will largely depend on available bandwidth and the required number of subchannels.

• Modulation type per subcarrier: This is trivial, because different modulation schemes will give different performances. Adaptive modulation and bit loading may be needed depending on the performance requirement. It is interesting to note that the performance of OFDM systems with differential modulation compares quite well with systems using non-differential and coherent demodulation. Furthermore, the computation complexity in the demodulation process is quite low for differential modulations.

OFDM has several significant benefits over single carrier QAM modulation which has lead to its adoption in many of the modern wireless standards, including ADSL, European Digital Video Broadcast, IEEE 802.11a/g/n (WiFi), WiMax, and 3GPP Long Term Evolution (LTE). First of all, OFDM enables easy equalization, solving many of the equalization complexity issues we encountered for single carrier QAM by working instead in the frequency domain. Second of all, the ability to control the size of the constellation on each subcarrier in OFDM also allows it to mimic the "waterfilling" construction which maximizes the amount of information which can be reliably transmitted over the channel: subcarriers (frequencies) with better channel gains can utilize higher order modulations and coding rates. A final significant benefit that we will encounter later in the course is that OFDM allows for an especially easy to implement and understand multiple access strategy OFDMA, in which different subcarriers (frequencies) are allocated to different users at different times. To understand how OFDM modems work, we must first revisit several properties of the discrete Fourier Transform, and its efficient implementation the Fast Fourier Transform.

Orthogonal Frequency Division Multiplexing (OFDM) is special form of multi-carrier modulation, patented in 1970. It is particularly suited for transmission over a dispersive channel. (See further discussion of MCM over wireless channel.)

In a multipath channel, most conventional modulation techniques are sensitive to intersymbol interference unless the channel symbol rate is small compared to the delay spread of the channel. OFDM is significantly less sensitive to intersymbol interference, because a special set of signals is used to build the composite transmitted signal. The basic idea is that each bit occupies a frequency-time window which ensures little or no distortion of the waveform. In practice, it means that bits are transmitted in parallel over a number of frequency-nonselective channels. Applications of OFDM are found in

- Digital Audio Broadcasting (DAB) and
- Digital Video Broadcasting over the terrestrial network: Digital Terrestrial Television Broadcasting (DTTB). In the DTTB OFDM transmission standard, about 2,000 to 8,000 subcarriers are used.
- UMTS. The UMTS Forum is selecting an appropriate radio solution for the third generation mobile standard, as a successor to GSM. OFDM is one of the five competing proposals.

- Wireless LANs. OFDM is used in HIPERLAN Phase II, which supports 20 Mbit/s in propagation environments with delay spreads up to 1 msecond.

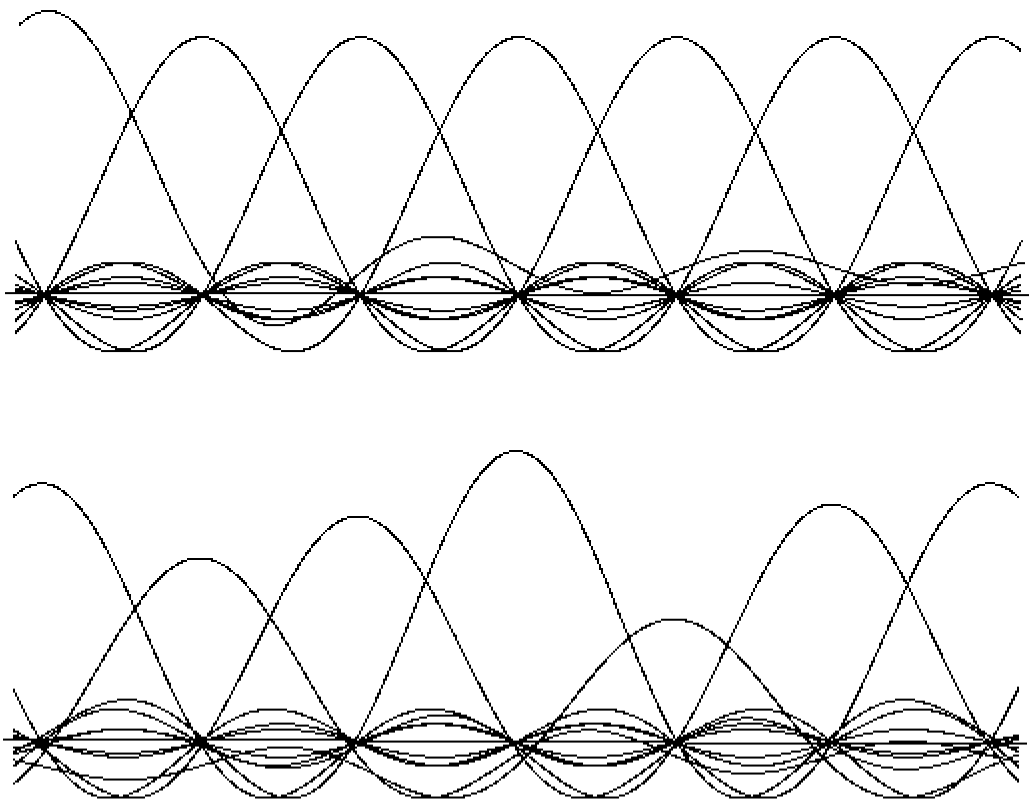


Figure 3.3: Signal spectrum of an OFDM signal.

The effect of multipath scattering on OFDM differs from what happens to other forms of modulation. A qualitative description and mathematical description of OFDM is presented by Dusan Matic. Jean-Paul Linnartz reviews the effects of a Doppler spread and the associated rapid channel variations. Dusan Matic also studied the system design aspects of OFDM at mm-wavelengths.

Coded OFDM

Multi-Carrier Modulation on its own is not the solution to the problems of communication over unreliable multipath channels. The channel time dispersion will excessively attenuate some subcarriers such that the throughput on these sub-channels would be unacceptable small. Only if the joint signal of many subcarriers is processed appropriately, the diversity advantages of MCM can be exploited. The need for coding across subcarriers was addressed by Sari et al. warning against overly enthusiastic pursuit of MCM. The advantages of frequency-domain implementations of equalizers (using an FFT) should not be mistaken for an "inherent" diversity gain of OFDM, which may not exist.

In an OFDM transmitter, blocks of k incoming bits are encoded into n channel bits. Before transmission, an n-point Inverse-FFT operation is performed. When the signals at the I-FFT output are transmitted sequentially, each of the n channel bits appears at a different (subcarrier) frequency. Such coding across subcarriers is necessary. If one subcarrier experiences deep fading, this leads to erasure of the bit on this subcarrier.

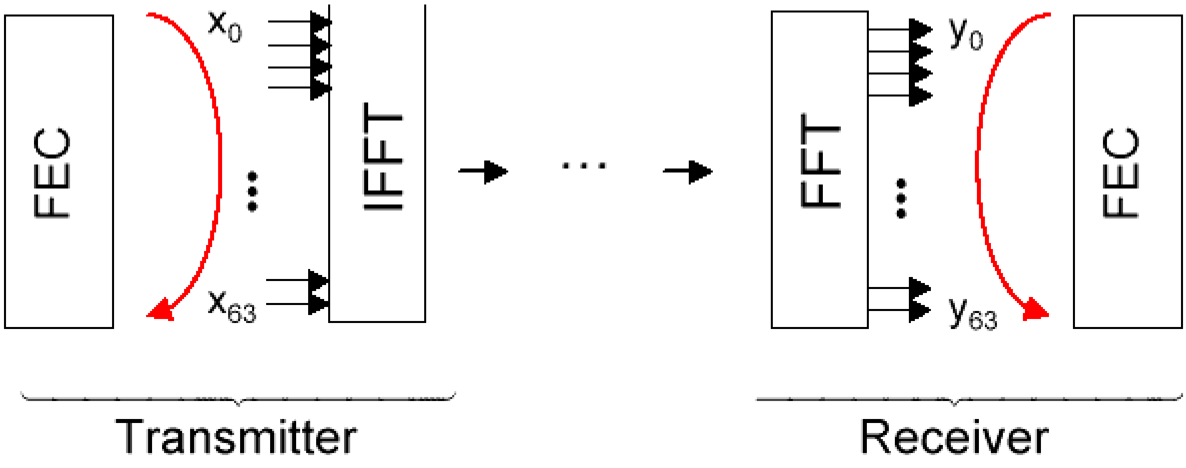


Figure 3.4: Transmitter and Receiver

But of course coding across subcarriers is not the only mechanism that can be invoked to combat dispersion or to exploit diversity. Other possibilities are

- Interleaving in frequency or time domain with coding in the other domain,
- The use of different signal constellations at different frequencies, i.e., adapting the subcarrier bit rate to the channel state,
- Signal spreading over various subcarriers, e.g., according to a linear matrix operation, as is proposed in Orthogonal Multi-Carrier Code Division Multiplexing.

If in a point-to-point MCM link, the receiver and the transmitter can cooperate by adaptively distributing of their power budget over the individual subcarriers. For instance, the signal-to-noise ratios selected according to Gallager's water-pouring theorem can (under certain conditions) be proved to be optimum. Efficient loading of the various subcarriers can for instance significantly enhance the performance of MCM over twisted pair telephone subscriber loops with crosstalk from other nearby copper pairs.

Implementational Aspects:

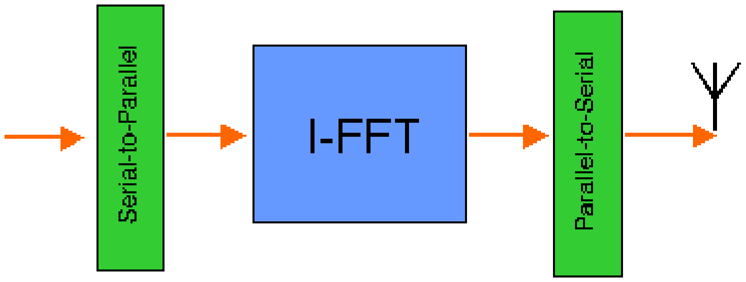


Figure 3.5: OFDM transmitter using an (inverse) Fast Fourier Transform (FFT).

- OFDM is not a constant-envelope modulation method. Therefore transmit power amplifiers must be highly linear.
- OFDM receiver performance is very sensitive to phase noise.
- Synchronization to an OFDM signal also requires frame synchronization, to support an FFT operation at the receiver.

What is OFDM?

OFDM is a form of multicarrier modulation. An OFDM signal consists of a number of closely spaced modulated carriers. When modulation of any form - voice, data, etc. is applied to a carrier, then sidebands spread out either side. It is necessary for a receiver to be able to receive the whole signal to be able to successfully demodulate the data. As a result when signals are transmitted close to one another they must be spaced so that the receiver can separate them using a filter and there must be a guard band between them. This is not the case with OFDM. Although the sidebands from each carrier overlap, they can still be received without the interference that might be expected because they are orthogonal to each another. This is achieved by having the carrier spacing equal to the reciprocal of the symbol period.

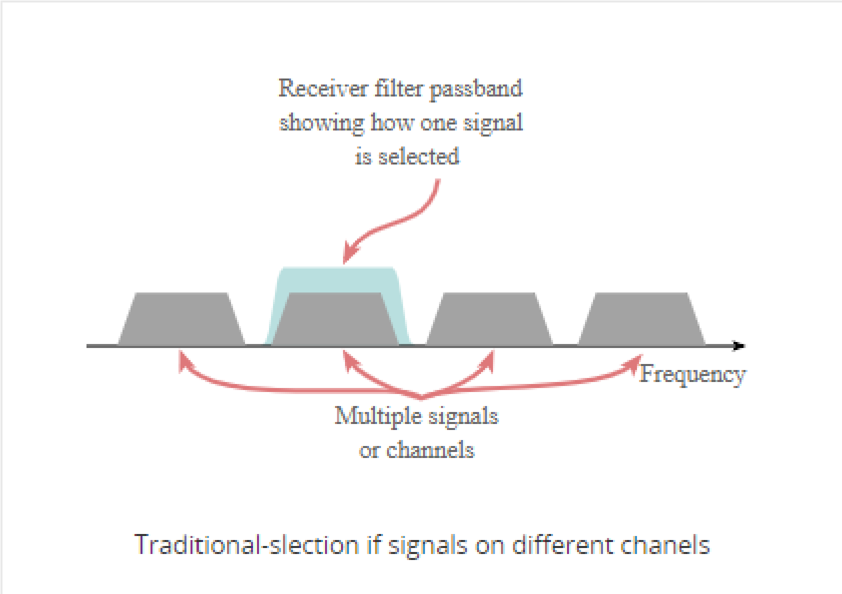


Figure 3.6: Signal at the Receiver End.

To see how OFDM works, it is necessary to look at the receiver. This acts as a bank of demodulators, translating each carrier down to DC. The resulting signal is integrated over the symbol period to regenerate the data from that carrier. The same demodulator also demodulates the other carriers. As the carrier spacing equal to the reciprocal of the symbol period means that they will have a whole number of cycles in the symbol period and their contribution will sum to zero - in other words there is no interference contribution.

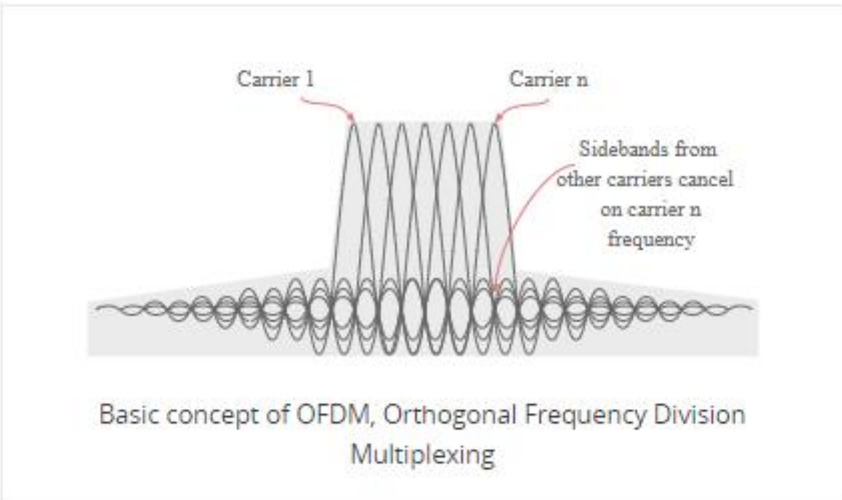


Figure 3.7: Basic concept of OFDM.

One requirement of the OFDM transmitting and receiving systems is that they must be linear. Any non-linearity will cause interference between the carriers as a result of inter-modulation distortion. This will introduce unwanted signals that would cause interference and impair the orthogonality of the transmission.

In terms of the equipment to be used the high peak to average ratio of multi-carrier systems such as OFDM requires the RF final amplifier on the output of the transmitter to be able to handle the peaks whilst the average power is much lower and this leads to inefficiency. In some systems the peaks are limited. Although this introduces distortion that results in a higher level of data errors, the system can rely on the error correction to remove them.

Data on OFDM:

The data to be transmitted on an OFDM signal is spread across the carriers of the signal, each carrier taking part of the payload. This reduces the data rate taken by each carrier. The lower data rate has the advantage that interference from reflections is much less critical. This is achieved by adding a guard band time or guard interval into the system. This ensures that the data is only sampled when the signal is stable and no new delayed signals arrive that would alter the timing and phase of the signal.

Guard interval on OFDM signals:

The distribution of the data across a large number of carriers in the OFDM signal has some further advantages. Nulls caused by multi-path effects or interference on a given frequency only affect a small number of the carriers, the remaining ones being received correctly. By using error-coding techniques, which does mean adding further data to the transmitted signal, it enables many or all of the corrupted data to be reconstructed within the receiver. This can be done because the error correction code is transmitted in a different part of the signal.

Key features of OFDM

The OFDM scheme differs from traditional FDM in the following interrelated ways:

- Multiple carriers (called subcarriers) carry the information stream
- The subcarriers are orthogonal to each other.

- A guard interval is added to each symbol to minimize the channel delay spread and intersymbol interference.

Frequency offset OFDM synchronization:

- It is particularly important that the demodulator in an OFDM receiver is able to synchronize accurately with the carriers within the OFDM signal. Offsets may arise for a number of reasons including any frequency errors between the transmitter and the receiver and also as a result of Doppler shifts if there is movement between the transmitter and receiver.
- If the frequency synchronisation is impaired, then the orthogonality of the carriers is reduced within the demodulation process and error rates increase. Accordingly it is essential to maintain orthogonality to reduce errors and maintain the performance of the link.
- First look at the way that sampling should occur. With the demodulator in synchronisation, all the contributions from the other carriers sum to zero as shown. On this way all the carriers are orthogonal and the error rate is at its minimum.
- OFDM demodulation showing how the interference from other carriers is minimised at the centre of the carrier frequency
- If a situation is encountered where the OFDM synchronisation for the frequency aspects are poor, then the demodulator will centre its samples away from the peak of the signal, and also at a point where the contributions from the other signals do not sum to zero. This will lead to a degradation of the signal which could in turn lead to an increase in the number of bit errors.
- OFDM demodulation where interference is high due to poor frequency synchronisation
- Clock offset OFDM synchronization
- It is also necessary to maintain OFDM synchronization in terms of the clock. Gain if the clock synchronisation is not accurate, sampling will be offset and again orthogonality will be reduced, and data errors will increase.

- When looking at OFDM synchronization with regard to the clock offset, the carrier spacing used within the receiver for sampling the received signal will be based upon the internal clock rate. If this differs from that used within the transmitter, it will be found that even if the first carrier within the multiplex is correct, then there will be a growing discrepancy with each carrier away from the first one. Even small levels of discrepancy will cause the error rate to increase.
- OFDM demodulation with clock synchronization misalignment : When using OFDM it is necessary to ensure that the synchronisation for both timing and frequency is accurate. By ensuring accurate synchronisation, it is possible to perform the optimum demodulation of the signal. Any misalignment causes the receiver to start to pick up the unwanted interference signals. Fortunately it is relatively easy to obtain accurate synchronisation signals as these are available from the network, and short term synchronisation can be generated internally.
- In telecommunications, orthogonal frequency-division multiplexing (OFDM) is a method of encoding digital data on multiple carrierfrequencies. OFDM has developed into a popular scheme for wideband digital communication, used in applications such as digital television and audio broadcasting, DSL internet access, wireless networks, power line networks, and 4G mobile communications.
- In coded orthogonal frequency-division multiplexing (COFDM), forward error correction (convolutional coding) and time/frequency interleaving are applied to the signal being transmitted. This is done to overcome errors in mobile communication channels affected by multipath propagation and Doppler effects. COFDM was introduced by Alard in 1986 [1] [2][3] for Digital Audio Broadcasting for Eureka Project 147. In practice, OFDM has become used in combination with such coding and interleaving, so that the terms COFDM and OFDM co-apply to common applications.[4]
- OFDM is a frequency-division multiplexing (FDM) scheme used as a digital multi-carrier modulation method. OFDM was introduced by Chang of Bell Labs in 1966.[5][6][7] Numerous closely spaced orthogonal sub-carrier signals with overlapping spectra are emitted to carry data.[8] Demodulation is based on Fast Fourier Transform algorithms. OFDM was improved by Weinstein and

Ebert in 1971 with the introduction of a guard interval, providing better orthogonality in transmission channels affected by multipath propagation.[9] Each sub-carrier (signal) is modulated with a conventional modulation scheme (such as quadrature amplitude modulation or phase-shift keying) at a low symbol rate. This maintains total data rates similar to conventional single-carrier modulation schemes in the same bandwidth.

- The main advantage of OFDM over single-carrier schemes is its ability to cope with severe channel conditions (for example, attenuation of high frequencies in a long copper wire, narrowband interference and frequency-selective fading due to multipath) without complex equalization filters. Channel equalization is simplified because OFDM may be viewed as using many slowly modulated narrowband signals rather than one rapidly modulated wideband signal. The low symbol rate makes the use of a guard interval between symbols affordable, making it possible to eliminate intersymbol interference (ISI) and use echoes and time-spreading (in analog television visible as ghosting and blurring, respectively) to achieve a diversity gain, i.e. a signal-to-noise ratio improvement. This mechanism also facilitates the design of single frequency networks (SFNs) where several adjacent transmitters send the same signal simultaneously at the same frequency, as the signals from multiple distant transmitters may be re-combined constructively, sparing interference of a traditional single-carrier system.

Orthogonality:

- Conceptually, OFDM is a specialized frequency-division multiplexing (FDM) method, with the additional constraint that all subcarrier signals within a communication channel are orthogonal to one another.
- In OFDM, the sub-carrier frequencies are chosen so that the sub-carriers are orthogonal to each other, meaning that cross-talk between the sub-channels is eliminated and inter-carrier guard bands are not required. This greatly simplifies the design of both the transmitter and the receiver; unlike conventional FDM, a separate filter for each sub-channel is not required.

- The orthogonality requires that the sub-carrier spacing is Hertz, where T_U seconds is the useful symbol duration (the receiver-side window size), and k is a positive integer, typically equal to 1. This stipulates that each carrier frequency undergoes k more complete cycles per symbol period than the previous carrier. Therefore, with N sub-carriers, the total passband bandwidth will be B = N*Δf (Hz).
- The orthogonality also allows high spectral efficiency, with a total symbol rate near the Nyquist rate for the equivalent baseband signal (i.e. near half the Nyquist rate for the double-side band physical passband signal). Almost the whole available frequency band can be used. OFDM generally has a nearly 'white' spectrum, giving it benign electromagnetic interference properties with respect to other co-channel users.
- A simple example: A useful symbol duration T_U = 1 ms would require a sub-carrier spacing of (or an integer multiple of that) for orthogonality. N = 1,000 sub-carriers would result in a total passband bandwidth of NΔf = 1 MHz. For this symbol time, the required bandwidth in theory according to Nyquist is (half of the achieved bandwidth required by our scheme), where R is the bit rate and where N = 1,000 samples per symbol by FFT. If a guard interval is applied (see below), Nyquist bandwidth requirement would be even lower. The FFT would result in N = 1,000 samples per symbol. If no guard interval was applied, this would result in a base band complex valued signal with a sample rate of 1 MHz, which would require a baseband bandwidth of 0.5 MHz according to Nyquist. However, the passband RF signal is produced by multiplying the baseband signal with a carrier waveform (i.e., double-sideband quadrature amplitude-modulation) resulting in a passband bandwidth of 1 MHz. A single-side band (SSB) or vestigial sideband (VSB) modulation scheme would achieve almost half that bandwidth for the same symbol rate (i.e., twice as high spectral efficiency for the same symbol alphabet length). It is however more sensitive to multipath interference.
- OFDM requires very accurate frequency synchronization between the receiver and the transmitter; with frequency deviation the sub-carriers will no longer be orthogonal, causing inter-carrier interference (ICI) (i.e., cross-talk between the sub-carriers). Frequency offsets are typically caused by mismatched transmitter

and receiver oscillators, or by Doppler shift due to movement. While Doppler shift alone may be compensated for by the receiver, the situation is worsened when combined with multipath, as reflections will appear at various frequency offsets, which is much harder to correct. This effect typically worsens as speed increases, and is an important factor limiting the use of OFDM in high-speed vehicles. In order to mitigate ICI in such scenarios, one can shape each sub-carrier in order to minimize the interference resulting in a non-orthogonal subcarriers overlapping. For example, a low-complexity scheme referred to as WCP-OFDM (Weighted Cyclic Prefix Orthogonal Frequency-Division Multiplexing) consists of using short filters at the transmitter output in order to perform a potentially non-rectangular pulse shaping and a near perfect reconstruction using a single-tap per subcarrier equalization. Other ICI suppression techniques usually increase drastically the receiver complexity.

CHAPTER 4

5G NR BASED SYSTEM MODEL

5G NR Transmission Frame:

When comparing 4G LTE with 5G NR, the most significant change is in frequency domain. NR supports frequencies above 6 GHz as well as frequencies below 6 GHz, and supports various subcarrier spacing (SCS) according to OFDM Numerology. OFDM Numerology is a newly designed concept to utilize the wide bandwidth of the millimeter-wave, and SCS of 30 kHz, 60 kHz, 120 kHz, and 240 kHz, including the existing 15 kHz are supported according to that. For each SCS, the number of slots and the number of OFDM symbols is determined in a frame. This paper considers SCS of 60 kHz for millimeter-wave MIMO systems. A subframe consists of four slots, each consisting of 14 OFDM symbols. When the system bandwidth is 100 MHz, the number of resource blocks (RBs) is 125 and the number of resource elements (REs) is 84000 in a subframe. Table 2 shows the NR transmission frame configuration. The Fig. 1 shows the RBs assigned to 10 UEs by proportional fair (PF) scheduling. Wireless mobile communication channels with wide bandwidth have frequency selective fading characteristics, so each UE is allocated suitable sections among 100 MHz

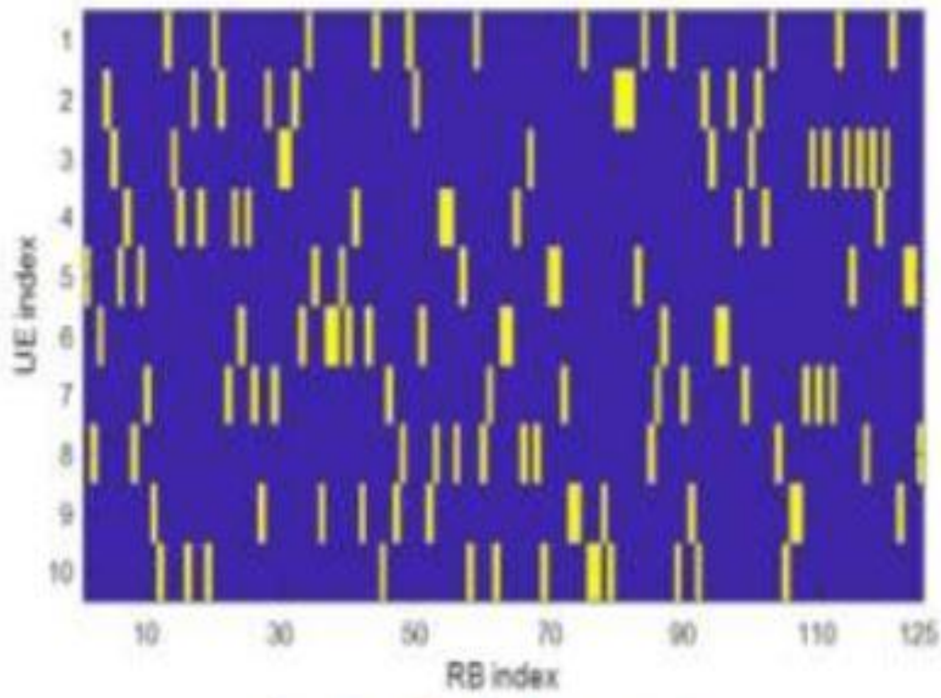


Figure 1. Scheduled RBs per UE

Figure 4.1: Scheduled RBs per UE

TABLE II. NR FRAME CONFIGURATION

Parameter	Value
SCS [kHz]	60
The number of SCs per RB	12
RB bandwidth [kHz]	720
System bandwidth [MHz]	100
The total number of SCs	1500
The total number of RBs	125
The number of slots	4
The number of symbols per slot	14

Spatial Multiplexing MIMO System Model This paper considers an uplink massive MIMO system based on 3GPP NR in the millimeter-wave channel. One BS equipped with uniform rectangular array (URA) having r v h N N N = · antennas and K UEs equipped with uniform linear array (ULA), N 1 t · , are considered, as shown in Fig. 2. This paper assumes that the number of antennas in base stations is greater than that of UE. Each UE can transmit L data streams to the base station at the same time. L is the number of data streams, and this paper considers L Nδ t . That is, the total number of data streams N s is K L· . The transmit signal k x by k-th UE can be expressed as follows:

$$x_k = \sqrt{p} H_k S_k + N_k \qquad (1)$$

Where p is the average transmit power. Nk is denotes Gaussian noise. N N t H C k · □ is the channel matrix from k-th UE to BS. It is expressed as follows:

$$H_k = \begin{bmatrix} h_{1,1}^k & h_{1,2}^k & \cdots & h_{1,N_r}^k \\ h_{2,1}^k & h_{2,2}^k & \cdots & h_{2,N_r}^k \\ \vdots & \vdots & \vdots & \vdots \\ h_{N_t,1}^k & h_{N_t,2}^k & \cdots & h_{N_t,N_r}^k \end{bmatrix} \qquad (2)$$

h is the channel impulse response (CIR) for a NLOS case or a LOS case. NLOS CIR and LOS CIR are given by:

$$h_{u,s}^{NLOS}(\tau,t) = \sum_{n=1}^{2}\sum_{i=1}^{3}\sum_{m\in R_i} h_{u,s}^{NLOS}(t)\delta(\Delta\tau) + \sum_{n=3}^{N} h_{u,s,n}^{NLOS}(t)\delta(\Delta\tau) \qquad (3)$$

$$h_{u,s}^{LOS}(\tau,t) = \sqrt{\frac{1}{K_R+1}} h_{u,s}^{NLOS}(\tau,t) + \sqrt{\frac{K_R}{K_R+1}} h_{u,s,l}^{LOS}(t)\delta(\Delta\tau) \qquad (4)$$

h and , ,LOS are the channel coefficients obtained by 12 step procedure. The field patterns of receive antenna element u and the field patterns of transmit antenna element s are considered. For the vertical and horizontal dimensions, azimuth angle of arrival, zenith angle of arrival, azimuth angle of departure, and zenith angle of departure are considered. The detailed step can be found in [10]. k S is data streams, 1 2 [, ,..., ,]T k L S s s s = , and l s is the transmit symbol. L is determined by the channel rank. An BS reports on the calculated rank indicator and channel quality indicator (CQI) to indicate a suitable uplink transmission data rate towards each UE.

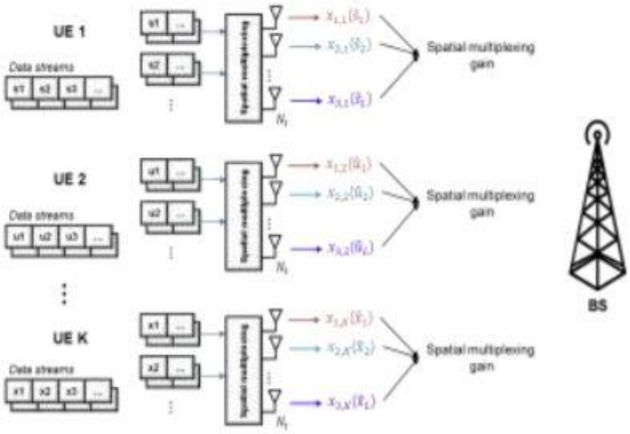


Figure 4.2: System model in uplink

CHAPTER 5

Software Introduction:

5.1. Introduction to MATLAB:

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn

and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

5.2 The MATLAB system:

The MATLAB system consists of five main parts

- **Development Environment**:

  This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library**:

  This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language**:

  This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both “programming in the small” to rapidly create quick and dirty throw-away programs, and “programming in the large” to create large and complex application programs.

- **Graphics:**

  MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

5.3 GRAPHICAL USER INTERFACE (GUI):

MATLAB’s Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function’s GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.

- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type

guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted, GUIDE opens a new (i.e., blank) window.

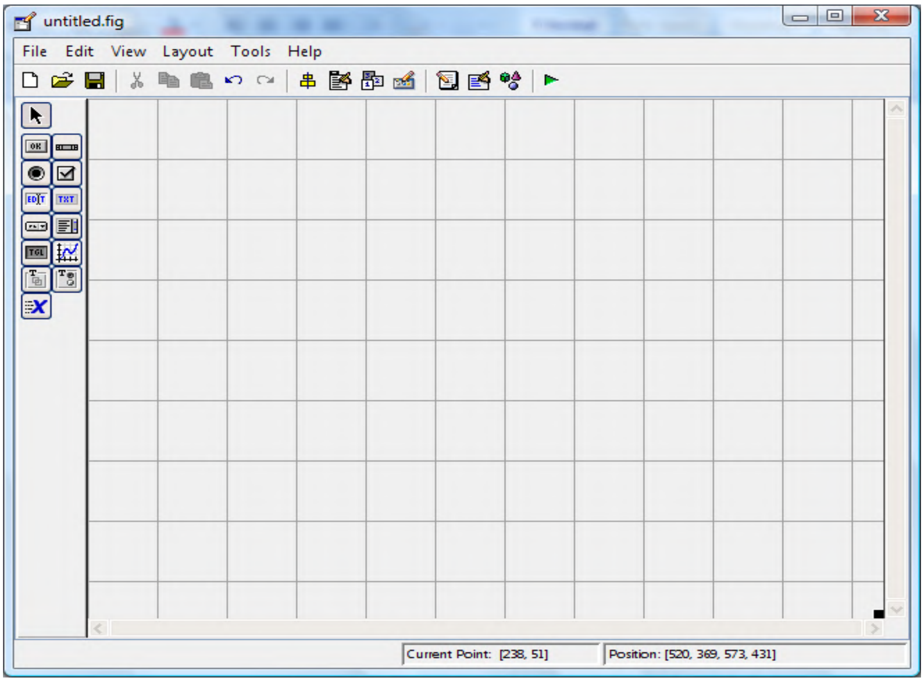


Figure 5.1: GUI(Graphical User Interface)

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

5.4 Getting Started

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

5.4.1 Introduction - describes the components of the MATLAB system.

5.4.2 Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

5.4.3 Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

5.4.4 Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

5.4.5 Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

5.5 DEVELOPMENT ENVIRONMENT

5.5.1Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

5.5.2 Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

5.5.3 Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

5.5.4 MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

5.5.5 Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window

- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window.In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type helpbrowser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

Display Pane - After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to runMATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

5.6 MANIPULATING MATRICES

5.6.1 Entering Matrices

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Dürer's matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]

MATLAB displays the matrix you just entered.

```
A = 16     3     2    13

     5    10    11     8

     9     6     7    12
```

4   15   14   1

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

5.6.2 Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

5.6.3 Operators

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

5.6.4 Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.14159265...
I	Imaginary unit, $\sqrt{-1}$
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

5.7 GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

5.7.1 Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

5.7.2 GUI Development Environment

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

The Implementation of a GUI

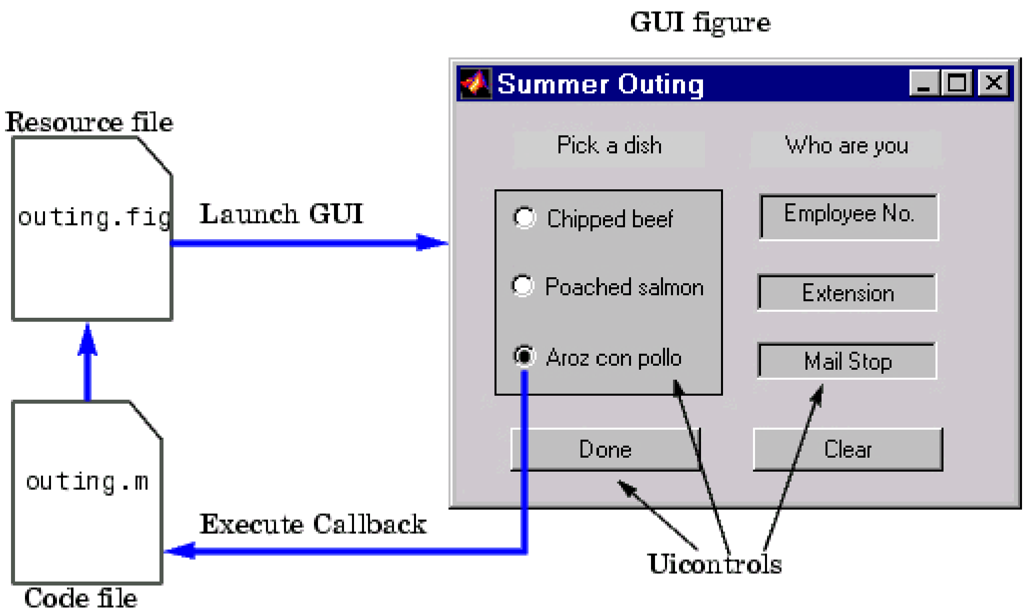
While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of itschildren (uicontrols and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and thecallbacks, which are defined as subfunctions. This M-file is referred to as theapplication M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.



5.7.3 Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

5.7.4 Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques

used in the application M-file.

Application Examples - a collection of examples that illustrate techniques

which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

5.7.5 User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- Listboxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)

button_state = get(h,'Value');

if button_state == get(h,'Max')

   % toggle button is pressed

elseif button_state == get(h,'Min')

   % toggle button is not pressed

end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);

a(:,:,2) = rand(16,128);

a(:,:,3) = rand(16,128);

set(h,'CData',a)
```

Radio Buttons

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)

set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)
off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];
mutual_exclude(off)
% Continue with callback
  .
  .
  .
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)
if (get(h,'Value') == get(h,'Max'))
  % then checkbox is checked-take approriate action
else
```

```
    % checkbox is not checked-take approriate action
end
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)
user_string = get(h,'string');
% proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)
user_entry = str2double(get(h,'string'));
if isnan(user_entry)
    errordlg('You must enter a numeric value','Bad Input','modal')
end
% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menubar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menubar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

Listboxes differentiate between single and double clicks on an item and set the figure SelectionType property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure SelectionType property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

```
val = get(h,'Value');
switch val
case 1
% The user selected the first item
case 2
% The user selected the second item
% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)
val = get(h,'Value');
string_list = get(h,'String');
selected_string = string_list{val}; % convert from cell array to string
% etc.
```

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes ButtonDownFcn property to define the callback.

3.7.6 Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

```
axes(handles.axes1)
```

makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axes. See GUI with Multiple Axes for and example that uses two axes.

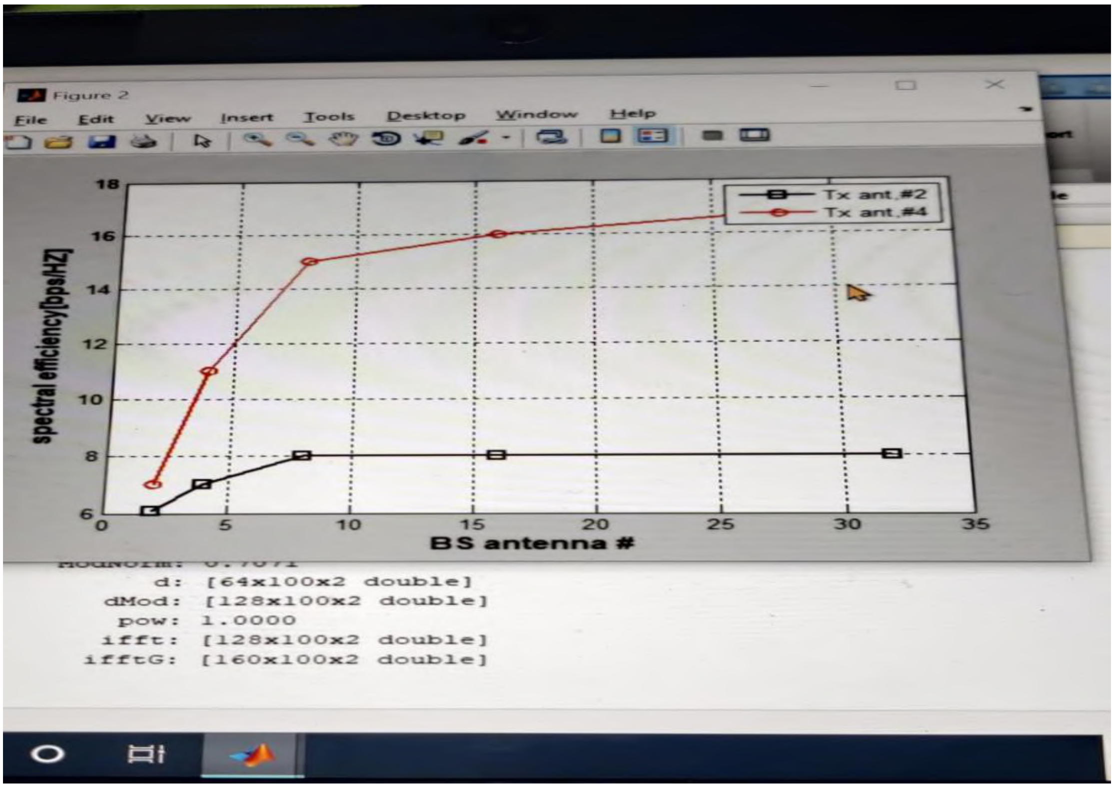
Figure

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control.

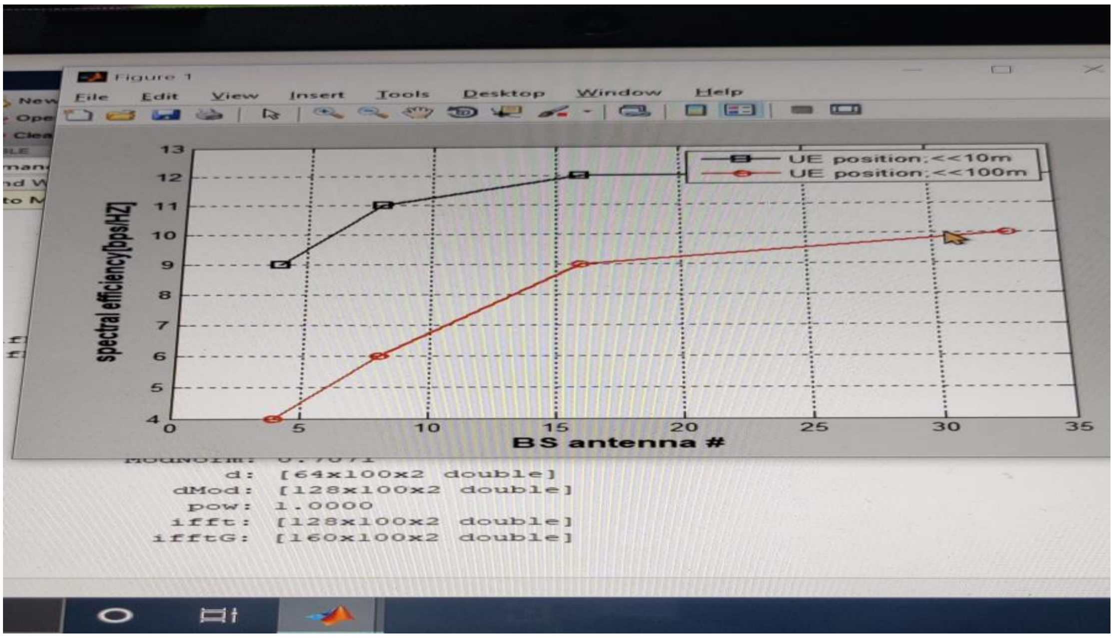
Chapter 6

RESULTS

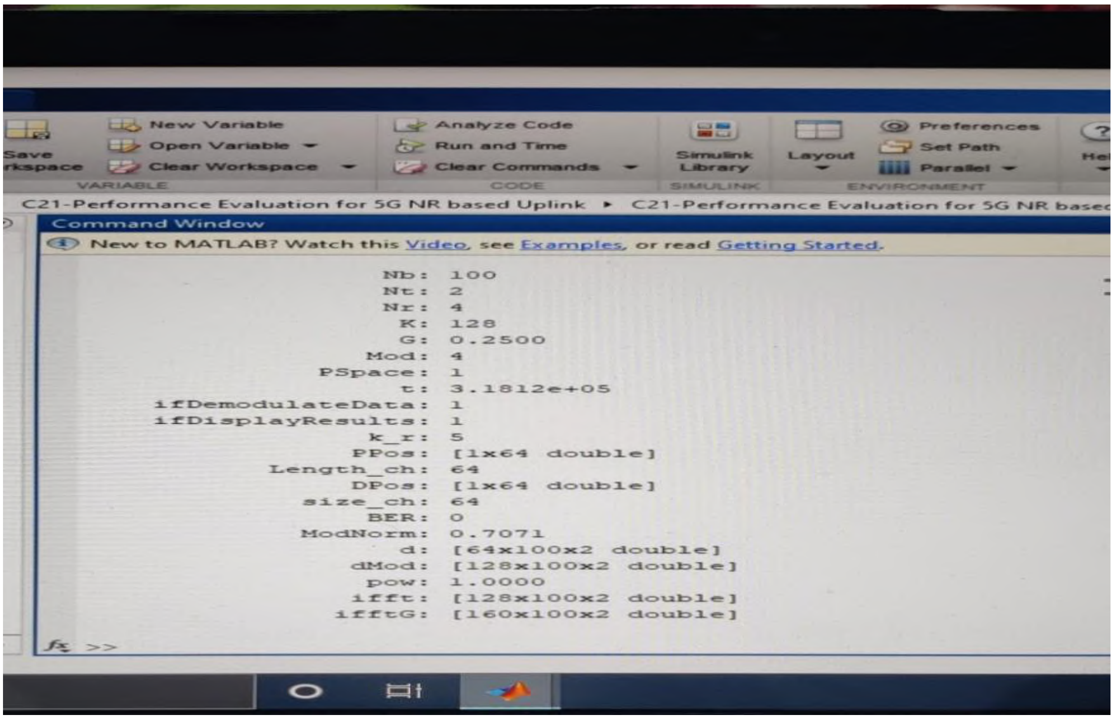
6.1 GRAPH 1:



6.2 GRAPH 2:



6.3 INPUTS OF PROGRAM:



Chapter 7

CONCLUSION

5G NR transmission frame and investigates the uplink millimeter-wave massive MIMO system under UMi scenario. The effect on the system performance according to the separation distance is investigated, and the system performance considering the number of transmit and receive antennas is also investigated. The simulation results show that high spectral efficiency can be obtained in a case where the separation distance between BS and UE is close. Also, as the number of transmit and receive antennas increases, the spectral efficiency that can be acquired increases by the spatial multiplexing MIMO technique. Based on the results of this study, further studies will be carried out to effectively estimate the millimeter wave channel and to reduce the excessive feedback overhead of the massive MIMO systems.

REFERENCES

[1] Thomas L. Marzetta, "Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas," IEEE Transactions on Wireless Communications, vol. 9, no. 11, November 2010

[2] H. Ji, et al., "Overview of Full-Dimension MIMO in LTE-Advanced Pro," IEEE commun. Magazine, vol. 55, no. 2, 2017

[3] X. Chen, J. Lu, P. Fan, and K. B. Letaief, "Massive MIMO Beamforming With Transmit Diversity for High Mobility Wireless Communications," IEEE Access, Vol. 5, pp. 23032 - 23045, October 2017

[4] A. L. Swindlehurst, E. Ayanoglu, P. Heydari, and F. Capolino, “Millimeter-wave massive MIMO: the next wireless revolution?” IEEE Commun. Mag., vol. 52, no. 9, pp. 56–62, Sep 2014

[5] A. Alkhateeb, G. Leus, and R. W. H. Jr., "Limited Feedback Hybrid Precoding for Multi-User Millimeter Wave Systems," IEEE Trans. Wireless Comm., vol. 14, no. 11, November 2015

[6] J. Noh, T. Kim, J.-Y. Seol, C. Lee, "Zero-Forcing Based Hybrid Beamforming for Multi-User Millimeter Wave Systems," IET Communications, vol. 10, Iss. 18, pp. 2670–2677, 2016

[7] R. W. H. Jr, N. González-prelcic, S. Rangan, W. Roh, and A. M. Sayeed, “An overview of signal processing techniques for millimeter wave MIMO systems,” IEEE J. Sel. Top. Signal Process., vol. 10, no. 3, pp. 436–453, Apr 2016

[8] E. Bjornson, et al., "Massive MIMO in Sub-6 GHz and mmWave: Physical, Practical, and Use-Case Differences," IEEE Wireless Communications (Early Access), pp. 2-10, January 2019

[9] 3GPP Technical Specification TS 38.211, “Physical channels and modulation (Release 15),” V15.0.0, 2017

[10] 3GPP Technical Reports TR 38.901, “Study on channel model for frequencies from 0.5 to 100 GHz (Release 14)”, V14.3.0, 2017

A

MAJOR PROJECT REPORT

On

IOT BASED TOLL BOOTH MANAGER SYSTEM

Submitted by

1) Ms. K. Poojitha (17K81A04L2) **2) Mr. Kanduri Karthik Reddy (17K81A04L3)**

3) Mr. Karnam Thanoj Kumar (17K81A04L4)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Esteemed Guidance of

Mr.S.P.MANIKANTA

Associate Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE
An Autonomous Institute

Dhulapally, Secunderabad – 500 100

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

AN UGC AUTONOMOUS INSTITUTE

Dhulapally, Secunderabad-500 100

NBA & NAAC A+ Accredited

Department Of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the project entitled "IOT BASED TOLL BOOTH MANAGER SYSTEM", is being submitted by 1.**Ms.K.Poojitha(17K81A04L2), 2.Mr.Kanduri Karthik Reddy(17K81A04L3), 3.Mr.Karnam Thanoj Kumar(17K81A04L4)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

PROJECT GUIDE	**HEAD OF THE DEPARTMENT**
Mr. S. P. MANIKANTA	**Dr. B. HARI KRISHNA**
Associate Professor	**Professor**
Department of ECE	**Department of ECE**

Internal Examiner   External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT K POOJITHA WITH ROLL NO.17K81A04L2, KANDURI KARTHIK REDDY WITH ROLL NO.17K81A04L3, KARNAM THANOJ KUMAR WITH ROLL NO.17K81A04L4, OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "IOT BASED TOLL BOOTH MANAGER SYSTEM" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of **Bachelor of Technology** in Department of ELECTRONICS AND COMMUNICATION ENGINEERING, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled "IOT BASED TOLL BOOTH MANAGER SYSTEM" is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

1) Ms. K. Poojitha(17K81A04L2)

2) Mr. Kanduri Karthik Reddy(17K81A04L3)

3) Mr. Karnam Thanoj Kumar(17K81A04L4)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to our College Management for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr. P. SANTOSH KUMAR PATRA**, our beloved **Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr. B. HARI KRISHNA**, **Head &Professor**, Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in our memory for years to come.

We would like to express our gratitude to our dept. project coordinator **Dr. A. ANAND, Professor** for his involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time to come.

We take it as privilege to thank our Project Coordinator, **Ms. P. PUSHPA**, **Assistant Professor**, Department of ECE for the ideas that led to complete the Project work and we also thank her for her continuous guidance, support and unfailing patience, throughout the course of this work.

Finally, we are highly grateful and obliged to our project guide **Mr. S. P.MANIKANTA, Associate Professor,** for his advice, supervision, and the vital contribution as and when required during this research work.

K. POOJITHA	K. KARTHIK REDDY	K. THANOJ KUMAR
(17K81A04L2)	(17K81A04L3)	(17K81A04L4)

ABSTRACT

Managing multiple toll booths is a very complicated task. We here propose a smart card based toll booth system that is monitored over IOT. The Internet server maintains all the data of user accounts and also their balance. All vehicle owners would possess a rfid based card that stores their account number. Our system at toll booths will monitor the cards scanned when a car arrives at the toll booth. The system now connects to the online server to check if the card is valid and if valid what is the balance. If user balance is sufficient, the user balance is deducted online and web system sends signal back to the card scanner system that the user has been billed. On receiving this signal, the system operates a motor to open the toll gate for that car. The system is controlled by a microcontroller to achieve this purpose. The microcontroller uses wifi connection to connect to the internet through which system interacts with web server to perform the online verification process. Also, system allows to store data of all the vehicles passed at particular time intervals for later reference and surveillance. This system thus automates the entire toll booth collection and monitoring process with ease using RFid plus IOT based system.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY OF COMMONLY USED TERMS:

Active Tag – An RFID tag that uses a battery to power its microchip and communicate with a reader. Active tags can transmit over the greatest distances (100+ feet). Typically, they can cost $20.00 or more and are used to track high value goods like vehicles and large containers of merchandise.

Agile Reader – A reader that can read different types of RFID tags – either made by different manufacturers or operating on different frequencies.

Antenna – A device for sending or receiving electromagnetic waves.

Anti-Collision – A feature of RFID systems that enables a batch of tags to be read in one reader field by preventing the radio waves from interfering with one another. It also prevents individual tags from being read more than once.

Automatic Data Capture (ADC) – Methods of collecting data and entering it directly into a computer system without human intervention. Automatic Identification (Auto-ID) Refers to any technologies for capturing and processing data into a computer system without using a keyboard. Includes bar coding, RFID and voice recognition.

Auto-ID Center – A group of potential RFID end users, technology companies and academia. The Auto-ID center began at the Massachusetts Institute of Technology (MIT) and is now a global entity. It is focused on driving the commercialization of ultra-low cost RFID solutions that use Internet like infrastructure for tracking goods throughout the global supply chain. The Auto-ID Center organization is now EPC global.

Electronic Product Code (EPC)

A standard format for a 96-bit code that was developed by the Auto-ID Center. It is designed to enable identification of products down to the unique item level. EPC's have memory allocated for the product manufacturer, product category and the individual item. The benefit of EPC's over traditional bar codes is their ability to be read without line of sight and their ability to track down to the individual item versus at the SKU level.

EPC global – The association of companies that are working together to set standards for RFID in the retail supply chain. EPC global is a joint venture between EAN International and the Uniform Code Council, Inc.

Radio Frequency Identification (RFID)

A method of identifying items uniquely using radio waves. Radio waves do not require line of site and can pass through materials like cardboard and plastic but not metals and some liquids.

Read Range

The distance from which a reader can communicate with a tag. Several factors including frequency used orientation of the tag, power of the reader and design of the antenna affect range.

Reader

Also called an interrogator. The RFID reader communicates via radio waves with the RFID tag and passes information in digital form to the computer system. Readers can be configured with antennas in many formats including handheld devices, portals or conveyor mounted.

Read Only Tags

Tags that contain data that cannot be changed. Read only chips are less expensive than read-write chips

Read-Write Tags

RFID chips that can be read and written multiple times. Read/Write tags can accept data at various points along the distribution cycle. This may include transaction data at the retail point of sale. They are typically more expensive than read only tags but offer more flexibility.

RFID Transponder

Another name for a RFID tag. Typically refers to a microchip that is attached to an antenna, which communicates with a reader via radio waves. RFID tags contain serial numbers that are permanently encoded, and which allow them to be uniquely id

"WORM" Chip (Write Once Read Many) and then becomes "Read only" afterward

CHAPTER-1
INTRODUCTION

1.1 OVERVIEW OF THE PROJECT:

With the growth in the number of vehicles the need for expansive roads catering to thousands of vehicles moving across India has become inevitable. However, considering the present situation the current toll system has several drawbacks. Due to the limited number of toll booths and slow collection process, the average waiting time per vehicle is 10 minutes. This results in losses worth thousands of crores of Rupees in terms of fuel wastage. This long wait time often results in drivers getting irritated resulting in verbal spats and physical fights among people and the toll attendants. Several such incidents have been reported in the press with some of these fights even resulting in the death of the toll plaza attendants. In addition, there are numerous cases of toll plaza accidents which happen due to the sudden lane changing by drivers for faster clearance. The major reason behind this is that, the security at the tolls is insufficient and it is beyond the traffic polices control to manage the vast number of vehicles. We keep hearing of many such incidents at toll plazas which mostly occur due to negligence either on the people's side or due to lack of control from the government agencies including the police. In case of events, where lives are lost, such losses are a life shattering experience.

As is well known, in such a scenario, the general public is a little hesitant in taking responsibilities of any such incident. Hence the government has to come up with an effective plan which bridges the gap between the toll management and the public expectation of the service that they experience. Introduction of an elective toll plaza operation plan by the government, its strict implementation and monitoring which would result in a more efficient and a more responsive and efficient system could be a good option for easing the challenges associated with the existing tolling process.

The requirements for new web applications supporting different types of devices an purposes are continuously growing. The main advantages of web application development as well as popular development features covering integration with different technologies are considered. Integration and possibilities of application of cloud-based web applications in real scenarios with different embedded Internet of Things (IoT) devices are considered and described. The design and implementation of a cloud-based web application supporting vehicle toll payment system using IoT device is presented and described. The development framework as well as featured and popular technologies used to realize a vehicle toll payment by IoT device are described. The concept of vehicle toll payment over an online payment system is also described. Processing, monitoring and control in the cloud-based web application of such payments using IoT devices are described and presented. Also, system allows to store data of all the vehicles passed at particular time intervals for later reference and surveillance. This system thus automates the entire toll booth collection and monitoring process with ease using RFID plus IoT based system.

1.2 OBJECTIVES OF THE STUDY:

The use of automated toll collection system in many metropolitan cities would be an efficient step towards the overcrowding of the city highways in heavy congestion of traffic. As we all know, transportation is the strength of our country's economy. There are various implementation, protocols in wireless sensor network and components such as RFID thus enabling reduction in operation costs and motivating cashless transactions. In case of manual toll collection system time consumption is much far worse as well as fuel depletion and most important is the environment, the amount of air pollution that is created at the toll booth site is at high level, so our developed system will reduce time wastage and not only reduce air pollution but also conserve fuel. The sole purpose of this paper is to reduce the hardships caused by manual toll collection system and pass the subject's vehicle through toll barrier in a matter of few seconds without halt.

Each and every day more and more vehicles are increasing rapidly and the graph of the rate of buying vehicles is exponential, which has become a major problem at the site of toll booths due to heavy traffic causing endless number of problems such as high petrol/diesel consumption leading to depletion of hydrocarbon deposits below earth's crust and also death causalities due to heavy traffic. The sole purpose of this paper is to motivate cashless transactions by installing automated e-toll collection system and the technology that we used is the use of RFID readers/tags. Basically to tackle this problem, the use of RFID tags that must be uniquely fixed onto subject's vehicle and RFID reader module must be fixed at e-toll tooth. When a subject's vehicle passes through the gate, the reader will detect an incoming frequency of 125 kHz of the RFID tag and read a unique no that has been assigned by govt. authority and the toll fee will be deducted from linked bank a/c or the e-wallet of system. This system is capable of saving time as well as fuel conservation which can save a lot of individual's economy. This particular system is far much better and very efficient towards people as they will not stay in a long and lengthy queue thus automated e-toll system will eliminate the hardships of people parking vehicles in a long queue. RFID has the potential of eliminating corruption at local level and also reduce operational costs as well as errors in human operations. WSN's i.e. wireless sensor networks are basically used in different scenarios such as home, office, healthcare, agriculture and also at toll collection plaza which can capture and transmit data from all incoming vehicles and outgoing vehicles because of their consistent and distinctive properties.

1.3 SCOPE OF THE STUDY:

- **IOT (internet of things)** based toll booth manager system is such type of a system that provides the automatic toll collection process with the help of internet resources and RFID (radio frequency identification) technology.

- If we talk about accountability, then manually toll collection process is very complicated because it is not so much easy to manage hundred and thousands of vehicles at same time.

- It required more man power as well as with more toll collection booth, which could be so much costly and difficult for toll collection department.

- On other hand, so much time is wasted by the vehicle owners when they are stand in so much long toll collection row.

- To manage all these above situations and to facilitate the vehicles owners, so many toll booth manager systems are available in market but that one's are so much costly as well as they are not so much efficient, reliable, precise and friendly to use.

- Here we have proposed a system that is called IOT based toll booth manager system which is so much efficient, reliable and friendly to use. In this system every vehicle owner possess a RFID based card, which would be scanned this card during pass through toll collection both.

1.4 MATERIAL REQUIREMENT:

1.4.1 HARDWARE REQUIREMENT:

- Arduino uno
- LCD
- Power Supply
- RFID
- Buzzer
- WIFI
- L293D
- Motor

1.4.2 SOFTWARE REQUIREMENT:

- Proteus software
- Arduino software

1.5 PROCUMENT OF EQUIPMENT:

We brought all the required hardware components in an electronic manufacturing shop andbefore assembling and interfacing hardware components to Arduino UNO micro-controller, we connected individual components to Arduino in Proteus Software and understood working of every individual component in Proteus Software.

Now, we assembled all components and interfaced to Arduino UNO micro-controller using connecting wires through bread board and we wrote program to work all the hardware components in Arduino IDE we understood the working of all components in a practical way.

We were able to monitor the toll booth system with better efficiency, with less time delay and the results were also satisfactory and we successfully got the output in mobile telnet app and LCD screen. We would like to improve the project in future for further developments.

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH AREA:

The ATC system in the paper "A review on automated toll collection system" the two types of study were compared namely the RFID based study and Leach-C based study showed promising and efficient way to develop the system but it also showed us lack of a point where the Leach-C system worked on digital image processing where the number plate of subject's vehicle was scanned using high-tech camera. Using the RFID based implementation and eliminating the DIP based implementation to make system better time efficient. The survey done in the paper named "A 5.8 GHz ISM band Microstrip antenna for RFID applications" showed us the path using and developing RFID antenna for scanning the details of subject's vehicles from a handy distance thus enhancing the distance of vehicle and RFID reader would cover a large area. Smart Toll Collection System based on IoT (IJSTE/ Volume 3 / Issue 12 / 002) All rights reserved by www.ijste.org 10 The review was done of the paper "Automated toll collection system using RFID" Enlighted us the various components that can used while developing our own system in time saving and in efficient way so as to make a hybrid system that is mentioned in the paper "A review on automated toll collection system". "A Survey in Image Mining" showed a path towards machine learning and data mining concepts and helped us in mining databases in the form of people's feedback.

2.2 REVIEW ON RELATED LITERATURE:

The first person to propose electronic toll system was William Vickrey. He proposed this system for Washington Metropolitan Area. He was also the recipient of Nobel Economics prize in 1959 [5] The transponder and reader technology spread all around Norway. The system was broached in the Bergen (1986). World"s first completely automated electronic toll collection system was introduced in Trondheim (1991). Norway too has electronic fee collection EFC. [5] In different places in the United States like California, Pennsylvania, Texas, Delaware and Florida, vehicles can pass through electronic toll collecting booths directly. For the same no of cars, operations in electronic toll booths take place faster as the users are preregistered. Toll collecting gates are usually broader than the original width of the road but this made it possible to fit them into smaller narrower roads. Although these limitations are present, if the time taken for operation at the toll booth is reduced, then the speed of operation can be increased. The greater the speed of operation, less number of total lanes was required and as more number of lanes started getting automated, there was a reduction in the overall short term cost of the project. Also if the long term cost of the project is considered, larger number of vehicles started using electronic toll and vehicles using manual toll reduced. [5] According to references, the methodology proposes use of Infrared sensors to gather real time data from the waste bins and that of the microcontroller board to communicate this information to the waste managers.

These sensors collect information about the objects, their surroundings and communicate this information to other stations, linked through wired or wireless networks. The system will check using the IR sensors when the bin becomes full and will notify the waste managers upon filling up of the bins and will provide an optimal and effective collection route. [10] According to reference, The initial technical realization of IoT was achieved by utilizing RFID (Radio Frequency Identification) technology, which was limited to identification, object tracking and extracting information of specific objects. However, The implemented IoT based system performs sensing, actuating, data gathering, storing, and processing by connecting physical or virtual devices to the Internet. [10] According to reference, To facilitate vehicle monitoring, toll collection and faithful vehicle authentication on the highways and to have an efficient usage of communication link between RF Modems over a wireless channel, a module is proposed. There are two types of implementation modules-the Vehicle Module (Active RFID Tag) and the Base Module. Microcontroller contains user-specific information associated with vehicle, such as the owners information with his billing address, vehicle engine number and vehicle registration number. The base module allows the base module to check the activities of vehicles in range, including the vehicles in range, their status, and the detailed information about any registered vehicle.[6] According to reference, Open Road Tolling uses video evidence to identify vehicle usage of a toll facility without the use of toll booths for toll collection without having to stop or even slow down to pay the toll.

The application, for Open Road Tolling, utilizes pattern recognition and image processing methods. This paper presents Open Road Tolling (ORT) using number plate recognition. The proposed Number Plate Recognition (NPR) techniques consist of mainly two modules: histogram based number plate localization and number plate recognition using template matching, thus making it simple & faster. AVI technology uses Laser or Radio Frequency (RF) method. Laser systems uses barcode attached to the vehicle & read by vehicle scanner as the vehicle passes through the toll lane. [7] According to reference, The German Federal Ministry of Transport had conducted a fee collection experiment on German motorways called ChipTicket systems. The ChipTicket system assumes that vehicles are equipped with a chip card, and an in-vehicle unit. The in-vehicle unit contains transmitting and receiving facilities for communication with the toll stations - the so-called charge collection stations. The chip card is a plastic card contains an integrated microprocessor. When a driver passes a charge collection station, payment is made automatically by the vehicle the chip ticket is made and the fee is registered and stored in the station computer.[8]

A microcontroller has been programmed to operate a remotely operated positioning system of a satellite. Earlier, if we wanted to get an exact angle of the satellite, it needed manual adjustment. To overcome this drawback, this paper was aimed at developing a system to remotely operate the satellite. The IR signal from the remote (Transmitter) is sent and is received by the IR sensor (Receiver) which has been interfaced with the main microcontroller. The data from the transmitter is sent in an encoded format which is received by the receiver sensor and is the suitably sent to the main microcontroller. Corresponding signal is sent by the microcontroller to the motor driver which in turn rotates the motor and thereby the satellite accordingly. [9] According to reference, the ESP826 is a Wi-Fi module that has very efficient on chip storage and processing capabilities with lower cost and higher quality. ESP is based on IEEE 802.11 Wi-Fi protocol and it supports various network protocols. For connecting the machine to server a level shifter is used. [12]

This system on chip (SOC) design has been provided with a TCP/IP protocol stack which is used to provide Wi-Fi connection to any microcontroller that has been used by the system. There are three different methods to implement ESP8266. Using AT commands for communication is the simplest way that can be used. AT commands can be sent via the computer through a USB to serial adapter cable to the controller for setup and testing purposes. The second way is peripheral mode by interfacing compatible microcontroller with the module. The third method can be implemented by directly programming the GPIO pins of the module and interfacing them with external peripherals and sensors. An ESP8266 device can be used as an access point or as a station or both at the same time. Usually the access point also has a network data connection.

Therefore itcan act as a bridge between the wireless network and the TCP/IP network which is the internet. Once it has been finalized which mode the device will be used in, then we set a global mode which indicates the state in which the device will be used (As an access point or a station or both). Wi-Fi module is connected to the controller through this level shifter and have used SPI protocol.. Here considering the need for consistency of data we have utilized TCP/IP network protocol for communicating with the server [11] The ESP8266 is designed to occupy minimal PCB area, without compromising on providing maximum inbuilt features. Cost Effective - Among the major advantages of the module include its price. You can get a module at lower than $5. Power - The power saving architecture operates in 3 modes: active mode, sleep mode and deep sleep mode, thus yielding very power efficient system. Ease of Interfacing - Module can be easily calibrated when required by reprogramming it for interfacing wide range of sensors.[13]

There are various of protocol is used in IoT (Internet of Things) devices. One of the most useful protocols is MQTT (Message Queuing Telemetry Transport). Second basic protocol is TCP. MQTT is broker based publishing/subscribing messaging protocol. It is open, lightweight, and very easy to implement. Moreover it has low bandwidth protocol.[14] The ATmega328 is the microcontroller used for this project. It shows important specifications like 14 digital input/output pins, 6 analog inputs, a USB connection, a 16 MHz ceramic resonator, an ICSP header, a power jack, and a reset button. The Arduino can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an ACto-DC adapter (wall-wart) or battery. The board can operate on an external supply of 6 to 20 volts. However it suffers form possibilities of overheating which may damage the board. The recommended range is 7 to 12 volts.[21] Controller –ATmega328 controller has been used which is an 8-bit microcontroller based on AVR. It has built-in communication protocol like I2C and UART and many other essential features. Real Time Clock-RTC module of DS1307 is connected to ATmega328 through two wire interface for reading the on and off time of the machine. Level shifter: MOSFET based level shifter is needed to drop down voltage in between, as ATmega328 operate on 5V and ESP-12 on 3.3V.[12]

2.3 CONCLUSION ON REVIEWS:

The ATC system in the paper “A review on automated toll collection system” the two types of study were compared namely the RFID based study and Leach-C based study showed promising and efficient way to develop the system but it also showed us lack of a point where the Leach-C system worked on digital image processing where the number plate of subject’s vehicle was scanned using high-tech camera. Using the RFID based implementation and eliminating the DIP based implementation to make system better time efficient. The survey done in the paper named “A 5.8 GHz ISM band Microstrip antenna for RFID applications” showed us the path using and developing RFID antenna for scanning the details of subject's vehicles from a handy distance thus enhancing the distance of vehicle and RFID reader would cover a large area.

The review was done of the paper "Automated toll collection system using RFID" Enlighted us the various components that can used while developing our own system in time saving and in efficient way so as to make a hybrid system that is mentioned in the paper "A review on automated toll collection system". "A Survey in Image Mining" showed a path towards machine learning and data mining concepts and helped us in mining databases in the form of people's feedback. The survey done in the paper named ATC turned out to be good step for developing ATC as we tried the best option in choosing the components in terms of reducing operation costs, time saving, flexible and also added the theft detection module in our system enabling security on high alert and detecting of stolen vehicles. During the period of learning and reviewing we also tried in improving and eliminating the costs of GSM kits embedded on the micro-controller board by replacing by use of private SMS gateway or govt. running GSM Company, which has reduced the operation cost much and has made our system more independent of using GSM kit.

CHAPTER-3

PROJECT DESIGN

3.1 OVERVIEW OF THE DESIGN:

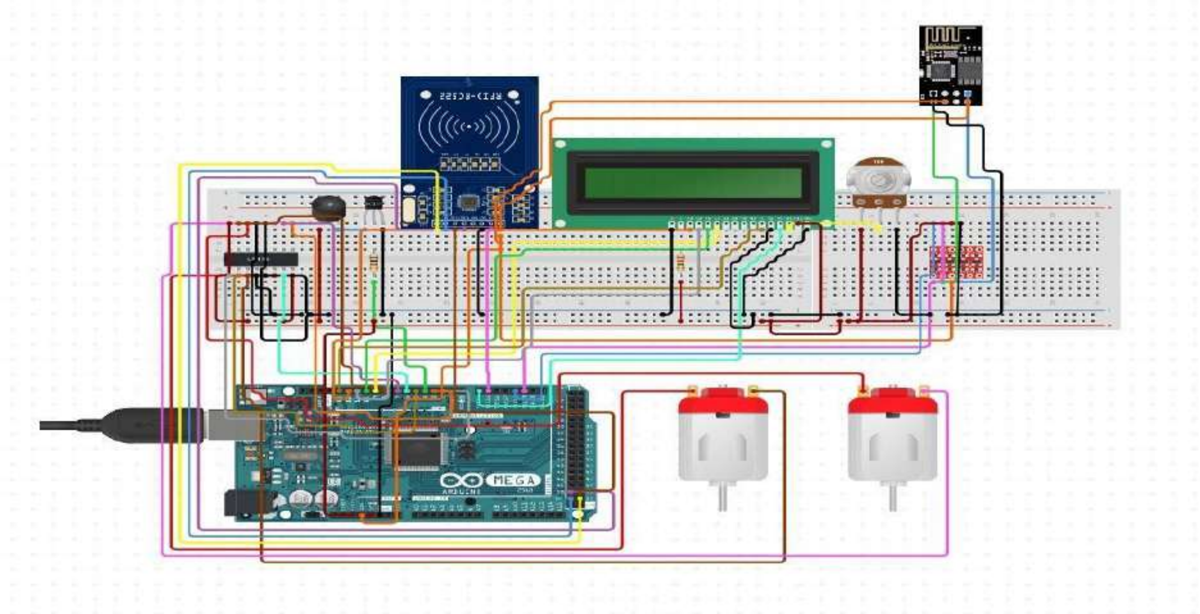


Fig.3.1 Overview of the design

3.2 EQUIPMENT ANALYSIS:

3.2.1 ARDUINO:

3.2.1.1 INTRODUCTION:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

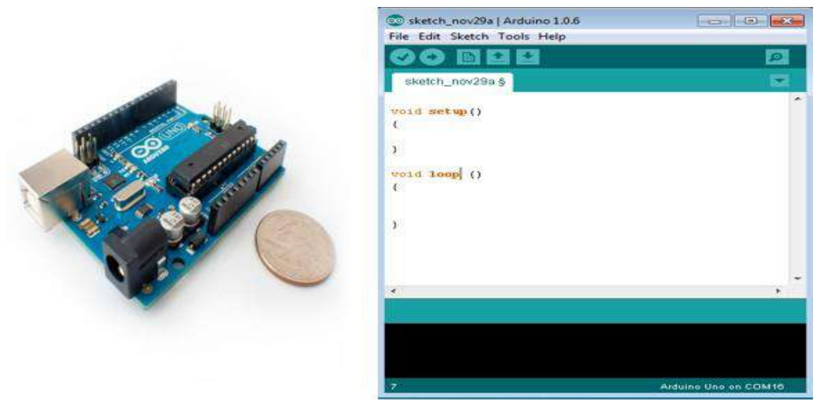


Fig 3.2.1.1 Arduino Uno

3.2.1.2 BOARD TYPES

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini-05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini-3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header

Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Lilypad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Lilypad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.2.1.2.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro-3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB

Lilypad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 3.2.1.2.2 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 3.2.1.2.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table 3.2.1.2.4 Arduino boards based on AT91SAM3X8E microcontroller

3.2.1.3 BOARD DESCRIPTION:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

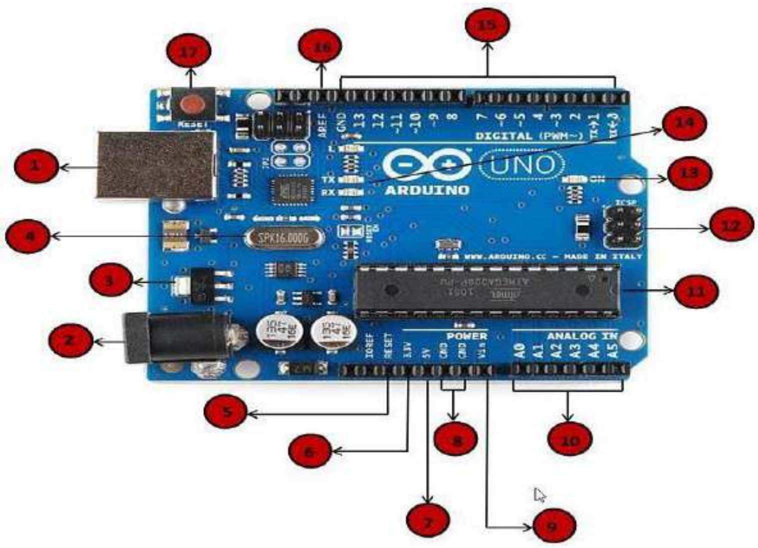


Fig.3.2.1.3 Board Description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.

5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator**

	This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Table 3.2.1.3 Board description of Arduino UNO

3.2.1.4 ARDUINO FAMILY

Arduino makes several different boards, each with different capabilities. In addition, part of being open-source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

Fig.3.2.1.4: Arduino Family

3.2.1.5 SHIELDS:

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

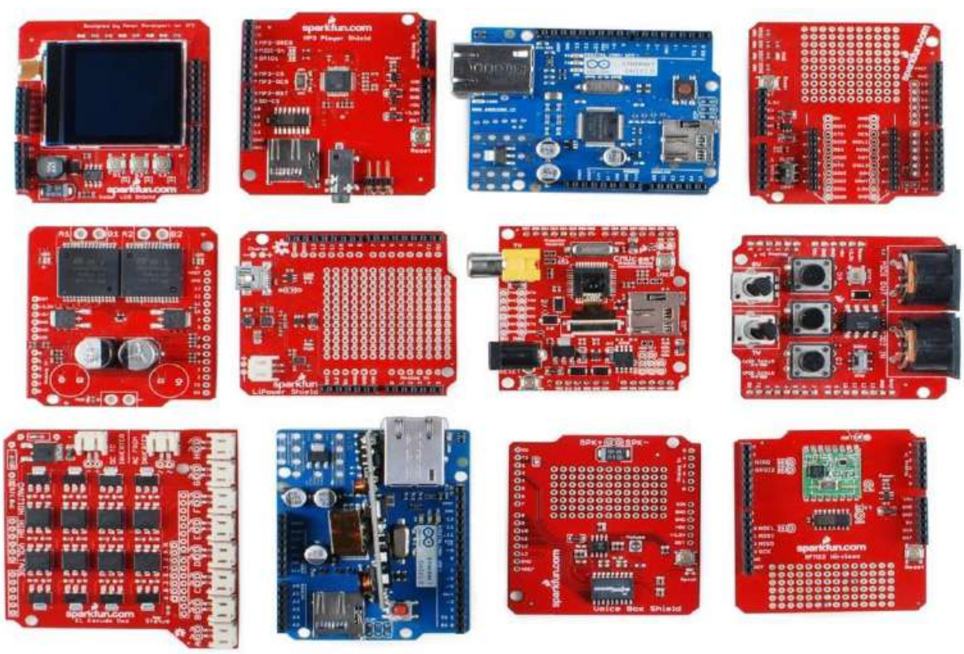


Fig.3.2.1.5: Arduino Shields

3.2.1.6 PIN DESCRIPTION OF ATMEGA328:

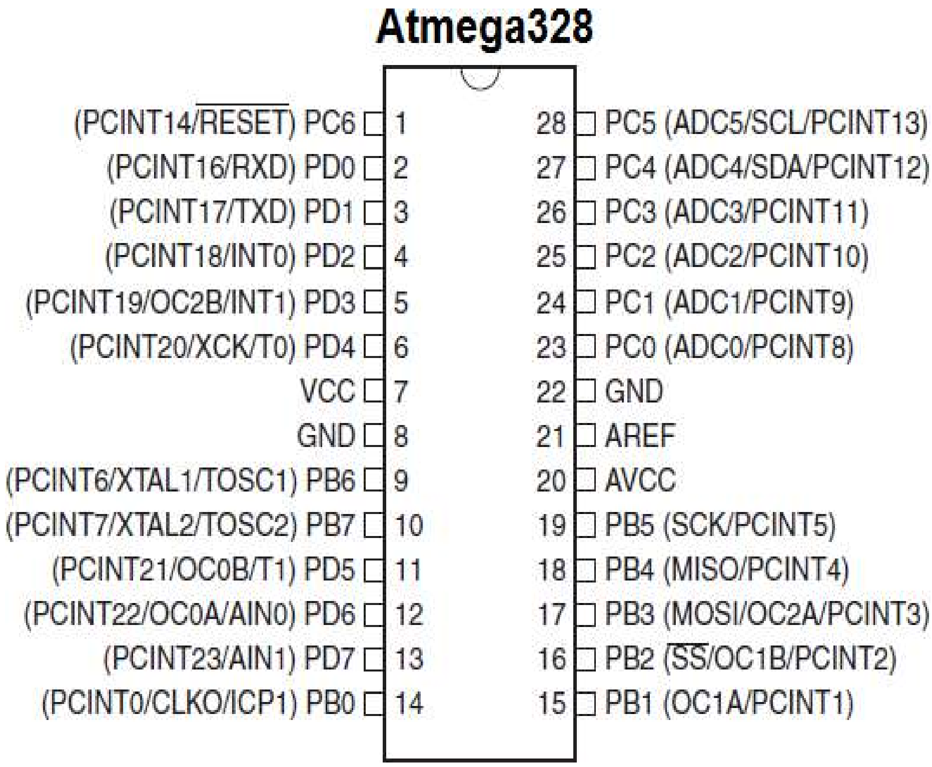


Fig.3.2.1.6 Pin description of ATMEGA328

3.2.1.7 APPLICATIONS:

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

3.2.2 16*2 LCD SCREEN DISPLAY

3.2.2.1 INTRODUCTION

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

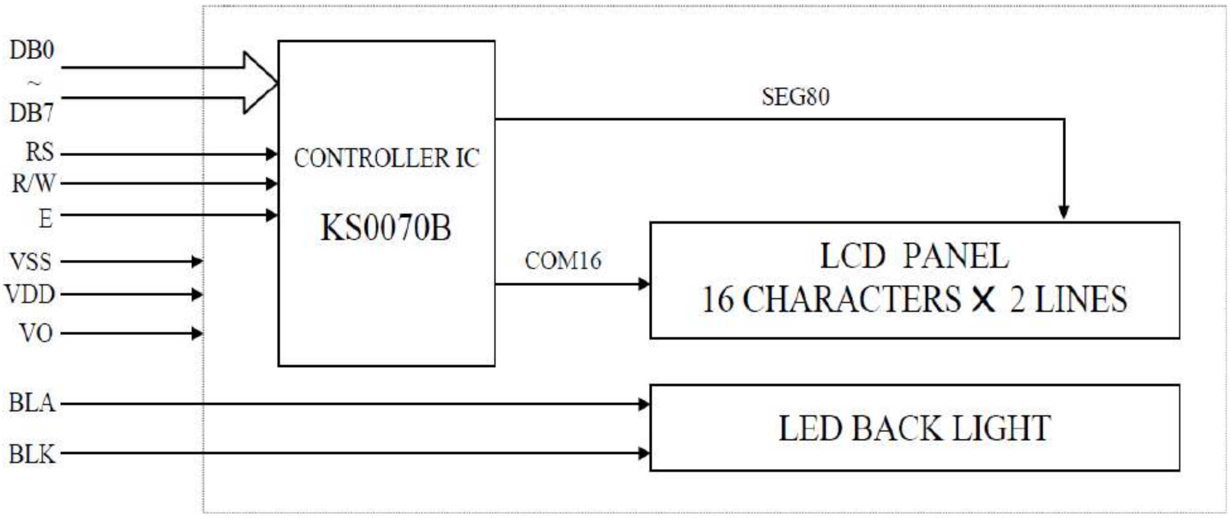


fig.3.2.2.1 Lcd block diagram

3.2.2.2 DISPLAY DATA RAM:

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user. Figure below will show you the DDRAM addresses of 2 Line LCD.

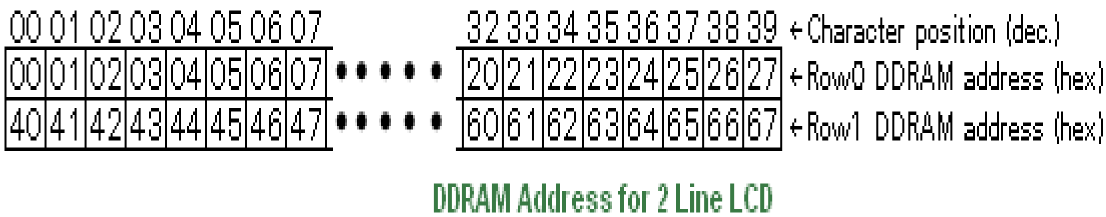


Fig.3.2.2.2 DDRAM addresses of 2 Line LCD

3.2.2.3 CHARACTER GENERATOR ROM:

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM.

3.2.2.4 BUS FLAG:

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e. BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

3.2.2.5 INSTRUCTION REGISTER (IR) AND DATA REGISTER (DR):

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g. LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD.

Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

3.2.2.6 16 X 2 ALPHANUMERIC LCD MODULE FEATURES:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

3.2.2.7 CIRCUIT DESCRIPTION:

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

3.2.3 BUZZER

3.2.3.1 MAGNETIC TRANSDUCER

Magnetic transducers contain a magnetic circuit consisting of a iron core with a wound coil and a yoke plate, a permanent magnet and a vibrating diaphragm with a movable iron piece. The diaphragm is slightly pulled towards the top of the core by the magnet's magnetic field. When a positive AC signal is applied, the current flowing through the excitation coil produces a fluctuating magnetic field, which causes the diaphragm to vibrate up and down, thus vibrating air. Resonance amplifies vibration through resonator consisting of sound hole(s) and cavity and produces a loud sound.

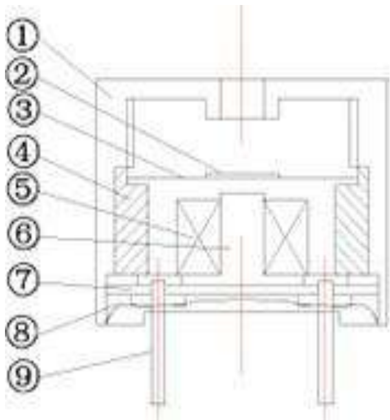


Fig.3.2.3.1 Magnetic Transducer

3.2.3.2 MAGNETIC BUZZER(SOUNDER)

Buzzers like the TMB-series are magnetic audible signal devices with built-in oscillating circuits. The construction combines an oscillation circuit unit with a detection coil, a drive coil and a magnetic transducer. Transistors, resistors, diodes and other small devices act as circuit devices for driving sound generators. With the application of voltage, current flows to the drive coil on primary side and to the detection coil on the secondary side. The amplification circuit, including the transistor and the feedback circuit, causes vibration. The oscillation current excites the coil and the unit generates an AC magnetic field corresponding to an oscillation frequency. This AC magnetic field magnetizes the yoke comprising the magnetic circuit. The oscillation from the intermittent magnetization prompts the vibration diaphragm to vibrate up and down, generating buzzer sounds through the resonator.

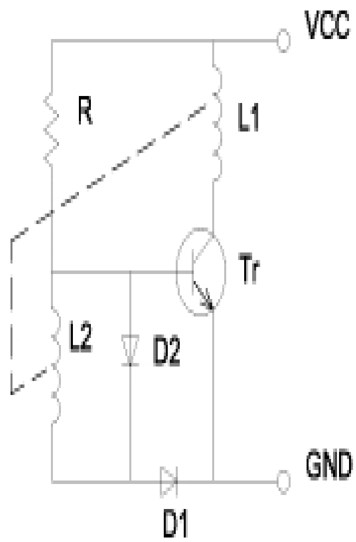


Fig.3.2.3.2 Magnetic Buzzer

3.2.3.3 SPECIFICATIONS:

Rated Voltage: A magnetic buzzer is driven by 1/2 square waves (V o-p).
Operating Voltage: For normal operating. But it is not guaranteed to make the minimum Sound Pressure Level (SPL) under the rated voltage.
Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.
Direct Current Resistance: The direct current resistance is measured by ammeter directly.
Sound Output: The sound output is measured by decibel meter. Applying rated voltage and 1/2 square waves, and the distance of 10 cm.
Rated Frequency: A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.
Operating Temp. : Keep working well between -30℃ and +70℃.
How to choose:

Driving methods: AX series with built drive circuit will be the best choice when we cannot provide frequency signal to a buzzer, it only needs direct current.

Dimension: Dimension affects frequency, small size result in high frequency.

Voltage: Depend on V o-p (1/2 square waves)

Fixed methods: From the highest cost to the lowest- DIP, wires/ connector, SMD.

Soldering methods: AS series is soldered by hand, the frequency is lower because of the holes on the bottom. On the other hand, we suggest AC series for the reflow soldering, the reliability is better.

3.2.3.4 HOW TO CHOOSE A BUZZER:

There are many different kinds of buzzer to choose, first we need to know a few parameters, such as voltage, current, drive method, dimension, mounting type, and the most important thing is how much SPL and frequency we want.

Operating voltage: Normally, the operating voltage for a magnetic buzzer is from 1.5V to 24V, for a piezo buzzer is from 3V to 220V. However, in order to get enough SPL, we suggest giving at least 9V to drive a piezo buzzer.

Consumption current: According to the different voltage, the consumption current of a magnetic buzzer is from dozens to hundreds of mill amperes; oppositely, the piezo type saves much more electricity, only needs a few mill amperes, and consumes three times current when the buzzer start to work.

Driving method: Both magnetic and piezo buzzer have self-drive type to choose. Because of the internal set drive circuit, the self-drive buzzer can emit sound as long as connecting with the direct current. Due to the different work principle, the magnetic buzzer need to be driven by 1/2 square waves, and the piezo buzzer need square waves to get better sound output.

Dimension: The dimension of the buzzer affects its SPL and the frequency, the dimension of the magnetic buzzer is from 7 mm to 25 mm; the piezo buzzer is from 12 mm to 50 mm, or even bigger.

Connecting way: Dip type, Wire type, SMD type, and screwed type for big piezo buzzer are usually seen.

Sound Pressure Level (SPL): Buzzer is usually tested the SPL at the distance of 10 cm, if distance double, the SPL will decay about 6 dB; oppositely, the SPL will increase 6 dB when the distance is shortened by one time. The SPL of the magnetic buzzer can reach to around 85 dB/ 10 cm; the piezo buzzer can be designed to emit very loud sound, for example, the common siren, are mostly made of piezo buzzer.

3.2.3.5 INTRODUCTION OF PIEZO BUZZER:

Piezo buzzers are simple devices that can generate basic beeps and tones. They work by using a piezo crystal, a special material that changes shape when voltage is applied to it. If the crystal pushes against a diaphragm, like a tiny speaker cone, it can generate a pressure wave which the human ear picks up as sound. Simple change the frequency of the voltage sent to the piezo and it will start generating sounds by changing shape very quickly!

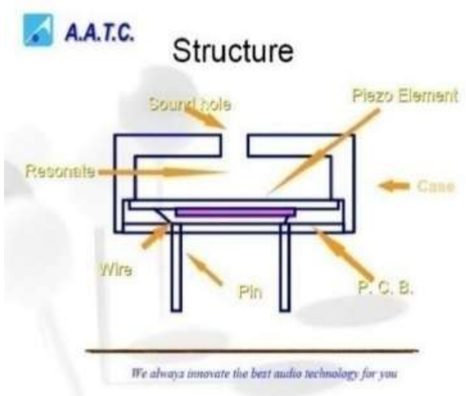


Fig.3.2.3.5 piezo buzzer

3.2.3.6 SPECIFICATIONS:

Rated Voltage: A piezo buzzer is driven by square waves(Vp-p).

Operating Voltage: For normal operating. But it is not guaranteed to make the minimum SPL under the rated voltage.

Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.

Capacitance: A piezo buzzer can make higher SPL with higher capacitance, but it consumes more electricity.

Sound Output: The sound output is measured by decibel meter.

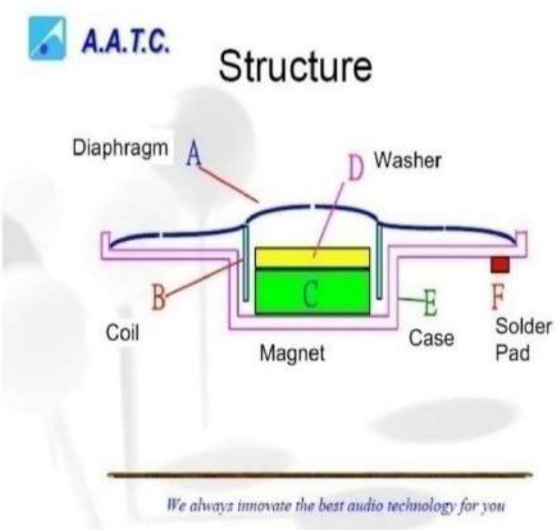


Fig. 3.2.3.6 structure of piezo buzzer

Applying rated voltage and square waves, and the distance of 10 cm.
Rated Frequency: A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.
Operating Temp.: Keep working well between -30℃ and +70℃.

3.2.4 L293D

3.2.4.1 INTRODUCTION

L293D is basically a high current dual motor driver/controller Integrated Circuit (IC). It is able to drive load having current up to 1A at the voltage ranging from 4.5V to 36V. Motor driver usually act as current amplifier because they receive a low current signal as an input and provides high current signal at the output.
L293D is basically a motor driver or controller. It has two built-in H-bridge circuits which are able to control two DC motors simultaneously in both clockwise and counter clockwise direction. It acts as an current high amplifier because it take low current signal at its input and provides higher current signal at the output in order to drives different load e.g. stepper motor & DC motors. Its features include large input voltage supply range, large output current high noise immunity input signals etc. Its common real-life applications include stepping motor drivers, relay drivers, DC motor drivers etc. L-293D motor driver/controller is shown in the figure given below.

Fig. 3.2.4.1 L293D Motor Driver

3.2.4.2 L293D PINS:

- L-293D has sixteen (16) pins, having different individual functions.
- All of these sixteen pins along with their serial no, are given in the table shown below.

3.2.4.3 L293D PIN FUNCTIONS:

- L 293D each pin has different tasks to perform when they are in working condition.

- The tasks assigned to each pin are given in the table shown below.

L293D Pins	
Pin. No	Pin Name
1	Enable 1,2
2	Input 1
3	Output 1
4	GND
5	GND
6	Output 2
7	Input 2
8	Vcc2
9	Enable 3,4
10	Input 3
11	Output 3
12	GND
13	GND
14	Output put 4
15	Input 4
16	Vcc1

Table 3.2.4.3 L293D Pin Functions

3.2.4.4 L293D PINOUT:

- Pinout diagram of any device presents the pins configuration through a completely labelled diagram.

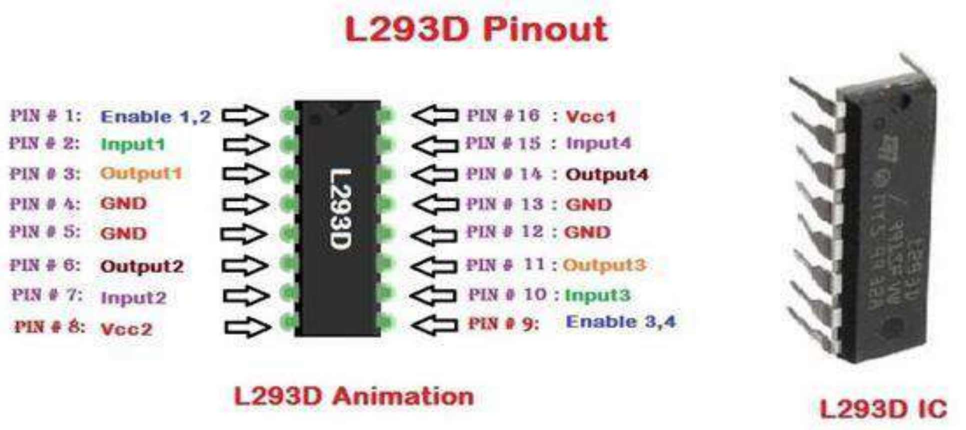


Fig. 3.2.4.4 L293D Pinout

3.2.5 MOTOR

Motors usually operates on this higher current. L-293D has to built-in H-Bridge driver circuits and is able to control two DC motors at a time in both clockwise and counter clockwise direction. It has two enable pins and they should be kept high in order to control the motor. By changing the polarity of applied signal motor can be rotated in either clockwise or counter clockwise direction. If L 293D enable pin is high, its corresponding driver will provide the desired out. If the enable pin is low, there will be no output. L-293D has different features including internal ESD protection, large voltage supply range, large output current per channel, high noise immunity input etc. L 293D plays a vital role in electronics era and has several different applications e.g., relay drivers, DC motor drivers, stepping motor drivers etc. The further detail about L 293D motor driver/controller will be given later in this tutorial.

3.2.6 SOFTWARE EXPLANATION:

- Arduino software
- Proteus simulation

3.2.6.1 ARDUINO SOFTWARE:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are
completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

3.2.6.2 WHY ARDUINO?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low-cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step-by-step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handy board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and
wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows. Wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers,

it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

- Open source and extensible software - The Arduino software is published as open-source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under

a Creative Commons license, so experienced circuit designers can make their own version Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money

- Getting Started with Arduino and Genuino products: -

Install the Arduino Software (IDE) on Windows PCs-
This document explains how to install the Arduino Software (IDE) on Windows machines.

✓ Download the Arduino Software (IDE)

✓ Proceed with board specific instructions.

3.2.6.3 HOW TO DOWNLOAD THE ARDUINO SOFTWARE (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.
When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

3.2.6.4 INSTALLATION:

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.3.2.6.4.1 Standard USB Cable (A plug to B plug)

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig. 3.2.6.4.2 Standard USB Cable (an A to Mini-B plug)

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

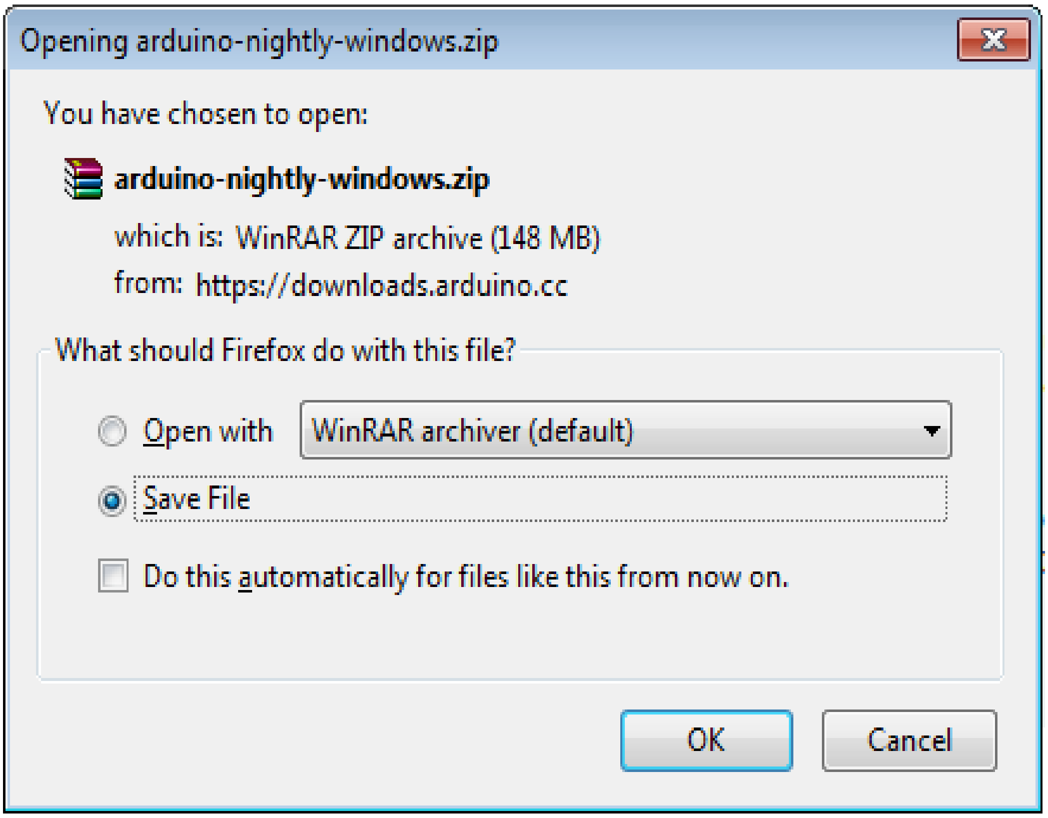


Fig. 3.2.6.4.3 Download Arduino IDE Software

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

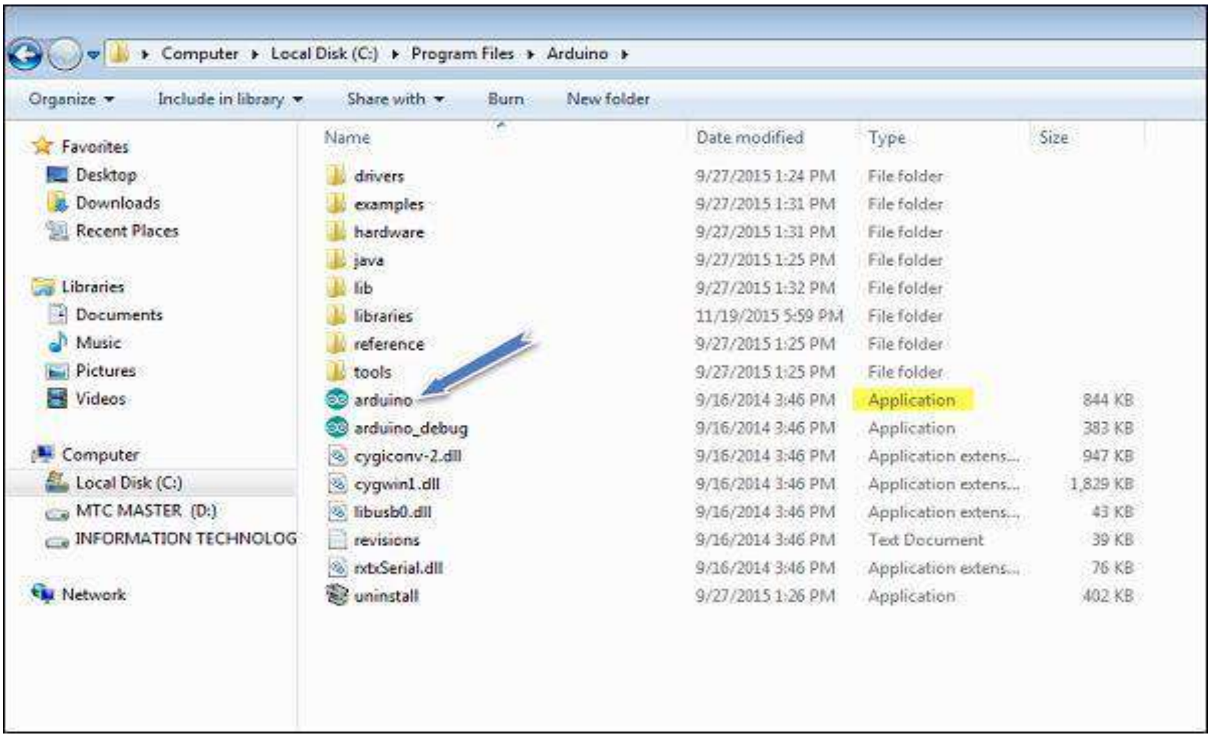


Fig. 3.2.6.4.4 Launch Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**

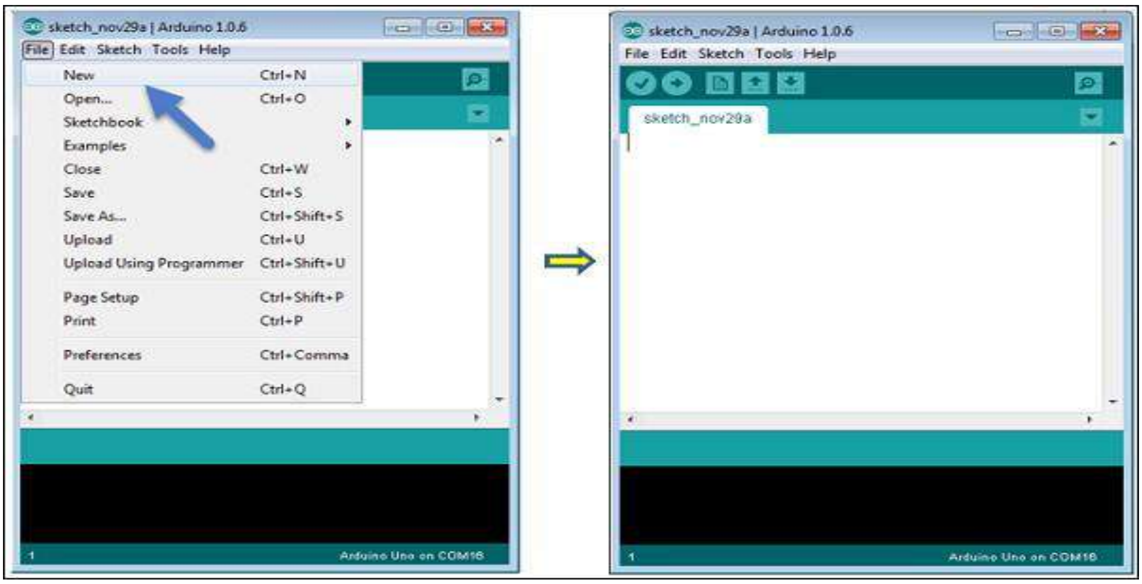


Fig. 3.2.6.4.5 Open your first project.

To open an existing project example, select File → Example → Basics → Blink.

.

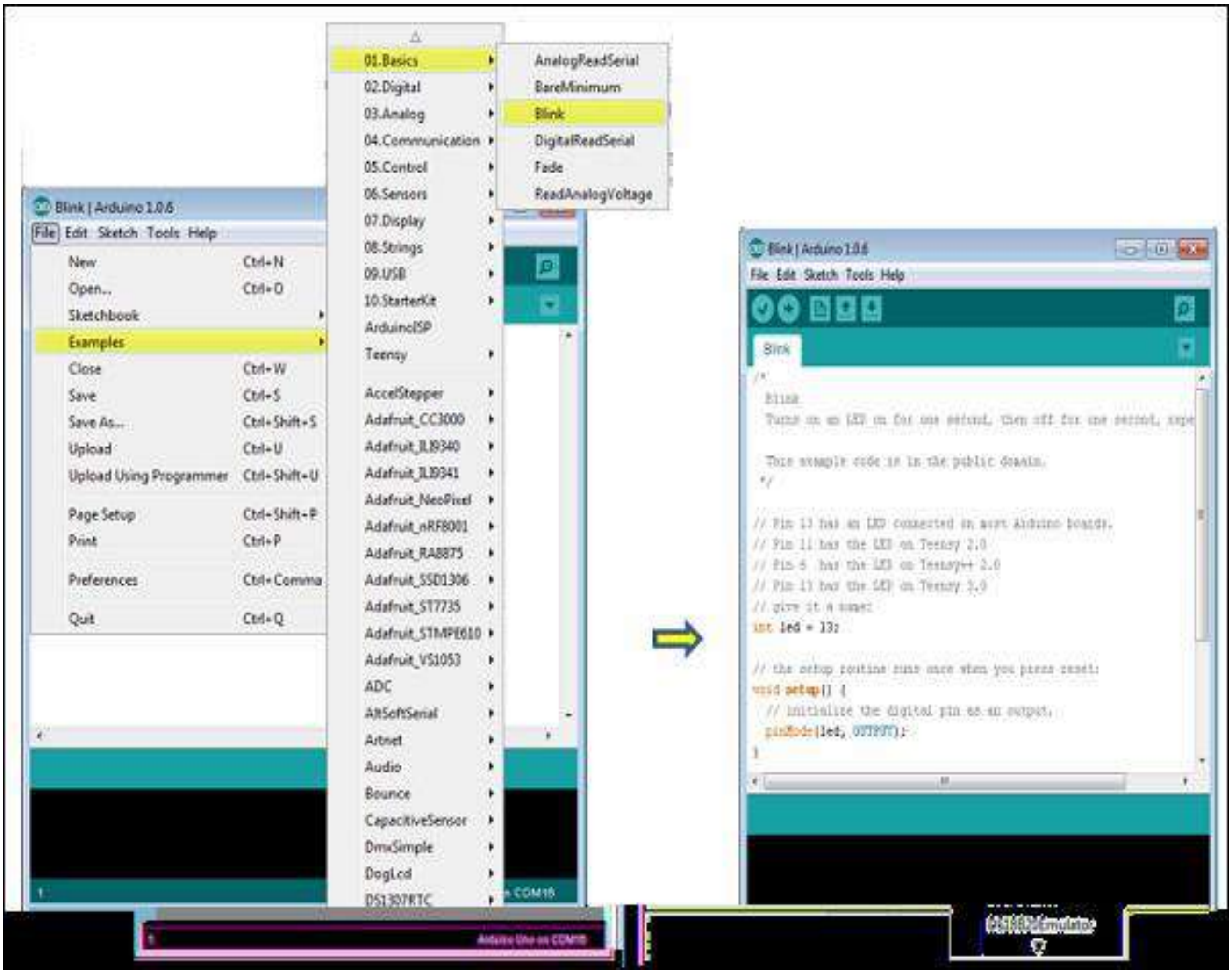


Fig. 3.2.6.4.6 Open your first program.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

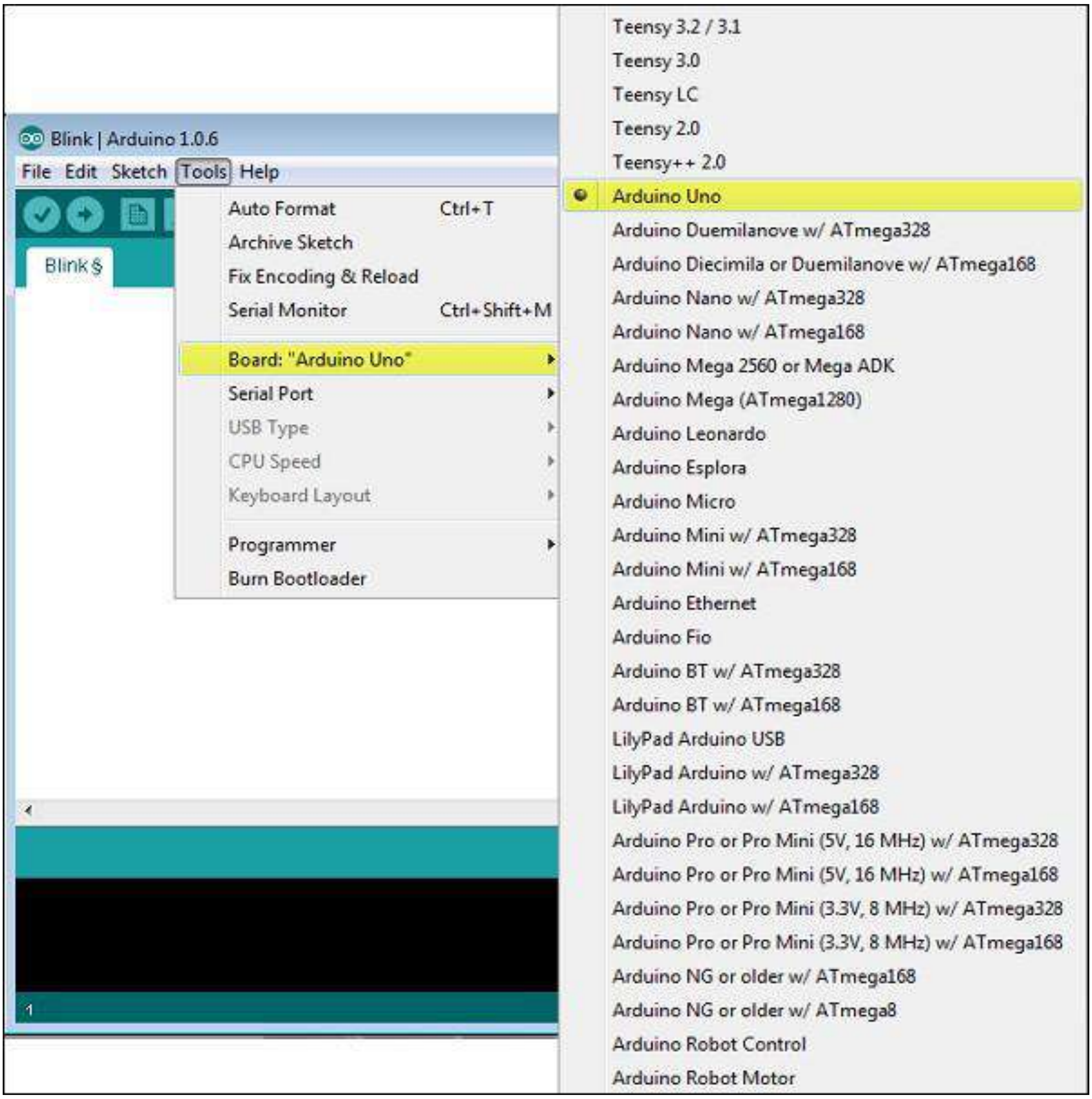


Fig. 3.2.6.4.7 Select your Arduino board.

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

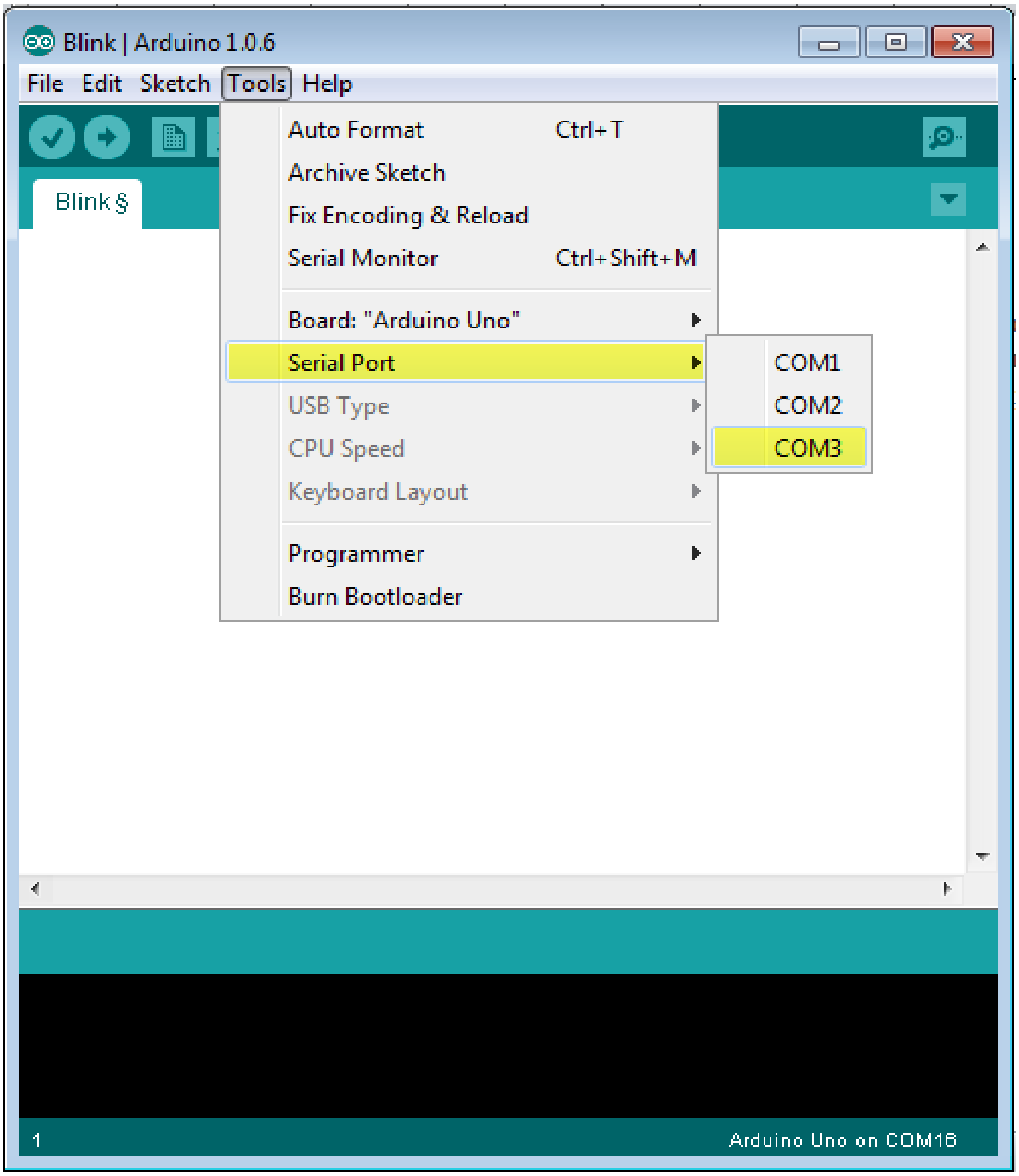


Fig. 3.2.6.4.8 Select your serial port.

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

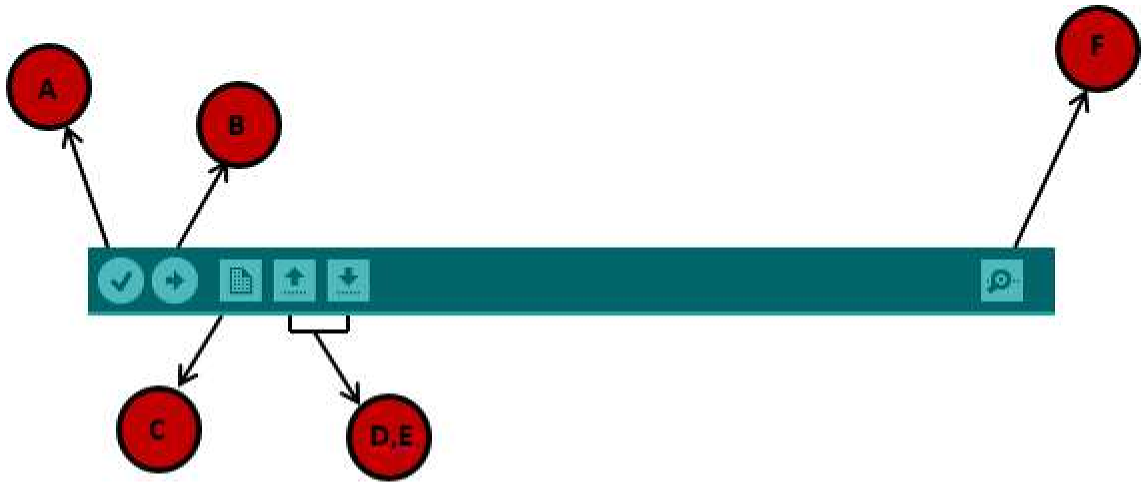


Fig. 3.2.6.4.9 Upload the program to your board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from board & send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

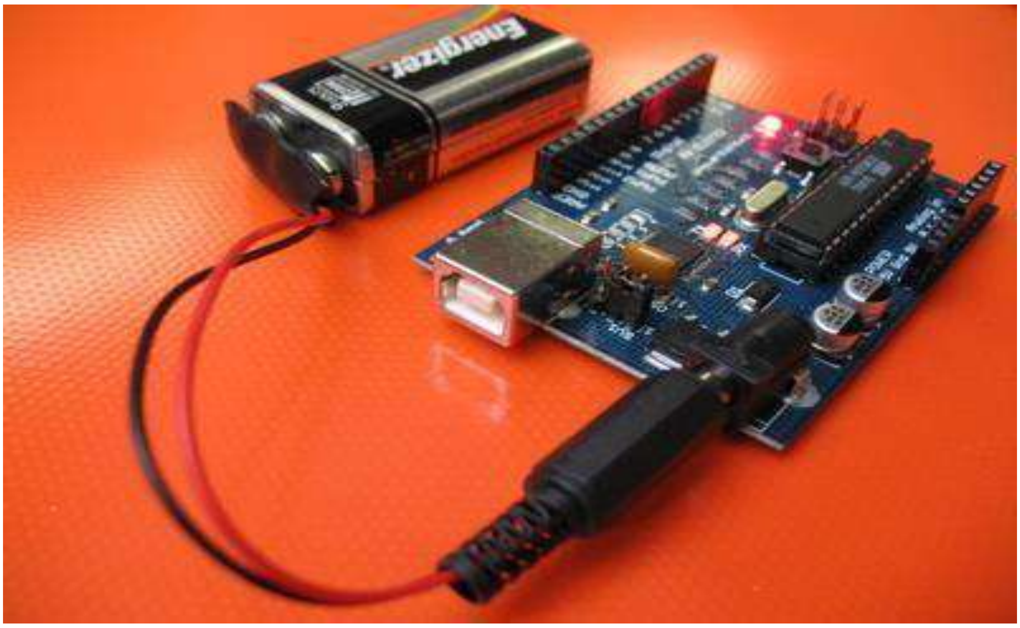


Fig. 3.2.6.4.10 Connecting a Battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called “sketches”, but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

Fig. 3.2.6.4.11 Example Program

3.2.7 PROTEUS:

PROTEUS:

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design:

Step 1: Open ISIS software and select New design in File menu

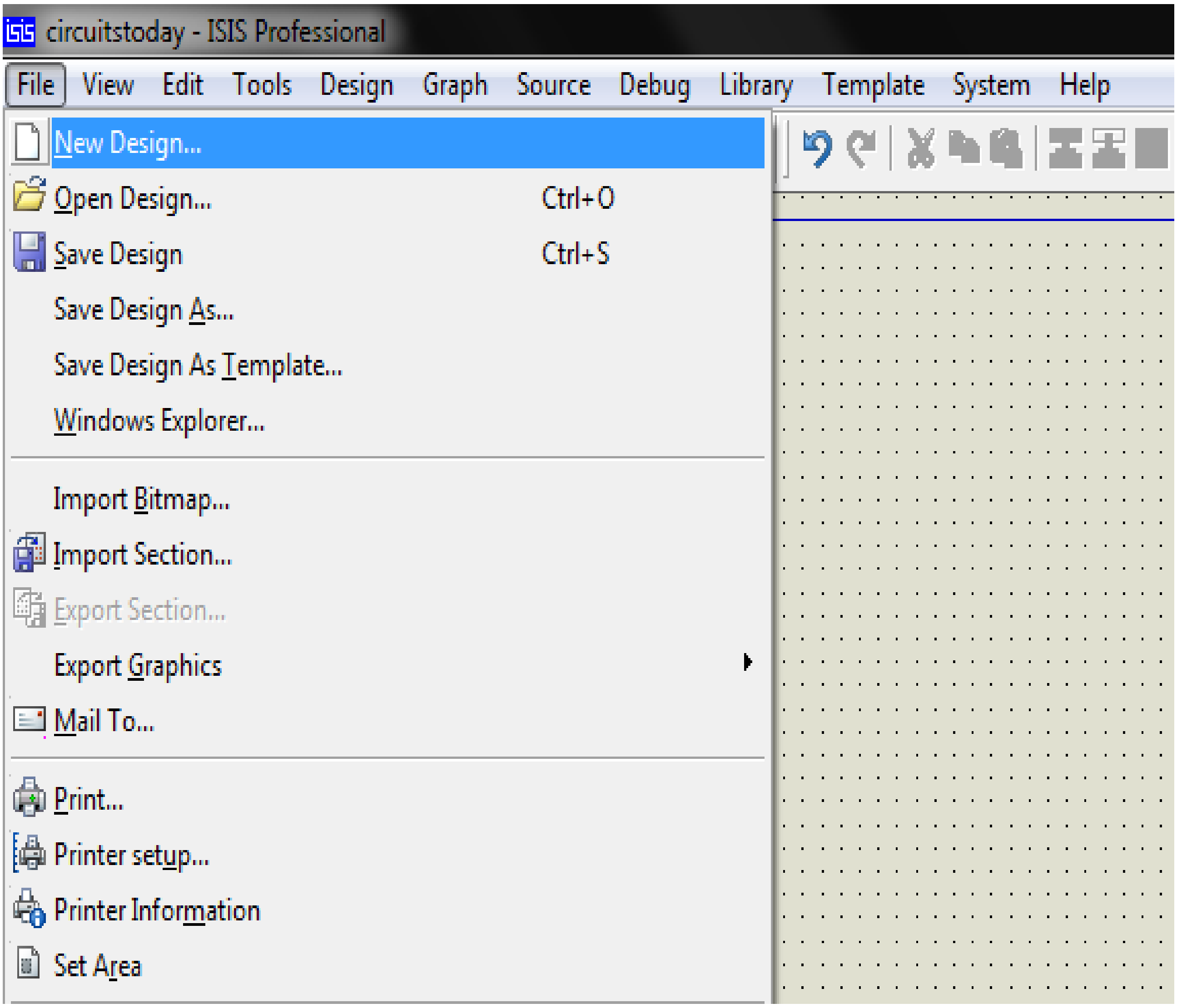


Fig. 3.2.7.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

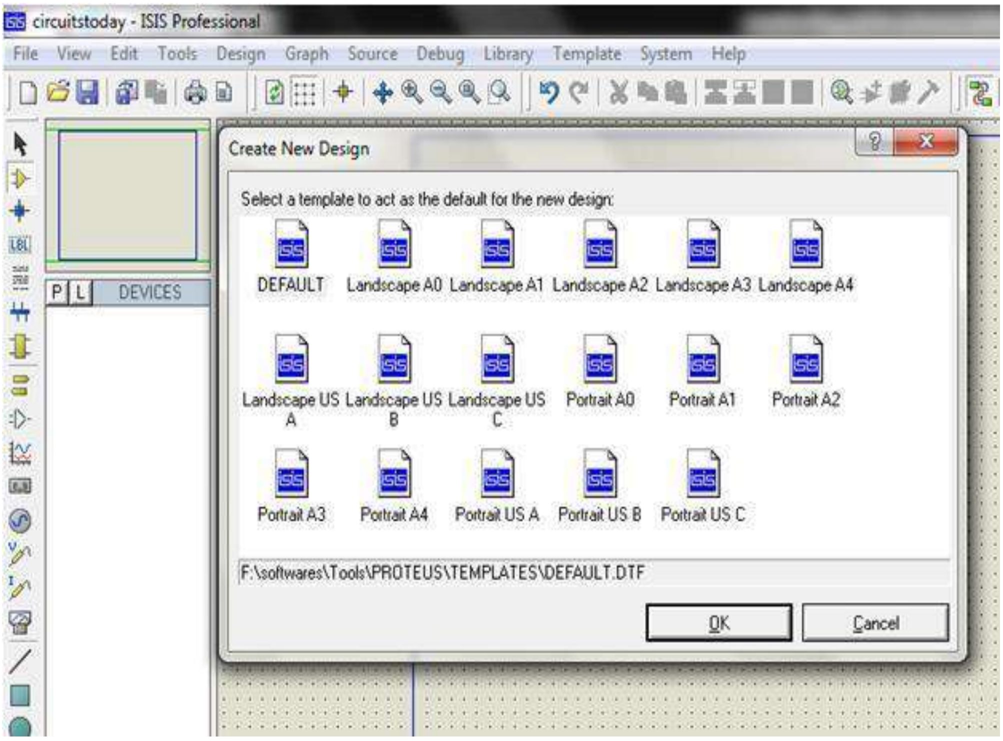


Fig. 3.2.7.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

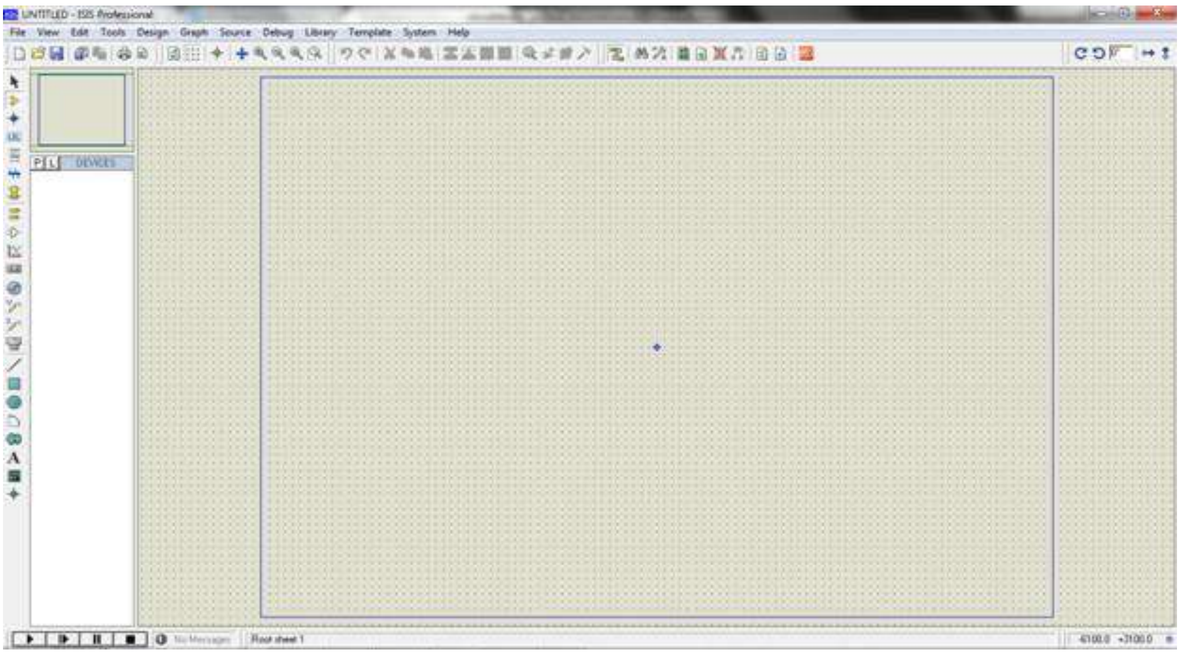


Fig. 3.2.7.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

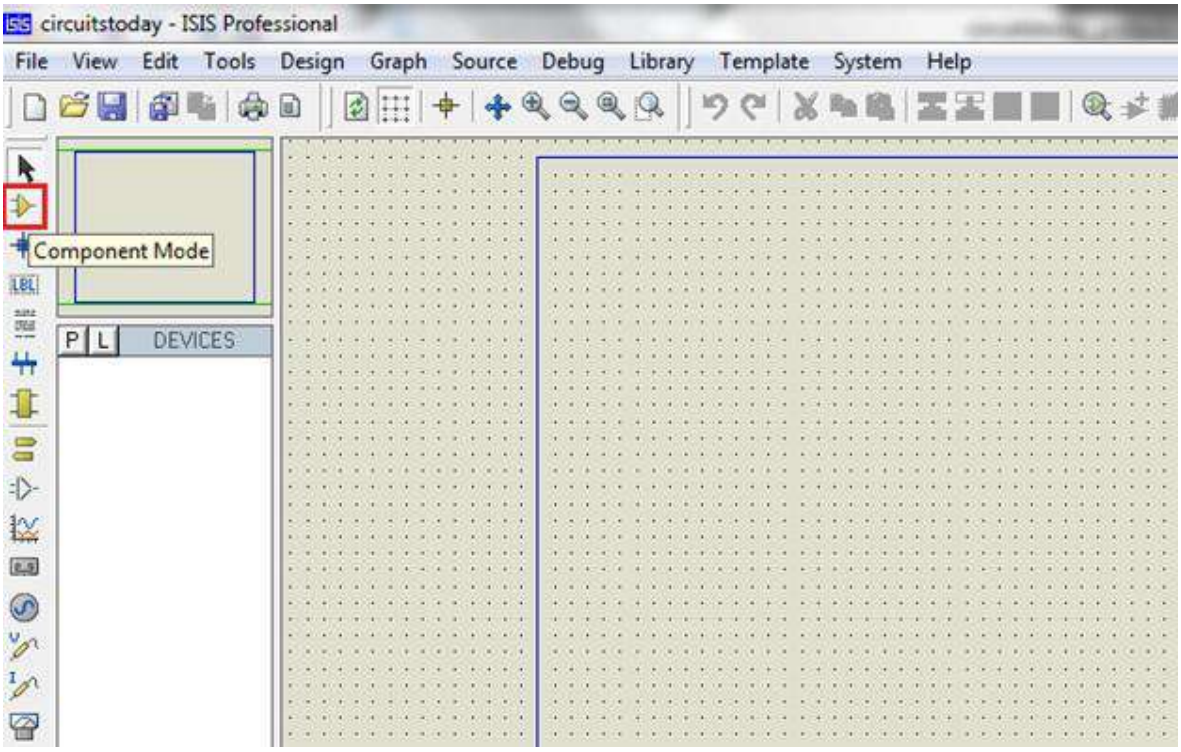


Fig. 3.2.7.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

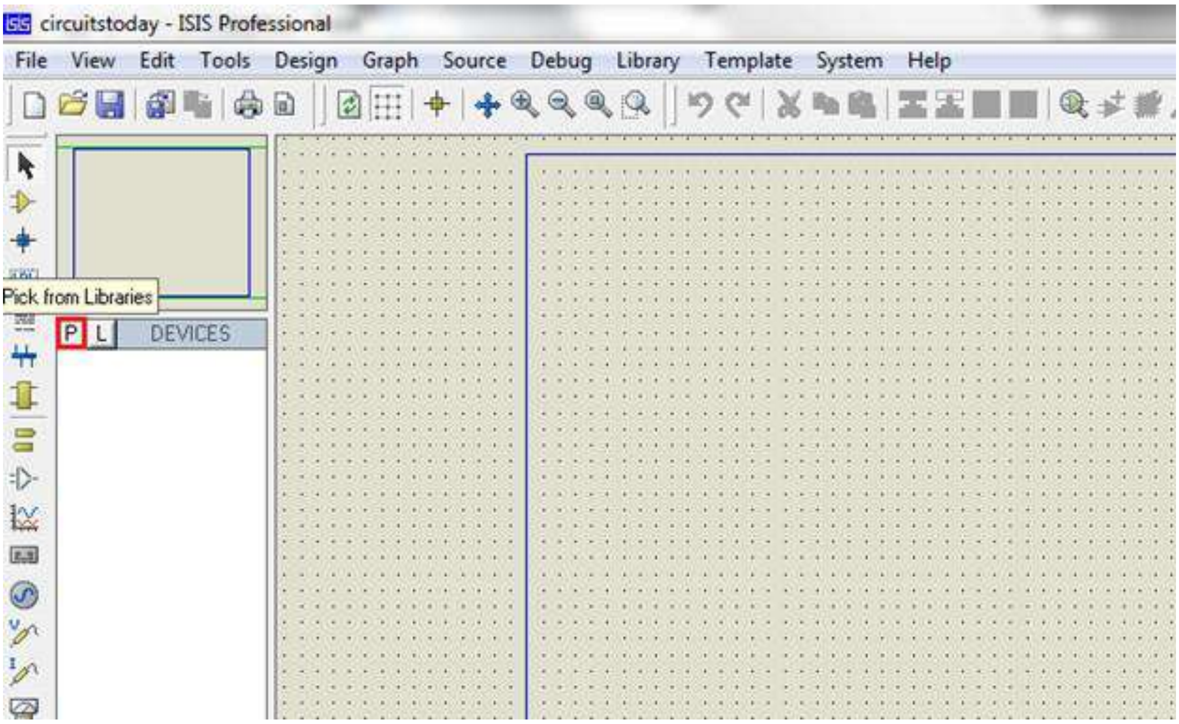


Fig. 3.2.7.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keyword's text box.

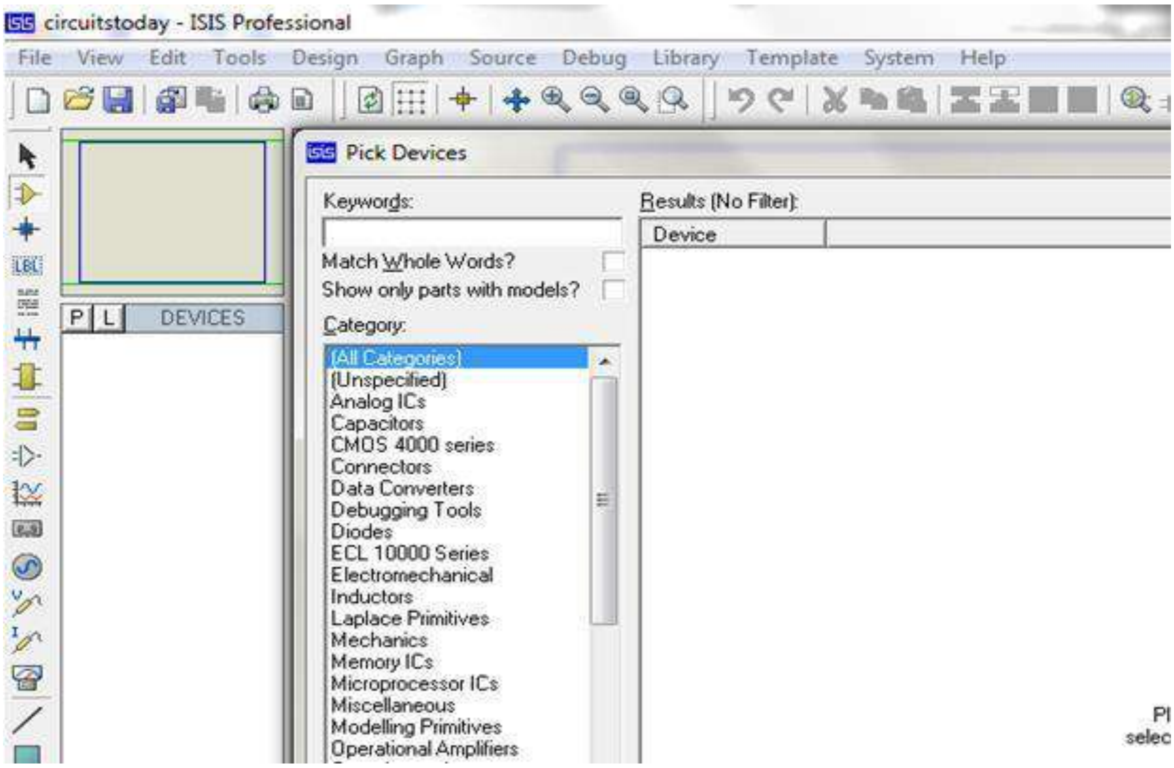


Fig. 3.2.7.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

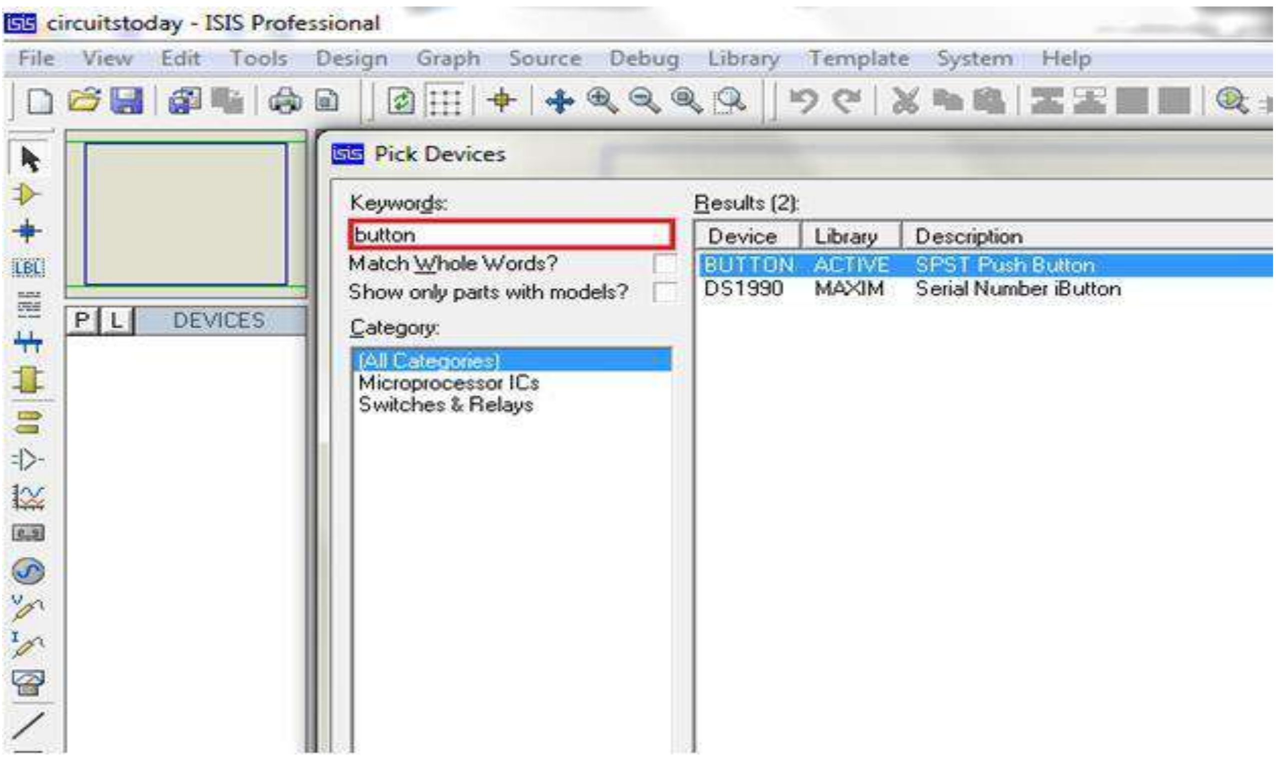


Fig. 3.2.7.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

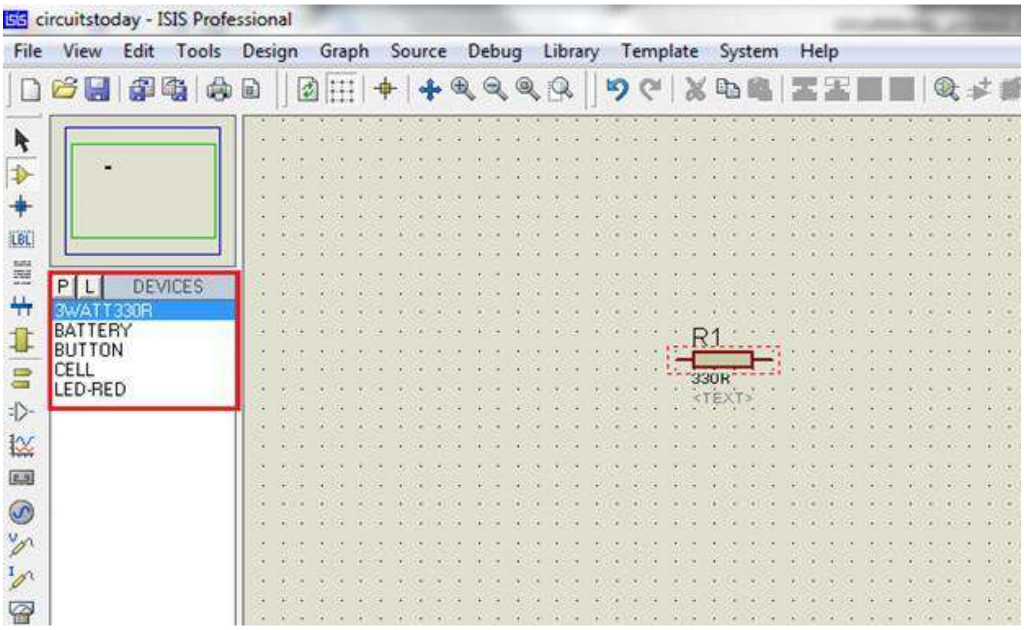


Fig. 3.2.7.8 Component Selection

Place all the required components and route the wires i.e., make connections. Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

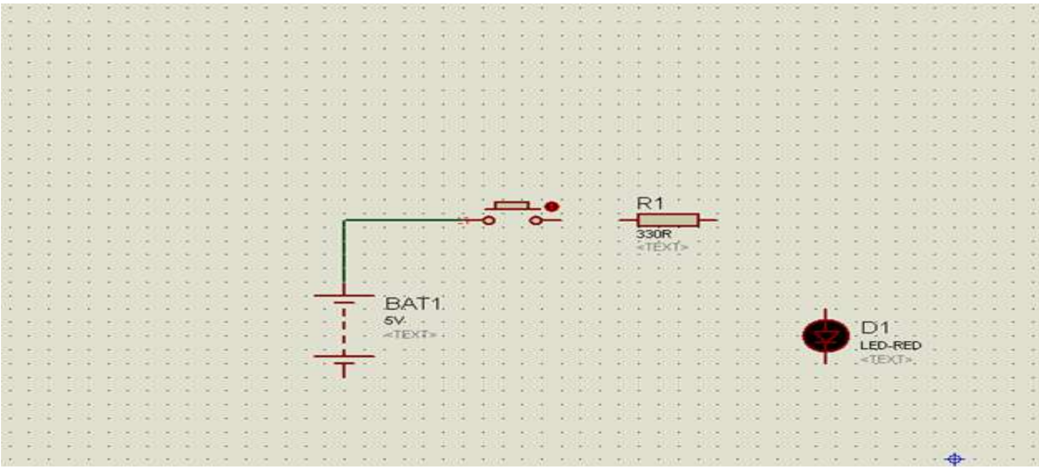


Fig. 3.2.7.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

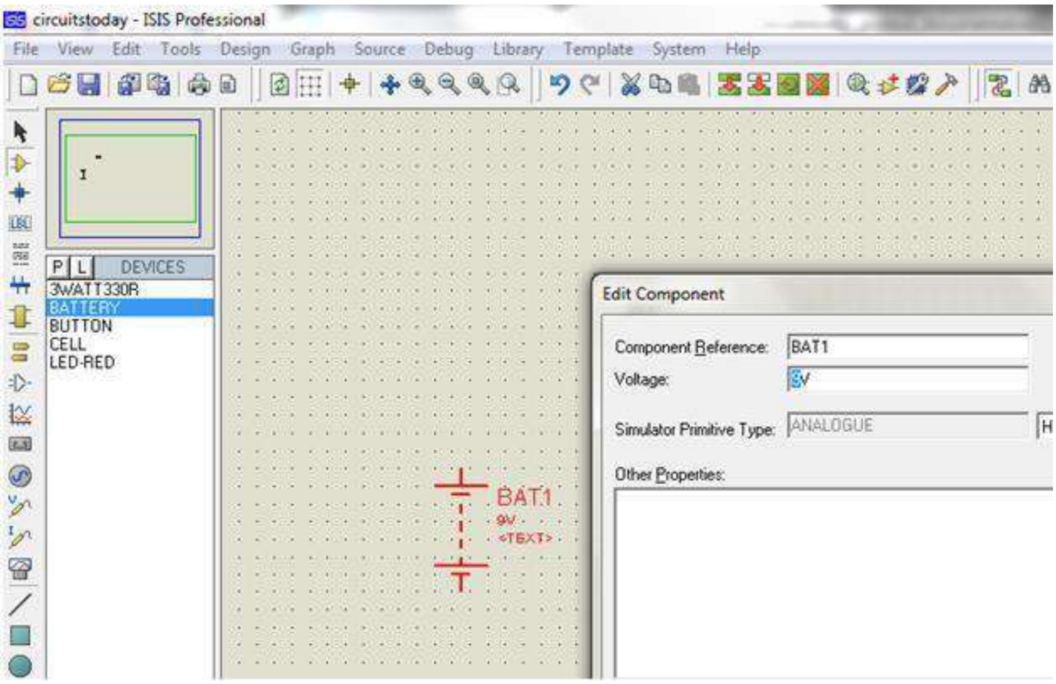


Fig. 3.2.7.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

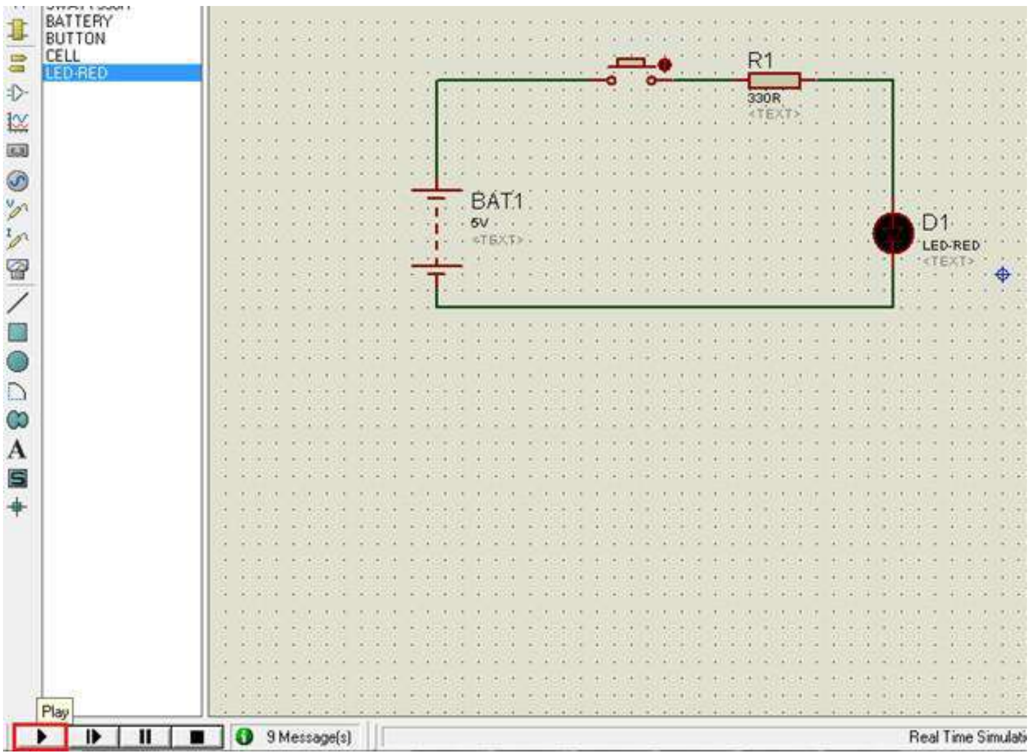


Fig. 3.2.7.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

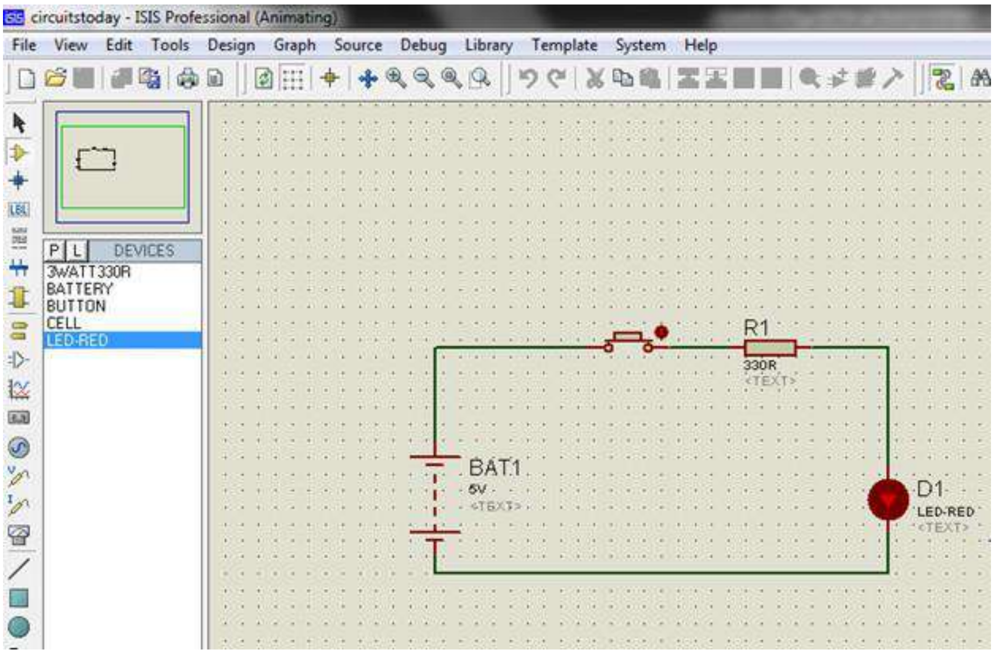


Fig . 3.2.7.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

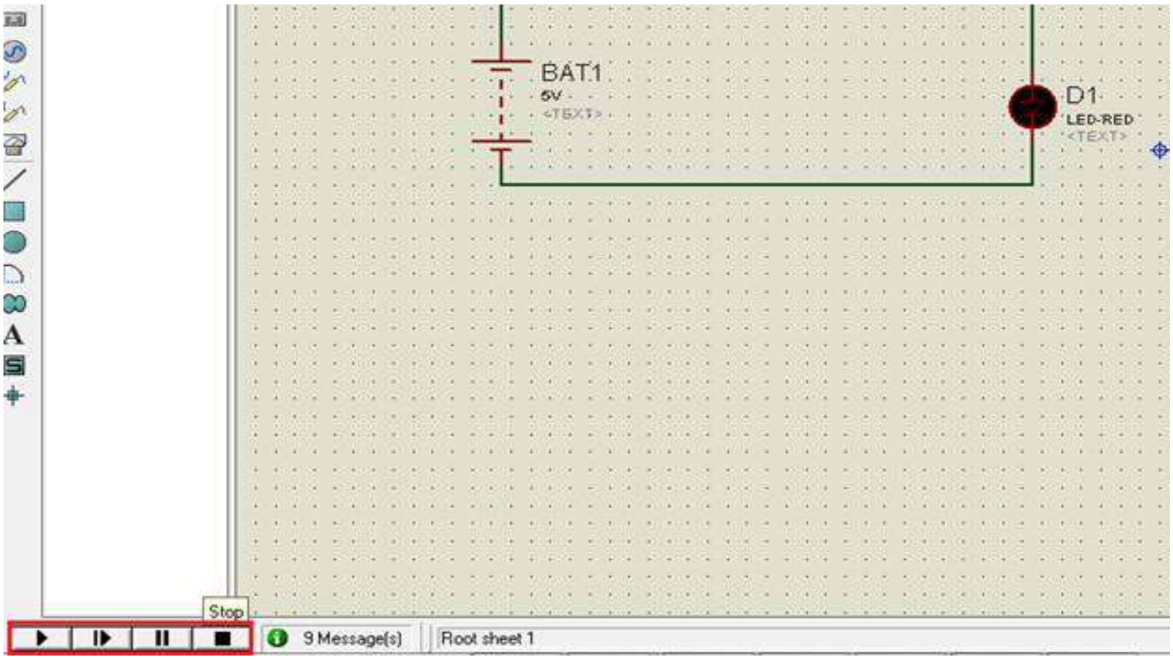


Fig. 3.2.7.13 Simulation Step-Pause-Stop Buttons

According to its developers, Things Speak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. Things Speak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates.

Things Speak was originally launched by iot Bridge in 2010 as a service in support of IoT applications.

Things Speak has integrated support from the numerical computing software MATLAB from MathWorks, allowing Things Speak users to analyse and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license from Math works.[citation needed].

Things Speak has a close relationship with Math works, Inc. In fact, all of the Things Speak documentation is incorporated into the Math works' Matlab documentation site and even enabling registered Math works user accounts as valid login credentials on the Things Speak website. The terms of service and privacy policy of ThingSpeak.com are between the agreeing user and Math works, Inc.

Setup Things Speak

Create a free MathWorks account or sign into Things Speak using an existing account.

Select the Things Speak channel you want your data to stream into. See Collect Data in a New Channel for help creating a new channel.

Record the following for the selected channel:

Channel ID, which is listed at the top of the channel view.

Write API key, which can be found on the API Keys tab of your channel view.

3.2.8 TELNET MOBILE APP:

Telnet is a client-server protocol, based on a reliable connection-oriented transport. Typically, this protocol is used to establish a connection to Transmission Control Protocol (TCP) port number 23, where a Telnet server application (telnetd) is listening. Telnet, however, predates TCP/IP and was originally run over Network Control Program (NCP) protocols.

Even though Telnet was an ad hoc protocol with no official definition until March 5, 1973,[3] the name actually referred to Teletype Over Network Protocol as the RFC 206 (NIC 7176) on Telnet makes the connection clear:[4]

The TELNET protocol is based upon the notion of a virtual teletype, employing a 7-bit ASCII character set. The primary function of a User TELNET, then, is to provide the means by which its users can 'hit' all the keys on that virtual teletype.[5]

Essentially, it used an 8-bit channel to exchange 7-bit ASCII data. Any byte with the high bit set was a special Telnet character. On March 5, 1973, a Telnet protocol standard was defined at UCLA[6] with the publication of two NIC documents: Telnet Protocol Specification, NIC 15372, and Telnet Option Specifications, NIC 15373.

Many extensions were made for Telnet because of its negotiable options protocol architecture. Some of these extensions have been adopted as Internet standards, IETF documents STD 27 through STD 32. Some extensions have been widely implemented and others are proposed standards on the IETF standards track (see below) Telnet is best understood in the context of a user with a simple terminal using the local Telnet program (known as the client program) to run a logon session on a remote computer where the user's communications needs are handled by a Telnet server program.

When Telnet was initially developed in 1969, most users of networked computers were in the computer departments of academic institutions, or at large private and government research facilities. In this environment, security was not nearly as much a concern as it became after the bandwidth explosion of the 1990s. The rise in the number of people with access to the Internet, and by extension the number of people attempting to hack other people's servers, made encrypted alternatives necessary.

3.3 DEFINE THE MODULES:

3.3.1 RFID (RADIO FREQUENCY IDENTIFICATION) MODULE

3.3.1.1 INTRODUCTION:

Radio Frequency Identification (RFID) technology has been attracting considerable attention with the expectation of improved supply chain visibility for both suppliers and retailers. It will also improve the consumer shopping experience by making it more likely that the products they want to purchase are available.

Recent announcements from some key retailers have brought the interest in RFID to the forefront. This guide is an attempt to familiarize the reader with RFID technology so that they can be asking the right questions when considering the technology.

3.3.1.2 WHAT IS RFID?

RFID (Radio Frequency Identification) is a method of identifying unique items using radio waves. Typical RFID systems are made up of 2 major components: readers and tags. The reader, sometimes called the interrogator, sends and receives RF data to and from the tag via antennas. A reader may have multiple antennas that are responsible for sending and receiving the radio waves. The tag, or transponder, is made up of the microchip that stores the data, an antenna, and a carrier to which the chip and antenna are mounted.

RFID technology is used today in many applications, including security and access control, transportation and supply chain tracking. It is a technology that works well for collecting multiple pieces of data on items for tracking and counting purposes in a cooperative environment.

3.3.1.3 THREE PRIMARY FREQUENCY BANDS HAVE BEEN ALLOCATED FOR RFID USE.

Low Frequency (125/134 KHz):

Most commonly used for access control and asset tracking.

Mid-Frequency (13.56 MHz):

Used where medium data rate and read ranges are required.

Ultra-High-Frequency (850 MHz to 950 MHz and 2.4 GHz to 2.5 GHz): offer the longest read ranges and high reading speeds.

Applications for RFID within the supply chain can be found at multiple frequencies and different RFID solutions may be required to meet the varying needs of the marketplace. Many of today's RFID technologies cannot reliably cover areas wider than 4 to 5 feet, making them unsuitable for wide openings that are the norm in manufacturing, distribution and stor receiving dock environments. Since UHF (Ultra High Frequency) can cover portals up to 9 feet wide it is gaining industry support as the choice bandwidth for inventory tracking applications including pallets and cases.

Technology providers are developing readers that work with multiple system protocols and frequencies so that users will be able to choose the RFID products that work best for their market and products.

3.3.1.4 RFID TAGS ARE FURTHER BROKEN DOWN INTO TWO CATEGORIES:

Active RFID Tags are battery powered. They broadcast a signal to the reader and can transmit over the greatest Distances (100+ feet). Typically, they can cost $4.00 - $20.00 or more and are used to track high value goods like vehicles and large containers of goods. Shipboard containers are a good example of an active RFID tag application

Passive RFID Tags do not contain a battery. Instead, they draw their power from the reader. The reader transmits a low power radio signal through its antenna to the tag, which in turn receives it through its own antenna to power the integrated circuit (chip). The tag will briefly converse with the reader for verification and the exchange of data. As a result, passive tags can transmit information over shorter distances (typically 10 feet or less) than active tags. They have a smaller memory capacity and are considerably lower in cost ($1.00 or less) making them ideal for tracking lower cost items

There are two basic types of chips available on RFID tags, Read-Only and **Read-Write**. Read only chips are programmed with unique information stored on them during the manufacturing process. The information on read-only chips can never be changed.
With Read-Write chips, the user can add information to the tag or write over existing information when the tag is within range of the reader. Read-Write chips are more expensive that Read Only chips. Another method used is something called a "WORM" chip (Write Once Read Many). It can be written once and then becomes "Read only" afterwards. This is a desirable format since companies will be able to write an EPC (electronic product code) to the tag when the product is produced and packaged.

3.3.1.5 HOW WILL RFID AFFECT OUR INDUSTRY?

RFID is expected to provide huge advantages to manufacturers by offering the tools to better plan production and respond more quickly to market demand. It will facilitate automation of inventory counts and speed shipping and receiving at the distribution level. For retailers, it will help to reduce stock-outs, enable product tracking and potentially reduce theft and streamline the POS function. RFID will also open other merchandising opportunities and help with the overall consumer buying experience.

Due to the current cost of the technology (both tags and infrastructure), the initial phase of adoption for retailers is at carton and pallet marking applications. The current technology being adopted for carton and pallet labelling is passive UHF tags (850 MHz – 950 MHz). As the cost of tags and readers comes down, a wider adoption at the item marking level will develop.
In order for RFID to grow quickly, it is important that standards be developed so that the technology providers are working toward a common goal of providing low cost and

compatible technologies. Not only will it drive down costs, but standards will also help users to reap the greatest benefit from their investment by providing value throughout the whole supply chain.

3.3.1.6 ORGANIZATIONS FOCUSED ON DEVELOPING RFID STANDARDS:

EPC global, Inc., a division of the Uniform Code Council, and its sponsors are working to standardize a new Electronic Product Code (EPC) as the next standard for identifying products. Their goal is not to replace existing bar code standards but to expand the information available down to unique identifiers for each marked item, and to enable more automatic reading. EPC utilizes the basic structure of the Global Trade Item Number (GTIN).

EPC global, Inc. has proposed open standards for tags and readers with the intention of bringing the costs down to a level where RFID tags could be applied to individual items. The work may lead to the creation of a new global Internet network that would allow companies to track items and enable end users to access the full benefits of RFID.

EPC global, Inc. has developed a specification for RFID tags to be used in the retail sector. The specification does not mandate what type of tag to be used but is intended to provide guidelines on data structure and how the tags should perform so that they can be used over a common platform.The specification requires that the chip contain an Electronic Product Code (EPC). The chip must be able to communicate according to an open standard and meet some minimum requirements so that it can be read by reading devices anywhere.

3.3.1.7 THE ELECTRONIC PRODUCT CODE (EPC):

The EPC is a number made up of a header and 3 sets of data as shown in the figure below. The header identifies the EPC version number – which will allow for different lengths or types of EPC later on. The second part of the number identifies the EPC manager – typically this would be the manufacturer of the item the EPC is attached to. The third part is called object class and refers to the exact type of product– most often the stock-keeping unit (SKU). The fourth series of numbers is the serial number that is unique to the item. (The second and third sets of data are similar in function to the numbers in UPC barcodes.)

ELECTRONIC PRODUCT CODE TYPE 1			
01.0000A89.00016F.000169DCO			
Header	EPC Manager	Object Class	Serial Number
8-bits	28-bits	24-bits	36-bits

Table 3.3.1.7 The Electronic Product Code (EPC)

Above is an example of a 96-bit EPC. It will allow sufficient capacity for 268 million companies. Each manufacturer will have the ability to create up to 16 million object classes with 68 billion serial numbers in each class. This should provide sufficient capacity to cover all products manufactured in the world for many years to come. As an interim step, the Auto-ID center is also proposing a 64-bit tag in order to minimize cost in the near term.

Potential Issues That Need Consideration When Choosing the Type of RFID and Method for Application to Your Products or Packaging.

Enthusiasm within the RFID industry has resulted in much hype about the technology over the past several years. As a result, it is important to embrace the technology with a bit of caution. The following are some of the issues that require close scrutiny when investigating RFID:

Tag Cost – This should not to be confused with chip cost. Although the goal is to bring the cost of the tag (chip and antenna) down to 5 cents, this goal is in the future since it both assumes manufacturing breakthroughs and is predicated on consumption in the billions of tags per year. Today, the cost is closer to "less than 50 cents" for a read/write solution in high (millions) volume. Ultimate tag cost will also be very much dependent on the type of chip required (read only versus read/write), size of the antenna needed and how it is packaged to meet a specific application.

Tag Size – Tag size is dependent on the read range desired. Although the chips are very tiny, they will not operate without being mounted to an antenna. The size of the antenna will determine the read distance performance of the tag so understanding the size of the antenna needed for the application is more important than the size of the chip alone.

Infrastructure Cost – Much focus appears to be placed on the tag cost since it is a recurring expenditure. Reader cost and infrastructure costs for implementing RFID must also be looked at very closely as well. Both the software systems requirements and physical environment, in which RFID is intended to be used, are critical to the ultimate performance of a system and may require changes to accommodate using it effectively. As an example, RFID chips cannot be read through metal objects. Other forms of electromagnetic interference may also impede performance of he technology and require changes to the physical environment where RFID will be used. The number and types of readers will also be a major expenditure depending on your application.

Read Distances – Read distances for RFID are very much dependent on the frequency chosen for the application. Tag orientation also affects the read range as the range diminishes as the tag is rotated from being perpendicular to the path to the reader. Reading reliability is quite good when labels are alone in a reader field like cases on a conveyor line, but less certain when the labels are randomly oriented as with labelled cases on a skid. The antenna size (both on the tag and the readers) will also be a determining factor. Hand held readers are not capable of using as much power as stationary readers and as result provide shorter read distances.

Government Regulation – Governments around the world regulate the use of the frequency spectrum. Different countries have already assigned certain parts of the spectrum for other uses and as a result, there is virtually no part of the spectrum that is available everywhere in the world for use by RFID. This means that a RFID tag may not work in all countries. As an example if you choose the Ultra High Frequency (UHF) frequency that Operates at 915MHz in the U.S. and you ship your product to Europe, they may not be able to be read it since Europe operates in the UHF spectrum at 869 MHz. This is an important consideration when operating in a global environment.

Anti-Collision – This is an important feature of RFID chips/readers since it will allow multiple tags to be read while grouped in one reader field. It is not available on all RFID tags but is an important feature if you are planning to use RFID for inventory counts, shipping and receiving where multiple tags need to be read at the same time.

Privacy Issues – Consumer groups have expressed concern over the potential (real or imagined) privacy invasion that might result with widespread RFID item marking. These groups are pushing for legislation that will require manufacturers to advise consumers that the products contain RFID devices and must provide a means so that the devices can be disabled at point of purchase. These issues are most prevalent at the item marking level and will have little impact on the implementation of carton and pallet labelling.

3.3.2 ESP8266 WIFI MODULE

3.3.2.1 INTRODUCTION

The **ESP8266 Wi-Fi Module** is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your **Wi-Fi** network. The **ESP8266** is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

3.3.2.2 CHARACTER CONFIGURATION:

- 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2),
- general-purpose input/output (16GPIO),
- Inter-Integrated Circuit (I^2C) serial communication protocol,
- analog-to-digital conversion (10-bitADC)
- Serial Peripheral Interface (SPI) serial communication protocol,
- I^2S (Inter-IC Sound) interfaces with DMA(Direct Memory Access) (sharing pins with GPIO),
- UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2),and
- Pulse-width modulation (PWM).

3.3.2.3 APPLICATIONS:

- Smart Home Appliances
- Improved Home safety and security
- Home air quality and water quality monitoring
- Natural Language-based voice assistants.

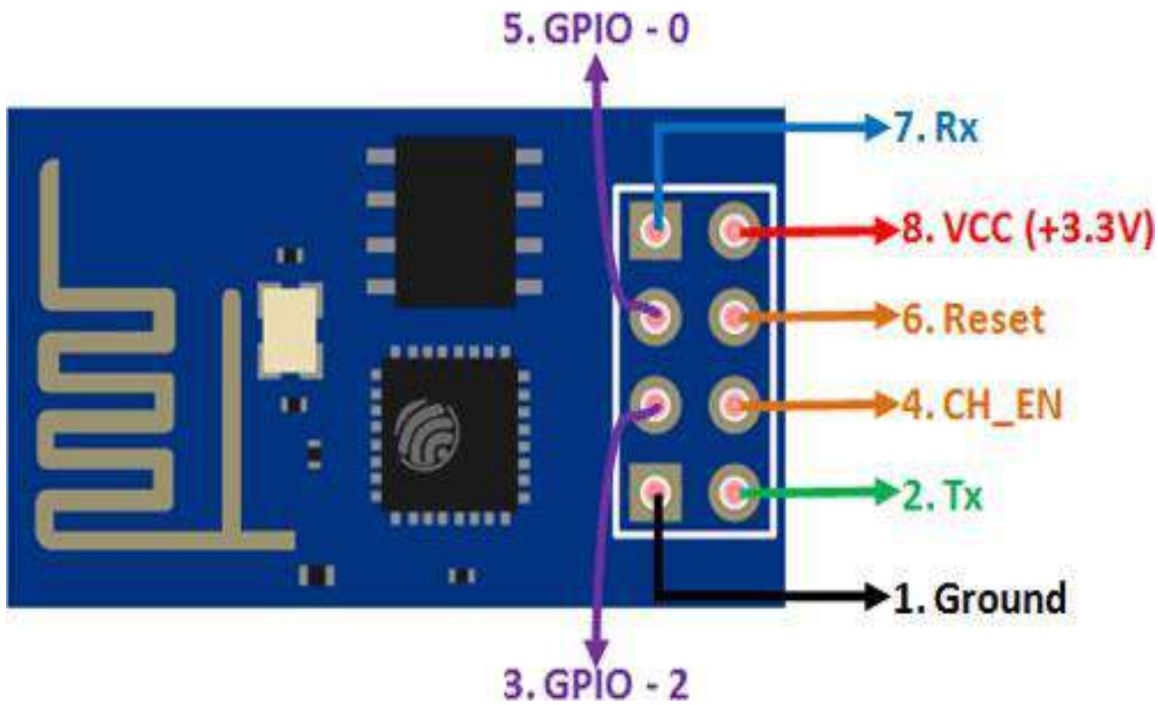


Fig.3.3.2.3 ESP8266 wifi module pin out

3.3.2.4 STRUCTURE AND CONFIGURATION:

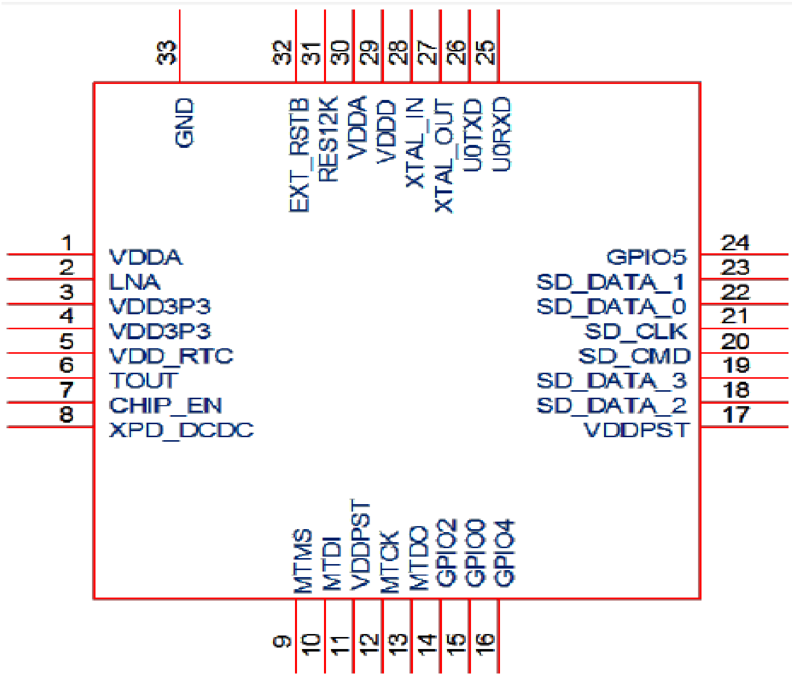


Fig.3.3.2.4 Structure and configuration

3.4 MODULE FUNCTIONALITIES:

3.4.1 RFID MODULE

3.4.1.1 Functionality of Radio Frequency Identification: The Transponder

The RFID transmitter is called a **transponder**, sometimes also a tag or simply a radio tag. It consists of an antenna (often a coil), a computer chip and a carrier material or housing. The chip is equipped with analog and digital circuits, and it also has a memory that can sometimes be written to several times. In some cases, the transponder is a semiconductor chip. Tags differ in several factors. Among other things, they have different sizes, lifetimes, clock frequencies, memory sizes or ranges.

The chips have a unique identity in the form of a **serial number** so that they can be clearly identified. The identity cannot be changed, which can prevent abuse. If it is a rewritable memory, the identity can be extended continuously and fully automatically by additional information (Auto-ID). In this respect, for example, previous locations of goods are interesting. RFID technology is also essential when it comes to choosing the right order picking procedure.

Tags usually do not have their **own energy supply**. In this case, the transponder is called passive. The reader delivers the necessary energy for data transmission to the tag. The antenna must be a coil that charges a capacitor by induction. Exceptions called active transponders have their own battery. Their use is worth it if the tag is to have a long range and various additional functions.

Transponders can be attached to or in the object to be marked. For example, they are stuck on (e.g. on pallets or cartons), embedded directly in objects (e.g. in screws, shoe soles, banknotes or documents) and even implanted in living beings (in pets behind the ear, in people's hands). Successful research and development have made it possible to use temperature-resistant transponders since 2006. Since then, they can even be cast in metals, which has various advantages: the risk of damage, wear and tear or loss is minimised and they are also invisible.

3.4.1.2 Functionality of Radio Frequency Identification: The Reader

A reader that must be in close range reads the tag's data using an alternating electromagnetic field or high-frequency radio waves. The receiver is also known as a **reader**. In addition to purely readout readers, there are also readers that can **write new information to the transponder's memory**. The transmitted radio waves are influenced by the tag because it reflects part of the energy. First, the unique identity and any other requested information are transmitted. In addition, different things can be stored.

The reader is a **combination of hardware and specially programmed software**. The **more range a reader should have, the more energy in the form of electrical current must be used for its use**. Typical ranges are between a few centimeters and a few meters. The smaller the range, the more precisely selected transponders can be addressed.

A longer range can be chosen, for example, if two objects should not be too close together. This is the case, for example, with containers containing flammable liquids and spark spraying equipment. Readers for radio frequency identification are available as handheld versions for manual operation, but can also take on various other forms, for example in the form of a gate through which goods flows.

3.4.2 ESP8266 WIFI MODULE

The ESP8266 WiFi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all WiFi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost-effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existance interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support. In the Documents section below you will find many resources to aid you in using the ESP8266, even instructions on how to transform this module into an IoT (Internet of Things) solution!

CHAPTER-4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM OF PROJECT:

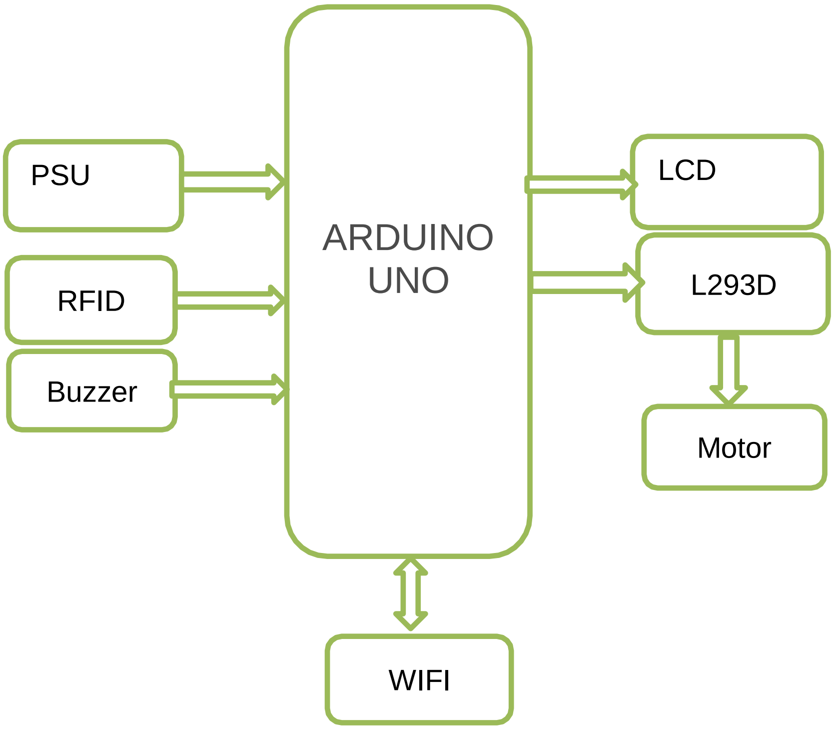


Fig.4.1: Block diagram of the project

4.2 FLOW CHART

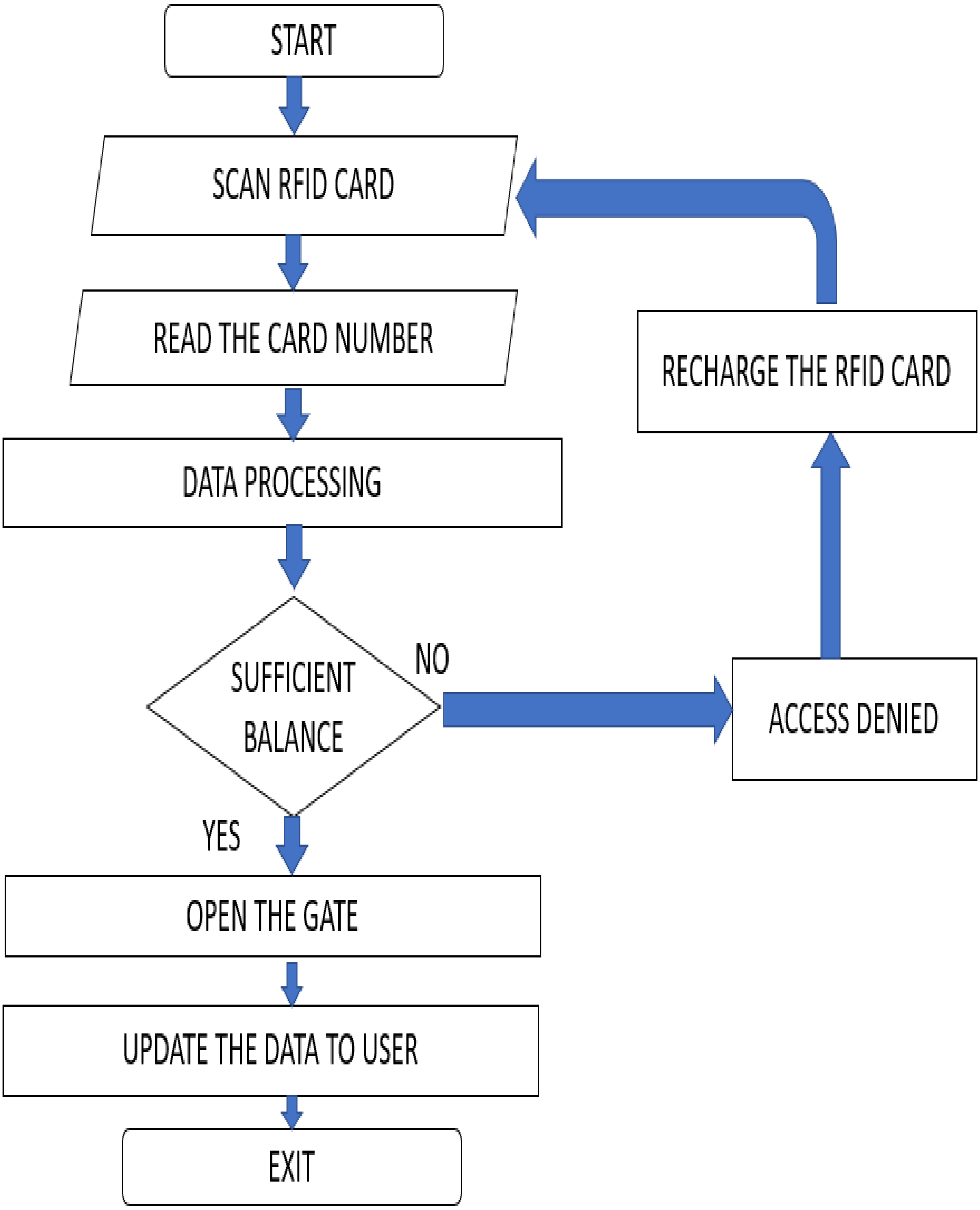


Fig.4.2 Flow Chart

4.3 IMPLEMENTATION STAGES

Stage-1: First on lcd we will get a message displaying waiting for link that indicates the wifi is not yet connected.

Fig.4.3.1 WAITING FOR LINK

Stage-2: In Mobile Telnet App, we will get a message like welcome when the wifi is connected to ESP8266 WIFI Module.

Fig.4.3.2 WELCOME

Stage-3: After the wifi is connected,we will get a message on LCD displaying Please Scan UR Card.

Fig.4.3.3 Please Scan UR Card

Stage-4: When we scan the card, if sufficient amount is available, we will get a message access granted for person and remaining balance will be displayed on LCD.

Fig.4.3.4 Access Granted

Stage-5: When we scan the card, if sufficient amount is not available, we will get a message access denied for person and balance will be displayed on LCD.

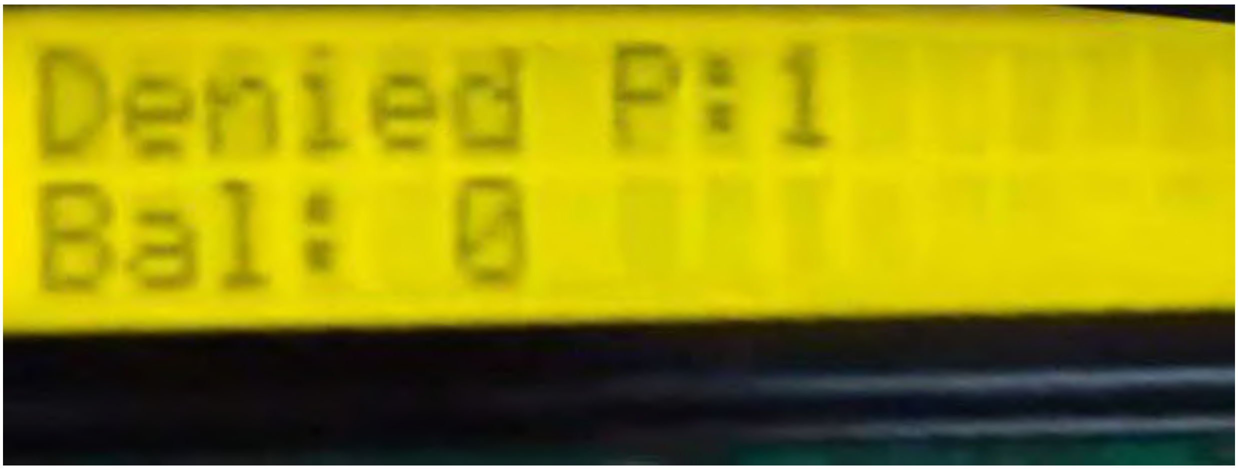


Fig.4.3.5 Access Denied

Stage-6: When card is recharged, amount will be added to card and display's person recharge done on LCD Screen.

Fig.4.3.6 Recharge Done

4.4 PROJECT CODE

```
#include<SoftwareSerial.h>
SoftwareSerial mySerial(2,3);
#include<LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);
int m1=4, m2=5;

char res[130];
char buff[130];

void sendwifi(String chr,unsigned int len)
{
  mySerial.print("AT+CIPSEND=0,");
  mySerial.println(len-1);
  delay(2000);
  mySerial.print(chr);
  delay(2000);
}

char t;

void setup()
{
  pinMode(m1,OUTPUT);
  pinMode(m2,OUTPUT);
  digitalWrite(m1,LOW);
  digitalWrite(m2,LOW);
  lcd.begin(16,2);
  lcd.clear();lcd.setCursor(0,0);
  lcd.print("Welcome");
  delay(1000);
  Serial.begin(9600);
  mySerial.begin(115200);
```

```
  mySerial.print("AT\r\n");
   delay(1000);
   mySerial.print("ATE0\r\n");
   delay(1000);
   mySerial.print("AT+CWMODE=3\r\n");
   delay(1000);
   mySerial.print("AT+CWSAP=\"Project\",\"project1235\",5,0\r\n\r\n");
   delay(1000);
   mySerial.print("AT+CIPMUX=1\r\n");
   delay(1000);
   mySerial.print("AT+CIPSERVER=1,23\r\n");
   delay(1000);
   lcd.clear();lcd.setCursor(0, 0);lcd.print("WATING FOR LINK");
   while(1)
   {
     if(mySerial.available())
     {
       //if(Esp.find("0,LINK"))
       if(mySerial.find("0,CONNECT"))
       {
          lcd.clear();lcd.setCursor(0, 0);lcd.print("LINK FOUND");
         break;
       }

     }
   }
   delay(1000);
   sendwifi("WELCOME \r\n",11);
   delay(1000);
}
char tt;
int i=100,j=100,m=100;
char rfid1[15];
void loop()
```

```
{
 lcd.clear();lcd.setCursor(0,0);lcd.print("Please Scan");
 lcd.setCursor(0,1);lcd.print("UR Card");
 delay(1000);
if(Serial.available())
{
 for(int k=0;k<12;k++)
 {
  while(!Serial.available());
   rfid1[k] = Serial.read();
   //Serial.print(rfid[k]);
 }
 if(strncmp(rfid1,"5400C1374CEE",12)==0)
 {
   i=i-20;
   if(i>=10)
   {
    lcd.clear();lcd.setCursor(0, 0);lcd.print("Access Grant P:1");
    lcd.setCursor(0,1);lcd.print("Bal: ");lcd.print(i);lcd.print("rs");
    delay(100);
    sprintf(buff,"Access Grant for Person: 1, Bal: %u\r\n",i);
    sendwifi(buff,strlen(buff));
    delay(100);
    digitalWrite(m1,LOW);
    digitalWrite(m2,HIGH);
    delay(1000);
    digitalWrite(m1,HIGH);
    digitalWrite(m2,LOW);
    delay(1000);
    digitalWrite(m1,LOW);
    digitalWrite(m2,LOW);
   }
 if(i<=10)
   {
```

```
    lcd.clear();lcd.setCursor(0, 0);lcd.print("Denied P:1");
    lcd.setCursor(0,1);lcd.print("Bal: 0");//lcd.print(i);lcd.print("rs");
    delay(100);
    sprintf(buff,"Access Denied for Person: 1, Bal: %u\r\n",0);
    sendwifi(buff,strlen(buff));
    delay(100);
    digitalWrite(m1,LOW);
    digitalWrite(m2,LOW);
    delay(3000);
   }
  }
 if(strncmp(rfid1,"5400C59D3C30",12)==0)
  {
   j=j-20;
   if(j>=10)
   {
    lcd.clear();lcd.setCursor(0, 0);lcd.print("Access Grant P:2");
    lcd.setCursor(0,1);lcd.print("Bal: ");lcd.print(j);lcd.print("rs");
    delay(100);
    sprintf(buff,"Access Grant for Person: 2, Bal: %u\r\n",j);
    sendwifi(buff,strlen(buff));
    delay(100);
    digitalWrite(m1,LOW);
    digitalWrite(m2,HIGH);
    delay(1000);
    digitalWrite(m1,HIGH);
    digitalWrite(m2,LOW);
    delay(1000);
    digitalWrite(m1,LOW);
    digitalWrite(m2,LOW);
   }
  if(j<=10)
   {
    lcd.clear();lcd.setCursor(0, 0);lcd.print("Denied P:2");
```

```
   lcd.setCursor(0,1);lcd.print("Bal: 0");//lcd.print(i);lcd.print("rs");
   delay(100);
   sprintf(buff,"Access Denied for Person: 2, Bal: %u\r\n",0);
   sendwifi(buff,strlen(buff));
   delay(100);
   digitalWrite(m1,LOW);
   digitalWrite(m2,LOW);
   delay(3000);
  }
 }
if(strncmp(rfid1,"5400C5CF431D",12)==0)
 {
  m=m-20;
  if(m>=10)
  {
   lcd.clear();lcd.setCursor (0, 0);lcd.print("Access Grant P:3");
   lcd.setCursor(0,1);lcd.print("Bal: ");lcd.print(m);lcd.print("rs");
   delay(100);
   sprintf(buff,"Access Grant for Person: 3, Bal: %u\r\n",m);
   sendwifi(buff,strlen(buff));
   delay(100);
   digitalWrite(m1,LOW);
   digitalWrite(m2,HIGH);
   delay(1000);
   digitalWrite(m1,HIGH);
   digitalWrite(m2,LOW);
   delay(1000);
   digitalWrite(m1,LOW);
   digitalWrite(m2,LOW);
  }
 if(m<=10)
  {
   lcd.clear();lcd.setCursor(0, 0);lcd.print("Access Denied P:3");
   lcd.setCursor(0,1);lcd.print("Bal: 0");//lcd.print(i);lcd.print("rs");
```

```
    delay(100);
    sprintf(buff,"Access Denied for Person: 3, Bal: %u\r\n",0);
    sendwifi(buff,strlen(buff));
    delay(100);
    digitalWrite(m1,LOW);
    digitalWrite(m2,LOW);
    delay(3000);
   }
  }
 }

 if(mySerial.available()> 0)
   {
     tt = mySerial.read();

  if( tt == '*')
  {
   i=i+100;
   lcd.clear();lcd.setCursor(0,0);lcd.print("Person 1");
    lcd.setCursor(0,1);lcd.print("Recharge Done");
    delay(500);
    sprintf(buff,"Recharge Done for 1 Rs: %u\r\n",100);
    sendwifi(buff,strlen(buff));
    delay(2000);
  }
  if( tt == '#')
  {
   j=j+100;
   lcd.clear();lcd.setCursor(0,0);lcd.print("Person 2");
    lcd.setCursor(0,1);lcd.print("Recharge Done");
    delay(500);
    sprintf(buff,"Recharge Done for 2 Rs: %u\r\n",100);
    sendwifi(buff,strlen(buff));
    delay(2000);
```

```
  }
    if( tt == '@')
  {
   m=m+100;
   lcd.clear();lcd.setCursor(0,0);lcd.print("Person 3");
    lcd.setCursor(0,1);lcd.print("Recharge Done");
    delay(500);
    sprintf(buff,"Recharge Done for 3 Rs: %u\r\n",100);
    sendwifi(buff,strlen(buff));
    delay(2000);
  }
 }
}
```

4.5 RESULTS

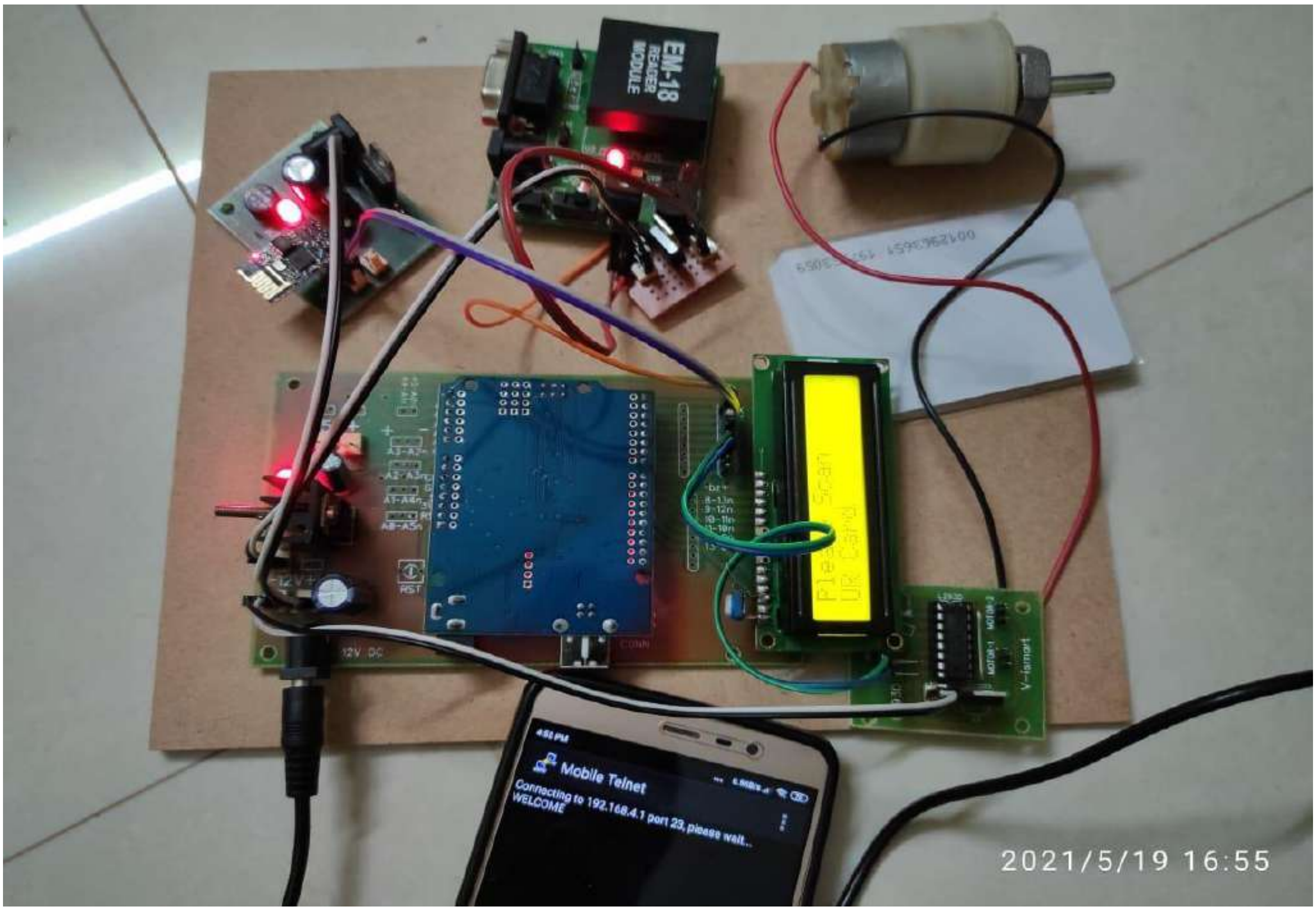


Fig.4.5 Final Output

CHAPTER-5

PROJECT TESTING

5.1 OVERVIEW OF TESTING METHODS

Today, governments want to take these ETC solutions one step further, with more sophisticated systems that not only eliminate the need for toll booths, but also enable governments to charge for road usage based on a variety of factors, including time of day, vehicle weight, distance driven, type of road and the pollution produced by the vehicle. These new systems enable governments to implement programs that charge tolls that better reflect the vehicle’s impact on road infrastructure, traffic and the environment, helping them:

- Reduce congestion during peak periods of the day by lowering tolls during low-traffic periods;
- Reduce congestion in specific areas by lowering tolls on specific roads;
- Encourage individuals and companies to purchase low-carbon-emission vehicles by charging higher tolls for vehicles that pollute more; and
- Transition the source of revenue financing road infrastructure from fuel-based taxes to toll on road usage, which will include any type of vehicle (from electric vehicles with small tolls to high-gas-consumption vehicles with high tolls).

Security: IoT is a combination of physical objects (“things”), sensors, embedded software and keeps a goal of effective connectivity and faithful data exchange. The current graphs imply that IoT market growing at remarkable rate. However, some of these devices suffer from limited memory, power consumption and processing power. These issues may cause IoT to become penetrable and hence security is at utmost priority to IoT domain.

As stated in a 2015 report by Hewlett-Packard on IoT research [3]:
1. Per 100, 70 devices still use un-encrypted network services
2. Per 100, 90 devices collected at least one piece of personally identifiable information (via device, cloud or mobile app)
3. Per 100, 70 devices (with cloud and mobile app components) enabled an attacker to identify valid user accounts through enumeration.

4. 80% of devices (with cloud and mobile app components) failed to require passwords of sufficient complexity

5.Behind every 10 device every 6 devices user interfaces (UI) were vulnerable

Following are the vulnerabilities of the IoT as per The Open Web Application Security Project (OWASP)

1) Insufficient protection compositions.

2) Dubiously secured Software/Firmware

3) Doubtful Network Services

4) Doubtful Web Interface

5) Worst Physical Security

6) Doubtful Cloud Interface

7) Doubtful Mobile Interface

8) Privacy Concerns

9) Doubtful Authentication/Authorization

10) Inadequate Transport Encryption Ongoing approaches to secure IoT have attempted to put a grip on communication protocol-based mechanisms, such as encryption for data-at-rest or in-transit.

But this itself is doubtful if the respective endpoints themselves are capable of being modified either by local access or remote connections. Gartner claims that by year 2020 more than 25% of identified attacks in a particular company will be on IoT devices or systems, even though IoT will only contribute to less than 10% of IT security budgets.

The testing methods of our project that is iot based toll booth manager system The requirements for testing this project is mobile app, wifi connection. First on lcd we will get a message displaying waiting for link that indicates the wifi is not yet connected. In Mobile Telnet App, we will get a message like welcome when the wifi is connected to ESP8266 WIFI Module.After the wifi is connected, we will get a message on LCD displaying Please Scan UR Card.When we scan the card, if sufficient amount is available, we will get a message access granted for person and remaining balance will be displayed on LCD. When we scan the card, if sufficient amount is not available, we will get a message access denied for person and balance will be displayed on LCD. When card is recharged, amount will be added to card and display's person recharge done on LCD Screen.

CHAPTER-6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

- The sole purpose of this project was to promote cashless transactions under Digital India initialization. Smooth trafficking at the toll plaza site corruption less transactions as well as reducing air pollution, stagnant long queue traffic will be reduced.

- The system has successfully overcome the drawbacks of the existing system by reducing the man power at the toll booth. It provides easy way of toll collection and maintenance of the information in a reliable and efficient manner.

- We aim to show that Toll booth can be completely managed by „Internet of Things" technology with the help of RFID reader. We present a related literature and market survey to show need of such design. We now discuss the potential directions for improving the design presented in this paper.

- **High data rates**: While the data rates of transmission and reception of data by Wi-Fi module and RFID reader are sufficient for our target application scenarios, we believe that with better access to the hardware functionality of RFID sensors, one can achieve higher data rates.

6.2 FUTURE ENHANCEMENT

- We can add GPS to our project to know live location, if anyone skips the toll booth.

- Voice System can be added in this project to indicate if the card is invalid or has insufficient balance.

- Cloud data can be handled using android application. Person will be granted permissions to register, recharge and renew toll account with the help of android application. Moreover, security system needs to be integrated within this module. Thus, when module is implemented on a greater level, it will result in cost reduction and increase in reliability.

PUBLICATION

Submitted Paper in the Conference ICISECT - 21 with Paper ID (ICISECT21-0003) and got Acceptance for the Paper.

REFERENCES

[1]. Sridevi, Vasantha Pai, Rakesh Y," IoT Based Unmanned Toll Booth Monitoring System" International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, (IJIREEICE), Volume 5, ISSUE 5, 2017.

[2]. R.Brindha, A.Jayanthi, A.Kalaivani, "IoT based Electronic Toll Collection System for Vehicles using Barcode System", International Journal of Electronics, Electrical and Computational System (IJEECS), Volume 7, Issue 3 March 2018.

[3]. Ganesh K. Andurkar, Vidya R. Ramteke, ‘‘Smart Highway Electronic Toll Collection System", International Journal of Innovative Research in Computer and Communication Engineering, Volume 3, Issue 5, May 2016.

[4]. P. S. Hanwate, Ashlesh Mandke, Narin Meher," Smart Toll Collection System based on IoT", IJSTE - International Journal of Science Technology & Engineering, Volume 3, Issue 12, June 2017.

[5]. Rana Pratap, Sanjeev, Niraj Kumar Tiwary, "IoT Based Smart Toll Booth System over Vahan 4.0 Cloud Database", International Journal of Science & Engineering (IJSE), Volume 4, Issue 8,2018.

[6]. Geethanjali N, Chiranth R, Chaitra J M, "IOT Based Intelligent Toll Gate System using Raspberry PI", International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume 6, Issue 2,2018.

[7]. Satyasrikanth P, Mahaveer Penna, Dileep Reddy Bolla, ‘‘AUTOMATIC TOLL COLLECTION SYSTEM USING RFID", International Journal of Computer Science and Mobile Computing, Volume 5, Issue 8, August- 2016.

[8]. Allen Francis Antony, Anju P, Gautham Raj P," Design of an Integrated Speed Tracking and Toll Management System for Express Highways", Proceedings of the IEEE 2017 International Conference on Computing Methodologies and Communication (ICCMC).

[9]. Amit Hatekar, Nikita Kokal, Tarun Jeswani, Pratik Lalwani," RFID Based Automatic Toll Collection System

BIBILOGRAPHY

[1] Sighila. P, VinithaValsan ,Preethibha .C,\ IoT Based RFID Gate Automation System", International Journal of Engineering Trends and Technology (IJETT), Vol. 36, Issue 9, June 2016.

[2] Ganesh K. Andurkar, Vidya R. Ramteke, \ Smart Highway Electronic Toll Col-lection System", International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certied Organization), Vol. 3, Issue 5, May 2015.

[3] Arduino Mega Front view",http://www.electroschematics.com/7963/ arduino-mega-2560-pinout/

[4] RFID Reader",https://communities.intel.com/thread/57844

[5] RFID",https://www.researchgate.net/publication235332138 Development of Java based RFID application programmable interface for heterogeneous RFID system

[6] LCD",https://www.researchgate.net/publication/235332138 Development of Java based RFID application programmable interface for heterogeneous RFID system

[7]GSM SIM900A Module",www.forum.researchdesignlab.com/datasheet/ modules/GSMsim900.pdf

[8] ESP 8266-(ESP 01)",www.sparkfun.com/products/13678

[9] Stepper motor",http://www.instructables.com/id/BYJ48-Stepper-Motor/

APPENDICES

This project shows, that the functionalities of the Toll booth can be completely managed using the 'Internet of Things' concept based on the RFID technology. In reference to survey of annual toll collection on toll plazas, conducted by government of Maharashtra in year 2010, we examine and report on the conditions of total time spent, for money transaction on toll plaza, by a particular vehicle and further evaluate total fuel wastage as well as human errors involvement, while doing so. Finally, to provide an optimal solution, we consent with the idea of making toll plazas completely automated, with assistance of IoT technology.

A

MAJOR PROJECT REPORT

On

IOT BASED MONITORING SYSTEM FOR COMATOSE PATIENT USING ARDUINO

Submitted by

1) Mr. D. BHAVISH (17K81A04J8) 2) Mr. MARTHA VAMASHI KRISHNA (17K81A04M5)
3) Mr. PAMPARI UMESH CHANDRA (17K81A04N5)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Esteemed Guidance of

Mr. P. PAVAN KUMAR

Assistant professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA & NAAC A+ Accredited

Department Of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the project entitled "IOT BASED MONITORING SYSTEM FOR COMATOSE SYSTEM USING ARDUINO", is being submitted by 1.**Mr.D.BHAVISH (17K81A04J8), 2.Mr.MARTHA VAMSHI KRISHNA (17K81A04M5), 3.Mr. PAMPARI UMESH CHANDRA (17K81A04N5)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Head of the Department

Mr.P.PAVAN KUMAR
Department of ECE

Dr. B. HARI KRRISHA
Department of ECE

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **D BHAVISH** WITH ROLL NO.**17K81A04J8, MARTHA VAMSHI KRISHNA** WITH ROLL NO.**17K81A04M5**, **PAMPARI UMESH CHANDRA** WITH ROLL NO.**17K81A04N5,** OF B. TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**IOT BASED MONITORING SYSTEM FOR COMATOSE PATIENTS USING ARDUINO**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. ★Kompally-500014.★

ORUGANTI VENKAT
DIRECTOR

TRAININGS & PLACEMENTS LASYA ITSOLUTION PVT. LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road,Secundrabad,500014
Email:contact@lasyainfotech.com, ov@lasyainfotech.com
Website: www.lasyainfotech.com | Contact: 7330666881/82/83/84/86

DECLARATION

We, the students of **Bachelor of Technology** in Department of ELECTRONICS AND COMMUNICATION ENGINEERING, session: <2017 – 2021>, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled "IOT BASED MONITORING SYSTEM FOR COMATOSE PATIENT USING ARDUINO" is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

1) **Mr. D. BHAVISH (17K81A04J8)**
2) **Mr. MARTHA VAMSHI KRISHNA(17K81A04M5)**
3) **Mr. PAMPARI UMESH CHANDRA(17K81A04N5)**

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr P. Santosh Kumar Patra**, **our beloved Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr B. HARI KRISHNA**, **Head &Professor**, Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in our memory for years to come.

We would like to express our gratitude to our dept. project coordinator **Dr A. Anand, Professor** for his involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time to come.

We sincerely thank our project internal guide, **Mr. P. Pavan Kumar, Assistant Professor,** Department of ECE and our class project coordinator **Ms. P. Pushpa**, **Assistant Professor,** Department of ECE for guidance and encouragement in carrying out this project work

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr Maalyadri**, **Professor in IT** and **Dr. R. Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

1. D.BHAVISH (17K81A04J8)
2. MARTHA VAMSHI KRISHNA (17K81A04M5)
3. PAMPARI UMESH CHANDRA (17K81A04N5)

ABSTRACT

Coma is a state of unconsciousness where the patient fails to respond. These patients need utmost care and 24*7 observations. This paper presents a continuous monitoring and recording of patient data without human intervention. If there is any sudden changes occur in the normal range of body parameters such as body temp falls or rise, blood pressure (B. P.) increases or decreases causing high or low B.P. where both are not stable conditions for better health, then it has facility to automatically alert the medical person. The development sensor identifies the patient development and furthermore produces an alarm message to the clinical individual.

One main area of research that has seen an adoption of the technology is the healthcare sector. The people in need of healthcare services find it very expensive this is particularly truein developing countries. As a result, this project is an attempt to solve a healthcare problem currently society is facing. The main objective of the project was to design a remote healthcare system. It's comprised of two main parts. The first part being, detection of patient's vitals using sensors, second for sending data to cloud storage. Remote viewing of the data enables a doctor or guardian to monitor a patient's health progress away from hospital premises. The Internet of Things (IoT) concepts have been widely used to interconnect the available medical resources and offer smart, reliable, and effective health care service to the patients. Health monitoring for coma living is one of the paradigms that can use the IoT advantages to improve the patient's lifestyle.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW OF THE PROJECT:

Health is a fundamental right of humans. Large number of people across the world are dying daily due to diseases. One major reason for this large number of deaths is the absence of timely and effective assistance. Continuous monitoring of coma patients can save lives considerably through the timely check-up. So, it very much needed that a system would be developed which would monitor the patients continuously and effectively.

The monitoring system is based on wireless sensor nodes. Failure of the node is mainly distributed to the system in the form of IoT, which still makes the system intentional. The use of the IoT protocol allows the integration of third - party application and devices. Advances in telecommunication and IT sector have given birth to Internet of Things (IoT). In today's world, the effective use of IoT technologies provided convenience to the physicians as well as patients, as they can be applied to various medical areas (such as real-time monitoring, patient information management, and healthcare management). The body sensor network (BSN) technology is one from the core technologies of IoT developments in healthcare system, in which a patient can be monitored by collection of tiny powered and lightweight wireless sensor nodes. To grab the benefit of emerged IoT in the field of healthcare, the consumers, physicians and patients need to think of some innovative and reliable methods. Due to the emergence of help of IoT's potential these people now can collect real-time raw data from unlimited number of patients for a continuous period of time through smart devices connected on an interconnected network. In last decades health issues are raising day to day life at very high speed every day. One of the major health issues is a coma. Coma is a deep state of persistent sleep in which a person cannot be awakened; he fails to respond normally too painful stimuli, light, or sound; lacks a normal wake-sleep cycle; and does not initiate voluntary actions. Comatose are not able to knowingly feel, speak or move. Comas can be caused by various things such as- a severe injury to the head that hurts the brain, infections in the brain, brain damage due to lack of oxygen for too long, taking too much medicine (overdose) or other drugs, may be due to chemical imbalances in the body from other illnesses. Sometimes the person in a coma state can respond to the external environment by voluntary movements such as he may open his eyes in response to external impulse. Even though an individual in a coma state appears normal but they could not respond to the external commands. Since the physical motion of persistent vegetative state individual are rare, there is a need for regular attention and care. Technology will always make things simpler and the human life much easier to survive. It affects the many zones of life; one of the important regions is medical field. Internet of Medical Things (IoMT) is an extension of the internet of things into the healthcare domain. Then IoMT is an ecosystem of interconnected sensors, wearable devices, medical devices, and clinical systems and applications that can connect to health care information technology systems using networking technologies.

It enables various healthcare applications to reduce healthcare costs, provide timely medical responses, and improves the quality of medical treatment. There are many applications of IoMT but one of the popular Example of IoMT is remote patient monitoring (RPM). The comatose health history will be perceived and studied at any time at any place and by doctor with the help of wireless technology. With the help of technology, we can permanently store patient health information on the server. This paper delivers a health status assisting system that identifies human body parameters such as blood pressure, body temperature, body movement and more information on the IoT server through Thing Speak. In emergency situations, this system automatically sent an alert SMS to the patient's caregivers, doctors if any strange data detected. Various sensors such as BP, temperature, acceleration and ultrasonic sensor, are used for a few seconds to collect body health parameter information for the proper treatment.

1.2 OBJECTIVES OF THE STUDY:

A health observing system comprises of variety of sensors connected to the patient and they communicate that data via the processing Things speak. In this project, ARDUINO is acts as a data junction node as well as a processor. The patient and doctor smart phone or computers are used as a monitoring device. The sensors are used to measure the health parameters of patient after these parameters are acts as readings and finally converted into signals. These signals are provided for processing to ARDUINO. Then ARDUINO displays the information on a monitor and also stores the information over the cloud with the help of IoT. This information can be accessed by the doctor on his phone/computer and get the notification. Also there is facility provided to send an alert message to the doctor or patient care giver if any abnormal data is detected. The workflow of the project is as- the sensors value are read and displayed on the monitor and stored in the cloud for future use. The proposed health monitoring system consists of different sensors which are divided into two categories. One is used for monitoring vitals of the comatose and second is used for detecting any physical changes occur in the comatose. Here, temperature and blood pressure are the two vitals recorded and monitored to understand health status of a comatose. The other sensors is PIR sensor which are used for detecting any physical changes that occurring comatose. These sign switch gives data are recorded and checked consistently to comprehend the body working. In the event that the arrangements of these detected signs which are outside ordinary ranges normally suggest the requirement for some consideration or conceivable departure to a more elevated level of treatment during which we alert the doctor.

1.3 SCOPE OF THE STUDY:

- IOT Monitoring proves really helpful when we need to monitor & record and keep track of changes in the health parameters of the patient over a period of time. So with the IOT health monitoring, we can have the database of these changes in the health parameters.
- Patient health parameter data is stored over the cloud. So it is more beneficial than maintaining the records on printed papers kept in the files. Or even the digital records which are kept in a particular computer or laptop or memory device like a pen- drive. Because there are chances that these devices can get corrupt and data might be lost. Whereas, in the case of IOT, cloud storage is more reliable and does have minimal chances of data loss.
- This system is powered by the Arduino-UNO it includes a blood pressure monitoring unit and an ultrasonic sensor to check urine, temperature sensor, motion sensor, and an LCD display.
- When we turn the system on, it gets connected to the website using WIFI, System monitor shows four signs namely heart rate, temperature, humidity, and urine output.
- In case if the patient regains consciousness and attempts to move, the sensor will detect the motion and update it over IOT and LCD. In this way, our system monitors the comatose patients.

1.4 MATERIAL REQUIREMENT:

1.4.1 HARDWARE REQUIREMENT:

- Arduino UNO
- Ultrasonic (US) sensor
- Liquid Cristal Display
- WIFI Module
- DHT11–Temperature and Humidity Sensor
- Blood Pressure Sensor
- PIR Sensor

1.4.2 SOFTWARE REQUIREMENT:

- ARDUINO IDE
- PROTEUS SOFTWARE

1.5 PROCUMENT OF EQUIPMENT:

We brought all the required hardware components in an electronic manufacturing shop and before assembling and interfacing hardware components to Arduino UNO micro- controller; we connected individual sensors to Arduino in Proteus Software and understood working of every individual sensor in Proteus Software.

Now, we assembled all sensors and interfaced to Arduino UNO micro-controller using connecting wires through bread board and we wrote program to work all the hardware sensors in Arduino IDE, we understood the working of all sensors in a practical way.

We calculated and got all health vital information of coma patient's and displayed on LCD and also updated on Server. So, doctor can check and track the patient's health status from any- where in the world

CHAPTER-2
LITERATURE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH AREA:

IoT based monitoring system for comatose patients using Arduino provides better healthcare service by improving the availability and transparency of the health data. However, it also poses serious threats to data security and privacy. This project aim to developing a system which gives body temperature and blood pressure using LM35 and blood pressure sensor respectively. These sensors interfaced with controller Arduino Uno board. Wireless data transmission done by Arduino through Wi-Fi module. This project proposes an internet so things (IOT) based real time patient monitoring system that is able to guarantee the integrity of the real time parameters.

The internet of things (IOT) has already changed the world and it influences both the way we live and work. The internet is gradually becoming a necessary and important tool in our everyday lives. The number of nodes connected to internet are increasing day by day and this burst amount of connections give rise to an emerging technology named as internet of things (IOT). It is consisting of smart devices connected to internet and embedded with processors and sensors. These sensors collect the data and send through Wi-Fi module where collected data is analyzed. This system also can be expanded by adding more e-health sensors to collect various health parameters. Research on reducing the jitter delay and eliminating the noise signal are also required to improve the performance of the proposed system. High blood pressure is a serious condition that can cause to damage heart and other organs and increase the risk of heart attack and stroke. It would be helpful to have a convenient way to automatically take many blood pressure readings throughout the day and over time to see how often one's blood pressure is high. temperature sensor is used for measurement of body temperature. Sensor is put in touch with body and it senses body temperature. It is calibrated linearly in Celsius. It has low self-heating capability.

2.2 REVIEW OF RELATED LITERATURE:

In the existing system, we use active network technology to network various sensors to a Patient. Various critical parameters are continuously monitored via sensors and reported to the

Doctors or Nurses in attendance for timely response in case of critical situations. The sensors are attached to the body of the patients without causing any discomfort to them. In this project we monitor the important physical parameters like body temperature, ECG, heart beat rate and blood pressure using the sensors which are readily available. Thus, the analog values that are sensed by the different sensors are then given to a microcontroller attached to it. The microcontroller processes these analog signal values of health parameters separately and converts it to digital values using ADC converter. Now, the digitalized values from more than one microcontroller are sent to the Central PMS. Each of the sensors attached microcontroller with a transceiver will act as a module which has its own unique ID. Each module transmits the data wirelessly to the gateway attached to the PC of the Central PMS.

The gateway is attached to the PC i.e. Central PMS which is situated in the medical center, is capable for selecting different patient IDs and allowing the gateway to receive different physical parameter values the patient specified by the ID. The software designed using Graphical User Interface (GUI) can operate on different physical parameters of each patient, consecutively with a specified time interval for each patient. The Existing framework gives the solution for just three well being parameters of the trance like state patients. This existing work utilizes the results of heartbeat sensor, Temperature sensor and body development sensor for detecting the health parameters suchas heartbeat rate, internal heat level, patient's body temperature and movements of the trance like state patients. In the event there is discover any variations from the results of health parameters of the patient suddenly the microcontroller sends an alarm message through the GSM device. It sends that information to the mobile number of the patients in charge person and guardian. Further- more in this framework LCD (Liquid Crystal Display) used to show the readings which are taken from the sensors. It has the accompanying drawbacks, for example, utilizing constrained detecting parameters, one of the chances that any irregular condition the message alert send through GSM module to the cell phone as SMS and also it showed in LCD board. These drawbacks could be overwhelmed by the proposed framework.

2.3 CONCLUSION ON REVIEWS:

To overcome all the problems in the existing system in monitoring health of the coma patients in the proposed system we introduced IoT based health monitoring system for coma patient's using Arduino.

Proposed system, consists of mainly two units- in the first unit, we connect all the sensors to coma patient's body and get all the vital health information from the sensors and Arduino UNO micro-controller will calculates all those information.

In the second unit, after calculating all the information by the Arduino UNO. It will display's all the health status of coma patients on the LCD and also updates all information in the Sever.

So, doctor can check the patient's health status from any part of the world.

CHAPTER-3
PROJECT DESIGN

3.1 OVER VIEW OF THE DESIGN

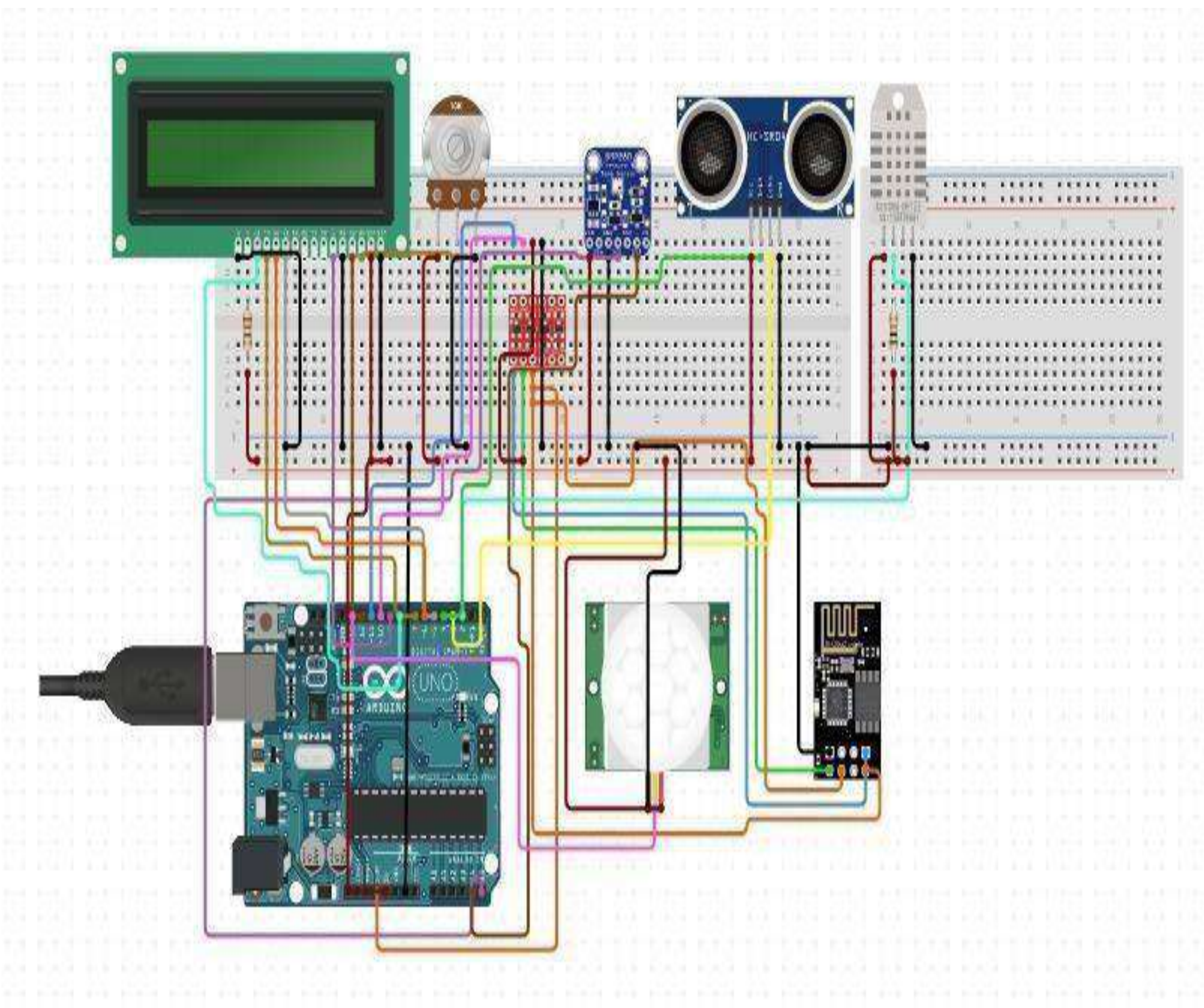


Fig 3.1 Over View Of The Design

3.2 BLOCK DIAGRAM OF PROJECT:

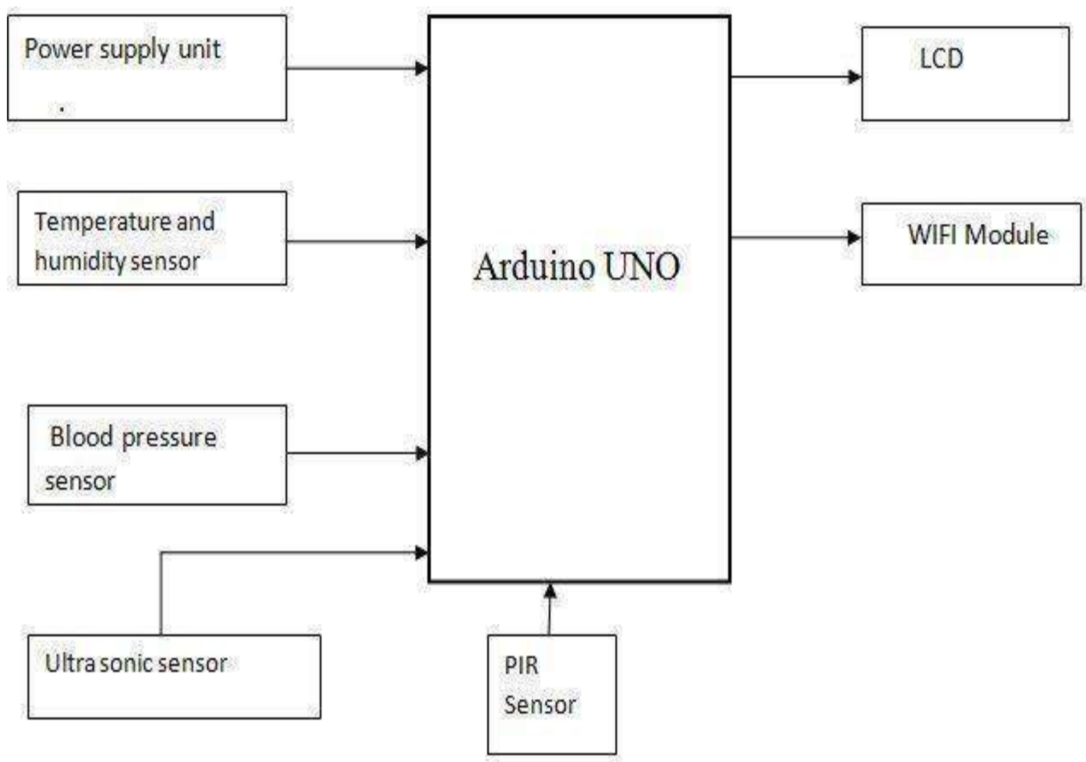


Fig.3.2 Block diagram of the project

3.3 HARDWARE REQUIREMENT:

- Arduino UNO
- ESP8266 Wi-Fi Module
- Liquid Cristal Display
- Ultrasonic Sensor
- DHT11–Temperature and Humidity Sensor
- PIR Sensor
- Blood Pressure Sensor

3.4 ARDUINO UNO

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The ATmega8 microcontroller contains 32 general purpose working registers. As shown in the below figure these registers are directly connected to ALU. Two registers can carry one single instruction consequently in one clock cycle.

Specifications:

Table 3.4: Atmega328 specifications

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 Ma
DC Current on 3.3V Pin	50 Ma
Flash Memory	32 KB (0.5 KB is used for Boot loader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read Analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloudand many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.

3.4.1 BOARD DESCRIPTION:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

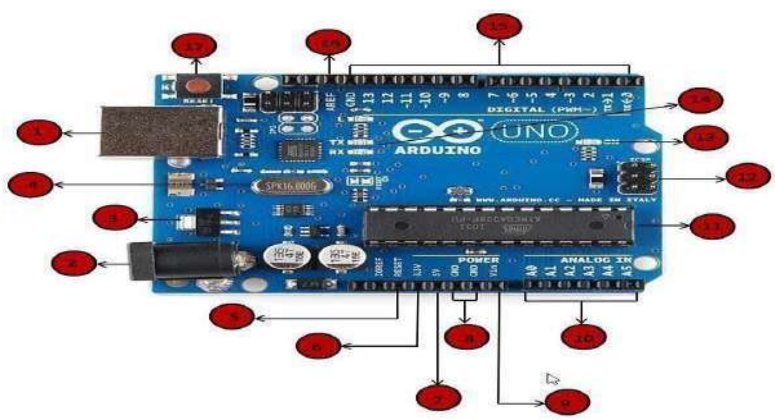


Fig 3.4.1 Board Description

TABLE 3.4.1 Board Description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit.

	• Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The

	TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

3.4.2 PIN DESCRIPTION OF ATMEGA328:

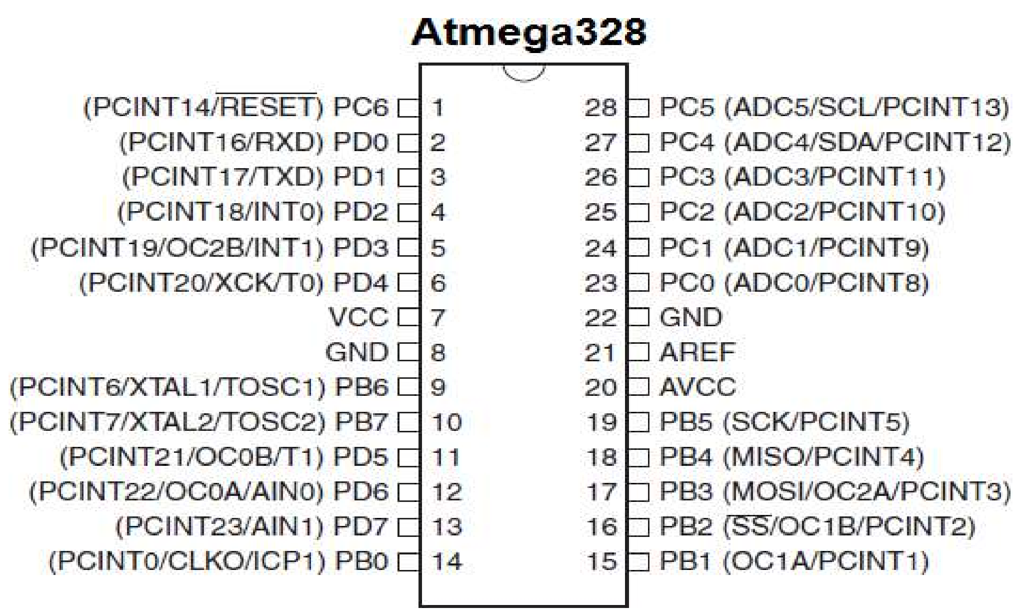


Fig 3.4.2 Pin description pf ATMEGA328

3.4.3 ADVANTAGES OF ARDUINO:

- It is cheap
- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
- For beginners, it is very simple to use.

3.4.4 APPLICATIONS:

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

3.5 WI-FI MODULE:

The ESP8266 is a low-cost Wi-Fi module that can be integrated easily into IoT devices. This will walk you through setting up ESP8266 Wi-fi module which can be used with Arduino. The ESP8266 comes in many models with different functionalities. We'll be focusing on the ESP8266 ESP-01 module, the most common and basic one available.

3.5.1 What is ESP8266?

The ESP8266 is a small Wi-Fi module built around the ESP8266 chip that can connect your microcontroller to the internet wirelessly for a very small cost. It can be a great option for Internet of Things (IoT) projects, but can be difficult to work with for beginner hobbyists who do not have prior experience with the module. In this tutorial, we hope to show you how to interface the ESP8266 with an Arduino and perform some basic functions like connecting it to a Wi-Fi network.

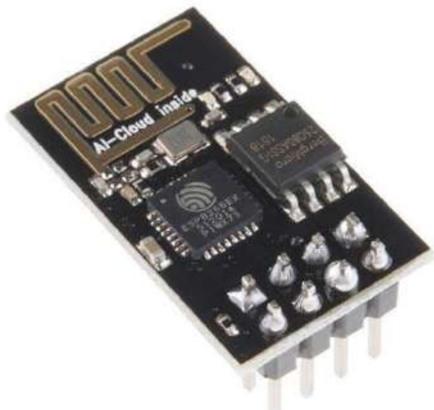


Fig 3.5.1 ESP8266 Wi-Fi Module

3.5.2 FEATURES OF ESP8266

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 1MB Flash Memory
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)

The first feature to notice about the ESP8266 is its awkwardly spaced header pins. The module has 8 pins that serve different functions, but they are packed in a 4×2 arrangement that makes plugging the module into a breadboard impossible. This means that to prototype projects on a breadboard, you'll need male-female jumper wires to connect the pins on the ESP8266 to rows on the breadboard. If you'd like to make your prototyping more compact, you can also purchase breadboard breakouts for the ESP8266 such as this one. For prototyping, I chose to just use jumper wires.

3.5.3 STRUCTURE AND PIN CONFIGURATION

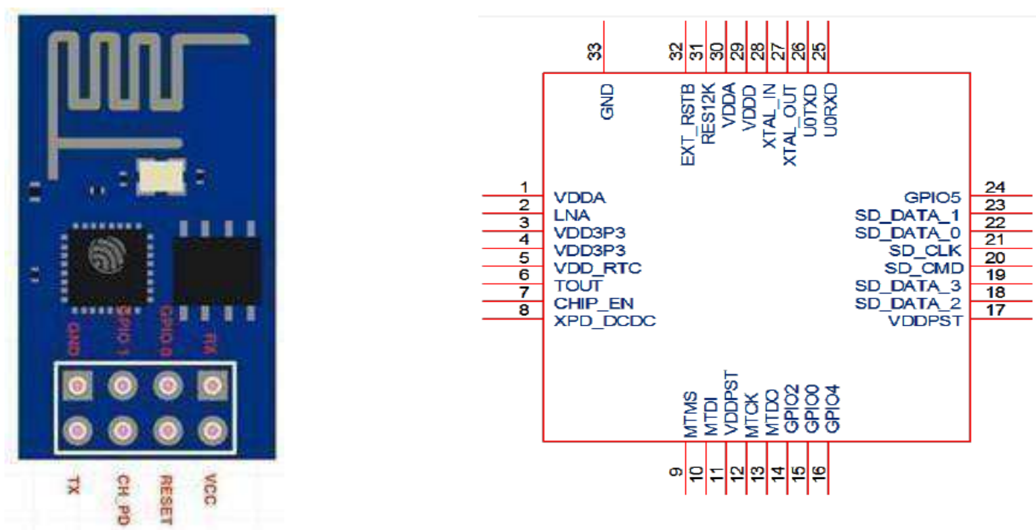


Fig 3.5.3 structure and pin configuration

RX: UART serial communication receive pin

GPIO 0: GPIO pin (unused in this project)

GPIO 1: GPIO pin (unused in this project)

GND: Connection to Ground

VCC: Connection to 3.3V VCC (VCC cannot exceed 3.3V!)

RESET: Reset pin (pull down to reset)

CH_PD: Chip enable and power down pin

TX: UART serial communication transmit pin

Note that the maximum voltage input for the ESP8266 is 3.3V. Any input voltage greater than 3.3V will damage the module! To program settings on the ESP8266, we'll first need to connect it to aserial terminal on a computer through which we can send it special commands. Settings that we'll have to program include, for example, the SSID and password for the wi-fi network the module will be connected to. To connect the ESP8266 to a computer and configure its settings, we'll need a USB to serial adapter with 3.3V logic, along with a serial terminal program. Fortunately for us, we have the Arduino and the Arduino IDE's serial monitor! This means that we'll just have to connect the ESP8266 module to the Arduino and upload a custom sketch to the Arduino.

3.5.4 HOW TO CONNECT ESP8266 WI-FI MODULE

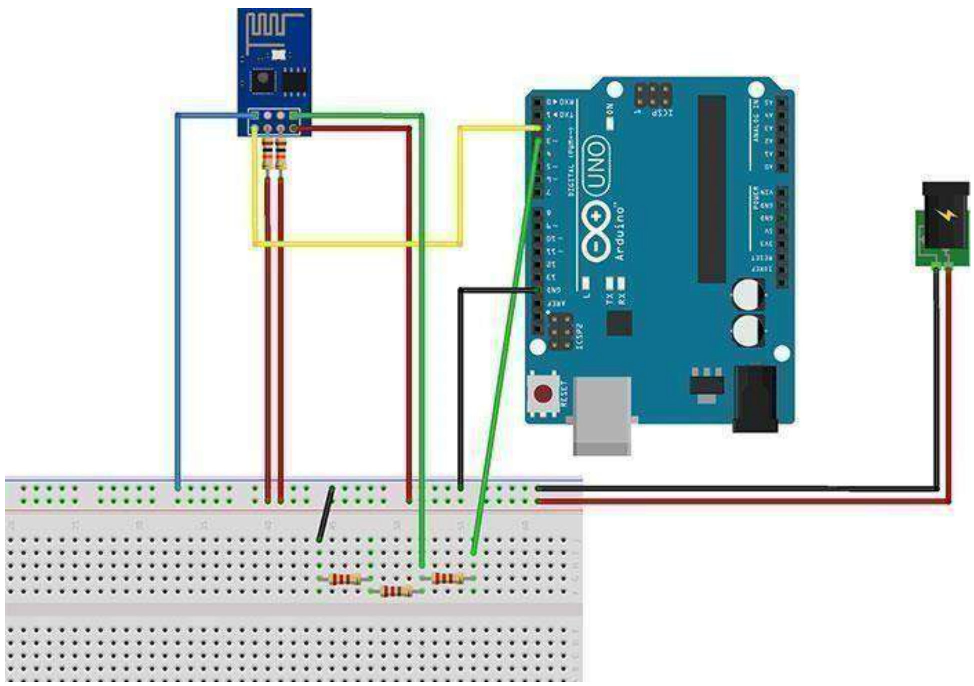


Fig.3.5.4 connection of Fig ESP8266 wi-fi module

You will need the following parts:

- Arduino Uno or similar module
- ESP8266 ESP-01 module
- 2 x 1kΩ resistors for CH_PD and RESET pull-up

- 3 x 220Ω resistors for serial line voltage divider
- 3.3V regulated power supply for ESP8266 (you cannot plug the power line into the Arduino's 3.3V out pin because the ESP8266 draws a lot of current!)
- Breadboard and jumper wires

A couple of features of this circuit stand out immediately.

First, the ESP8266 module is powered by a 3.3V regulated power supply. As described earlier, the ESP8266 has a maximum voltage input rating of 3.3 volts. Using a power supply rated higher than this recommended 3.3 volts could fry the module! When connecting the ESP8266 to this power supply, you should not only connect the VCC and Ground pins to the power supply lines, but also connect the power supply's ground line to the Arduino's ground pin. Since the Arduino will be powered by the USB connection to the laptop, creating a common ground essentially creates a common reference (you can think of it as a baseline) to compare voltages and thereby interpret digitalhigh and low signals.

Second, the receive (RX) line for the ESP8266 module is connected to the output of a resistor voltage-divider circuit. We do this to shift the serial communication logic level (the highs and lows of the digital signals that make up the serial communications) from a logic high of 5 volts on the Arduino to a logic high of 3.3 volts on the ESP8266 module. Again, the ESP8266 is specified with 3.3V logic, so connecting the module's receive line directly to the Arduino's transmit line could damage the device. We do not need to shift levels on the ESP8266's transmit line because the module's 3.3V logic high is a high enough voltage to also register as a logic high on the Arduino.

Finally, there are two pull-up resistors on the ESP8266's CH_PD and RESET pins. These are specified by the module's creators to ensure that the device functions properly while it is being used. To reset the ESP8266 or disable it, these pins must be pulled down, but since we do not want those processes to occur, we will instead pull these pins high through some 10kΩ pull-up resistors.

3.5.5 APPLICATIONS

- IOT Projects
- Access Point Portals
- Wireless Data logging
- Smart Home Automation

- Learn basics of networking
- Portable Electronics
- Smart bulbs and Sockets

3.6 16 * 2 ALPHANUMERIC LCD

3.6.1 WHAT IS LCD

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

3.6.2 BLOCK DIAGRAM:

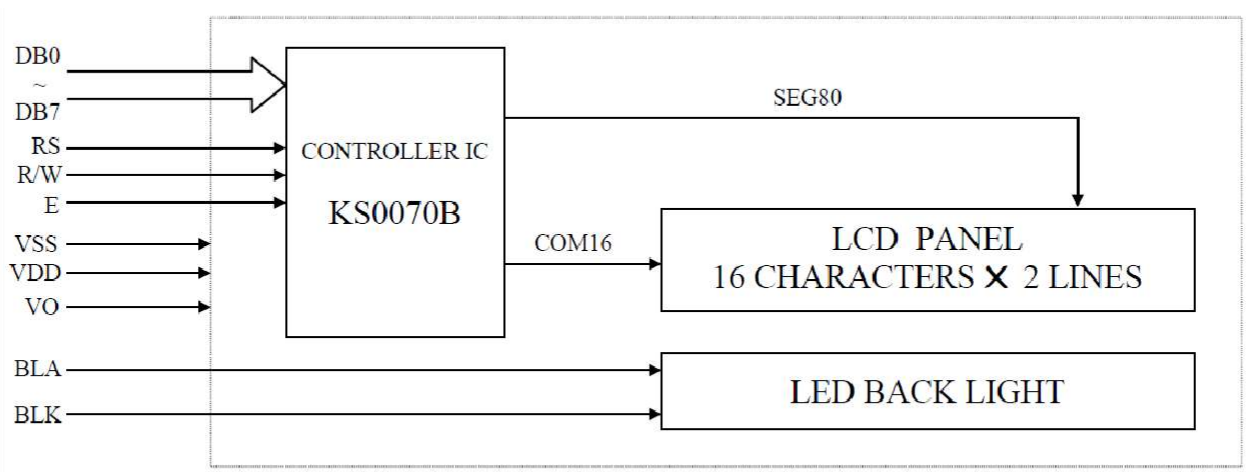


Fig 3.6.2 Block diagram of 16*2 LCD

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.

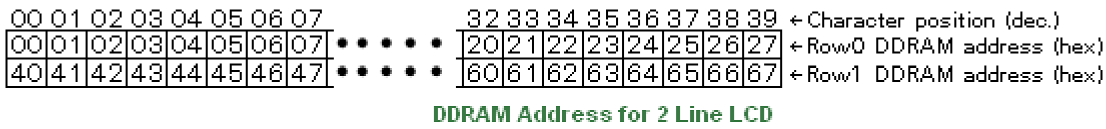


Fig.3.6.3 DDRAM addresses of 2 Line LCD

3.6.3 CHARACTER GENERATOR ROM:

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM.

3.6.4 BUSY FLAG:

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e. BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

3.6.5 INSTRUCTION REGISTER (IR) AND DATA REGISTER (DR):

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g. LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD. Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

3.6.6 16 X 2 ALPHANUMERIC LCD MODULE FEATURES:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

3.6.7 SCHEMATIC DIAGRAM:

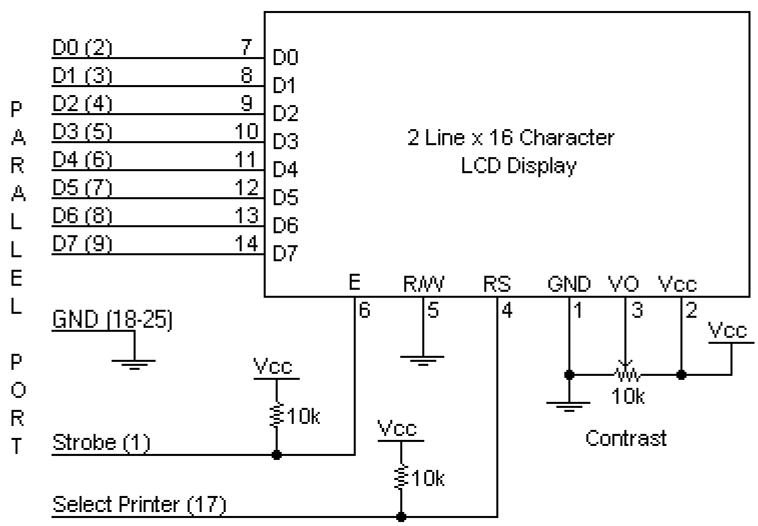


Fig 3.6.7 Schematic Diagram of LCD

3.6.8 SPECIFICATIONS:

Table 3.6.8 Specifications of 16*2 LCD Display

Connector Pin Assignment:

Pin	*Symbol*	*Function*	*Pin*	*Symbol*	*Function*
14	DB7	Data Bus Line	6	E	Enable
13	DB6	Data Bus Line	5	R/W	Read/Write
12	DB5	Data Bus Line	4	RS	Register select input
11	DB4	Data Bus Line	3	Vcontrast	Contrast (0V for max contrast)
10	DB3	Data Bus Line	2	**GND (0V)**	Power supply
9	DB2	Data Bus Line	1	**Vcc +5V**	Power supply
8	DB1	Data Bus Line			
7	DB0	Data Bus Line			

Note1: Pin 1 is +5V and pin 2 is GND! This is different from most other HD44780 compatible models.
Note2: Pin 14 is marked with a "1" on the back of the PCB. Do not get confused by this.

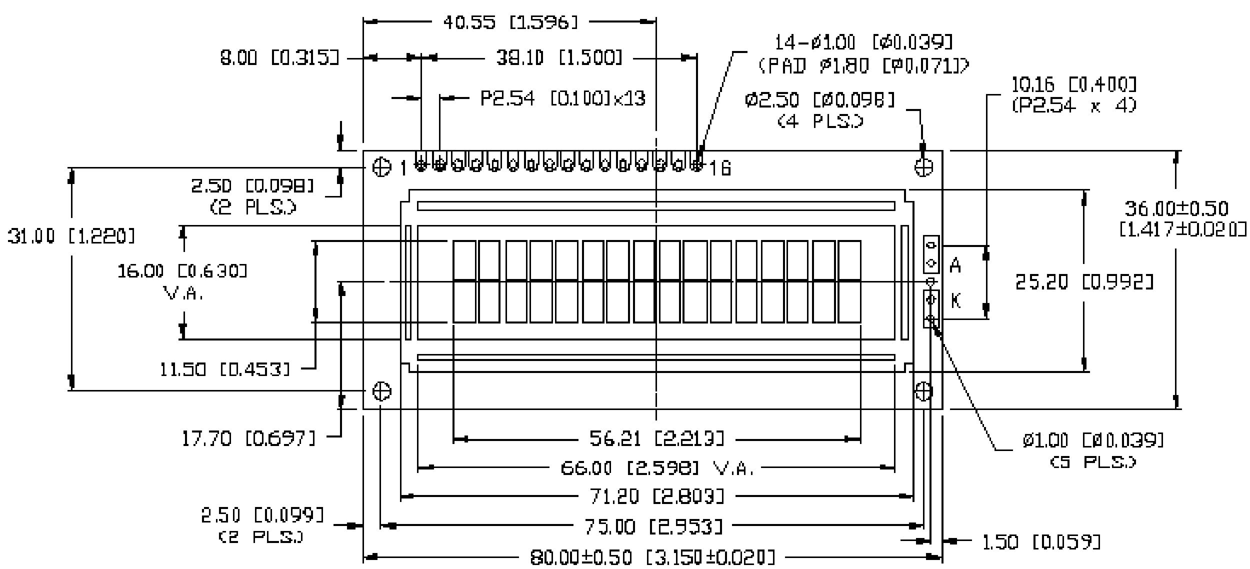


Fig.3.6.9 Connector pin diagram

3.6.9 CIRCUIT DESCRIPTION:

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore, by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore, we hard wire the R/W line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result, we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly. The 2-line x 16-character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

3.6.10 APPLICATIONS

- In most of the applications that's have only small values to show, uses the LCD.
- Most of the commercial meters use this module to represent the data output.
- In the toys and developing projects, it is still vastly in use.
- In black and white printers, it helps to show the printer settings and status.

3.7 ULTRASONIC SENSOR

3.7.1 WHAT IS ULTRA SONIC SENSOR?

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal.

Fig 3.7.1 Ultrasonic sensor

3.7.2 WORKING PRINCIPLE

As shown above the **HC-SR04 Ultrasonic (US) sensor** is a 4pin module, whose pin names are VCC, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications, where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver.

The sensor works with the simple high school formula that

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below

Fig 3.7.2 Ultrasonic sensor working principle

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic-wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

3.7.3 PIN CONFIGURATION

Fig 3.7.3 Pin Configuration

Table 3.7.3 Pin Configuration of Ultrasonic Sensor

Pin Number	Pin Name	Description
1	VCC	The VCC pin powers the sensor, typically with +5V
2	Trigger	TriggerpinisanInputpin.Thispinhastobekepthighfor10usto initialize measurement by sending US wave.
3	Echo	EchopinisanOutputpin.Thispingoeshighforaperiodoftimewhich willbeequaltothetime takenfortheUSwavetoreturnbacktothe sensor.
4	Ground	This pin is connected to the Ground of the system.

3.7.4 HOW TO USE THE HC-SR04 ULTRASONIC SENSOR

HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used.

Power the Sensor using a regulated +5V through the VCC ad Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to bemade high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor.

The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information, the distance is measured as explained in the above heading.

3.7.5 SPECIFICATIONS

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz

3.7.6 APPLICATIONS

- Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot path finding robot etc.
- Medical ultrasonography.

- Burglar Alarms.
- Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water

3.8 DHT11 TEMPERATURE AND HUMIDITY SENSOR

3.8.1 WHAT IS DHT11 SENSOR?

DHT11 is a low-cost digital sensor for sensing temperature and humidity. This sensor can be easily interfaced with any micro-controller such as Arduino, Raspberry Pi etc… to measurehumidity and temperature instantaneously. DHT11 humidity and temperature sensor is availableas a sensor and as a module

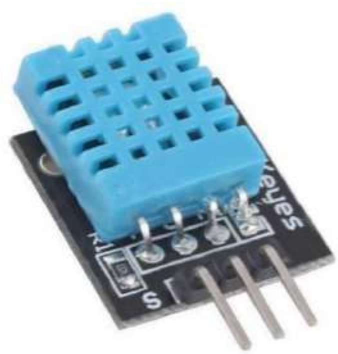


Fig 3.7.1 DHT11 Sensor

3.8.2 WORKING PRINCIPLE

DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels.

The IC measure, process this changed resistance values and change them into digital form. For measuring temperature this sensor uses a Negative Temperature coefficient thermistor, which causes a decrease in its resistance value with increase in temperature. To get larger resistance value even for the smallest change in temperature, this sensor is usually made up of semiconductor ceramics or polymers. The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy.

Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1Hz.i.e., it gives one reading for every second. DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA.

DHT11 sensor has four pins- VCC, GND, Data Pin and a not connected pin. A pull-up resistor of 5k to 10k ohms is provided for communication between sensor and micro-controller.

3.8.3 PIN CONFIGURATION

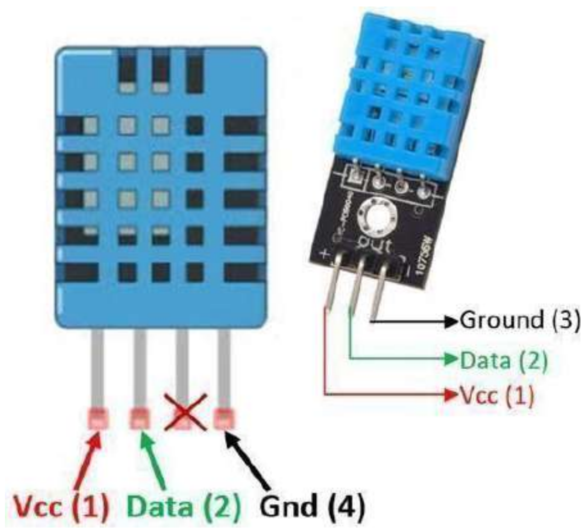


Fig 3.8.3 Pin Configuration

Table 3.8.3 Pin Configuration of DTH11 Sensor

No:	Pin Name	Description
For DHT11 Sensor		
1	VCC	Power supply 3.5V to 5.5V
2	Data	Outputs both Temperature and Humidity through serial Data
3	NC	No Connection and hence not used
4	Ground	Connected to the ground of the circuit
For DHT11 Sensor module		
1	VCC	Power supply 3.5V to 5.5V

2	Data	Outputs both Temperature and Humidity through serial Data
3	Ground	Connected to the ground of the circuit

3.8.4 HOW TO USE

The DHT11 Sensor is factory calibrated and outputs serial data and hence it is highly easy to set it up. The connection diagram for this sensor is shown below.

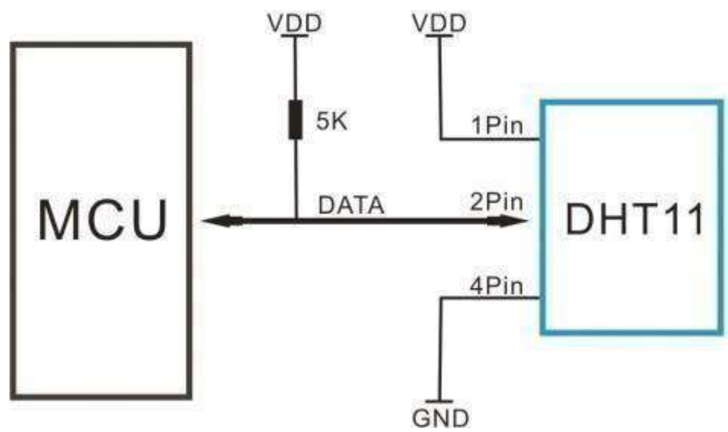


Fig 3.8.4 Connection of DHT11 Sensor

As you can see the data pin is connected to an I/O pin of the MCU and a 5K pull-up resistor is used. This data pin outputs the value of both temperature and humidity as serial data. If you are trying to interface DHT11 with Arduino then there are ready-made libraries for it which will give you a quick start.

If you are trying to interface it with some other MCU then the datasheet given below will come in handy. The output given out by the data pin will be in the order of 8bit humidity integer data + 8bit the Humidity decimal data +8bit temperature integer data + 8bit fractional temperature data +8bit parity bit. To request the DHT11 module to send these data the I/O pin has to be momentarily made low and then held high as shown in the timing diagram below.

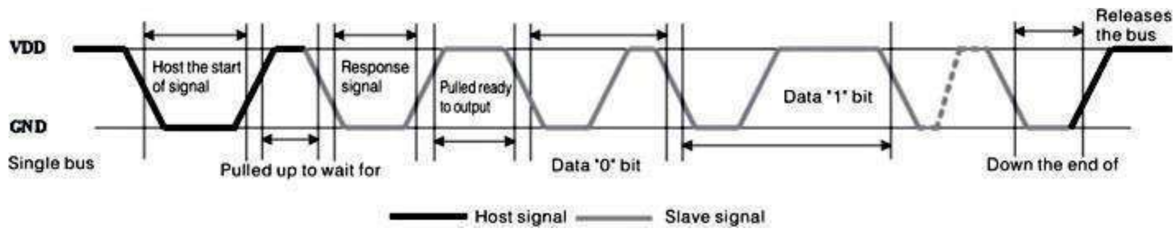


Fig 3.8.4 Timing Diagram of DHT11 Sensor

3.8.5 SPECIFICATIONS

- Operating Voltage: 3.5V to 5.5V
- Operating current: 0.3mA (measuring) 60uA (standby)
- Output: Serial data
- Temperature Range: 0°C to 50°C
- Humidity Range: 20% to 90%
- Resolution: Temperature and Humidity both are 16-bit
- Accuracy: ±1°C and ±1%

3.8.6 APPLICATIONS

- Measure temperature and humidity
- Local Weather station
- Automatic climate control
- Environment monitoring

3.9 PIR SENSOR

3.9.1 WHAT IS PIR SENSOR?

PIR is a short name for PASSIVE INFRA RED Sensor. PIR sensor is used to detect the motion of humans or any particular movement of a particle with up to distance of 10meters at an angle of+150 to -150 degrees.

Fig 3.9.1 PIR Sensor

3.9.2 WORKING PRINCIPLE

The PIR sensors are more complicated than the other sensors as they consist of two slots. These slots are made of a special material which is sensitive to IR. The Fresnel lens is used to see that the two slots of the PIR can see out past some distance. When the sensor is inactive, then the two slots sense the same amount of IR. The ambient amount radiates from the outdoors, walls or room etc.

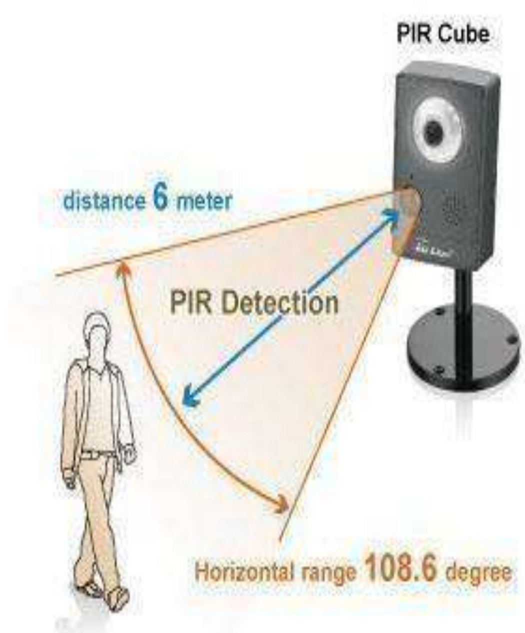


Fig 3.9.2.1 Working principle of PIR sensor

When a human body or any animal passes by, then it intercepts the first slot of the PIR sensor. This causes a positive differential change between the two bisects. When a human body leaves the sensing area, the sensor generates a negative differential change between the two bisects. The infrared sensor itself is housed in a hermetically sealed metal to improve humidity/temperature/noise/immunity. There is a window which is made of typically coated silicon material to protect the sensing element.

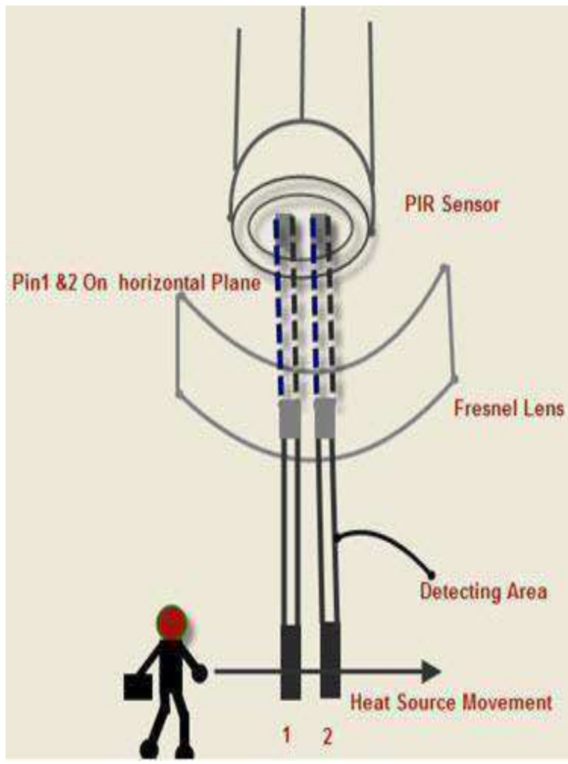
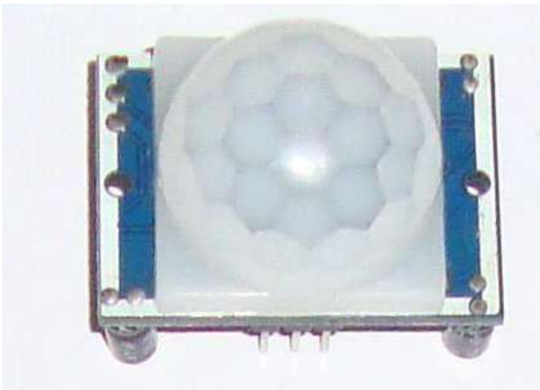


Fig 3.9.2.2 Working principle of PIR sensor

Basically, motion detection use light sensors to detect either the presence of infrared light emitted from a warm object or absence of infrared light when a object interrupts a beam emitted by another part of the device.

A PIR sensor detects the infrared light radiated by a warm object. It consists of pyro electric sensors which introduce changes in their temperature (due to incident infrared radiation) into electric signal. When infrared light strikes a crystal, it generates an electrical charge. Thus a PIR sensor can be used to detect presence of human beings within a detection area of approximately 14 meters

3.9.3 PIN CONFIGURATION



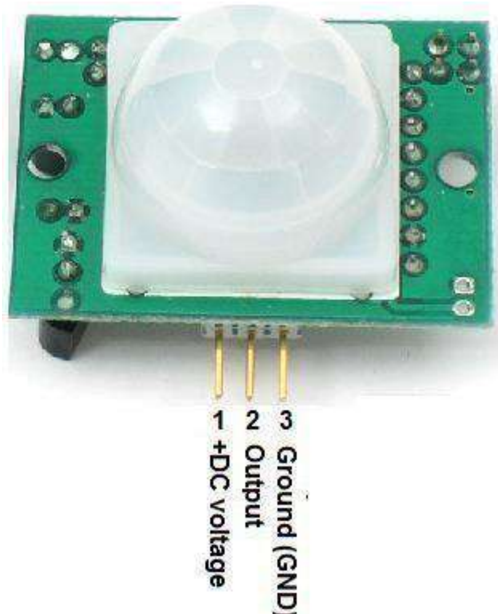


Fig 3.9.3 Pin Configuration

Table 3.9.3 Pin Configuration of PIR Sensor

Pin Number	Pin Name	Description
1	VCC	Inputvoltageis+5Vfortypicalapplications.Canrangefrom 4.5V-12V
2	High/Low Output (Dout)	Digital pulse high (3.3V) when triggered (motion detected) digital low(0V) when idle(no motion detected)
3	Ground	Connected to ground of circuit

3.9.4 SPECIFIACTIONS

- Complete with PIR, Motion Detection.
- Dual Element Sensor with Low Noise and High Sensitivity.
- Supply Voltage – 5V.
- Delay Time Adjustable.
- Standard TTL Output.

3.9.5 APPLICATIONS

- All outdoor Lights
- Lift Lobby
- Multi Apartment Complexes
- Common staircases
- For Basement or Covered Parking Area
- Shopping Malls
- For garden lights

3.10 BLOOD PRESSURE SENSOR

3.10.1 WHAT IS BLOOD PRESSURE SENSOR?

The Blood Pressure Sensor is a non-invasive sensor designed to measure human blood pressure.

Fig 3.10.1 Blood Pressure Sensor

It measures systolic, diastolic and mean arterial pressure utilizing the oscillometric method. Pulse rate is also reported.

3.10.2 WORKING PRINCIPLE

3.10.2.1 Using manual Machine:

Usually, pressure cuff linked to a mercury column is used to measure the blood pressure. Here, the doctor manually pumps the cuff to increase the pressure on the artery. Then using stethoscope, the noise of the blood rushing through the artery.

Fig 3.10.2 Manual BP Machine

3.10.2.2 Using Automated Machine

In automatic Blood Pressure measurement system, instead of mercury a pressure sensor is used to detect the pressure in the artery and give output. This digital output is displayed on the monitor. This monitor has an onboard processor to process the output given by pressure sensor, record results and display them on the digital read-out screen.

Fig 3.10.2.2 Automated BP Machine

3.10.3 SPECIFICATIONS

- Pressure range: 0 mmHg to 258 mmHg
- Maximum pressure without permanent damage: 1550 mmHg
- Typical accuracy: ±1 mmHg
- Temperature compensated: -20°C to 85°C
- Sensing element: SSCMRRN005PGAA5
- Combined linearity and hysteresis: typical ±0.25%
- Response time: 1 millisecond

CHAPTER-4
SOFTWARE DESCRIPTION

4.1 SOFTWARE EXPLANATION:

SOFTWARE REQUIREMENTS:

- Arduino software
- Programming language
- Proteus simulation

4.1.1 ARDUINO SOFTWARE:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments.

A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon asit reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments.

All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

4.1.2 WHY ARDUINO?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step-by-step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handy board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows, wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.

- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

Open source and extensible hardware - The plans of the Arduino boards are published under

- a Creative Commons license, so experienced circuit designers c an make their own version
- Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money
- Getting Started with Arduino and Genuino products: -

Install the Arduino Software (IDE) on Windows PCs-

This document explains how to install the Arduino Software (IDE) on Windows machines.

- ✓ Download the Arduino Software (IDE)
- ✓ Proceed with board specific instructions.

4.1.3 HOW TO DOWNLOAD THE ARDUINO SOFTWARE (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

4.1.4 INSTALLATION:

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.4.1.4.1 Standard USB Cable (A plug to B plug)

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig.4.1.4.2 Standard USB Cable (an A to Mini-B plug)

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

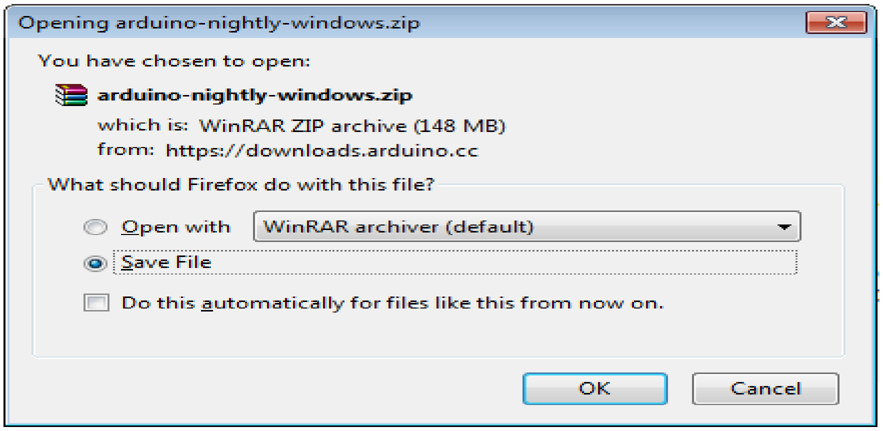


Fig.4.1.4.3 Download Arduino IDE SoftwareStep 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

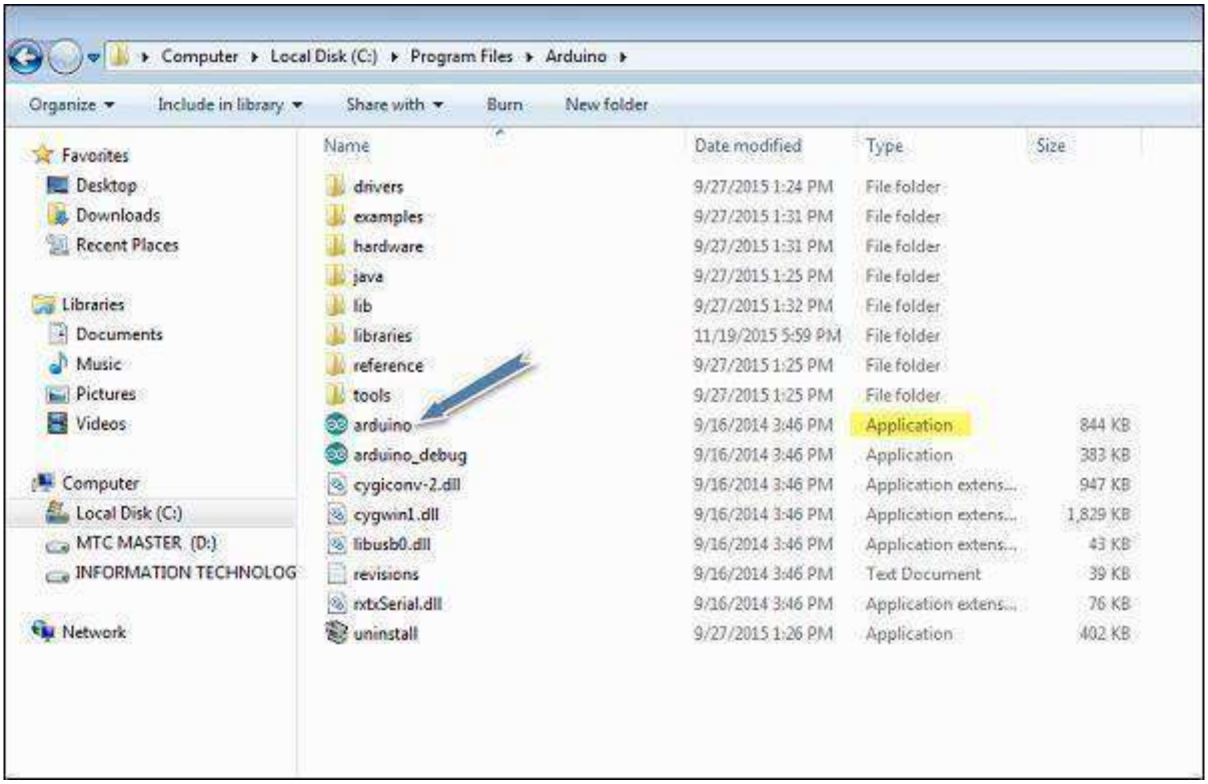


Fig.4.1.4.4 Launch Arduino IDEStep 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File →**New**

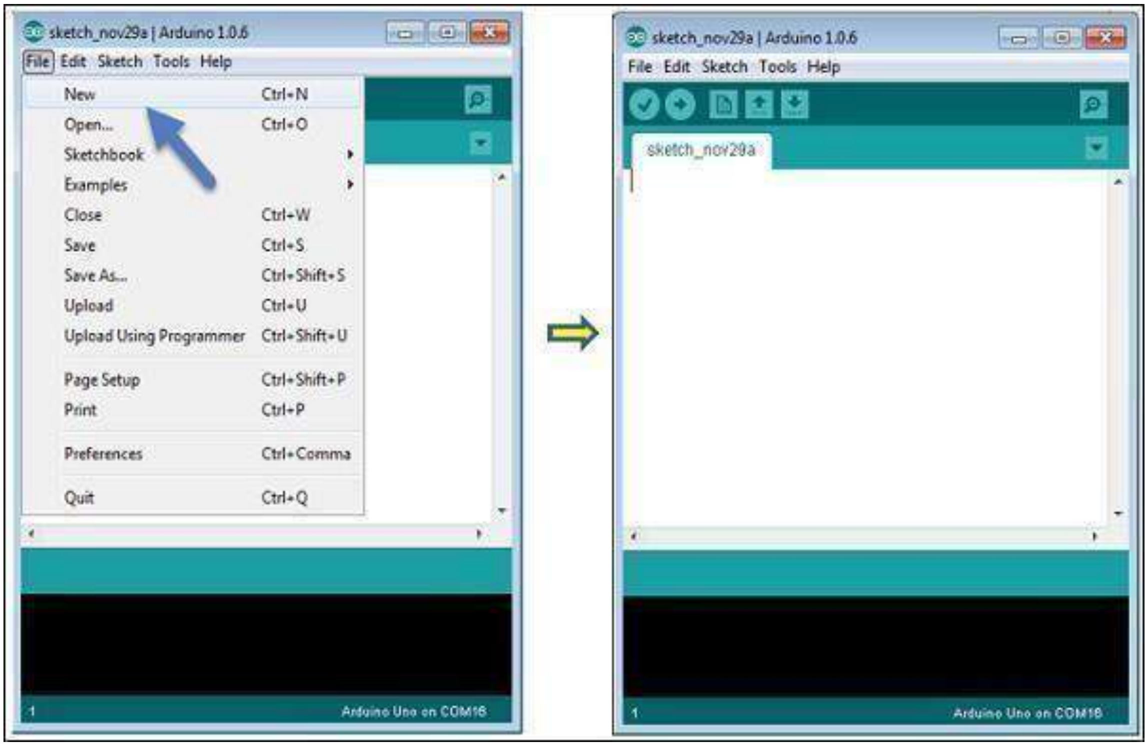
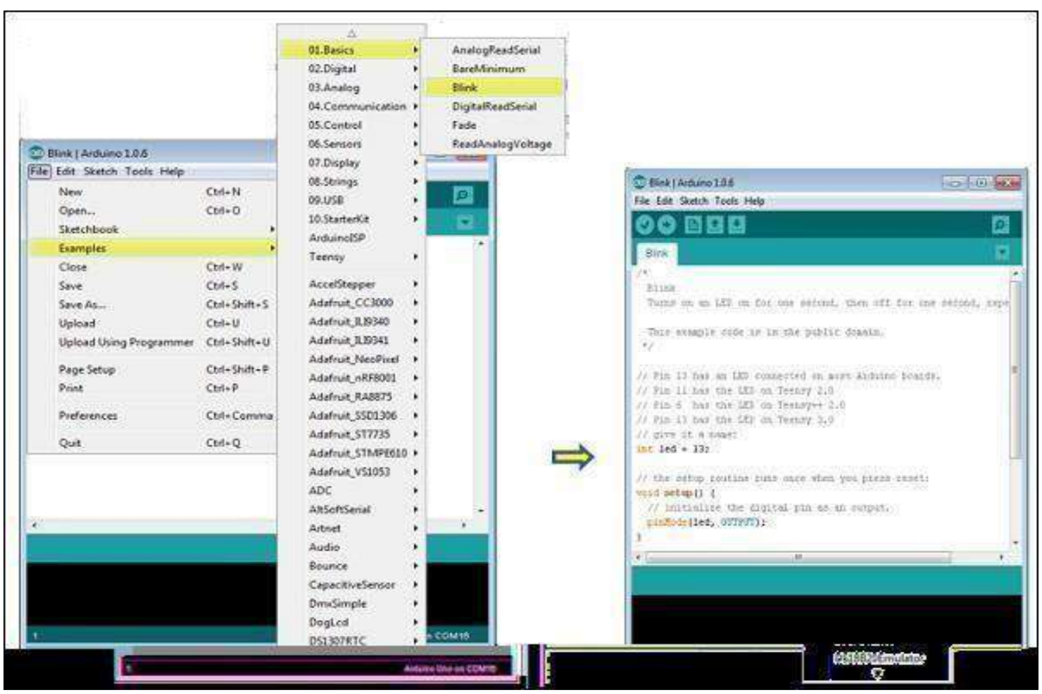


Fig.4.1.4.5 Open your first project.

To open an existing project example, select File → Example → Basics → Blink.



.

Fig.4.1.4.6 Open your first program.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

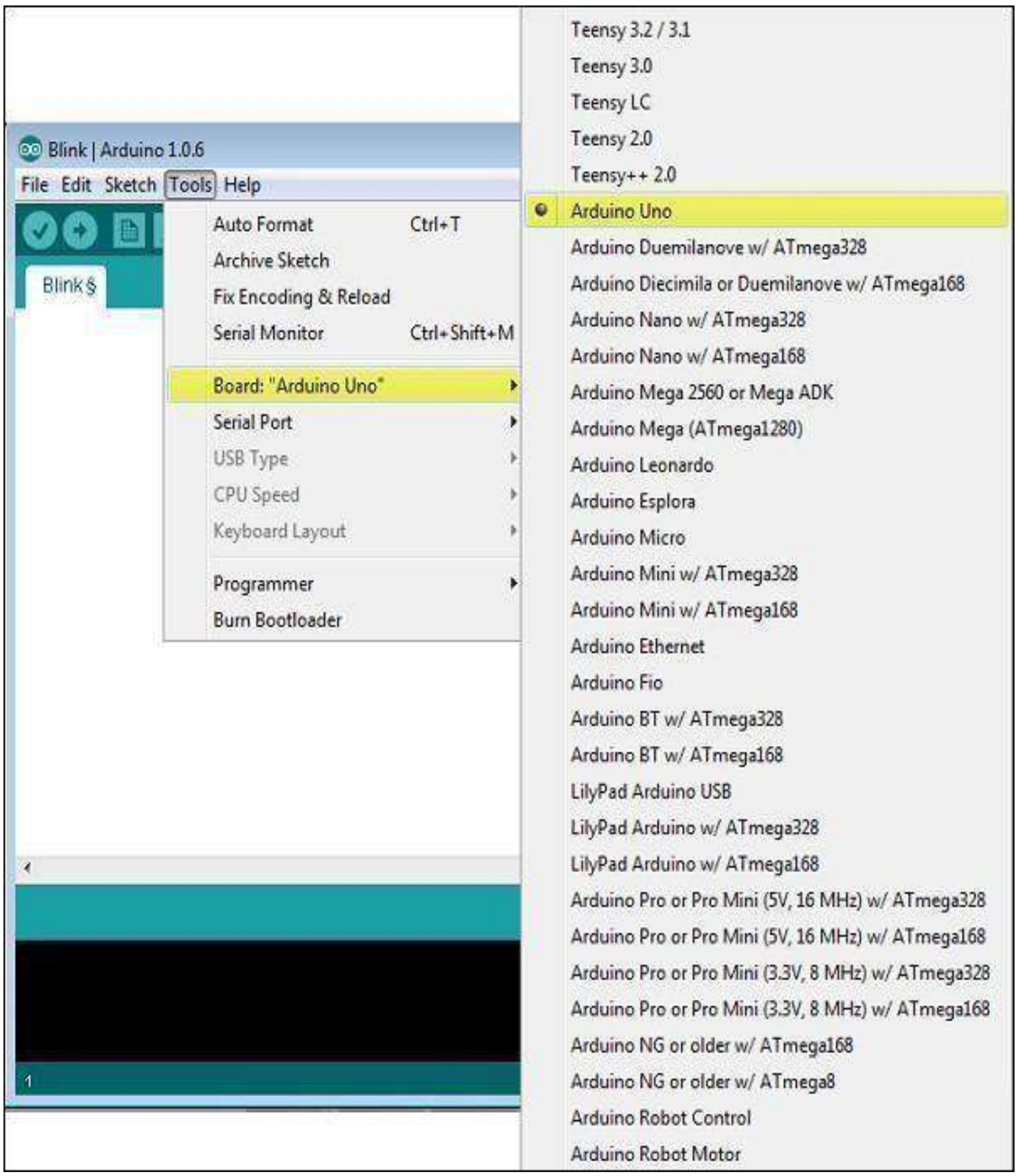


Fig.4.1.4.7 Select your Arduino board.

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports).To find out, you can disconnect your Arduino board and re-open the menu, the entry thatdisappears should be of the Arduino board. Reconnect the board and select that serial port.

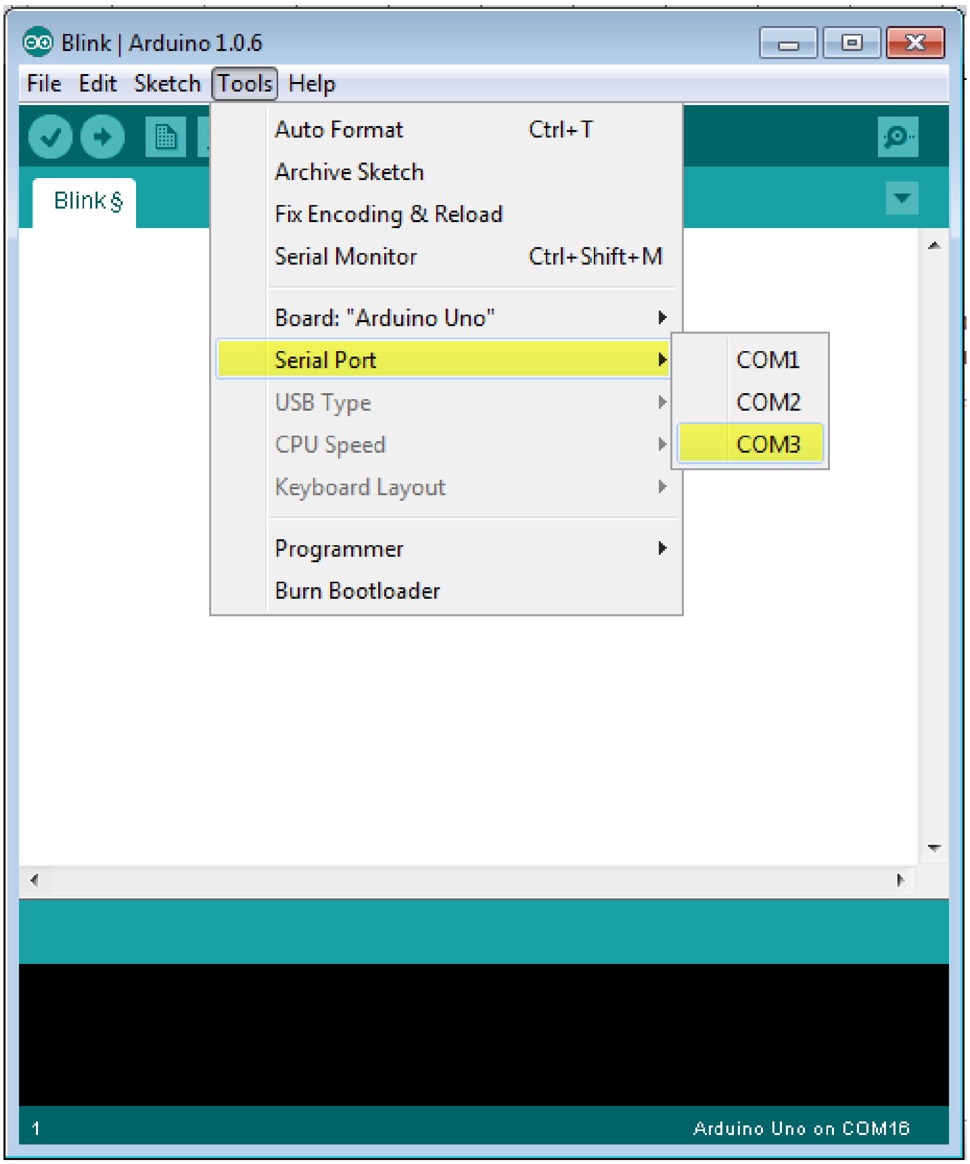


Fig.4.1.4.8 Select your serial port.

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

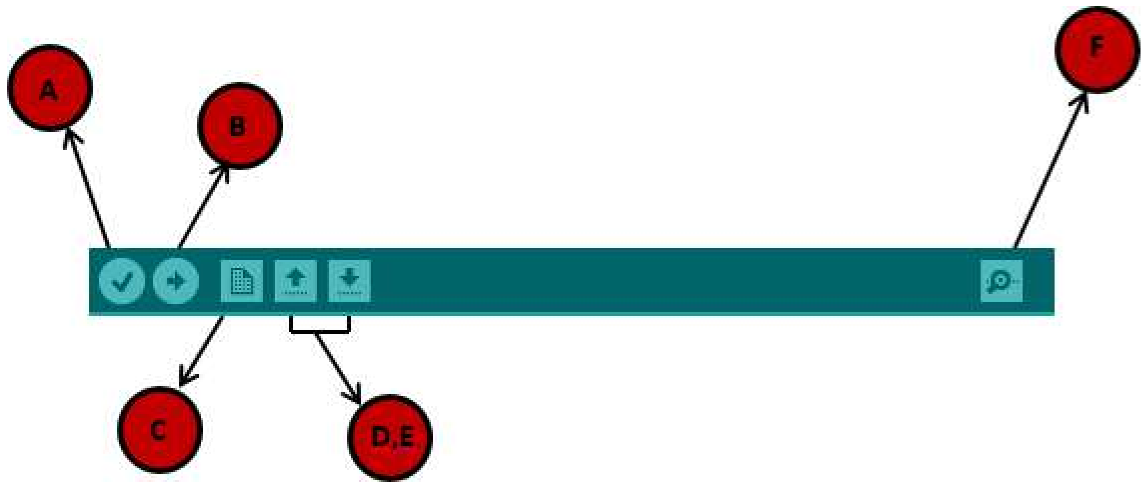


Fig.4.1.4.9 Upload the program to your board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from board & send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and GND connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

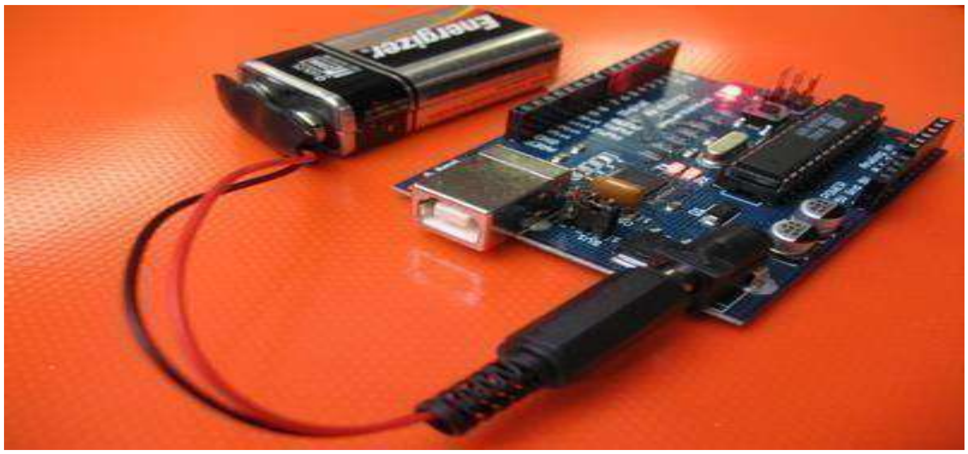


Fig.4.1.4.10 Connecting a Battery

.

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called "sketches", but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

Fig.4.1.4.11 Example Program

4.1.5 MC PROGRAMMING LANGUAGE: EMBEDDED C :

This is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements. Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

4.2 PROTEUS:

PROTEUS:

Proteus is a simulation and design software tool developed by Lab Centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline "From concept to completion".

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integratedcircuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design:

Step 1: Open ISIS software and select New design in File menu

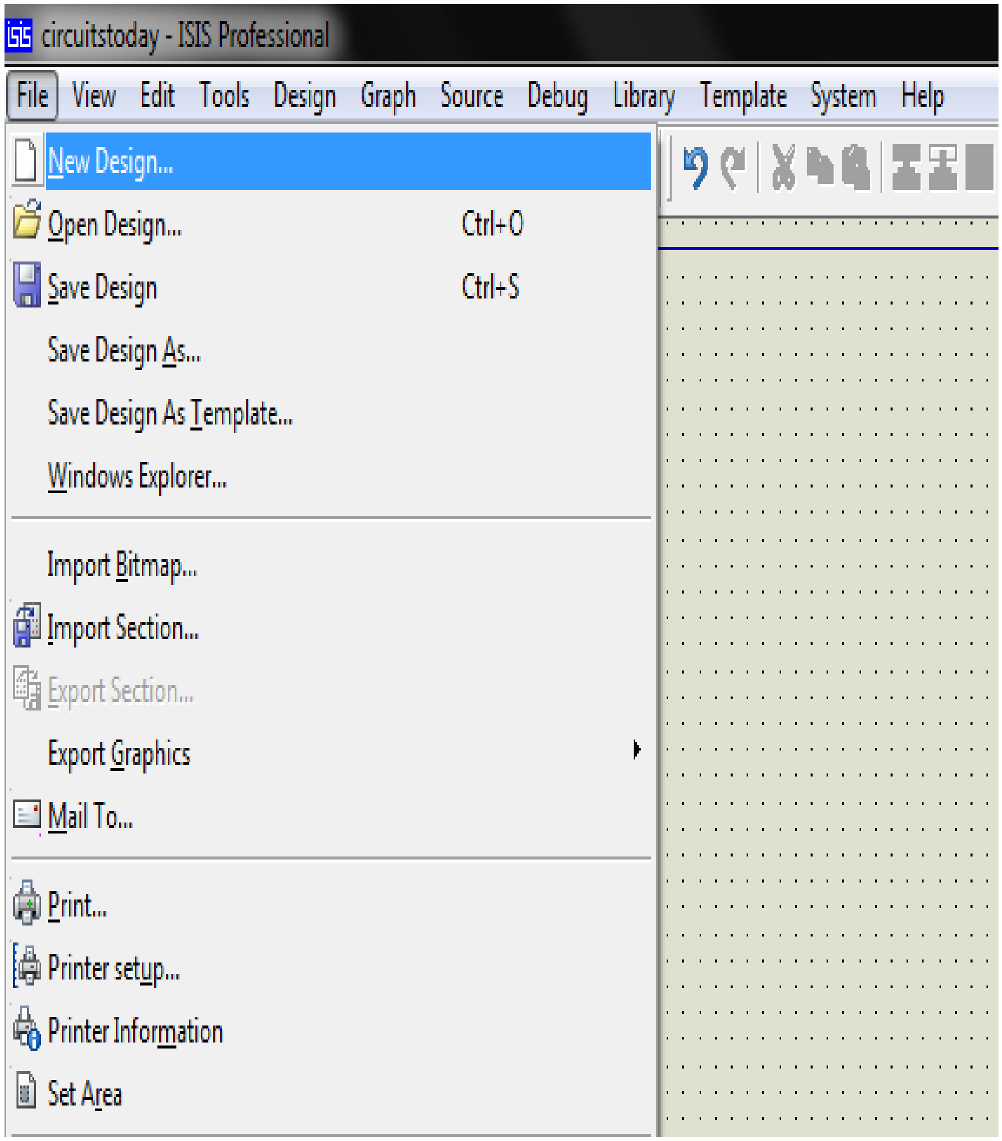


Fig.4.2.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

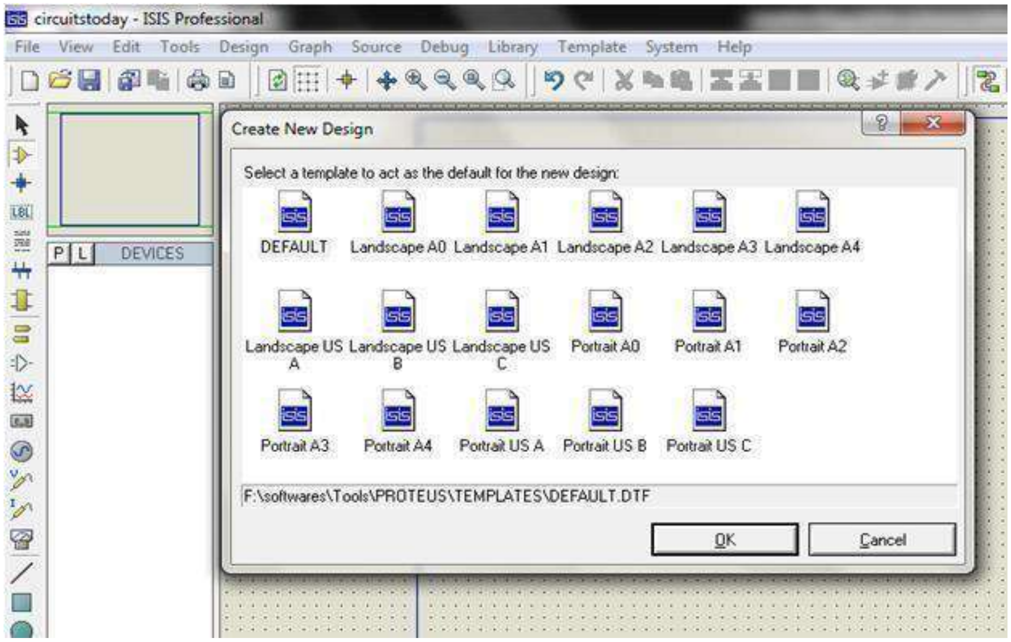


Fig.4.2.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

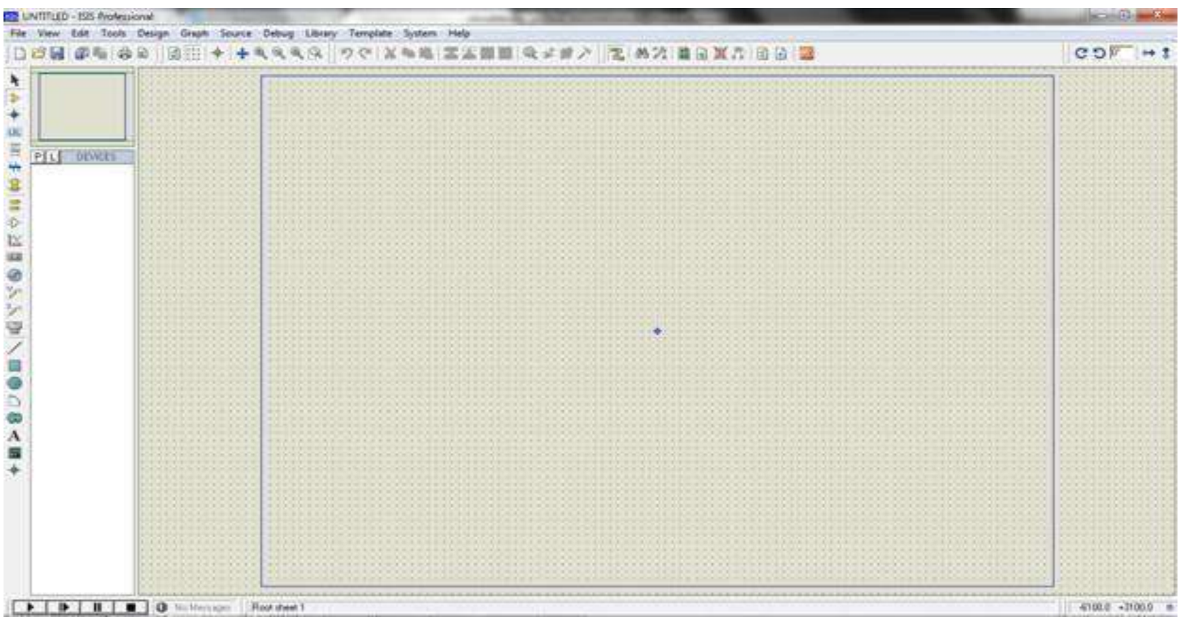


Fig.4.2.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

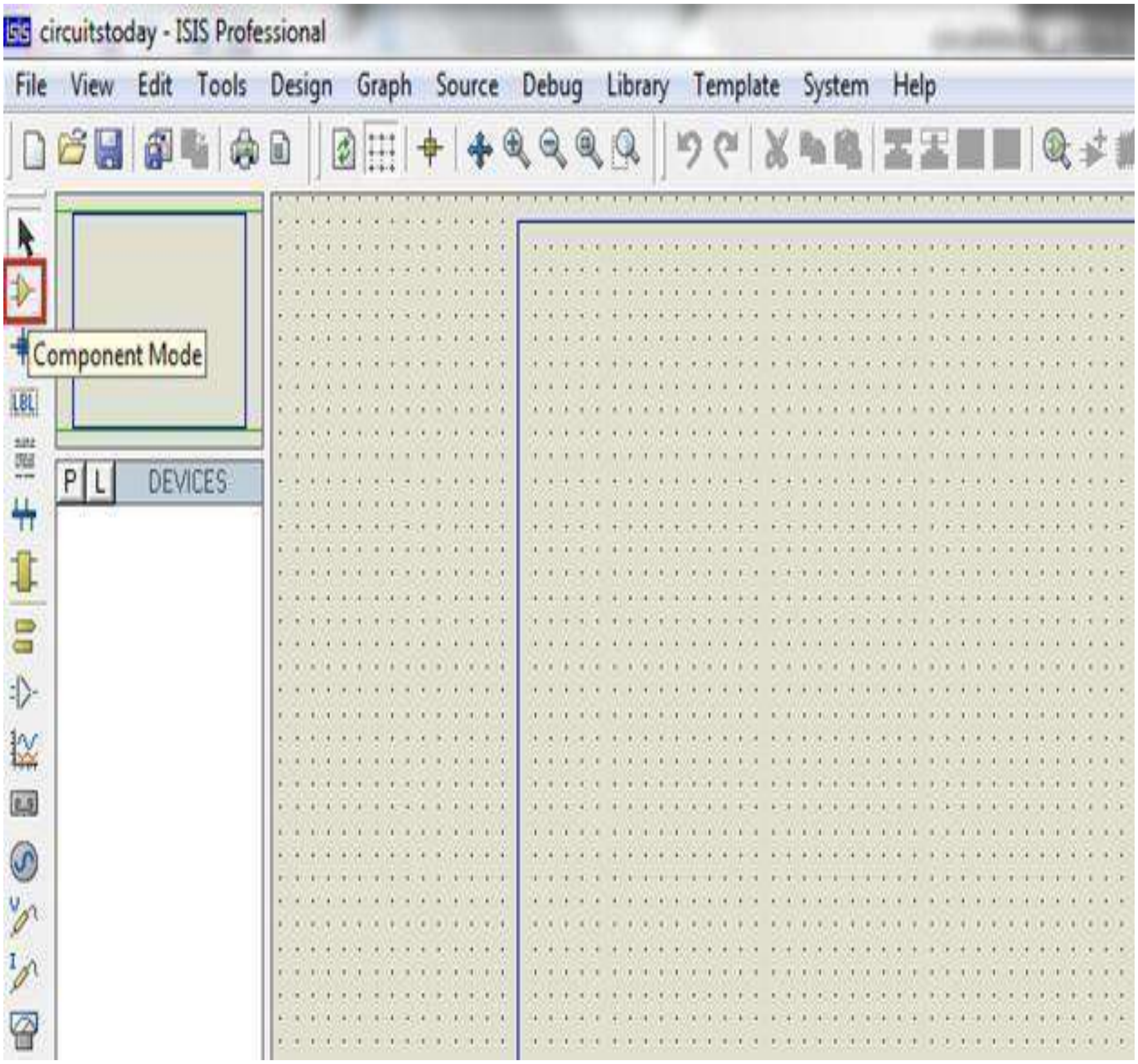


Fig.4.2.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

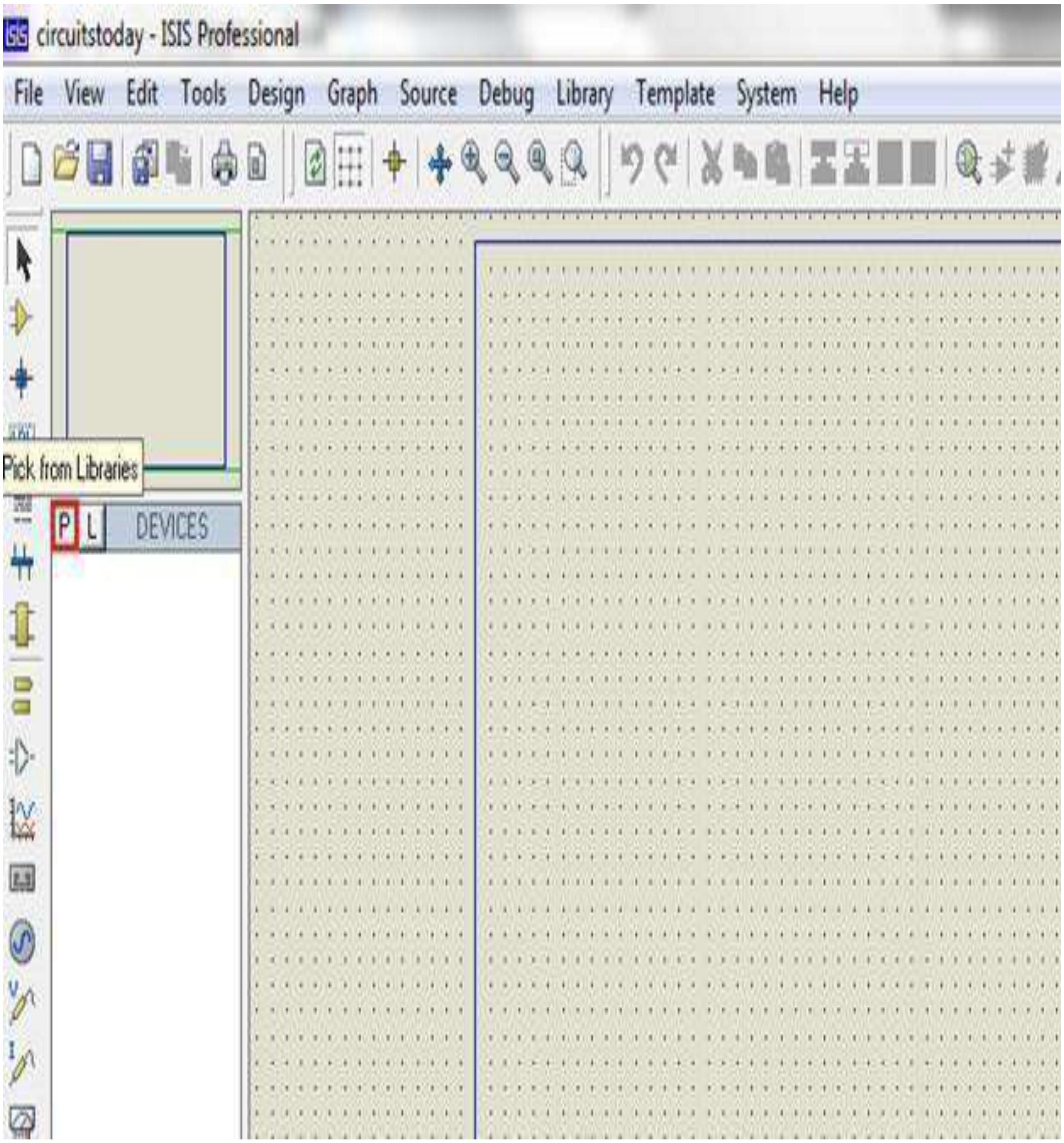


Fig.4.2.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

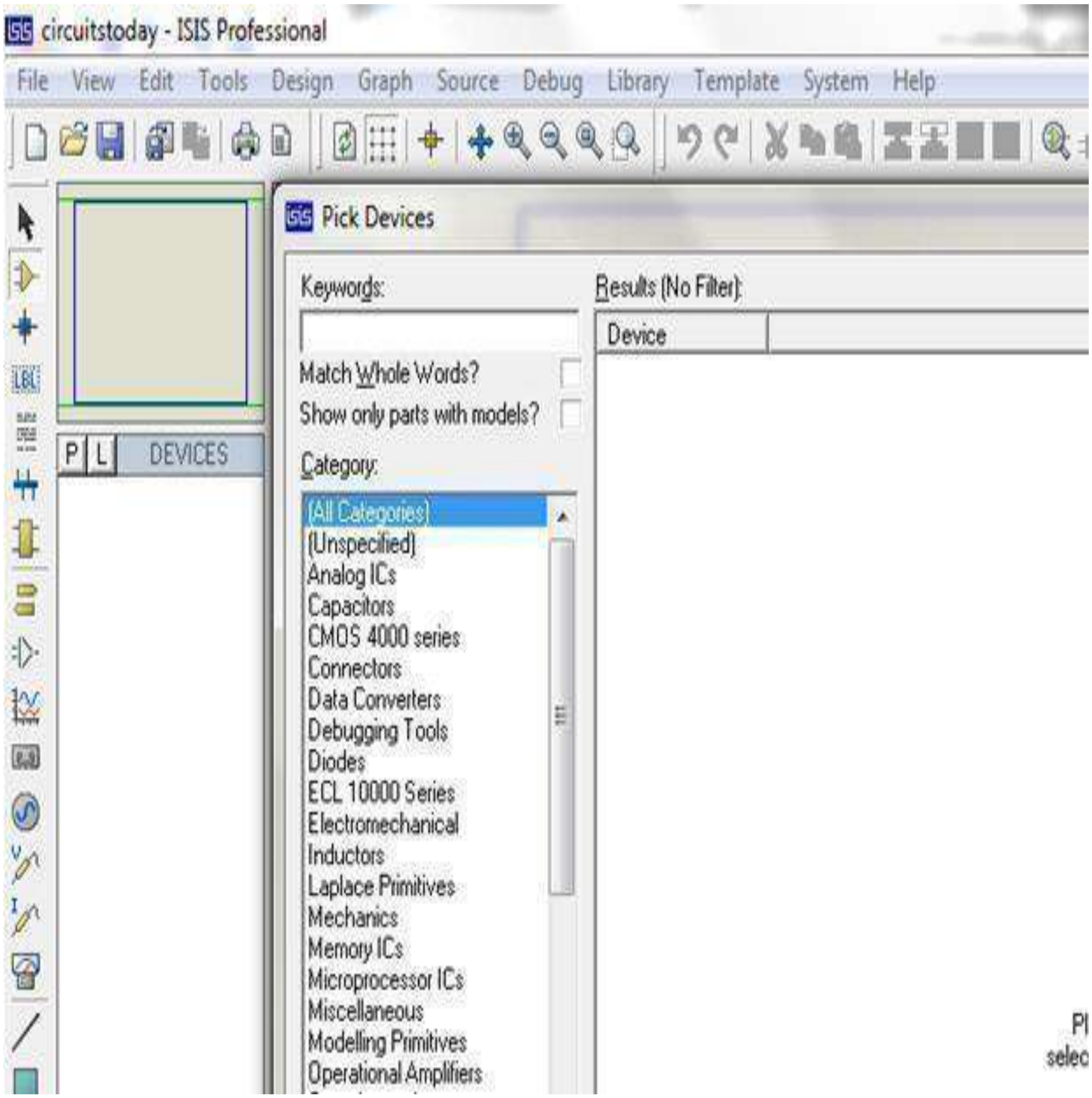


Fig.4.2.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

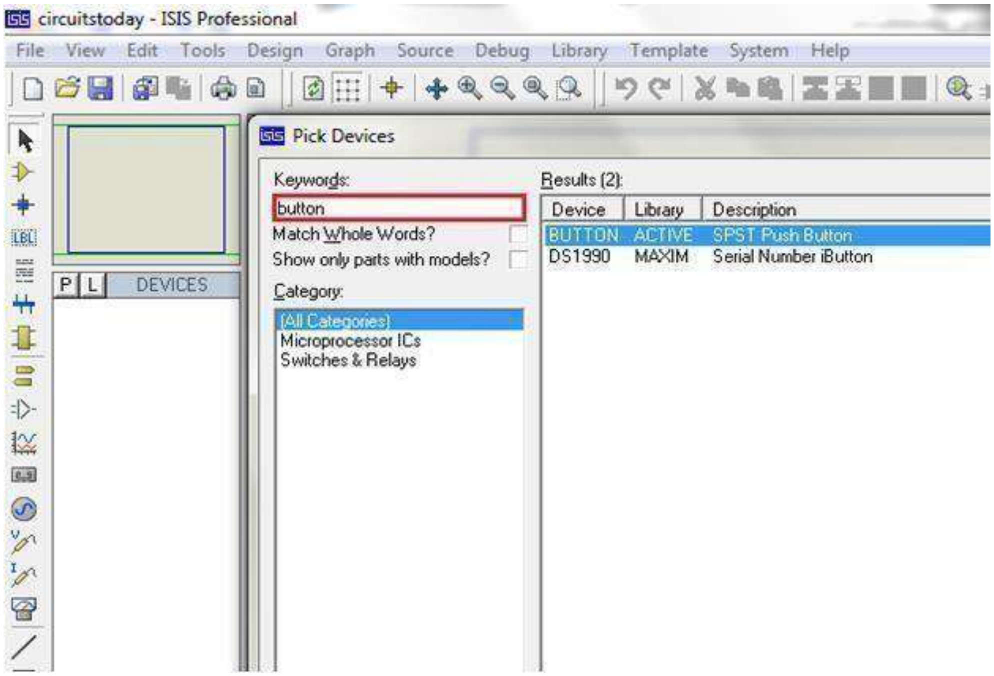


Fig4.2.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

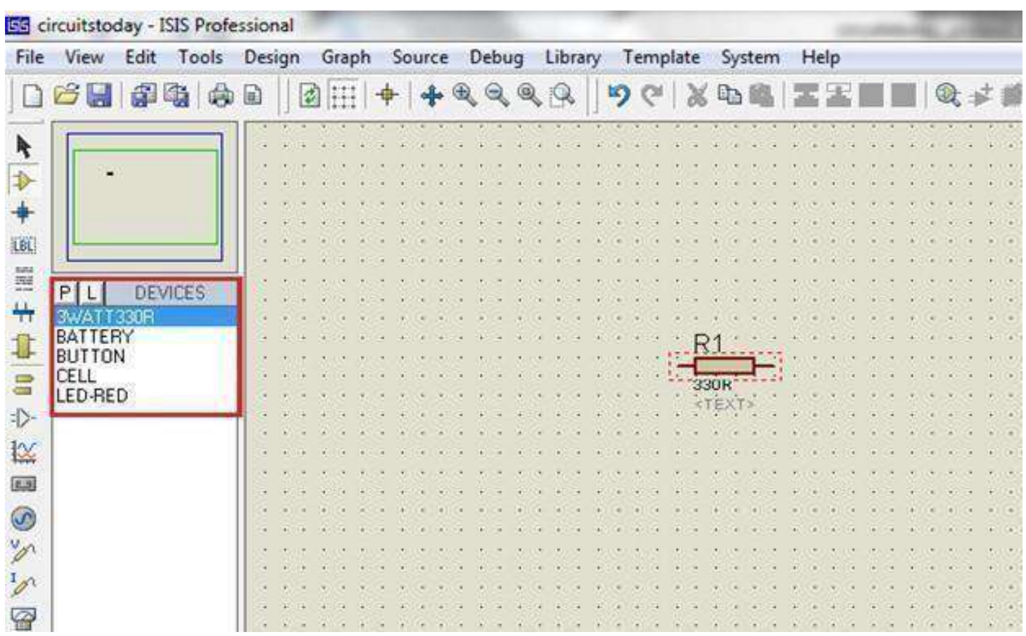


Fig.4.2.8 Component Selection

Place all the required components and route the wires i.e., make connections. Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

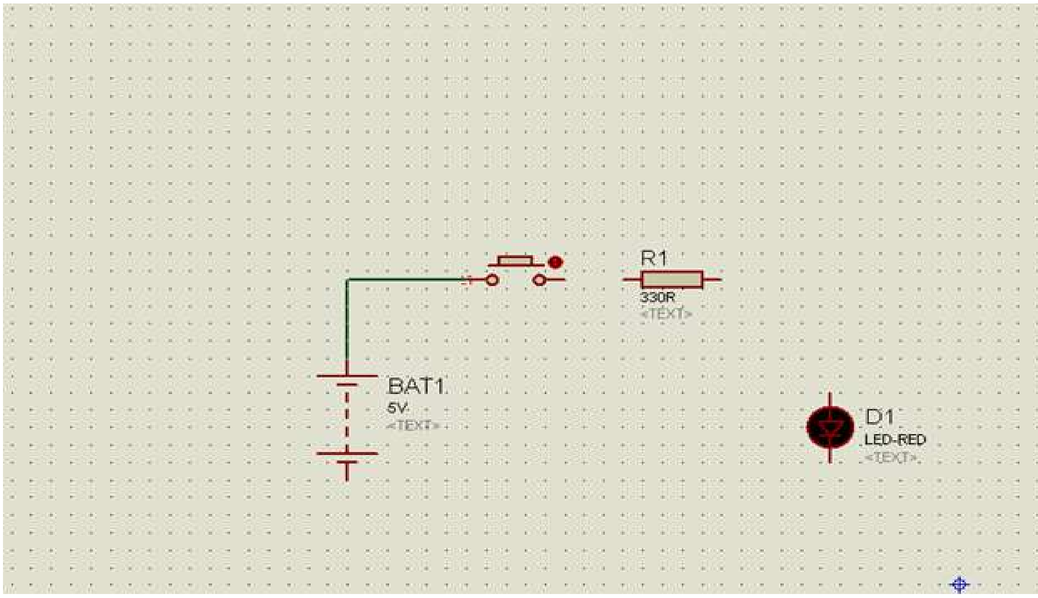


Fig.4.2.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

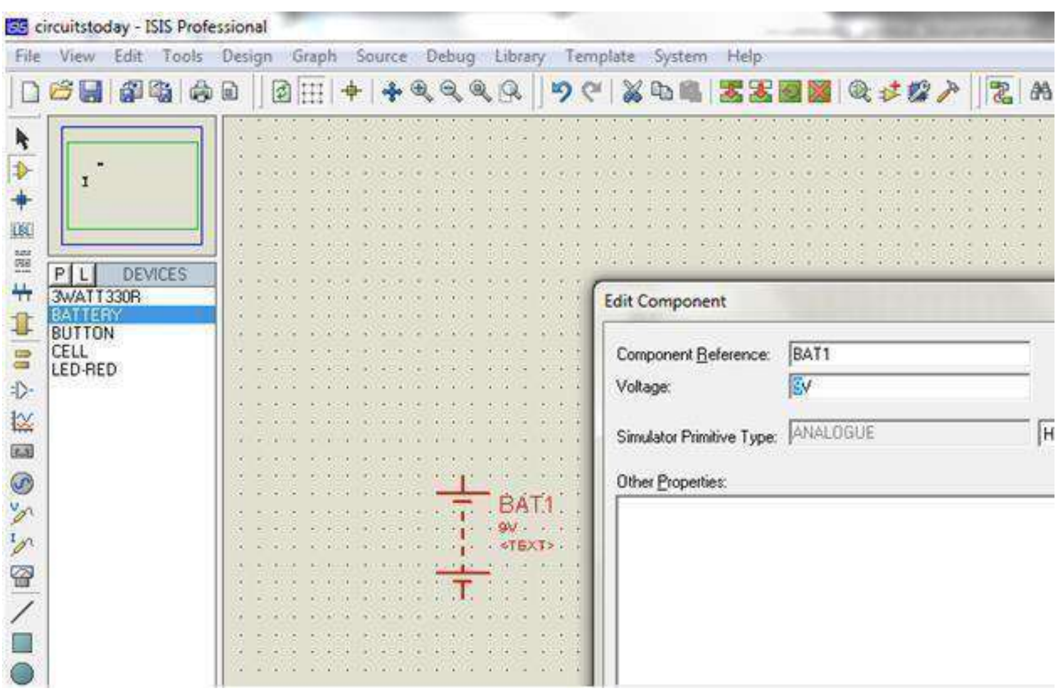


Fig.4.2.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

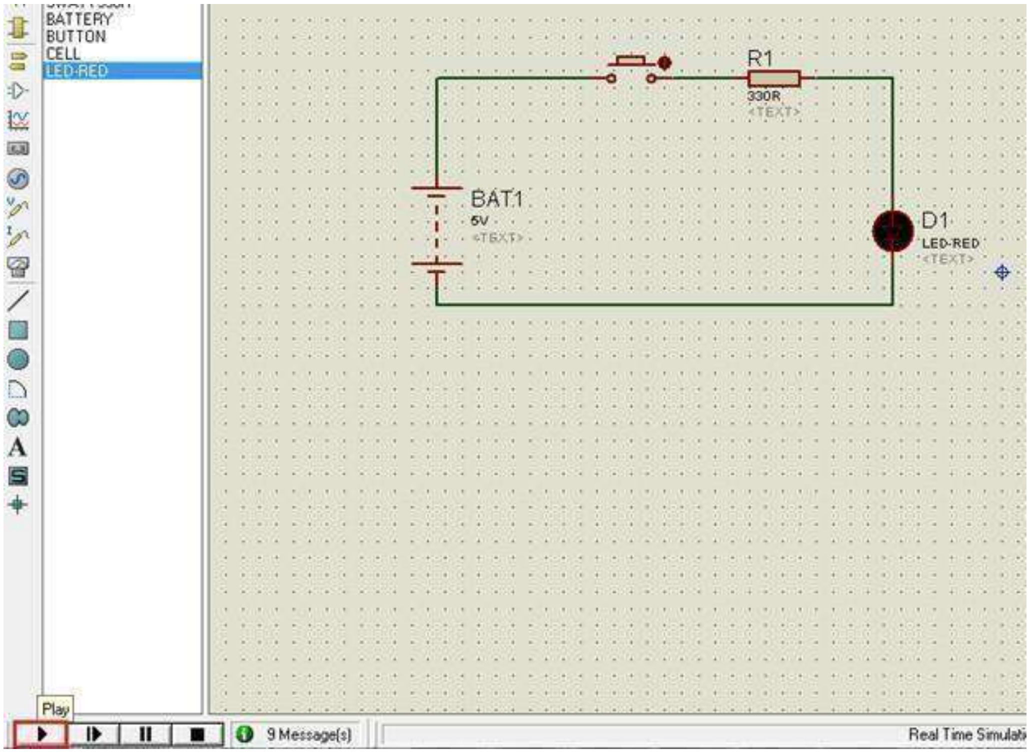


Fig.4.2.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

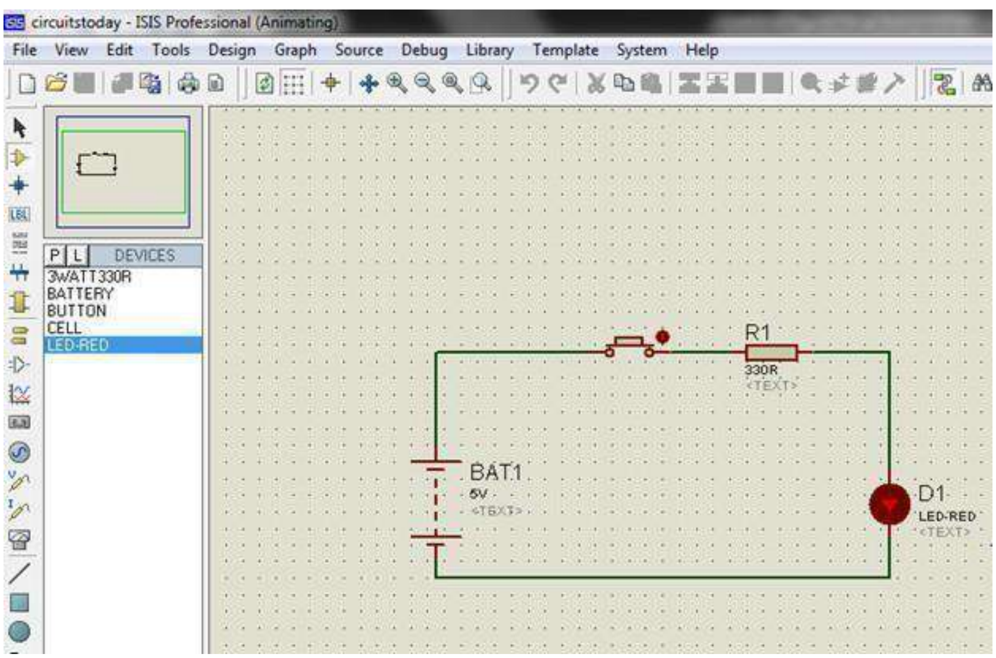


Fig4.2.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

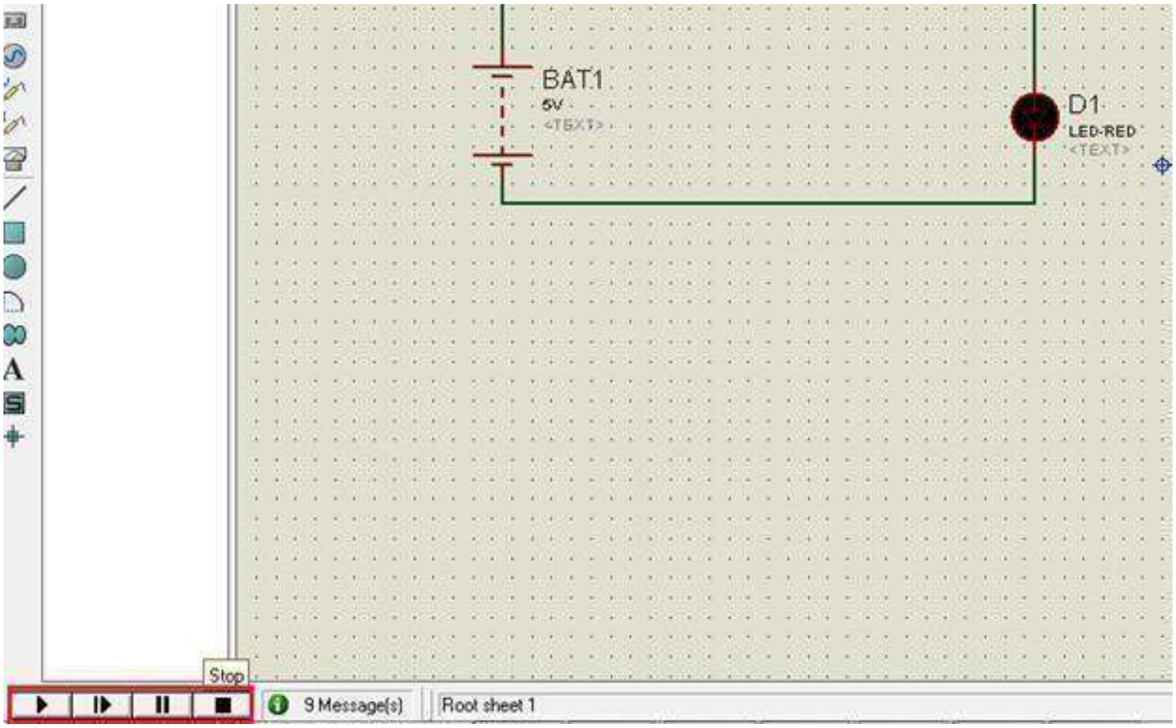


Fig.4.2.13 Simulation Step-Pause-Stop Buttons

According to its developers, Things Speak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. Things Speak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates.

Things Speak was originally launched by IoT Bridge in 2010 as a service in support of IoT applications.

Things Speak has integrated support from the numerical computing software MATLAB from MathWorks, allowing Things Speak users to analyze and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license from Math works.[citationneeded].

Things Speak has a close relationship with Math works, Inc. In fact, all of the Things Speak documentation is incorporated into the Math works' Matlab documentation site and even enabling registered Math works user accounts as valid login credentials on the Things Speak website. The terms of service and privacy policy of ThingSpeak.com are between the agreeing user and Math works, Inc.

Setup Things Speak

- Create a free MathWorks account or sign into Things Speak using an existing account.
- Select the Things Speak channel you want your data to stream into. See Collect Data in a New Channel for help creating a new channel.
- Record the following for the selected channel:
- Channel ID, which is listed at the top of the channel view.
- Write API key, which can be found on the API Keys tab of your channel view.

CHAPTER – 5
SOFTWARE TESTING AND CODE IMPLEMENTATION

5.1 SOFTWARE TESTING

Software Testing is a process of executing the application with an intent to find anysoftware bugs. It is used to check whether the application met its expectations and all the functionalities of the application is working. The final goal of testing is to check whether the application is behaving in the way it is supposed to under specified conditions. All aspects of the code are examined to check the quality of application. The primary purpose of testing is to detect software failures so that defects may be uncovered and corrected. The test cases are designed in such way that scope of finding the bugs is maximum.

5.2 TESTING LEVELS

There are various testing levels based on the specificity of test.

- **Unit testing**: Unit testing refers to tests conducted on a section of code in order to verify the functionality of that piece of code. This is done at the function level.

- **Integration Testing**: Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Its primary purpose is to expose the defects associated with the interfacing of modules.

- **System Testing**: System testing tests a completely integrated system to verify that the system meets its requirements.

- **Acceptance testing**: Acceptance testing tests the readiness of application, satisfying all requirements.

- **Performance testing:** Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or device such as response time or millions of instructions per second etc.

5.3 SYSTEM TEST CASES

A test case is a set of test data, preconditions, expected results and post conditions, developed for a test scenario to verify compliance against a specific requirement. We have designed and executed a few test cases to check if the project meets the functional requirements.

5.4 CODE IMPLEMENTATION

5.4.1 PROJECT CODE

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2,3);


#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);




String readvoice;
char buff[200],k=0;
char c=0,c1=0;
 unsigned char rec = 0,buf[120],buf1[120],buf3[120];
 unsigned int count=0,ik=0,ct=0,value=0;
void upload1(unsigned char *chr,unsigned char *chr1,unsigned char *chr2,unsigned char *chr3);
 char res[130];
char check(char* ex,int timeout)
{
  int i=0;
```

```
int j = 0,k=0;
while (1)
{
  sl:
  if(mySerial.available() > 0)
  {
    res[i] = mySerial.read();
    if(res[i] == 0x0a || res[i]=='>' || i == 100)
    {
      i++;
      res[i] = 0;break;
    }
    i++;
  }
  j++;
  if(j == 30000)
  {
    k++;
    //Serial.Sprintln("kk");
    j = 0;
  }
  if(k > timeout)
  {
   // Serial.println("timeout");
    return 1;
```

```
      }
   }//while 1
   if(!strncmp(ex,res,strlen(ex)))
   {
     //Serial.println("ok..");
     return 0;
    }
   else
   {
    // Serial.print("Wrong ");
    // Serial.println(res);
     i=0;
     goto sl;
    }
 }

 void serialFlush(){
   while(Serial.available() > 0) {
   char t = Serial.read();
   }
 }



 int temp=A4,pir=6,buz=7;
 const int trigPin = 6;
```

```
const int echoPin = 5;
long duration;
int distance;
const char* ssid = "project";
const char* password = "project123";
void setup() {
  char ret;
  pinMode(buz, OUTPUT);
  pinMode(pir, INPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  digitalWrite(buz,HIGH);

  Serial.begin(9600);
  mySerial.begin(115200);//esp
  lcd.begin(16,2);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("WELCOME"); delay(2000);

 /*     st:
  mySerial.println("ATE0");
  ret = check((char*)"OK",50);
  mySerial.println("AT");
  ret = check((char*)"OK",50);
```

```
 if(ret != 0)
 {
   delay(1000);
  goto st;
 }

  lcd.clear();lcd.setCursor(0, 0);lcd.print("CONNECTING");
 mySerial.println("AT+CWMODE=1");
 ret = check((char*)"OK",50);
cagain:

 serialFlush();
 mySerial.print("AT+CWJAP=\"");
 mySerial.print(ssid);
 mySerial.print("\",\"");
 mySerial.print(password);
 mySerial.println("\"");
 if(check((char*)"OK",300))goto cagain;
 mySerial.println("AT+CIPMUX=1");
 delay(1000);
 lcd.clear();lcd.setCursor(0, 0);lcd.print("CONNECTED");delay(1000);*/

}
```

```
void loop()
{
  int pd = digitalRead(pir);
  int td = analogRead(temp)/10;
  td = td + 20;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
 // Serial.print("Distance: ");
 // Serial.print(distance);
// Serial.println(" cm");

  if(distance < 10 )
  {
     // digitalWrite(buz,LOW);
     lcd.clear();lcd.setCursor(0,1);lcd.print(" FULL");
     upload1(distance,td,pd,buf);delay(2000);
  }
  if(td > 35 )
  {
```

```
    // digitalWrite(buz,LOW);
    lcd.clear();lcd.setCursor(0,1);lcd.print(" HIGH TEMP");
    upload1(distance,td,pd,buf);delay(2000);
  }
  if(pd == HIGH )
  {
    // digitalWrite(buz,LOW);
    lcd.clear();lcd.setCursor(0,1);lcd.print(" PERSON DETECTED");
    upload1(distance,td,pd,buf);delay(2000);
  }

        delay(2000);

   if (Serial.available()> 0)
   {
    delay(10);
     while(!Serial.available());
     c = Serial.read();
     lcd.setCursor(0, 0);lcd.print("T:");lcd.print(td);
     lcd.setCursor(6, 0);lcd.print("P:");lcd.print(pd);
     lcd.setCursor(11, 0);lcd.print("D:");lcd.print(distance);

     lcd.setCursor(0, 1);lcd.print("c:");lcd.print(c);
     if(c == '*')
       {       while(!Serial.available());
```

```
c1    = Serial.read();
lcd.setCursor(5, 1);lcd.print("c1:");lcd.print(c1);
  if(c1 == '+'|| c1 == '-' )
{                      while(!Serial.available());
        buf[0] =        Serial.read();
                     while(!Serial.available());
        buf[1] =        Serial.read();
                     while(!Serial.available());
        buf[2] =        Serial.read();
                     while(!Serial.available());
        buf[3] =        Serial.read();
                     while(!Serial.available());
        buf[4] =        Serial.read();
        buf[5] = 0;
        value = ((buf[0] - '0')*10000)+((buf[1] - '0')*1000)+((buf[2] - '0')*100)+((buf[3] -
'0')*10)+((buf[4] - '0'));
        lcd.setCursor(10, 1);lcd.print("v:");lcd.print(value);
        if(value > 120)
        {
                if(value%10 == 0)
               {
                       strcpy(buf,"120_80");
                      lcd.clear(); lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("120_80");
                       upload1(distance,td,pd,buf);
                        delay(2000);
```

```
}
if(value%10 == 1)
{
    strcpy(buf,"125_70");
   lcd.clear(); lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("125_70");
     upload1(distance,td,pd,buf);
     delay(2000);
}
if(value%10 == 2)
{
    strcpy(buf,"110_80");
   lcd.clear(); lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("110_80");
     upload1(distance,td,pd,buf);
     ;delay(2000);
}
if(value%10 == 3)
{
    strcpy(buf,"115_82");
    lcd.clear();lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("115_82");
    upload1(distance,td,pd,buf);
     delay(2000);
}
if(value%10 == 4)
{
    strcpy(buf,"130_90");
```

```
        lcd.clear(); lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("130_90");
        upload1(distance,td,pd,buf);
         delay(2000);
}
if(value%10 == 5)
{
        strcpy(buf,"125_92");
        lcd.clear(); lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("125_92");
        upload1(distance,td,pd,buf);
         delay(2000);
}
if(value%10 == 6)
{
        strcpy(buf,"135_82");
        lcd.clear();lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("135_82");
        upload1(distance,td,pd,buf);
         delay(2000);
}
if(value%10 == 7)
{
        strcpy(buf,"127_87");
        lcd.clear();lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("127_87");
        upload1(distance,td,pd,buf);
         delay(2000);
}
```

```
if(value%10 == 8)
{
        strcpy(buf,"132_89");
       lcd.clear(); lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("132_89");
       upload1(distance,td,pd,buf);
         delay(2000);
}
if(value%10 == 9)
{
        strcpy(buf,"121_72");
       lcd.clear(); lcd.setCursor(0, 8);lcd.print("BP:");lcd.print("121_72");
        upload1(distance,td,pd,buf);
         delay(2000);
}

 if(distance < 10 )
    {
       // digitalWrite(buz,LOW);
       lcd.clear();lcd.setCursor(0,1);lcd.print(" FULL");
       upload1(distance,td,pd,buf);delay(2000);
    }
   if(td > 35 )
    {
       // digitalWrite(buz,LOW);
        lcd.clear();lcd.setCursor(0,1);lcd.print(" HIGH TEMP");
```

```
        upload1(distance,td,pd,buf);delay(2000);
      }
      if(pd == HIGH )
      {
        //   digitalWrite(buz,LOW);
        lcd.clear();lcd.setCursor(0,1);lcd.print(" PERSON DETECTED");
        upload1(distance,td,pd,buf);delay(2000);
      }

    }
   }

  }

}

}

char bf2[50];
void upload1(unsigned char *chr,unsigned char *chr1,unsigned char *chr2,unsigned char *chr3)
{
```

```
  lcd.clear();lcd.setCursor(0, 1);lcd.print("UPLOADING");
  delay(2000);
  serialFlush();
  mySerial.println("AT+CIPSTART=4,\"TCP\",\"embeddedspot.top\",80");


     delay(8000);
     sprintf(buff,"GET
http://embeddedspot.top/iot/storedata.php?name=iot306&s1=%u&s2=%u&s3=%u&s4=%s\r\n\r\
n",ch r,chr1,chr2,chr3);
     serialFlush();
     sprintf(bf2,"AT+CIPSEND=4,%u",strlen(buff));
     mySerial.println(bf2);


       delay(5000);



        serialFlush();
        mySerial.print(buff);


           delay(2000);


           mySerial.println("AT+CIPCLOSE");
      lcd.setCursor(0, 1);lcd.print("UPLOADDED");
}
```

CHAPTER-6
PROJECTIMPELEMENTATION

6.1 FLOW CHART

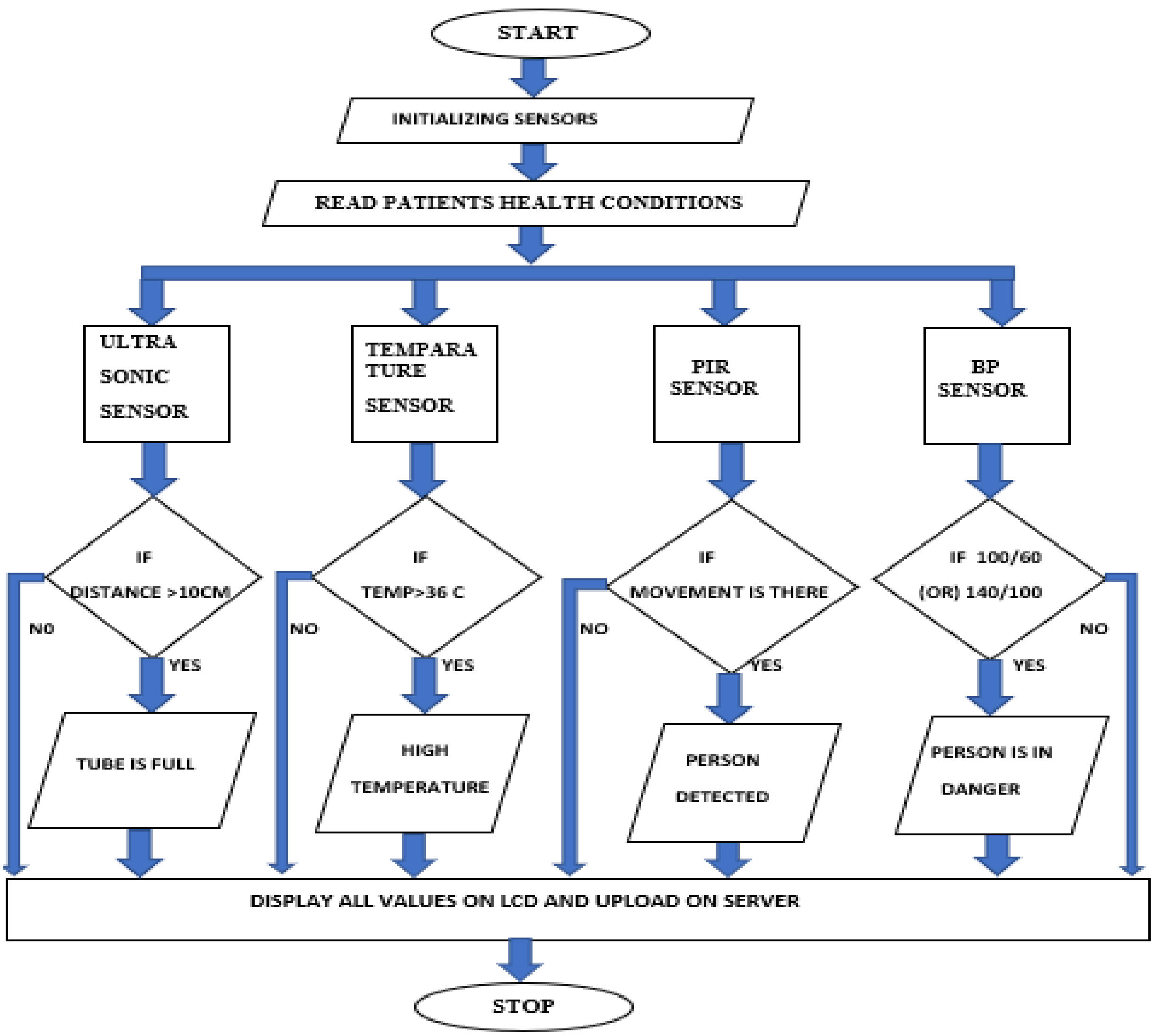


Fig 6.1 Flow Chart

6.2 IMPLEMENTATION STAGES

Stage 1:

When we switch on the power supply, the circuit will be activated and LCD displays as WELCOME.

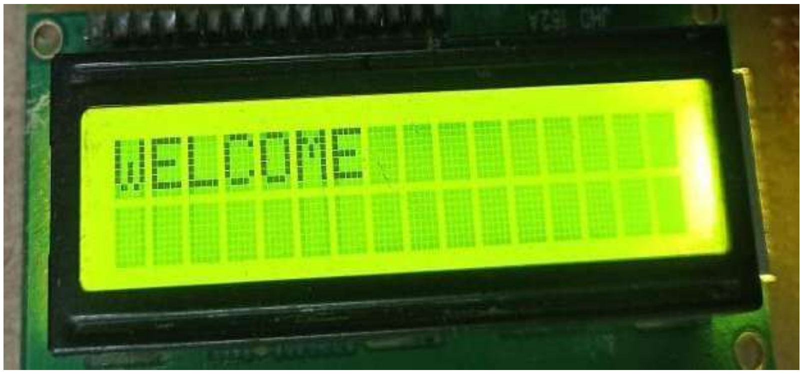


Fig 6.2.1 Starting of the operation

Stage 2:

To connect to wi-fi, we need to enter Required SSID name and its password in order establish internet connection to the system.

Fig 6.2.2.1 Status of the internet connection as connecting

After entering correct required SSID name and its password, the status of the internet connection will be displayed as CONNECTED on LCD.

Fig 6.2.2.2 Status of internet connection as connected

Stage 3:

In this stage, Sensors like temperature sensor, ultrasonic sensor, pir sensor and blood pressure sensor calculates the coma patient's vital health parameters and sends that information to Arduino and displays status on LCD.

Fig 6.2.3 Ultrasonic Sensor

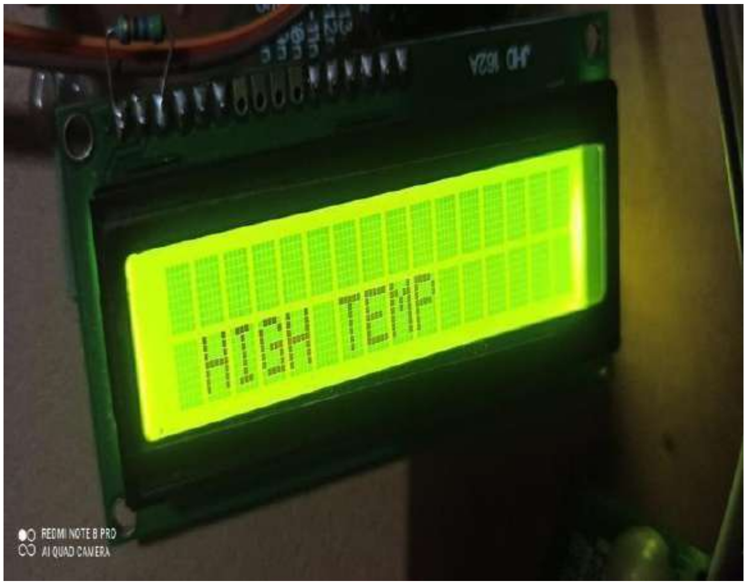


Fig 6.2.4 Temperature Sensor

Fig 6.2.5 PIR Sensor

Stage 4

After calculating all vital health parameters, these values are also uploaded into the server and that status is displays on LCD

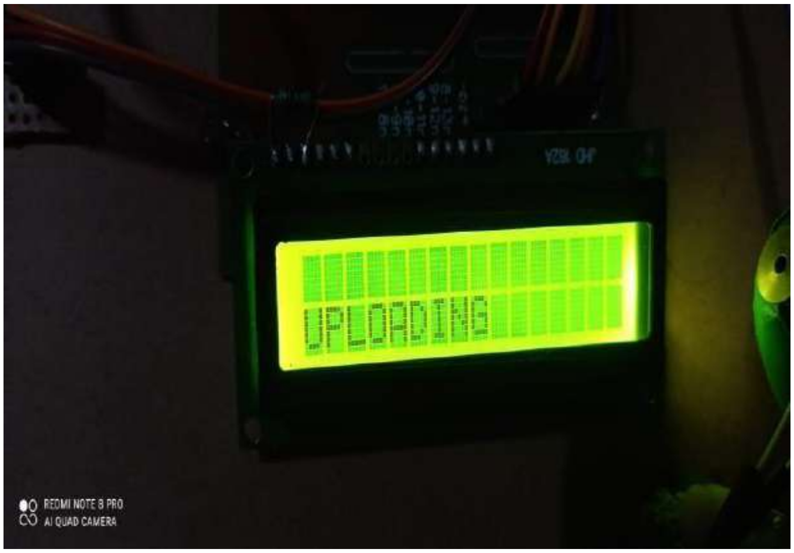


Fig6.2.6 Server Conformation

In this way, all the vital health parameter is calculated and also all calculated information is also uploaded in the things Speak Server. So that doctor can check and analyze patients health condition and can give any prescription to patient from anywhere in the world.

6.3 RESULT:

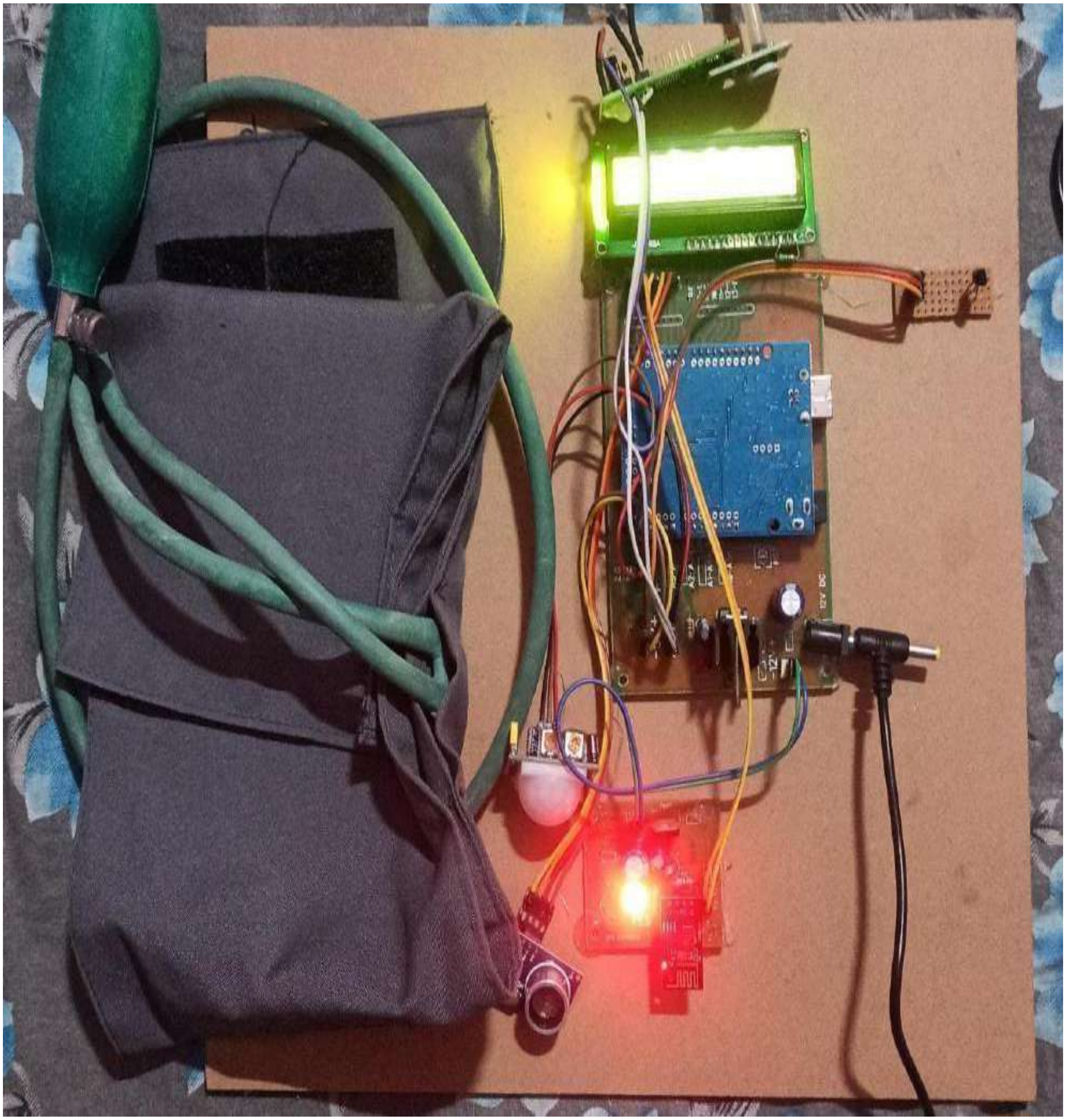


Fig 6.3 Final Output

S.No	S1	S2	S3	S4	Date
1	0	26	0		2021-05-25 13 56 36
2	5	28	0		2021-05-25 13 56 17
3	9	30	0		2021-05-23 20 14 32
4	16	32	0	120/80	2021-05-23 20 11 49
5	23	34	1		2021-05-23 20 11 28
6	29	36	0		2021-05-23 20 11 07
7	37	31	0		2021-05-23 18 47 45
8	42	29	0		2021-05-23 18 47 24
9	31	35	0		2021-05-23 18 47 03
10	26	30	0	120/80	2021-05-19 21 17 44
11	19	27	1		2021-05-19 21 17 23

Fig. 6.3.1 Tabular view of Health Parameters

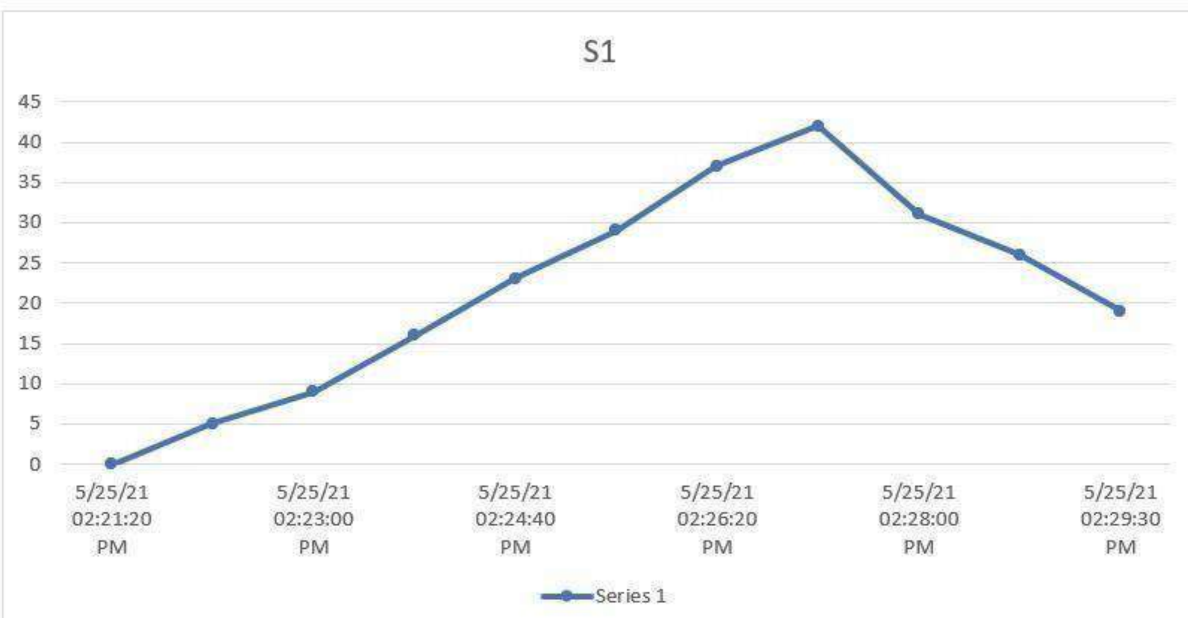


Fig 6.3.2 Ultrasonic sensor graph

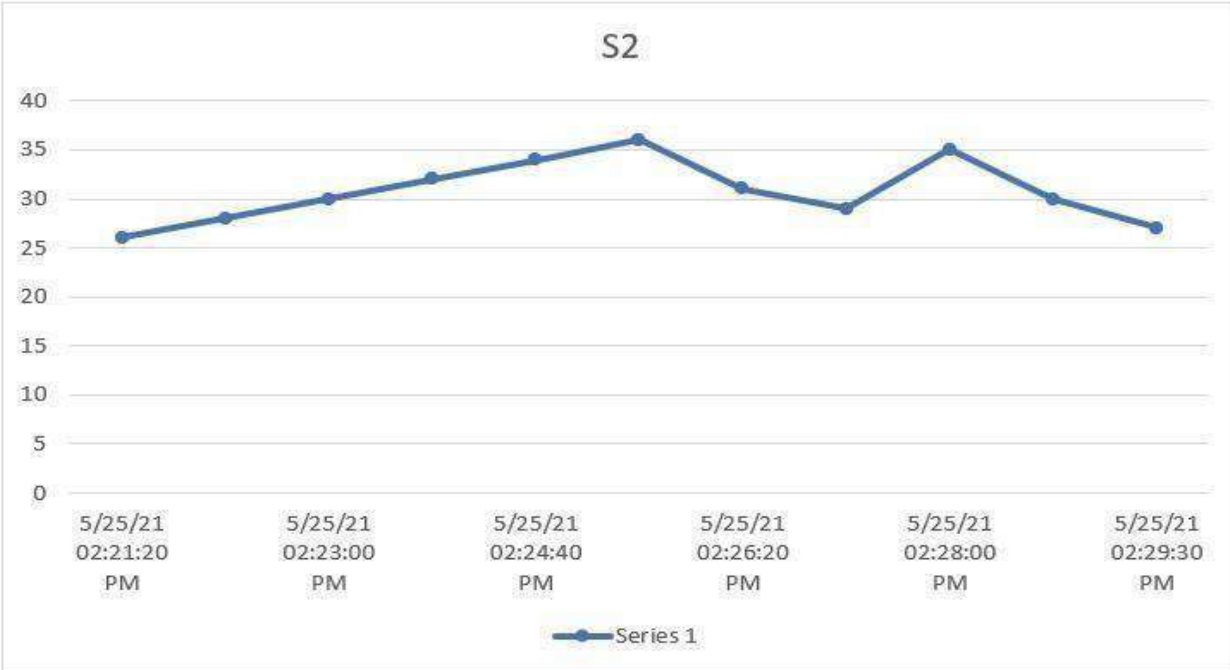


Fig 6.3.3 Temperature sensor graph

Fig 6.3.4 PIR sensor graph

CHAPTER-7
APPLICATIONS AND ADVANTAGES

7.1 APPLICATION OF THE PROJECT:

- IoT has applications in healthcare that benefit patients, families, physicians, hospitals and insurance companies. IoT for Patients - Devices in the form of wearables like fitness bands and other wirelessly connected devices like blood pressure and heart rate monitoring cuffs, glucometer etc.
- IoT is undoubtedly transforming the healthcare industry by redefining the space of devices and people interaction in delivering healthcare solutions. IoT has applications in healthcare that benefit patients, families, physicians, hospitals and insurance companies.
- **IoT for Patients** - Devices in the form of wearables like fitness bands and other wirelessly connected devices like blood pressure and heart rate monitoring cuffs, glucometer etc. give patients access to personalized attention. These devices can be tuned to remind calorie count, exercise check, appointments, blood pressure variations and much more.
- IoT has changed people's lives, especially elderly patients, by enabling constant tracking of health conditions. This has a major impact on people living alone and their families. On any disturbance or changes in the routine activities of a person, alert mechanism sends signals to family members and concerned health providers.
- **IoT for Physicians -** By using wearables and other home monitoring equipment embedded with IoT, physicians can keep track of patients' health more effectively. They can track patients' adherence to treatment plans or any need for immediate medical attention. IoT enables healthcare professionals to be more watchful and connect with the patients proactively. Data collected from IoT devices can help physicians identify the best treatment process for patients and reach the expected outcomes.
- **IoT for Hospitals -** Apart from monitoring patients' health, there are many other areas where IoT devices are very useful in hospitals. IoT devices tagged with sensors are used for tracking real time location of medical equipment like wheelchairs, defibrillators, nebulizers,

Oxygen Pumps and other monitoring equipment. Deployment of medical staff at different locations can also be analyzed real time.

- The spread of infections is a major concern for patients in hospitals. IoT-enabled hygiene monitoring devices help in preventing patients from getting infected. IoT devices also help in asset management like pharmacy inventory control, and environmental monitoring, for instance, checking refrigerator temperature, and humidity and temperature control.

7. 2 ADVANTAGES OF THE PROJECT:

- This system is less costly more reliable and efficient as compared to other systems.
- 24 * 7 monitoring of coma patient.
- Continuous recording of patient data after particular time period.
- Power consumption less.
- No need of human attention for continuous observation of coma patient.
- High speed communication between doctor and patient through Things Speak cloud.

CHAPTER-8
CONCLUSION AND FUTURE ENCHANCEMENT

8.1 CONCLUSION:

The Internet of Things is considered now as one of the feasible solutions for any remote value tracking especially in the field of health monitoring. It facilitates that the individual prosperity parameter data is secured inside the cloud, stays in the hospital are reduced for conventional routine examinations and most important that the health can be monitored and disease diagnosed by any doctor at any distance.

8.2 FUTURE SCOPE:

- For further implementation, in emergency situations, this system can automatically send a warning message or call to alert the nearest hospital as well as to the ambulance if any abnormal data is identified in monitoring.
- The advanced development for the designed model is to add more parameters for monitoring the health status of patient.
- In future improvement to this designed model is to include MEDIBOX. This system can be used by the paralyzed patients by sending or giving reminder alert to take their medicines or dosage on time.
- Another extension to this system is to add web camera, after that anyone can monitor patient world-wide at any time.

PUBLICATION

Submitted Paper in the Conference ICSMEC – 21 with Paper ID (ICSMEC21-0081) and got Acceptance of paper.

REFERENCES

1. Dr R. Josphine Leela M.E, P K. Hamsageetha, P. Monisha, S. Yuvarani, "Body Movement and Heart Beat Monitoring For Coma Patient Using IoT", International Journal of Innovative Research in Science, Engineering and Technology, Vol. 7, Special Issue 2, March2018.

2. Geethanjali R., Majidha Fathima K. M., Harini S., Sabitha M., "Health monitoring for coma patients", International Journal of Emerging Research & Development, Volume 2, Issue 3,2019.

3. S. Sandeep, Dr. P. Esther Rani, G. Sumalatha, "Monitoring of Health Parameters by Using Raspberry Pi", International Journals of Advanced Research in Computer Science and Software Engineering, (Volume-8, Issue-4),2018.

4. Sneha Chowdary Koganti, Dr. H N Suma, Appaji M. Abhishek "Analysis and Monitoring of Coma Patients using Wearable Motion Sensor System", International Journal of Science and Research (IJSR), Volume 4 Issue 9, September2015.

5. Emna Mezghani, Ernesto Exposito, and Khalil Drira, "A Model Driven Methodology for the Design of Autonomic and Cognitive IoT-Based Systems: Application to Healthcare", IEEE transactions on emerging topics in computational Intelligence, VOL. 1, NO. 3, JUNE2017.

6. K. Shankar and P. Eswaran. "An Efficient Image Encryption Technique Based on Optimized Key Generation in ECC Using Genetic Algorithm", Advances in Intelligent Systems and Computing, Springer, Volume: 394, Page(s): 705-714, 2016.

7. B. S. Murugan, Mohamed Elhoseny, K. Shankar, J. Uthayakumar, "Region-based scalable smart system for anomaly detection in pedestrian walkways", Computers and Electrical Engineering, Volume. 75, Pages 146–160, May 2019.

8. K. Shankar and P. Eswaran. “RGB Based Multiple Share Creation in Visual Cryptography with Aid of Elliptic Curve Cryptography”, China Communications, Volume. 14, Issue. 2, page(s): 118-130, February 2017.

9. Krishna C S, “Health care monitoring system based on IoT”, second IEEE International Conferences on computational system and information technology sustainable solution Apr2017

10. Shivleela Patil, Dr S anjay Pardeshi “Health monitoring system using IoT” International, Research journal of Engineering and Technology, volume:05 Issue:04, Apr-2018.

11. Zankhana Mehul Kalarthi, “Smart Health Care System Using Internet of Things”, IJRET: International Journal of Research in Engineering and Technology eISSN:2319- 1163|pISSN: 2321-7308 Apr-2017.

12. R Nikhil Nair, K. A. Kiran “secure system for human health monitoring” International Conference on Intelligent Computing and control systems, Apr2017.

13. Kansal, N., & Dhillon, H. S. (2011). Advanced Coma Patient Monitoring System. International Journal of Scientific & Engineering Research, 2(6).

14. Konganti, S. C., Suma, H. N., & Abhishek, A. M. (2015). Analysis and monitoring of coma patients using wearable motion sensor system. Int. J. Sci. Res, 4(9), 1154 1158.

15. Agrawal, D., Singh, P. K., Sinha, S., Gupta, D. K., Satyarthee, G. D., & Misra, M. C. (2015). Remaining unconscious: The burden of traumatic brain injuries in India. Journal of neurosciences in rural practice, 6(04), 520522

16. A. M. Nia, M. Mozaffari-Kermani, S. SurKolay, A. Raghunathan, and N. K. Jha, “Energy efficient long-term continuous personal health monitoring,” IEEE Trans. Multi-Scale Computing Systems, vol. 1, no. 2, pp. 85–98, 2015.

17. H. Ghayvat, J. Liu, S. C. Mukhopadhyay, and X. Gui, “Wellness sensor networks: A proposal and implementation for smart home for assisted living,” IEEE Sensors Journal, vol. 15, no. 12, pp. 7341–7348, 2015.

18. S. C. Mukhopadhyay, "Wearable sensors for human activity monitoring: A review," IEEE Sensors Journal, vol. 15, no. 3, pp. 1321–1330, 2015.

19. "Olympic medical institute validates Polar RS800 running computer and training system," https://www.polar.com/usen/about polar/news/polar R800, accessed: 10-9-2016.

20. O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," IEEE Communications Surveys & Tutorials, vol. 15, no. 3, pp. 1192–1209, 2013.

21. W. Gao, S. Emaminejad, H. Y. Y. Nyein, S. Challa, K. Chen, A. Peck, H. M. Fahad, H. Ota, H. Shiraki, and D. Kiriya, "Fully integrated wearable sensor arrays for multiplexed in situ perspiration analysis," Nature, vol. 529, no. 7587, pp. 509–514, 2016.

22. A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," IEEE Trans. Systems, Man, and Cybernetics, vol. 40, no. 1, pp. 1–12, 2010.

23. Lydia, E. L., Shankar, K., Ilayaraja, M., & Kumar, K. S. Technological Solutions for Health Care Protection and Services Through Internet Of Things (IoT). International Journal of Pure and Applied Mathematics,118(7),277-283.

24. Aleksander Kotevski, Natasa Koceska and Saso Koceski, "E-health Monitoring System", International Conference on Applied Internet and Information Technologies, 2016.

25. S. Sampeter and S. Padmavath, "An improved health monitoring system for coma patients using internet of things", (IJETIE), vol. 5, pp. 408-412, 2019.

BIBILOGRAPHY

1. https://www.researchgate.net/publication/338246586_Developing_IoT_Based_Smart_Health_Monitoring_Systems_A_Review

2. https://www.researchgate.net/publication/338246586_Developing_IoT_Based_Smart_Health_Monitoring_Systems_A_Review

3. https://www.academia.edu/35340974/An_IoT_based_Patient_Health_Monitoring_System_using_Arduino_Uno

4. https://www.ijert.org/an-iot-based-health-monitoring-system-using-arduino-uno

5. http://cloudstechnologies.in/cloudtechadmin/basepaperfiles/15952991258.%20iot%20based%20 health%20monitoring%20system.pdf

6. http://www.journaleca.com/gallery/jeca-2138.35-f.pdf

7. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3606060

8. https://ieeexplore.ieee.org/document/9074174

9. https://core.ac.uk/download/pdf/132492107.pdf

10. http://thingspeak.com

A

PROJECT REPORT

On

DESIGN OF A LASER COMMUNICATION LINK BETWEEN SATELLITES IN A CONSTELLATION

Submitted by

1) Mr. Kushal Pooppal (17K81A04L8)

2) Mr. Amith Reddy Mosali (17K81A04N0)

3) Mr. Nikhil Mishra (17K81A04N4)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Esteemed Guidance of

Ms.Pushpa P

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATIONS

ST.MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500100

NBA & NAAC A+ Accredited

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE
UGC AUTONOMOUS

Dhulapally, Secunderabad-500 100
NBA & NAAC A+ Accredited

BONAFIDE CERTIFICATE

This is to certify that the project entitled "**DESIGN OF A LASER COMMUNICATION LINK BETWEEN SATELLITES IN A CONSTELLATION**", is being submitted by **1.Mr. Kushal Pooppal(17K81A04L8) , 2.Mr. Amith Reddy Mosali (17K81A04N0), 3.Mr. Nikhil Mishra (17K81A04N4**) in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATIONS ENGINEERING** is recorded of bonafide work carried out by them. The results embodied in this report have been verified and found satisfactory.

< Signature>
Ms.Puspa P
Department of ECE

Head of the Department
Dr. B. HARI KRISHNA
Department of ECE

Internal Examiner

External Examiner

Place: Hyderabad

Date: 28 June 2021

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **KUSHAL POOPPAL** WITH ROLL NO.**17K81A04L8**, **MOSALI AMITH REDDY** WITH ROLL NO.**17K81A04N0**, **NIKHIL MISHRA** WITH ROLL NO.**17K81A04N4**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING** DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM **AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED **"DESIGN OF A LASER COMMUNICATION LINK BETWEEN SATELLITES IN CONSTELLATION"** AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEAVORS.

Lasya IT Solutions Pvt. Ltd. ★Kompally-500014.★

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014

Email : contact@lasyainfotech.com, ov@lasyainfotech.com

Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/88

DECLARATION

We, the student of Bachelor of Technology in Department of 'Electronics and Communications Engineering', session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled "**DESIGN OF A LASER COMMUNICATION LINK BETWEEN SATELLITES IN A CONSTELLATION**" is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

Kushal Pooppal (17K81A04L8)
Amith Reddy Mosali (17K81A04N0)
Nikhil Mishra (17K81A04N4)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna**, **Head &Professor**, the Dept. of Electronics and Communications Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our Dept. project coordinator Dr.A.Anand, Professor** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide **Ms. Pushpa P**, **Assistant Professor,** Department of ECE and our class coordinator **Ms. Pushpa P, Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level Project Coordinator **Project Coordinators Dr. Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. Kushal Pooppal
2. Amith Reddy Mosali
3. Nikhil Mishra

ABSTRACT

Satellites are the most important nodes in the global communication infrastructure on and off-earth. They have made communication in today's world more versatile. satellites helped in connecting stations and people from different parts of the globe. Despite it's popular applications, it's rarely available in rural areas. This can be solved by the use of Satellite Constellations which form a grid of networks in space, A satellite constellation is a group of artificial satellites working together as a system. Unlike a single satellite, a constellation can provide permanent global or near-global coverage. To facilitate high-bandwidth data transfer between satellites a novel Laser Communication Link needs to be designed that automatically acquires and tracks the signal to the next satellite ,thus allowing the data to be transferred within the constellation and then to the land to receive the data at high speeds. A laser communication link combined with satellite constellation grid will truly provide global high-speed internet access with future applications in LunarNet

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF GRAPHS

GLOSSARY OF COMMONLY USED TERMS:

Satellite - An artificial body placed in orbit around the earth or moon or another planet in order to collect information or for communication.

Satellite constellation - A satellite constellation is a group of artificial satellites working together as a system. Unlike a single satellite, a constellation can provide permanent global or near-global coverage, such that at any time everywhere on Earth at least one satellite is visible.

Laser - A device that generates an intense beam of coherent monochromatic light (or other electromagnetic radiation) by stimulated emission of photons from excited atoms or molecules.

Acquisition - The process or technique of acquiring the required object within a particular range.

Tracking - The Process of following the acquired target for a particular period of time for a particular purpose.

LDR - It is a passive component that decreases resistance with respect to receiving luminosity on the component's sensitive surface.

Morse Code - An alphabet or code in which letters are represented by combinations of long and short light or sound signals

PWM - Pulse-width modulation, or pulse-duration modulation, is a method of reducing the average power delivered by an electrical signal, by effectively chopping it up into discrete parts, it is a technique for getting analog results with digital means.

Throughput - is the actual amount of data that is successfully sent/received over the communication link.

BPS - A measure of data transmission speeds, the amount of bits transferred in a single second.(Bits per second).

WPS - A measure of data transmission speeds, the amount of words transferred in a single second.(Words per second).

TX - Abbreviation for Transmitter

RX - Abbreviation for Receiver

I2C - I2C is a serial communication protocol, so data is transferred bit by bit along a single wire (the SDA line) and the other is SCL

LOS - Abbreviation for Line of Sight.

CHAPTER-1
INTRODUCTION

1.1 OVERVIEW OF THE PROJECT:

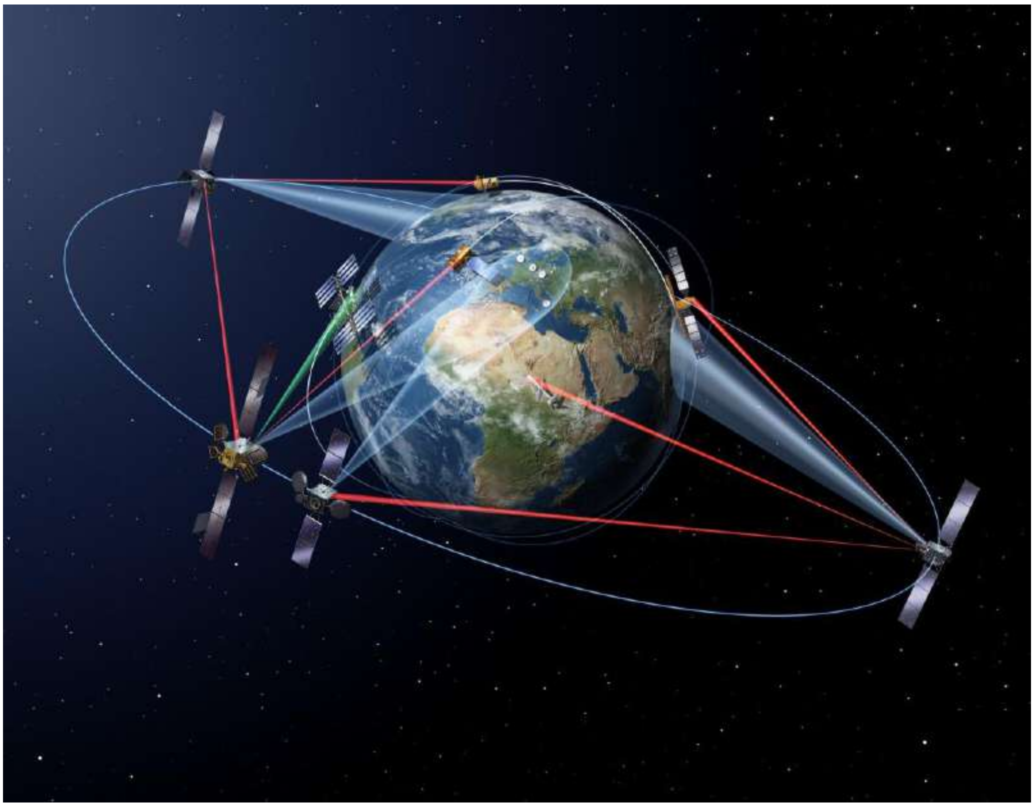


Fig 1.1: Global Space Segment of Communication

The demand for data—raw facts—and its successful evolution to information—arranged in context—is growing, arguably at an exponential rate. This is true for the civil, commercial, and security sectors (i.e., defense; specifically, the U.S. Department of Defense). Civil space activities include those to explore space and advance human understanding; commercial activities are those where private companies and industries provide services with the intent of making a profit [1]. During 1991's Operation DESERT STORM in Iraq, the U.S. Department of Defense had access to ninety-nine mega-bits per second of bandwidth. In 2003, access for Operations ENDURING FREEDOM in Afghanistan and IRAQI FREEDOM had grown to thirty-two-hundred mega-bits per second. In 2007, U.S. Department of Defense planners projected the need for sixteen giga-bits per second by 2010 [2]. A rudimentary analysis concludes that the U.S. Department of Defense requires an order of magnitude increase in bps every nine years. The U.S. Department of Defense's demand for data, information, and intelligence is exceeding its satellite communications capacity, whether owner-operator or contracted. Transitioning to laser-based satellite communications, in part due to its orders of magnitude greater throughput, is becoming necessary for the U.S. Department of Defense. Laser communications could provide from ten to one-hundred times better data rates than radio, due to the higher bandwidth [3]. This manuscript examines the technological maturation of employing lasers as the signal carrier for satellite communications linking terrestrial and space systems. The purpose of the manuscript is to inform the U.S. Department of Defense initial capabilities documents (ICDs) or key performance parameters (KPPs) for near-future satellite acquisition and development. By appreciating the history and technological challenges of employing lasers, rather than traditional radio frequency sources for satellite uplink and downlink signal carriers, this manuscript recommends ways for the U.S. Department of Defense to employ lasers to transmit and receive high bandwidth and large-throughput data from moving platforms that need to retain low probabilities of detection, intercept, and exploit (e.g., carrier battle group transiting to a hostile area of operations, unmanned aerial vehicle collecting over adversary areas). The manuscript also intends to identify commercial sector early-adopter fields and those fields likely to adapt to laser employment for transmission and receipt.

1.2 OBJECTIVES OF THE STUDY:

1. **Compare** existing system with Laser Comms
2. **Develop**: Signal Acquisition and Tracking for TX-RX
3. **Prototype**: Demonstrate LC between Sats using COTS
4. **Design**: An Implementation for Laser Comms in Satellite Constellation

1.3 SCOPE OF THE STUDY:

Satellites have made communication in today's world more versatile. A satellite constellation is a group of artificial satellites working together as a system. Unlike a single satellite, a constellation can provide permanent global or near-global coverage.

The present inter-sat communications is being carried out by RF(Radio Frequency) signals, which have its drawbacks in aspects like security, bandwidth, throughput, etc compared to laser communication.

To facilitate high-bandwidth data transfer between satellites a Laser Communication Link needs to be designed that automatically acquires and tracks the signal to the next satellite ,thus allowing the data to be transferred within the constellation and then to the land to receive the data at high speeds. A laser communication link combined with satellite constellation grid will truly provide global high-speed internet access with future applications in LunarNet.

The areas where the project focuses is in designing and developing techniques like Acquisition and tracking to successfully transmit data through laser using morse code and the chosen modulation techniques. These methods have to be followed to keep the satellites in Line of Sight throughout the period of communication for uninterrupted data transfer.

This project when implemented in the real world using viable equipment and technology can drastically improve the communication from the space segment.

1.4 MATERIAL REQUIREMENT:

1.4.1 HARDWARE REQUIREMENT:

- Arduino UNO - 2
- LASER Diode 650nm 5mw - 1
- Servos SG 90 - 2
- LDRs or Photoresistor - 3
- Perfboard
- Breadboard - 2
- Jumper Wires
- USB Type A Cable - 2

1.4.2 SOFTWARE REQUIREMENT:

- Proteus software
- Arduino IDE
- MATLAB
- Visual Studio Code

1.4 PROCUREMENT OF EQUIPMENT:

All the equipment used in the project are Commercially available off the shelf parts(COTS) procured from tech stores.

The assembling is done on our own expertise and connections are made with great care following all safety precautions

The coding for the project, that is the code to be dumped in arduino, is developed using Arduino IDE and integrated into the project.

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH AREA:

This section begins with a quote succinctly introducing the history of laser employment, followed by chronological articulation before and through the twenty-first century. Following brief chronology, brief paragraphs introduce historical applications, finalizing the preparation for discussion of the results.

HISTORY OF LASER

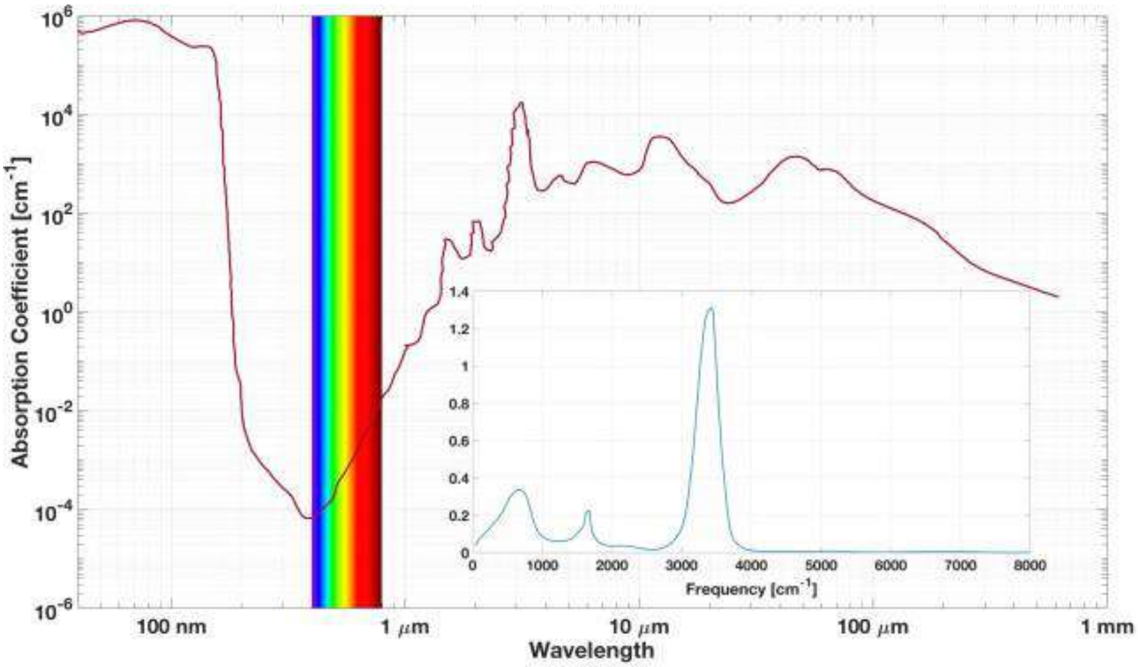


Fig 2.1: Attenuation of optical communication

Employment as Link Source Relevant laser applications have a long, interesting pedigree that harks back to the beginning of the space era. “The registered history of laser technologies for space application starts with the first laser echoes reflected off the Moon in 1962. Since then, photonic technologies have become very prominent in most technical development. Their presence has also dramatically increased in space applications thanks to the many advantages they present over traditional equivalent devices, such as the immunity against electromagnetic interference, as well as their efficiency and low power consumption. Lasers are one of the key components in most of those applications.” [4,5] The application of lasers for communications across great distances has an even longer pedigree, although the notions were not actualized until the last century.

Prior to the Twenty-First Century Shortly after WWII, Arthur C. Clarke wrote about employing light beams to transmit information [6]. His foresight was not realized until the 1960s with the development of the laser. Furthermore, it was not demonstrated as intended until the 1995 bidirectional ground-to-orbit laser communication demonstrations. The program successfully used lasers to uplink and downlink to the engineering test satellite-VI in elliptical geosynchronous transfer orbit, sending and receiving at a 1 mega-bits per second rate [7]. This unhurried pace (50 years between idea and demonstration: 30 years between technology development and adaptation) exemplifies the field of laser communications. Before considering lasers for uplinks and downlinks [8], lasers were employed for geodesy first [9–22] and also orbital determination [23–33]. The 1990s jump-started scientific examination of space and satellite laser communications [34–53]. Over roughly the next decade, the U.S. national air and space administration, and the European and Japanese space agencies, completed successful trials of space-based laser communications [34]. In 2001, the European space agency's semiconductor intersatellite link experiment and advanced relay and technology mission satellite received fifty mega-bits per second from hich is depicted in Figure 1. That same year, the geosynchronous lightweight technology experiment also successfully demonstrated bi-directional laser communications, and this time added an aircraft to the reception station list [35]. 2.1.2. Early Twenty-First Century Figure 1. satellite uplink and downlink with terrestrial stations from [8]: 0.6 m transmitter uplinking at 514.5 nanometers and 1.2 m receiver downlinking at 830 nanometers. The advanced relay and technology mission satellite laser communications capability was supplied by the optical payload laser experiment and the laser-utilizing communications equipment. Both systems provided source laser beams, at ten milli-watts and forty milli-watts, respectively. However, a lesson learned from the semiconductor intersatellite link experiment was that even with the laser-utilizing communications equipment beam's quadruple power, it failed to provide sufficient irradiance at ground-to-geosynchronous orbit ranges to perform initial acquisition [37]. However, the semiconductor intersatellite link experiment demonstrated a downlink bit error rate of 10-10 [37]. The semiconductor intersatellite link experiment did however reinforce a foundational advantage of laser communications over radio frequency communications: that laser communications requires less size, weight, and power for a given data rate [36]. Demonstrated years later, a two—point-five giga-bits per second communications link with a roughly three decibel link margin across a range of forty-two thousand kilometers, a capability that can be made using a two point-two meter antenna for Ka-band (weighing one-hundred fifty-three kilograms), a one-point nine meter antenna for a millimeter band (weighing one-hundred thirty-two kilograms), and a ten centimeters optical system for a one-point five-five micrometers wavelength (weighing sixty-five kilograms) [37]. In 2013, the U.S. national air and space administration began examining laser communications in earnest, focused in support of further exploration of Mars and beyond. That year the administration launched the lunar laser communications demonstrator [42,43], a high-rate demonstration of space laser communications [42]. “The Lunar Laser Communication Demonstration (LLCD) was conducted on NASA’s Lunar Atmosphere and Dust Environment Explorer (LADEE) satellite that launched in late 2013 [41]. The LLCD payload Figure 1. Satellite uplink and downlink with terrestrial stations from [8]: 0.6 m transmitter uplinking at 514.5 nanometers and 1.2 m receiver downlinking at 830 nanometers. 2.1.2. Early Twenty-First Century The advanced relay and technology mission satellite laser communications capability was supplied by the optical payload laser experiment and the laser-utilizing communications equipment. Both systems provided source laser beams, at ten milli-watts and forty milliwatts, respectively. However, a lesson learned from the semiconductor intersatellite link experiment was that even with the laser-utilizing communications equipment beam’s quadruple power, it failed to provide sufficient irradiance at ground-to-geosynchronous orbit ranges to perform initial acquisition [37]. However, the semiconductor intersatellite link experiment demonstrated a downlink bit error rate of 10−10 [37]. The semiconductor intersatellite link experiment did, however, reinforce a foundational advantage of laser communications over radio frequency communications: that laser communications requires less size, weight, and power for a given data rate [36]. Demonstrated years later was a 2.5 giga-bits per second communications link with a roughly three decibel link margin across a range of forty-two-thousand kilometers, a capability that can be made using a two-point-two meter antenna for Ka-band (weighing one-hundred fifty-three kilograms), a one-point nine meter antenna for a millimeter band (weighing one-hundred thirty-two kilograms), and a ten centimeter optical system for a 1.55 micrometer wavelength (weighing sixty-five kilograms) [37]. In 2013, the U.S. national air and space administration began examining laser communications in earnest, focused in support of the further exploration of Mars and beyond. That year, the administration launched the lunar laser communications demonstrator [42,43], a high-rate demonstration of space laser communications [42].

2.2 REVIEW ON RELATED LITERATURE:

Localization of underwater sensors is an important part of UOWNs for many applications such as resource exploration, surveillance, underwater environment monitoring, and disaster prevention. The large propagation delay of acoustic channels and high attenuation of RF/optical channels pose significant challenges for underwater localization. The major challenges for underwater localization are

Deployment of nodes: Most of the localization algorithms depend on the distribution of sensor nodes and the anchor nodes to form a network [44], [336]. • Mobility of the nodes: Due to the uncontrollable phenomena such as winds, turbulence, and current, the underwater sensor nodes inevitably drift from its position. The location of anchor nodes on the surface buoys can be accurately measured by using GPS but the location of the underwater nodes cannot be precisely measured.

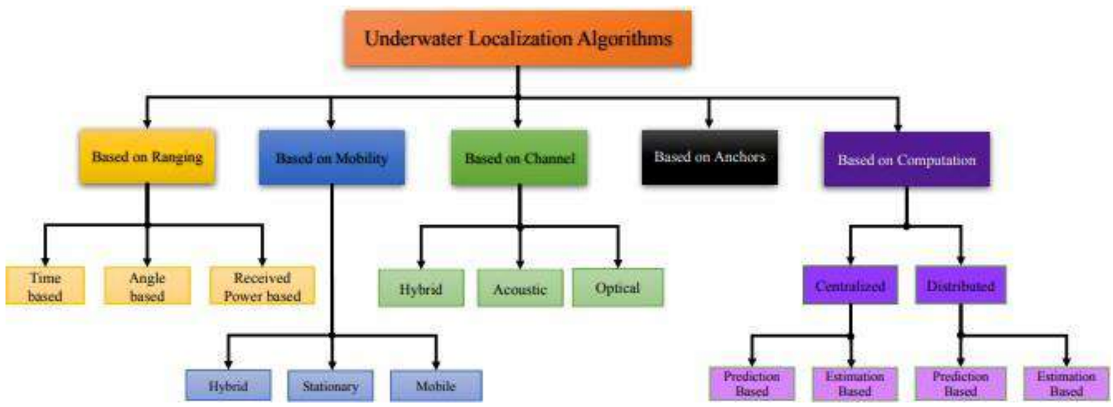


Fig 2.2: Taxonomy of underwater localization algorithms

Variations in the underwater wireless communication channel are very severe for all types of carrier waves. The effects of attenuation, absorption, reflection, scattering, and noise do not allow for accurate range measurements, thus reflecting large localization estimation error [45]. Synchronization: As the GPS signals are not available in the underwater environment, it is hard to achieve the time synchronization between the sensor nodes. Thus, if the time of arrival based ranging is used, this miss-synchronization will lead to large localization errors.

2.3 CONCLUSION ON REVIEWS:

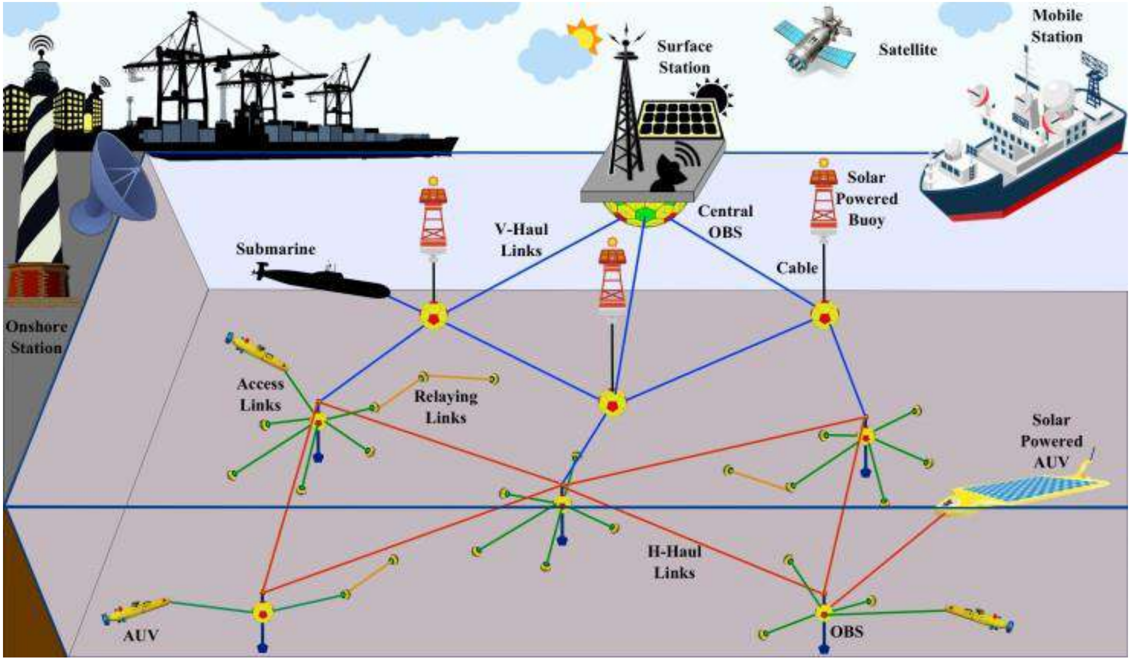


Fig 2.3 Illustration of underwater submarine optical comms

In this paper, we have presented a comprehensive survey of underwater optical wireless networks (UOWNs) research. This survey covers different aspects of cutting-edge UOWNs from a layer by layer perspective. Firstly, each layer of UOWNs such as physical, data link, networking, transport, and application layers are briefly presented and then localization techniques for UOWNs are surveyed. We started with defining different possible architectures for UOWNs and then the issues related to each layer are thoroughly discussed. Besides providing the technical background on UOWNs, we have also provided details on the challenges to design a practical UOWN. Additionally, localization is an important task where the location of the underwater optical sensor node can be used for node tracking, intruder detection, and data tagging. To reach this goal, communication, networking, and localization in UOWNs require more research efforts. In short, this survey can help the novice readers to get an insight of each layer and localization of UOWNs which can lead to the development of practical UOWNs.

CHAPTER-3

PROJECT DESIGN

3.1 OVERVIEW OF THE DESIGN:

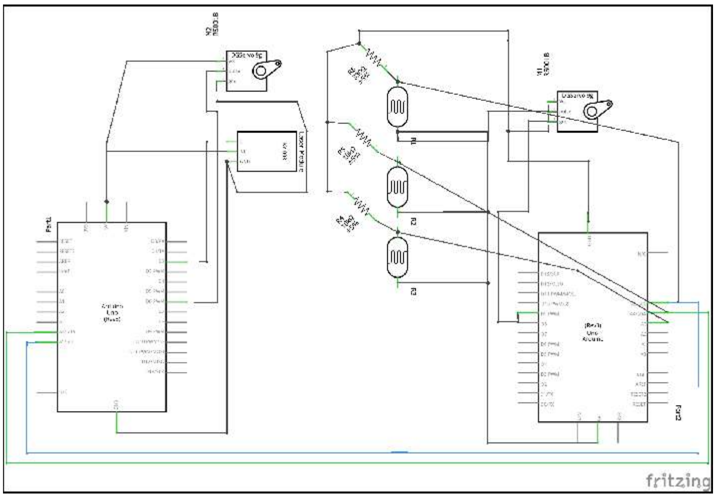


Fig 3.1 Schematic of the project

3.2 EQUIPMENT ANALYSIS:

3.2.1 ARDUINO:

3.2.1.1 INTRODUCTION

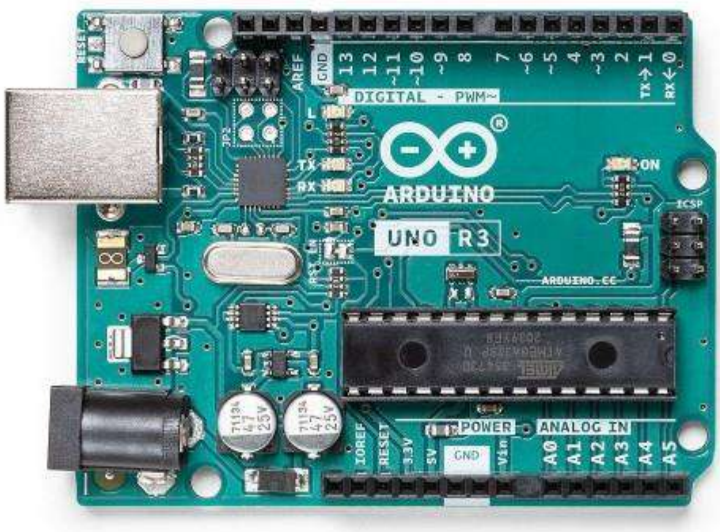


Fig 3.2 Arduino UNO Board Top View

Arduino is a prototype platform (open-source) based on easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connecting to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

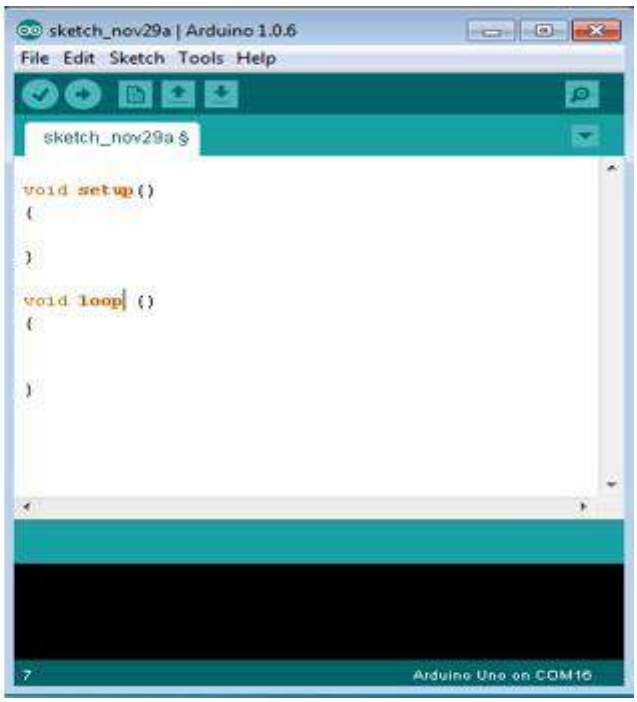


Fig 3.3 Arduino IDE View

3.2.1.2 BOARD TYPES

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

Fig 3.4 Different Arduino Board Types

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Arduino boards based on ATMEGA328 microcontroller

Board Name	Op Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header

Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Arduino boards based on AT91SAM3X8E microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have the majority of these components in common.

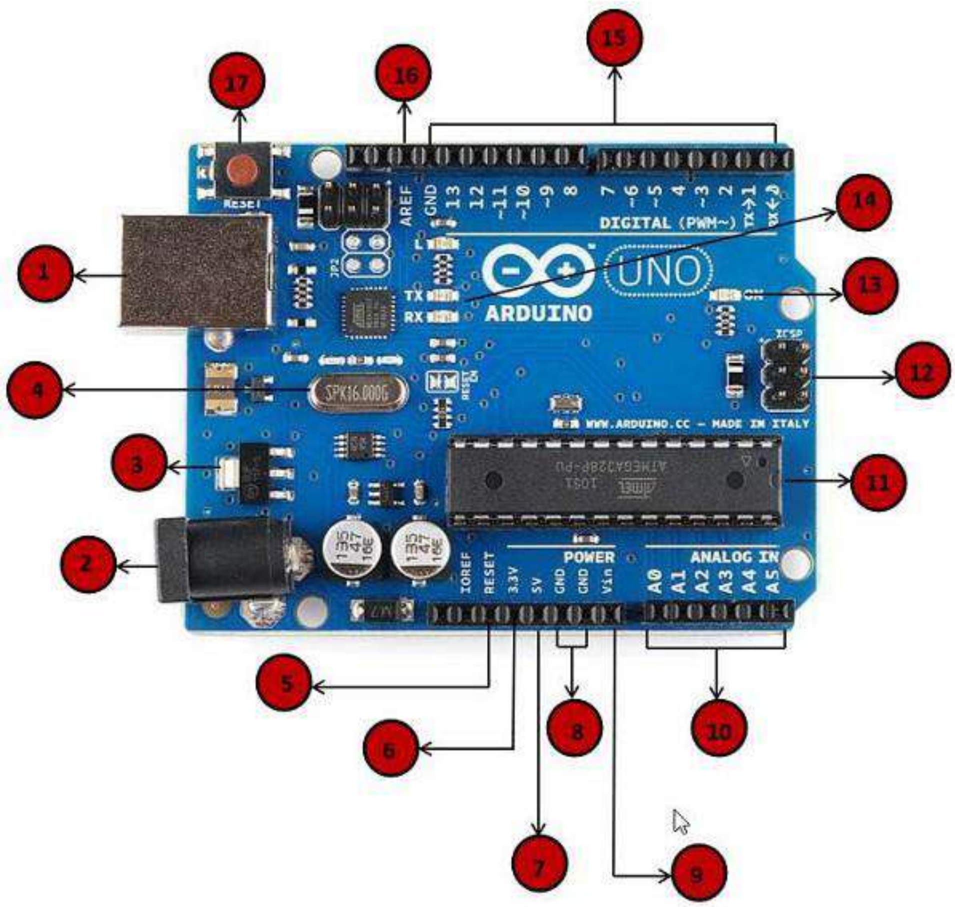


Fig 3.5 Arduino Pin Description

Table 4: Arduino Board Description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the

	reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.

<table>
<tr><td>12</td><td>ICSP pin

Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.</td></tr>
<tr><td>13</td><td>Power LED indicator

This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.</td></tr>
<tr><td>14</td><td>TX and RX LEDs

On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.</td></tr>
<tr><td>15</td><td>Digital I/O

The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.</td></tr>
</table>

16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

3.2.1.4 ARDUINO FAMILY

Arduino makes several different boards, each with different capabilities. In addition, part of being open-source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

Fig.3.5: Arduino Family

3.2.1.5 SHIELDS:

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

Fig.3.6: Arduino Shields

3.2.1.6 PIN DESCRIPTION OF ATMEGA328:

Fig.3.7 Pin description of ATMEGA328

3.2.1.7 APPLICATIONS:

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

3.2.2 LDR

A photo resistor is an electronic component whose resistance decreases with increasing incident light intensity. It can also be called a light-dependent resistor (LDR), or photoconductor.

Other light dependent resistors, or photo resistors have been made using materials including Cadmium Sulphide, Lead Sulphide and the more commonly used semiconductor materials including Ge ,Si and GaAs.

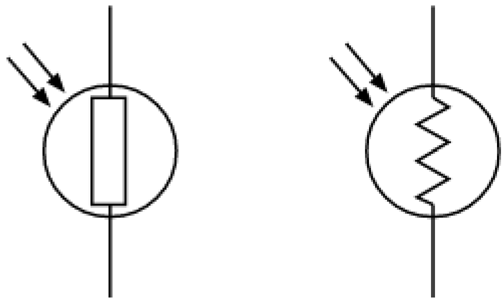


Fig 3.8: - Schematic Symbol of Photoresistor/LDR

The photoresistor, or Light Dependent Resistor, finds many uses as a low cost photosensitive element and was used for many years in photographic light meters as well as other applications.such as flame, smoke, and burglar detectors, card readers and lighting controls for street lamps. Units for the light intensity are Lux or Lumens.

1. **Basic structure:**

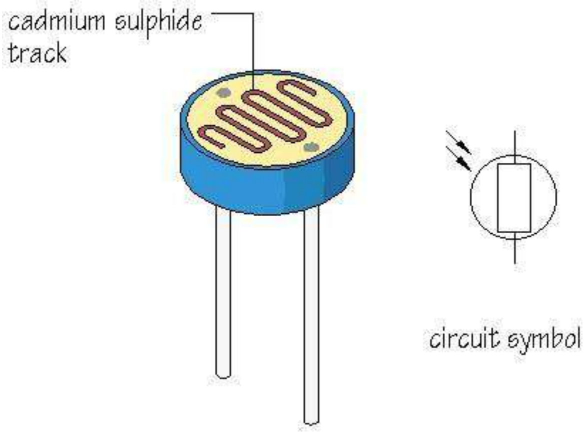


Fig 3.9 LDR Internal Layout

Although there are many ways in which LDR's or photo resistors can be manufactured, there are naturally a few more common methods that are seen. Essentially the LDR or photoresistor consists of a resistive material sensitive to light that is exposed to light. The photo resistive element comprises a section of material with contacts at either end. Although many of the materials used for light dependent resistors are semiconductors, when used as photo resistors, they are used only as a resistive element and there are no p-n junctions. Accordingly the devices are purely passive.

A typical structure for a Light Dependent Resistor uses an active semiconductor layer that is deposited on an insulating substrate. The semiconductor is normally lightly doped to enable it to have the required level of conductivity. Contacts then placed either side of the exposed area. In many instances the area between the contacts is in the form of zig zag, or inter digital pattern. This maximizes the exposed area and by keeping the distance between the contacts small it enhances the gain.

It is also possible to use a polycrystalline semiconductor that is deposited onto a substrate such as ceramic. This makes for a very low cost light dependent resistor.

2. Operation

Light Dependent Resistor made of a high resistance semiconductor, if light falling on the is of high enough efficiently, photons absorbed by the semiconductor give bound electrons enough energy to jump into the conduction band. The resulting free electron (and its hole partner) conduct electricity, thereby lowering resistance.

In intrinsic devices, the only available electrons are in the valence band, and hence the photon must have enough energy to excite the electrons across the entire band gap. Extrinsic devices have impurities added , which have a ground state energy closer to the conduction band, since the electrons don't have so far to jump, lower energy photons (i.e. longer wavelengths and lower frequencies) will suffice to trigger the device.

3. Characteristics of LDR:

The characteristics of LDR are shown below. Here the resistance variations are shown as a function of illumination. The resistance of LDR decreases with increasing incident light intensity.

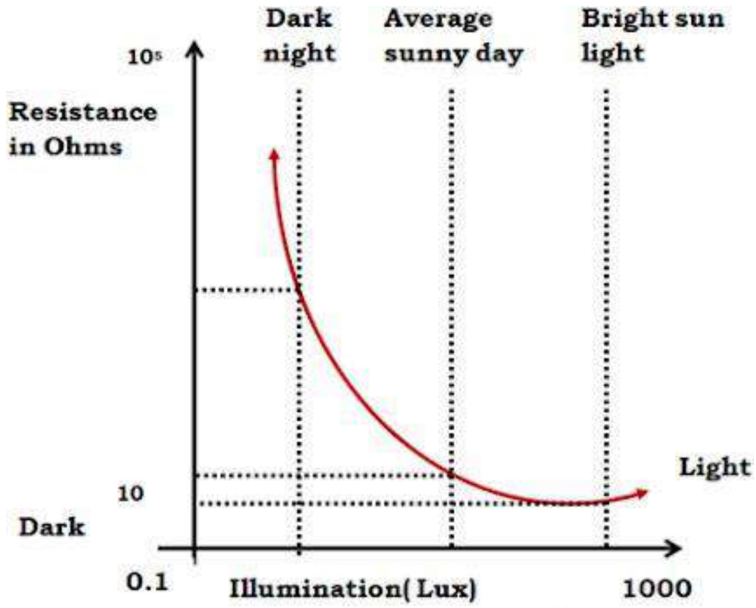


Fig 3.10: - Characteristics of LDR

4. LDR Applications:

Photoresistors come in many types. Inexpensive cadmium sulfide (CdS) cells can be found in many consumer items such as camera light meters, clock radios, alarm devices (as the detector for a light beam), nightlights, outdoor clocks, solar street lamps, and solar road studs, etc.Photoresistors can be placed in streetlights to control when the light is on. Ambient light falling on the photoresistor causes the streetlight to turn off. Thus energy is saved by ensuring the light is only on during hours of darkness.

Photoresistors or LDRs are also used in laser-based security systems to detect the change in the light intensity when a person/object passes through the laser beam.They are also used in some dynamic compressors together with a small incandescent or neon lamp, or light-emitting diode to control gain reduction. A common usage of this application can be found in many guitar amplifiers that incorporate an onboard tremolo effect, as the oscillating light patterns control the level of signal running through the amp circuit.The use of CdS and CdSe[3] photoresistors is severely restricted in Europe due to the RoHS ban on cadmium.

Lead sulfide (PbS) and indium antimonide (InSb) LDRs (light-dependent resistors) are used for the mid-infrared spectral region. Ge:Cu photoconductors are among the best far-infrared detectors available, and are used for infrared astronomy and infrared spectroscopy.

1. Camera light meters.
2. Clock radios.
3. Security alarms.
4. Optical switches.
5. Far infrared detector.
6. Streetlights.

5. **Testing Circuit of LDR:**

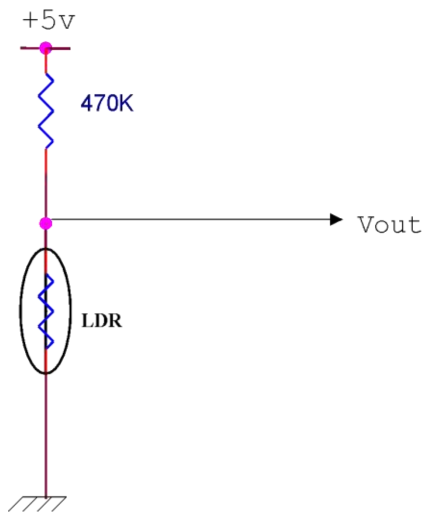


Fig 3.11: - Testing Circuit of LDR

3.2.3 SERVO MOTOR

A **servo motor** is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor which runs through a **servo mechanism**. If a motor is powered by a DC power supply then it is called a DC servo motor, and if it is an AC-powered motor then it is called an AC servo motor. For this tutorial, we will be discussing only about the **DC servo motor working**. Apart from these major classifications, there are many other types of servo motors based on the type of gear arrangement and operating characteristics. A servo motor usually comes with a gear arrangement that allows us to get a very high torque servo motor in small and lightweight packages. Due to these features, they are being used in many applications like toy cars, RC helicopters and planes, Robotics, etc.

Servo motors are rated in kg/cm (kilogram per centimeter) most hobby servo motors are rated at 3kg/cm or 6kg/cm or 12kg/cm. This kg/cm tells you how much weight your servo motor can lift at a particular distance. For example: A 6kg/cm Servo motor should be able to lift 6kg if the load is suspended 1cm away from the motors shaft, the greater the distance the lesser the weight carrying capacity. The position of a servo motor is decided by electrical pulse and its circuitry is placed beside the motor.

Servo Motor Working Mechanism

It consists of three parts:

1. Controlled device
2. Output sensor
3. Feedback system

It is a closed-loop system where it uses a positive feedback system to control motion and the final position of the shaft. Here the device is controlled by a feedback signal generated by comparing output signal and reference input signal.

Here the reference input signal is compared to the reference output signal and the third signal is produced by the feedback system. And this third signal acts as an input signal to control the device. This signal is present as long as the feedback signal is generated or there is a difference between the reference input signal and reference output signal. So the main task of servomechanism is to maintain the output of a system at the desired value in the presence of noises.

Servo Motor Working Principle

A servo consists of a Motor (DC or AC), a potentiometer, gear assembly, and a controlling circuit. First of all, we use gear assembly to reduce RPM and to increase torque of the motor. Say at the initial position of the servo motor shaft, the position of the potentiometer knob is such that there is no electrical signal generated at the output port of the potentiometer. Now an electrical signal is given to another input terminal of the error detector amplifier. Now the difference between these two signals, one comes from the potentiometer and another comes from other sources, will be processed in a feedback mechanism and output will be provided in terms of error signal. This error signal acts as the input for the motor and motor starts rotating. Now the motor shaft is connected with the potentiometer and as the motor rotates so does the potentiometer and it will generate a signal. So as the potentiometer's angular position changes, its output feedback signal changes. After sometime the position of potentiometer reaches a position that the output of potentiometer is the same as the external signal provided. At this condition, there will be no output signal from the amplifier to the motor input as there is no difference between external applied signal and the signal generated at potentiometer, and in this situation the motor stops rotating.

Interfacing Servo Motors with Microcontrollers:

Interfacing hobby Servo motors like s90 servo motors with MCU is very easy. **Servos have three wires coming out of them**. Out of which two will be used for Supply (positive and negative) and one will be used for the signal that is to be sent from the MCU. An **MG995 Metal Gear Servo Motor** which is most commonly used for RC cars, humanoid robots etc. The picture of MG995 is shown below:

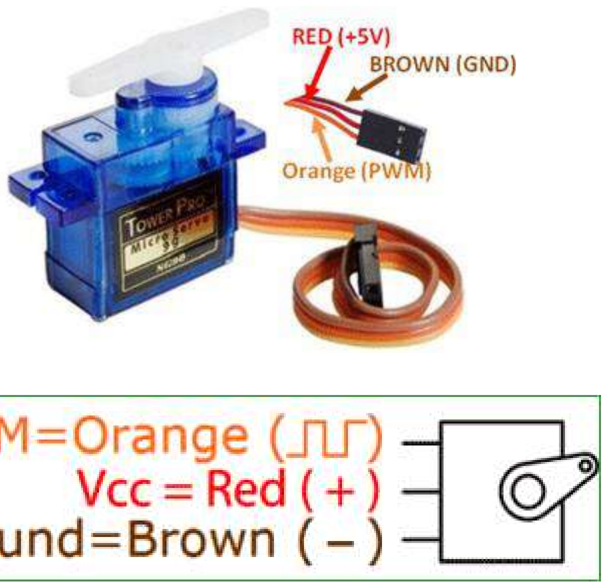


Fig 3.12 Servo Motor PWM Wiring

The color coding of your servo motor might differ hence check for your respective datasheet.

All servo motors work directly with your +5V supply rails but we have to be careful on the amount of current the motor would consume if you are planning to use more than two servo motors a proper servo shield should be designed.

Controlling Servo Motor:

All motors have three wires coming out of them. Out of which two will be used for Supply (positive and negative) and one will be used for the signal that is to be sent from the MCU.

Servo motor is controlled by PWM (Pulse width Modulation) which is provided by the control wires. There is a minimum pulse, a maximum pulse and a repetition rate. Servo motor can turn 90 degrees from either direction from its neutral position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position, such as if the pulse is shorter than 1.5ms the shaft moves to 0° and if it is longer than 1.5ms then it will turn the servo to 180°.

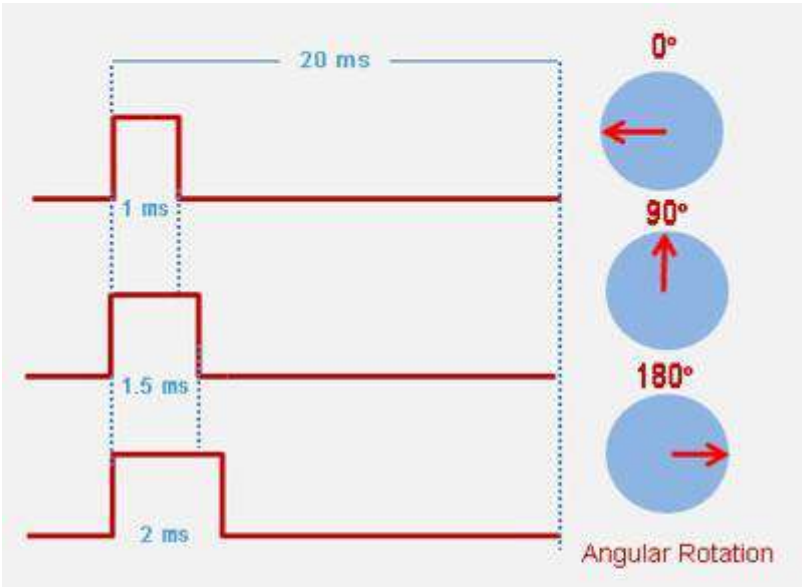


Fig 3.13: PWM Pulsing time series graph

Servo motor works on **PWM (Pulse width modulation)** principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically, servo motor is made up of **DC motor which is controlled by a variable resistor (potentiometer) and some gears**. High speed force of the DC motor is converted into torque by Gears. We know that

WORK= FORCE X DISTANCE,

in DC motor Force is less and distance (speed) is high and in Servo, force is High and distance is less. The potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on the required angle.

Servo motor can be rotated from 0 to 180 degrees, but it can go up to 210 degrees, depending on the manufacturing.

This degree of rotation can be controlled by applying the **Electrical Pulse** of proper width, to its Control pin. Servo checks the pulse every 20 milliseconds. The pulse of 1 ms (1 millisecond) width can rotate the servo to 0 degrees, 1.5ms can rotate to 90 degrees (neutral position) and 2 ms pulse can rotate it to 180 degree.

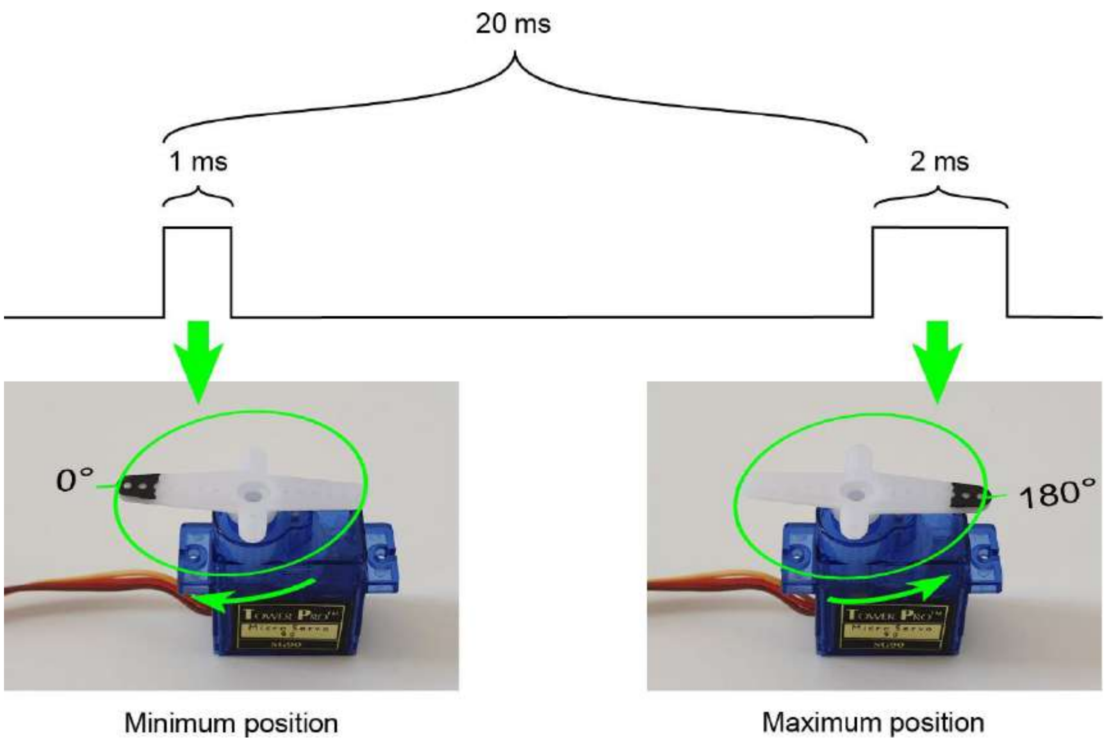


Fig 3.14 Servo Pulse Width with Angle

All servo motors work directly with your +5V supply rails but we have to be careful about the amount of current the motor would consume if you are planning to use more than two servo motors a proper servo shield should be designed.

3.2.3.2 SPECIFICATION

Dimensions & Specifications
A (mm) : 32
B (mm) : 23
C (mm) : 28.5
D (mm) : 12
E (mm) : 32
F (mm) : 19.5
Speed (sec) : 0.1
Torque (kg-cm) : 2.5
Weight (g) : 14.7
Voltage : 4.8 - 6

Table 4: SG90 Specification

3.2.4 LASER DIODE

Fig 3.15: Laser Diode

A Laser Diode is a semiconductor device similar to a light-emitting diode (LED). It uses a p-n junction to emit coherent light in which all the waves are at the same frequency and phase. This coherent light is produced by the laser diode using a process termed as "Light Amplification by Stimulated Emission of Radiation", which is abbreviated as LASER. And since a p-n junction is used to produce laser light, this device is named as a laser diode. Before we learn more about the working process of a laser diode, let's look at how laser light is different from other types of light, and its advantages.

A laser diode is electrically a PIN diode. The active region of the laser diode is in the intrinsic (I) region, and the carriers (electrons and holes) are pumped into that region from the N and P regions respectively. While initial diode laser research was conducted on simple P-N diodes, all modern lasers use the double-hetero-structure implementation, where the carriers and the photons are confined in order to maximize their chances for recombination and light generation. Unlike a regular diode, the goal for a laser diode is to recombine all carriers in the I region, and produce light. Thus, laser diodes are fabricated using direct band-gap semiconductors. The laser diode epitaxial structure is grown using one of the crystal growth techniques, usually starting from an N doped substrate, and growing the I doped active layer, followed by the P doped cladding, and a contact layer. The active layer most often consists of quantum wells, which provide lower threshold current and higher efficiency

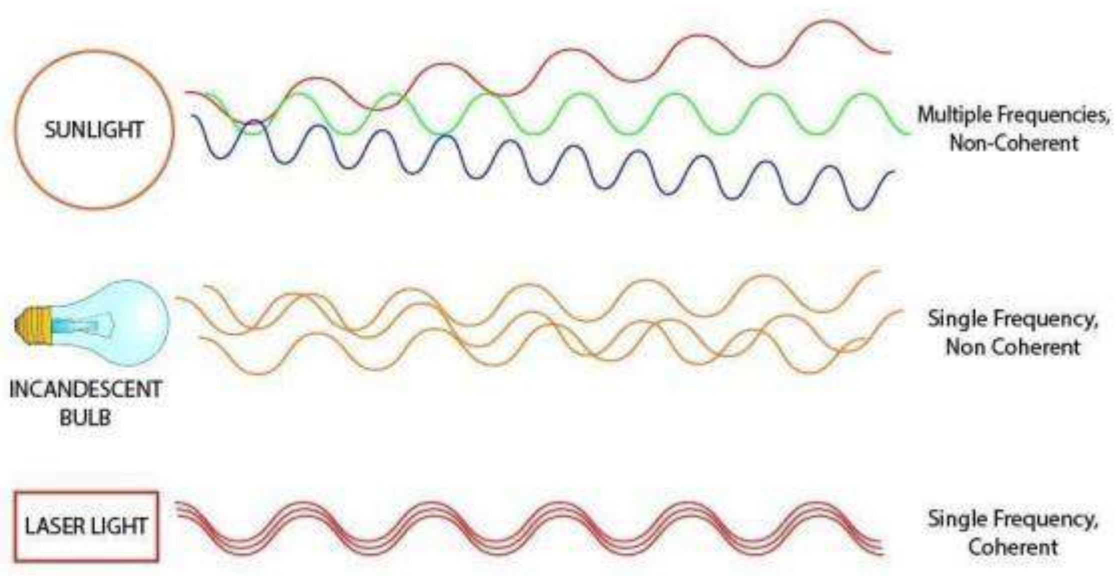


Fig 3.16 Coherency in Optical Waves

The light from sunlight or from most of the artificial light sources contains waves of multiple wavelengths and they are out of phase with each other. The light waves from monochromatic light sources like incandescent bulbs also are not in phase with each other. In contrast to the previous light sources, laser diodes produce a narrow beam of laser light in which all the light waves have similar wavelengths and they travel together with their peaks lined up. This is why laser beams are very bright, and can be focused over a very tiny spot.

Of all the devices that produce laser light, laser diodes or semiconductor lasers are the most efficient and they come in smaller packages. So they are widely used in various devices like laser printers, barcode readers, security systems, Autonomous vehicles (LIDAR), Fiber optic communications etc.

3.2.4.1 WORKING OF LASER DIODE

The working of a laser diode takes place in three main steps:

ENERGY ABSORPTION

The laser diode consists of a p-n junction where holes and electrons exist. (Here, a hole means the absence of an electron). When a certain voltage is applied at the p-n junction, the electrons absorb energy and they transition to a higher energy level.

Holes are formed at the original position of the excited electron. The electrons stay in this excited state without recombining with holes for a very small duration of time, termed as "recombination time" or "upper-state lifetime". The recombination time is about a nanosecond for most laser diodes.

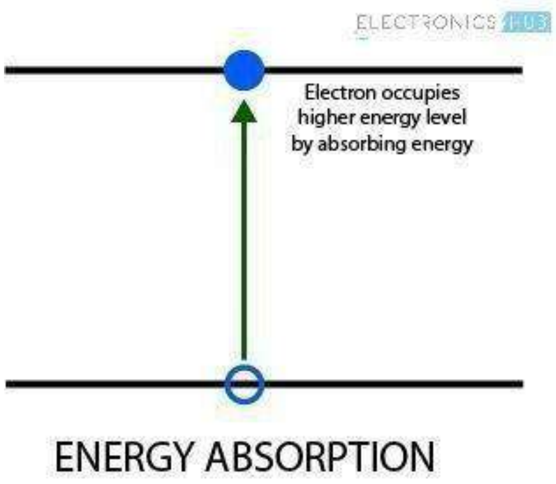


Fig 3.17 Energy Absorption

SPONTANEOUS EMISSION

After the upper-state lifetime of excited electrons, they recombine with holes. As the electrons fall from higher energy level to a lower energy level, the difference in energy is converted into photons or electromagnetic radiation. This same process is used to produce light in LEDs. The energy of the emitted photon is given by the difference between the two energy levels.

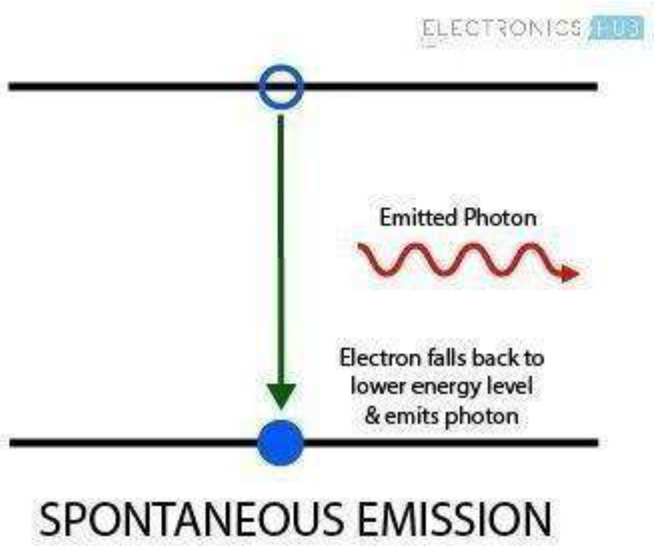


Fig 3.18 Spontaneous Emission

STIMULATED EMISSION

We need more coherent photons from the laser diode than the ones emitted through the process of spontaneous emission. A partially reflecting mirror is used on either side of the diode so that the photons released from spontaneous emission are trapped in the p-n junction until their concentration reaches a threshold value. These trapped photons stimulate the excited electrons to recombine with holes even before their recombination time. This results in the release of more photons that are in exact phase with the initial photons and so the output gets amplified. Once the photon concentration goes above a threshold, they escape from the partially reflecting mirrors, resulting in a bright monochromatic coherent light.

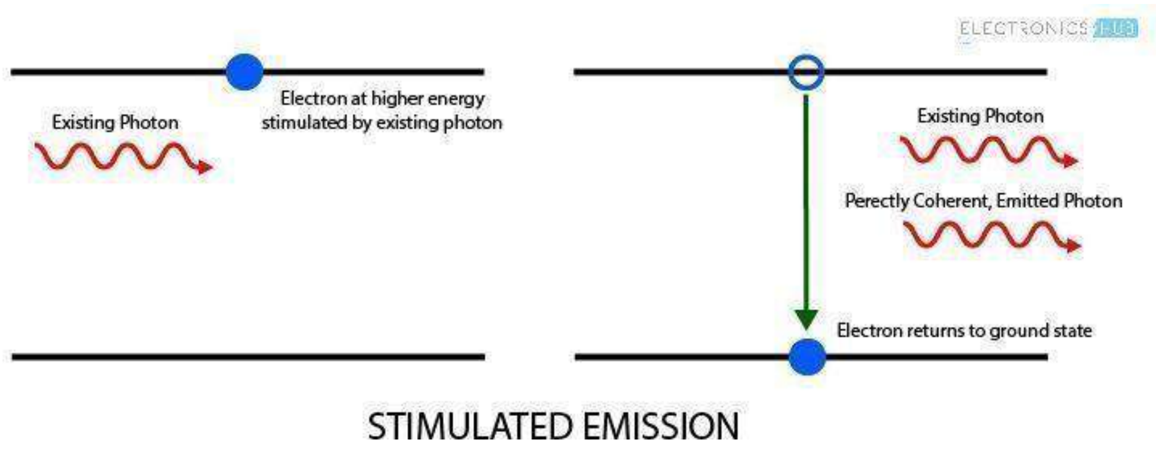


Fig 3.19 Stimulated Emission

CONSTRUCTION OF A LASER DIODE

A simple semiconductor laser diode is made up of the following parts in order:

- Metal Contact
- P-type Material
- Active/Intrinsic Region (N-type Material)
- N-type Material
- Metal Contact

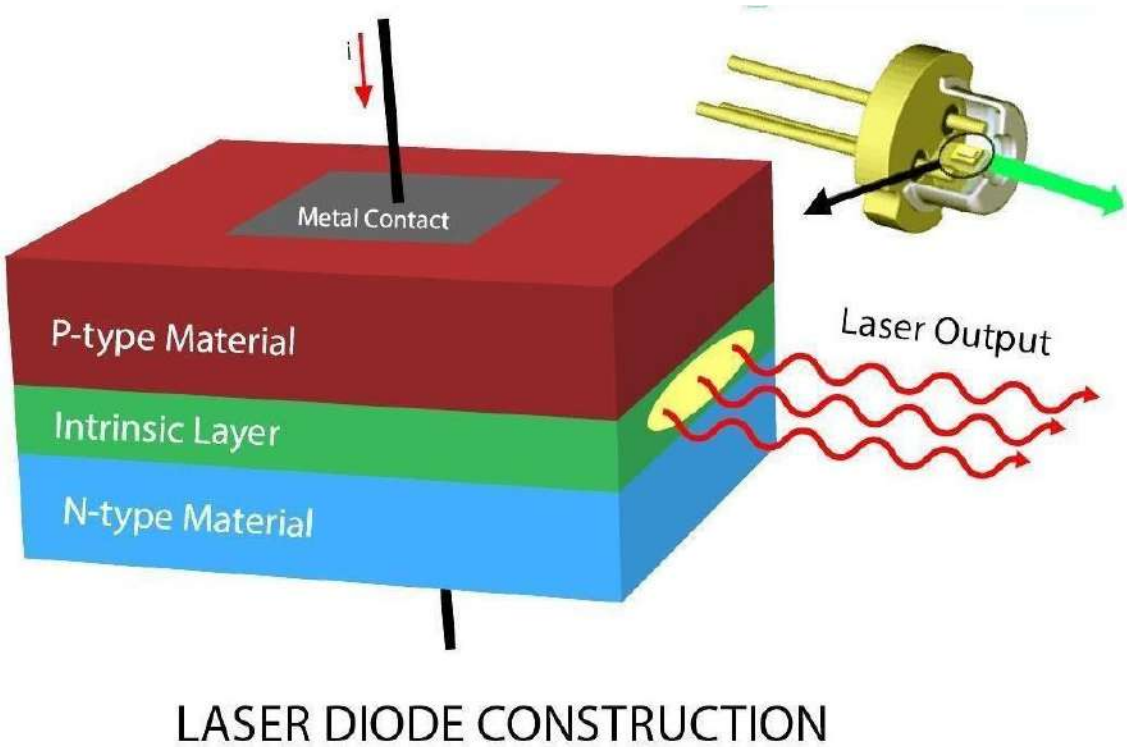


Fig 3.20 Laser Diode Construction

The input terminals are connected to metal plates which are sandwiched to the n-type and p-type layers. This type of laser diode is also called a “Homojunction Laser Diode”. The intrinsic region between the p-type and n-type material is used to increase the volume of the active region, so that more holes and electrons can accumulate at the junction. This allows more electrons to recombine with holes at any instant of time, resulting in better output power. The laser light is emitted from the elliptical region. This beam from the laser diode can be further focused using an optical lens. This entire PIN diode (P-type, Intrinsic, N-Type) arrangement is enclosed normally in a metal casing.

3.2.4.2 TYPES OF LASER DIODES

Double Heterostructure Laser Diode

In this type of laser diodes, an additional confinement layer of a different material is sandwiched between the two p-type and n-type materials. Each junction between different materials is called a heterostructure. Because of the presence of two heterostructures, this type of laser diode is named as a double heterostructure (DH) laser diode. The advantage of this DH laser diode is that the active region is confined to a thin layer which gives better optical amplification.

Quantum Well Laser Diode

The quantum well laser diode has a very thin middle layer, which acts as a quantum well. The electrons will be able to use quantum energy levels when transitioning from higher energy level to lower energy level. This gives a better efficiency for this type of laser diode.

Separate Confinement Heterostructure Laser Diode

The thin middle layer in the quantum well laser diode is very small for confining emitted light effectively. To compensate this, in the separate confinement heterostructure laser diode, another two layers are added over the three initial layers. These layers have a lower refractive index and help in confining the emitted light effectively.

Vertical Cavity Surface Emitting Laser Diode (VCSEL)

All the previously discussed laser diodes, the optical cavity is placed perpendicular to the current flow. In a vertical cavity surface emitting laser diodes, however, the optical cavity is along the axis of current flow. The partially reflecting mirrors are placed near the ends of the optical cavity.

Other Types

- Quantum Cascade Laser Diode
- Interband Cascade Laser Diode
- Distributed Bragg Reflector Laser Diode
- Distributed Feedback Laser Diode
- External Cavity Diode Laser
- Vertical External Cavity Surface Emitting Laser Diode (VCSEL)

3.2.4.3 Laser Diode P-I Characteristics

The diagram below is a graphical plot between output optical power on the y-axis and the current input to the laser diode on x-axis.

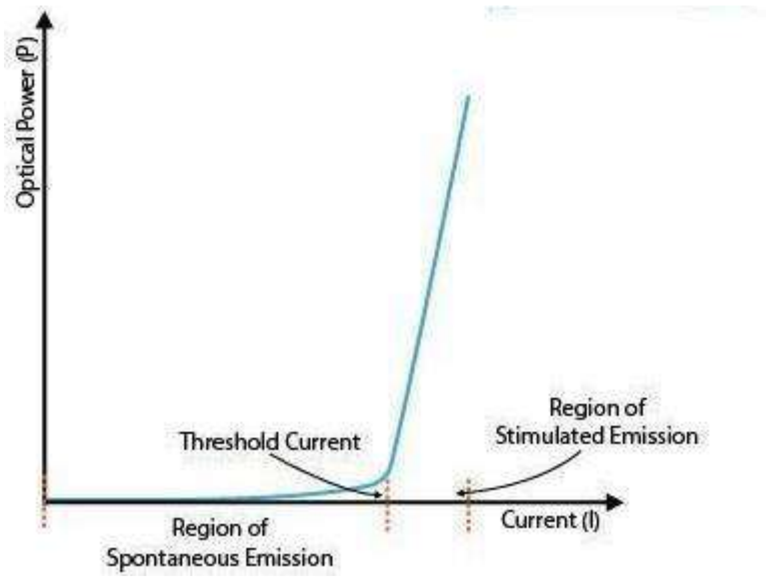


Fig 3.21 Laser Diode P-I Graph

As we increase the current flow to the laser diode, the optical power of output light gradually increases up to a certain threshold. Until this point, most of the light emitted is due to spontaneous emission. Above this threshold current, the process of stimulated emission increases. This causes the power of output light to increase a lot even for smaller increases in input current. The output optical power also depends on temperature and it reduces with decrease in temperature.

How to make a Laser Diode Driver Circuit?

Laser diodes require complex drive circuitries that involve feedback loops by measuring output optical power, temperature, voltage and input current. But for controlling a laser diode used in applications where high accuracy is not required, a simple laser diode driver circuit can be constructed using LM317 voltage regulator IC. Below is the diagram.

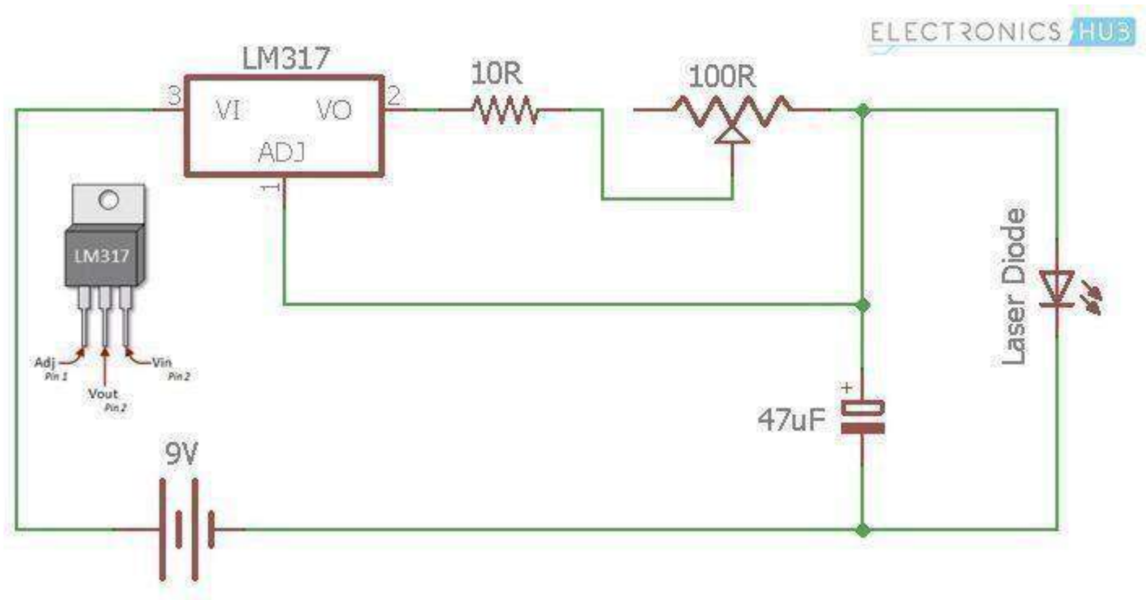


Fig 3.22 Laser Driver Circuit

The LM317 is configured to function as a constant current source. The output current depends on the value of resistance between Vout and Vadj of LM317 (Pin2 & Pin1). So adjusting the 100R potentiometer will change the output current that flows into the laser diode. The 10R resistor is used to prevent large currents from flowing when the value of 100R potentiometer is at zero. The 47uF capacitor is used to absorb any battery voltage spikes.

3.2.4.4 SPECIFICATION

Supply Voltage	5v
Power	5mW
Current	30mA
Wavelength	650 nm
Colour	Red
Module Size	15 x 24 x 9mm

Table: Laser Diode Specification

3.2.5 SOFTWARE EXPLANATION

3.2.5.1 Proteus

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

3.2.5.2 About Proteus

It is a software suite containing schematic, simulation as well as PCB designing.

ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.

The designer can also develop 2D drawings for the product.

3.2.5.3 Features

ISIS has a wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semiconductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the footprints of different categories of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in ISIS can be directly transferred to ARES.

3.2.5.4 Starting New Design

Step 1: Open ISIS software and select New design in File menu

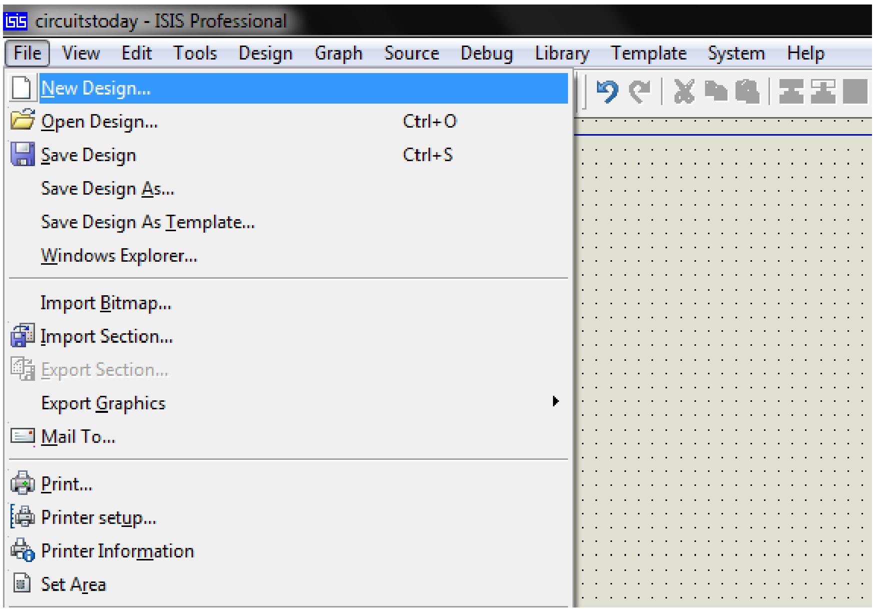


Figure 3.23 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

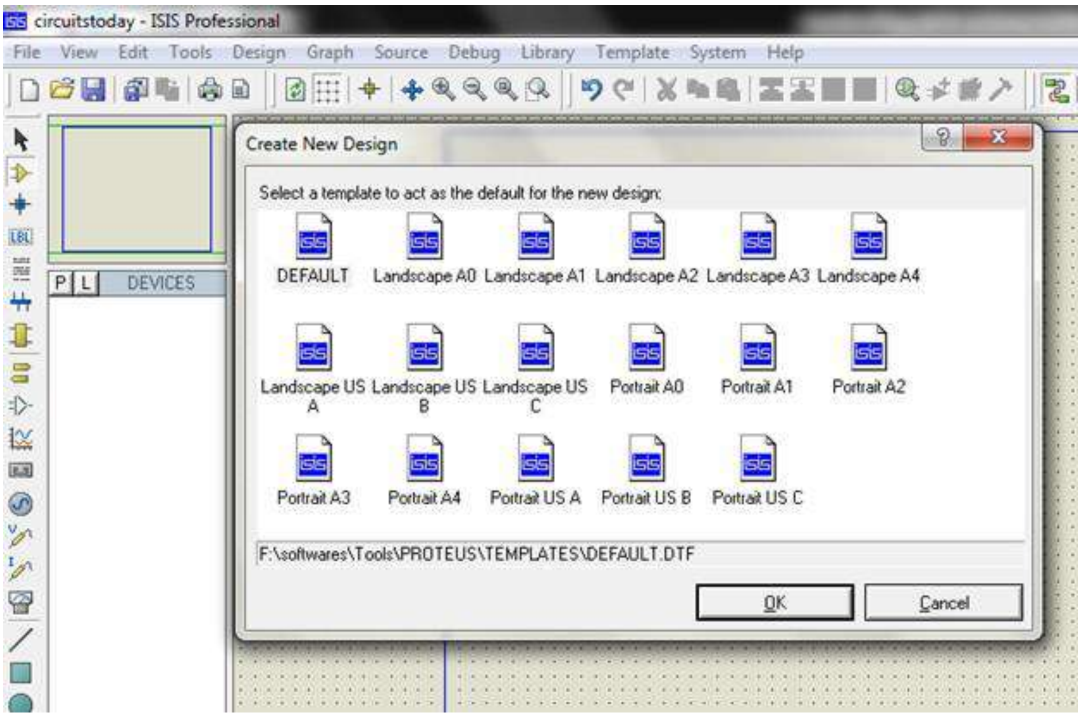


Figure3.24 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

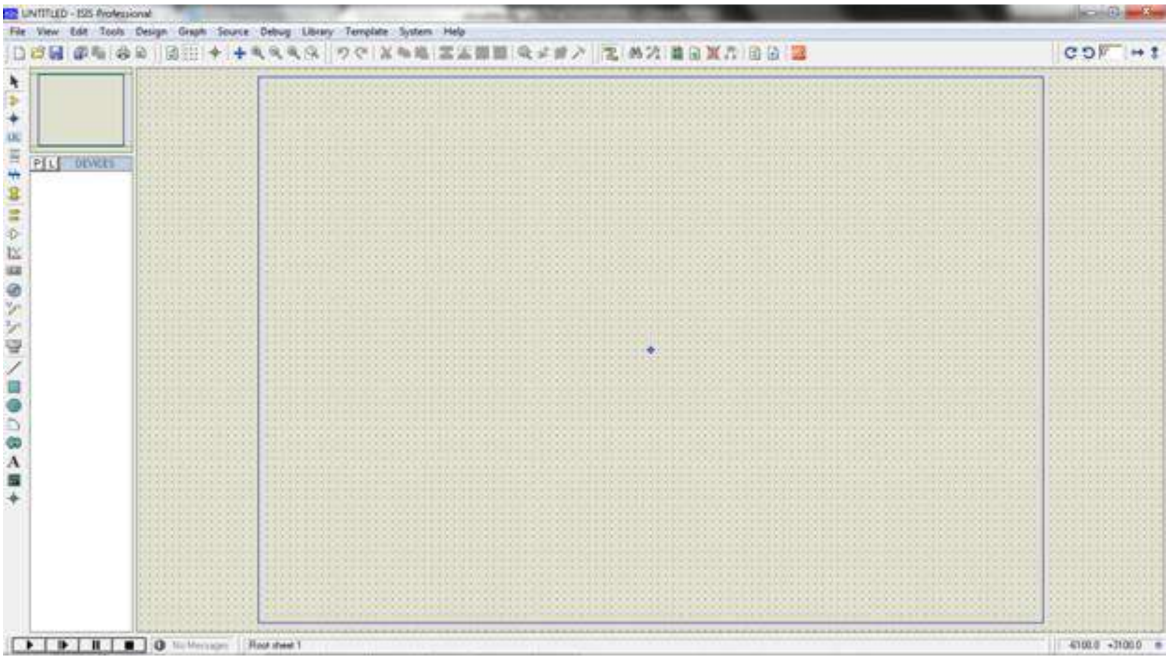


Figure 3.25 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

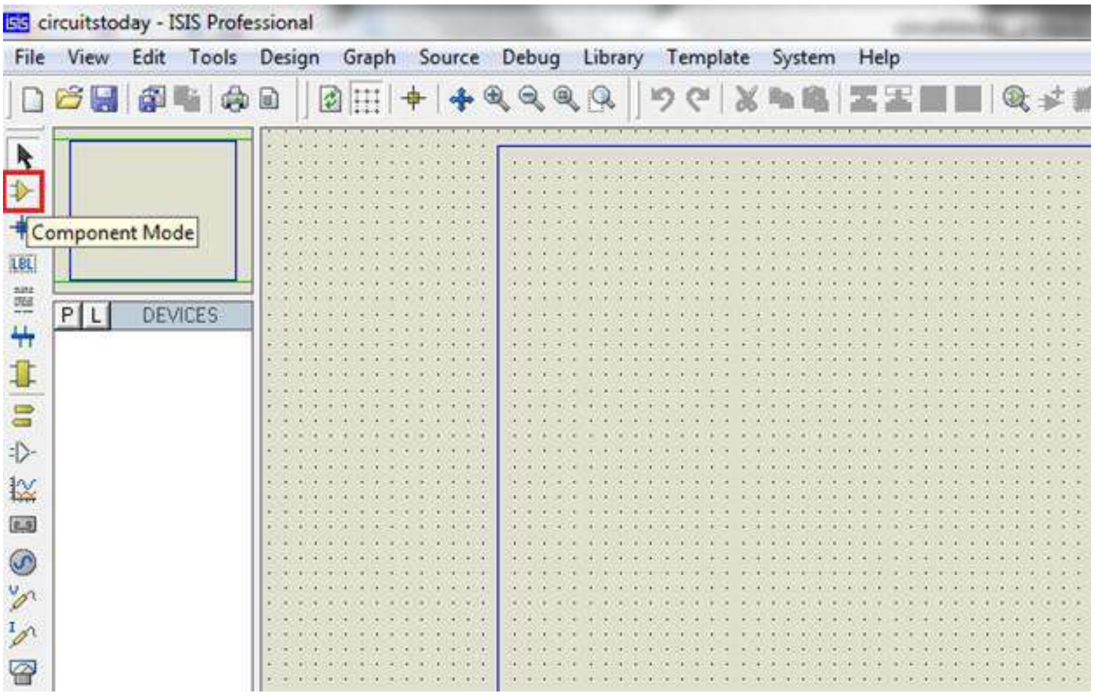


Figure 3.26 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

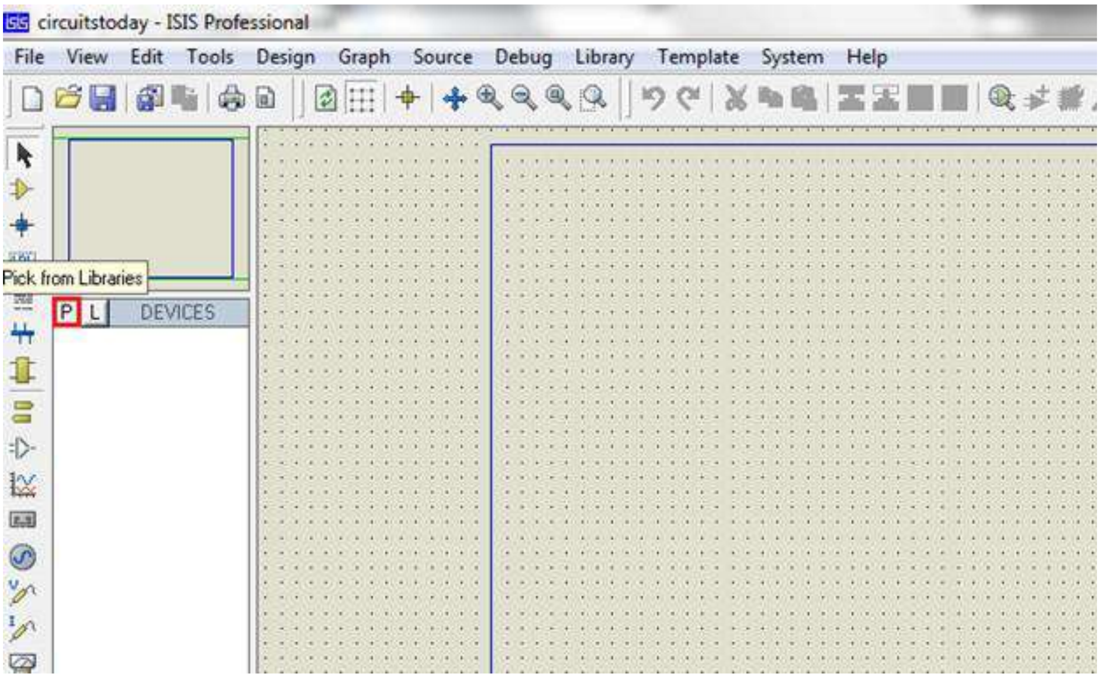


Figure 3.27 Pick from Libraries

Step 6: Select the components from categories or type the part name in the text box.

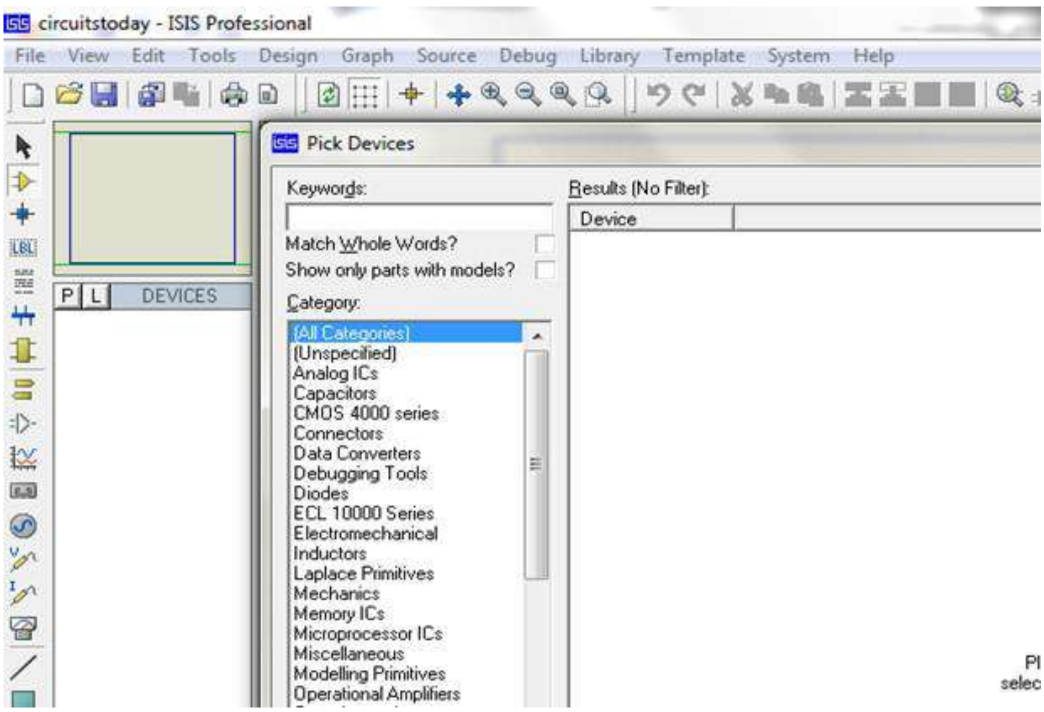


Figure 3.28 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

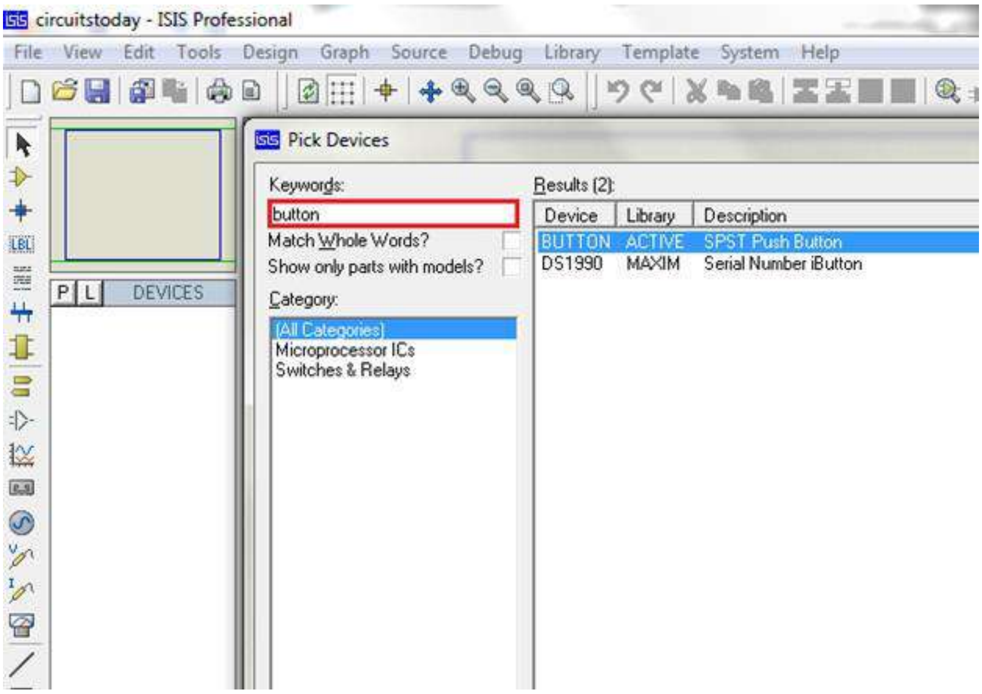


Figure 3.29 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

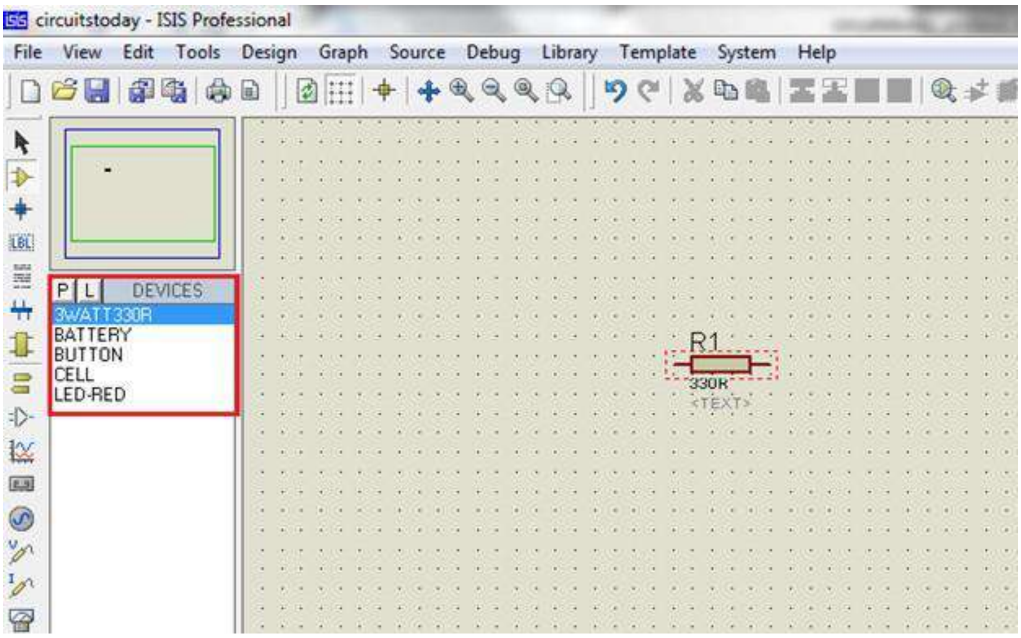


Figure 3.30 Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows it to connect through wires. Left click from one terminal to another to make a connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

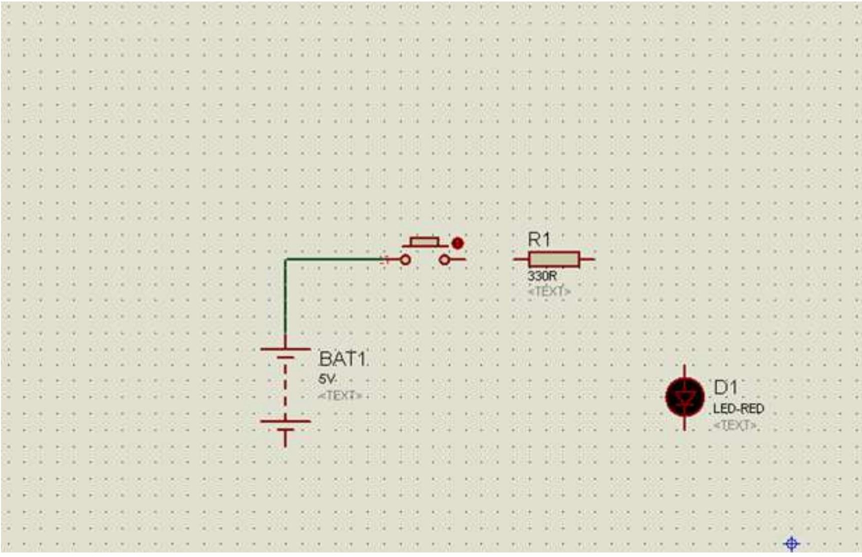


Figure 3.31 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok

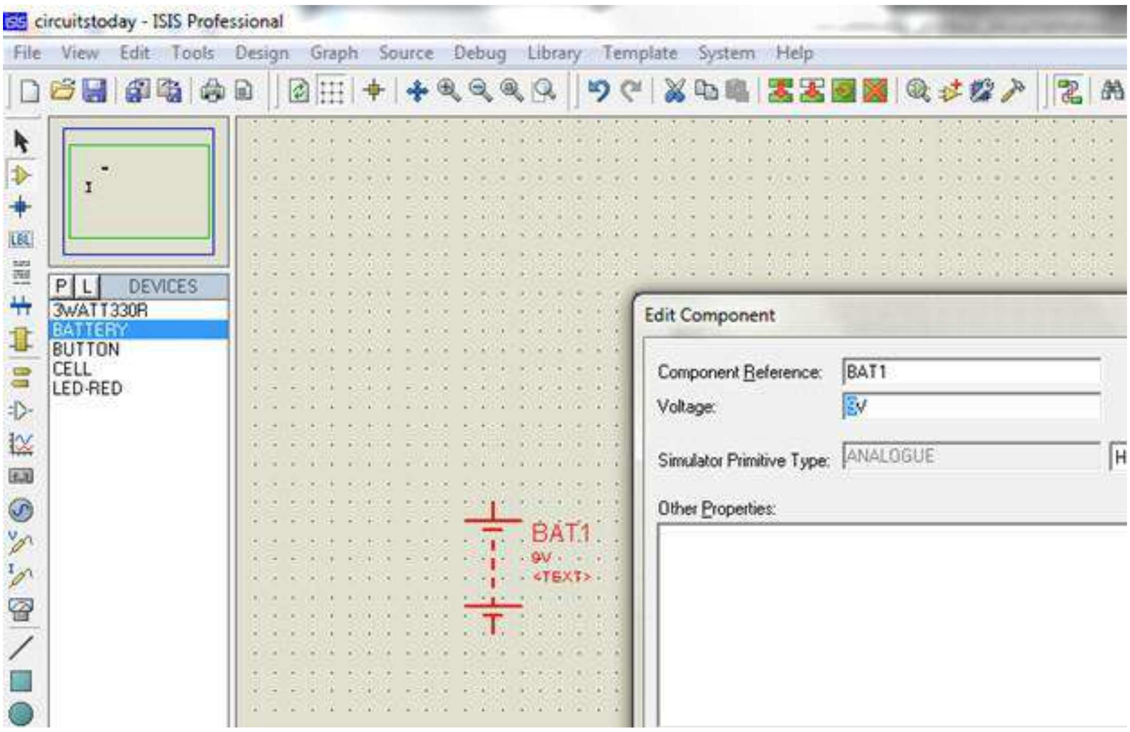


Figure 3.32 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

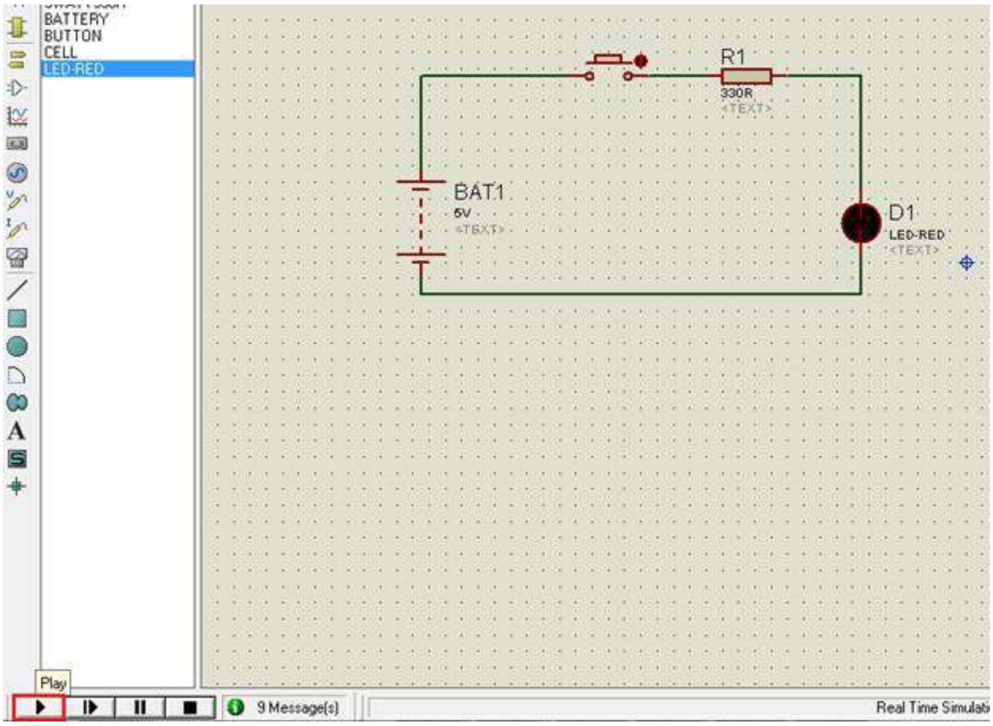


Figure 3.33 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make the LED glow.

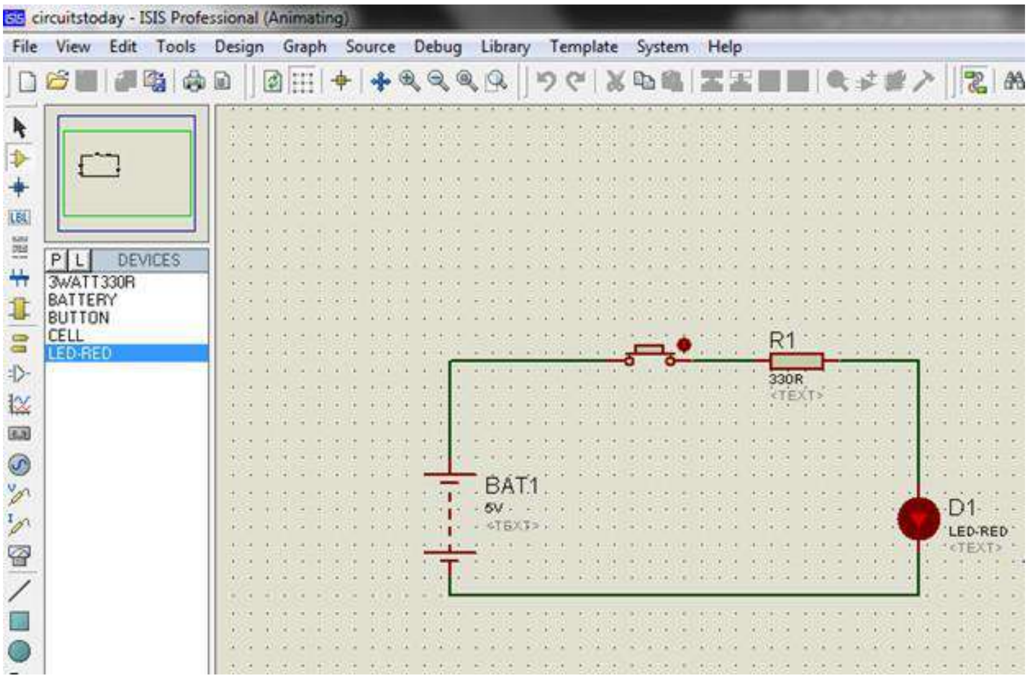


Figure 3.34 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

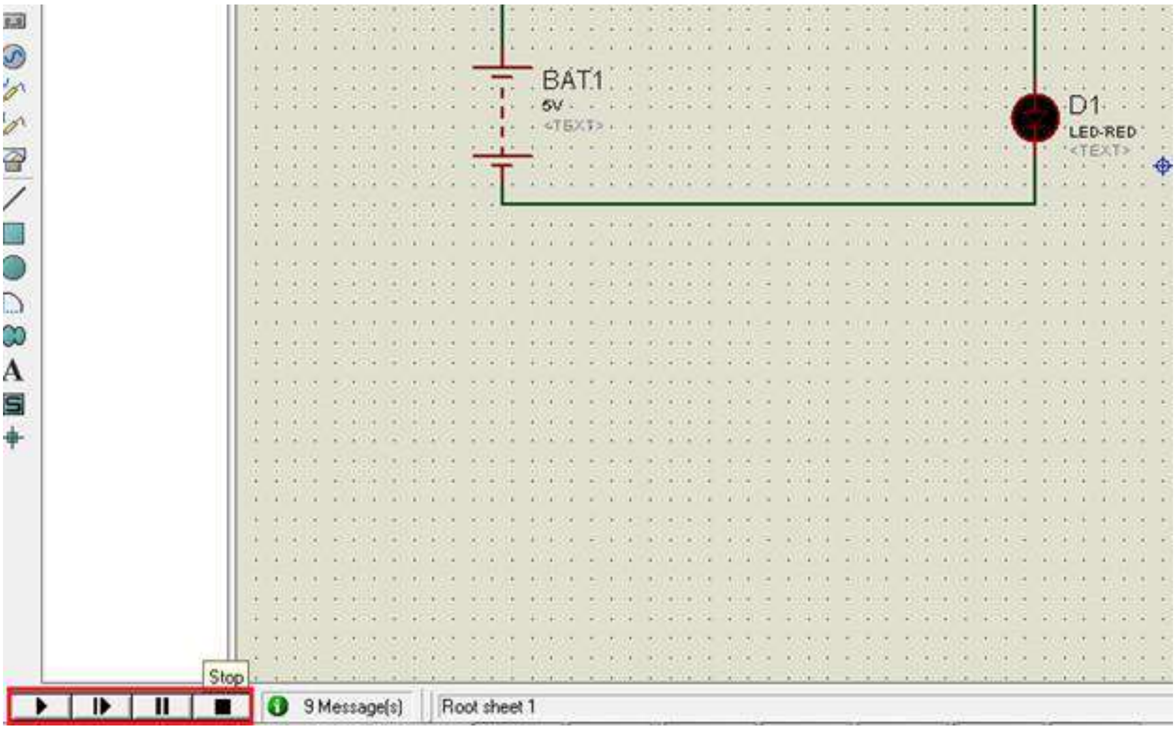


Figure 3.35 Simulation Step-Pause-Stop Buttons

3.2.5.2 Arduino IDE

The **Arduino Integrated Development Environment (IDE)** is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Arduino Pro IDE	
Developer(s)	Arduino Software
Preview release	v0.0.2 / 28 October 2019; 3 months ago[8]
Repository	github.com/arduino/Arduino
Written in	C, C++
Operating system	Windows, macOS, Linux
Platform	IA-32, x86-64, ARM
Type	Integrated development environment
License	LGPL or GPL license
Website	blog.arduino.cc/2019/10/18/arduino-pro-ide-alpha-preview-with-advanced-features/

In October 2019 the Arduino organization began providing early access to a new Arduino Pro IDE with debugging[9] and other advanced features.

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

4.3.1 Installation

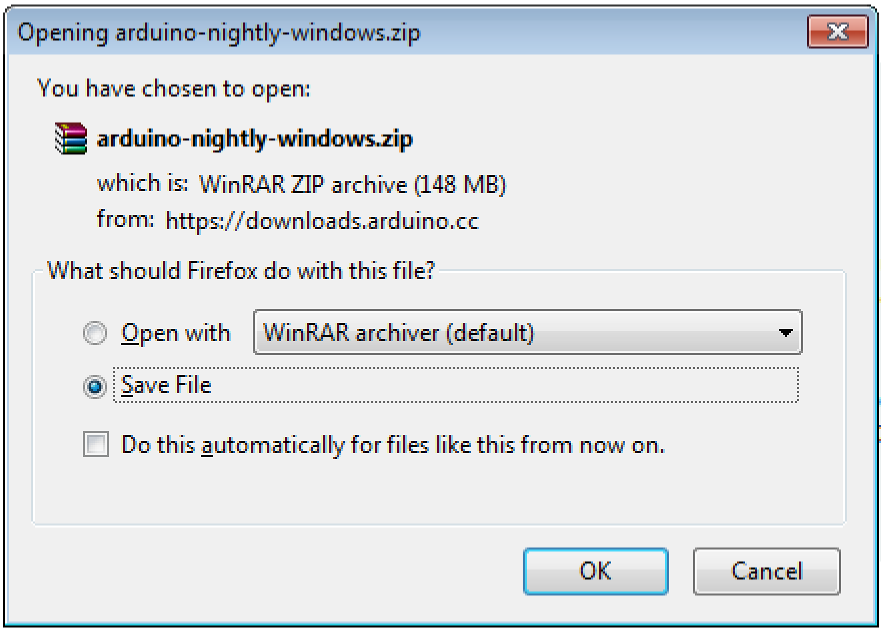
In this section, we will learn in easy steps how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



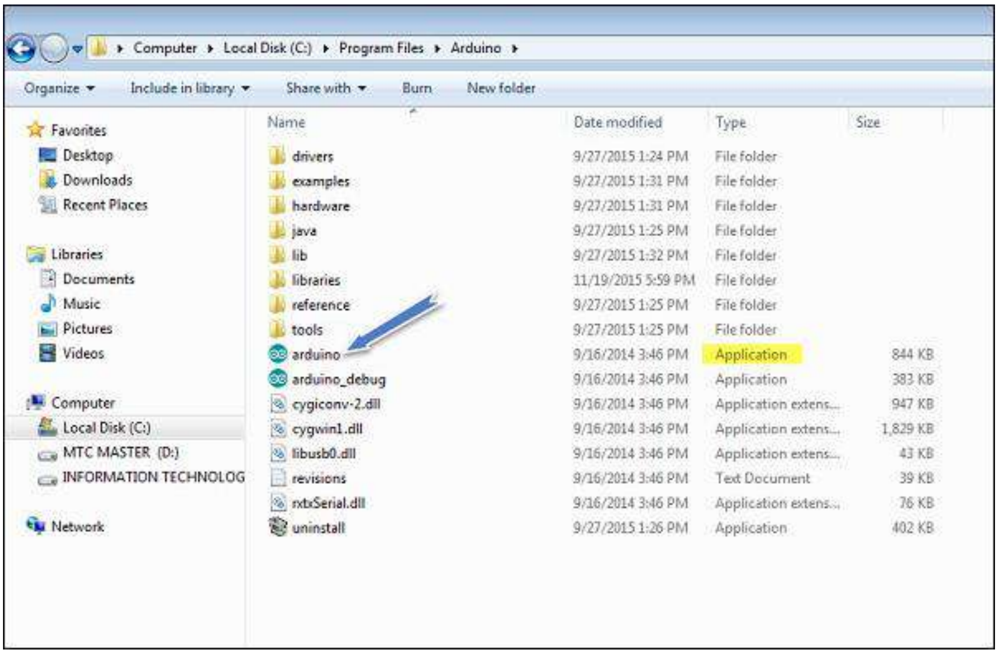
Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

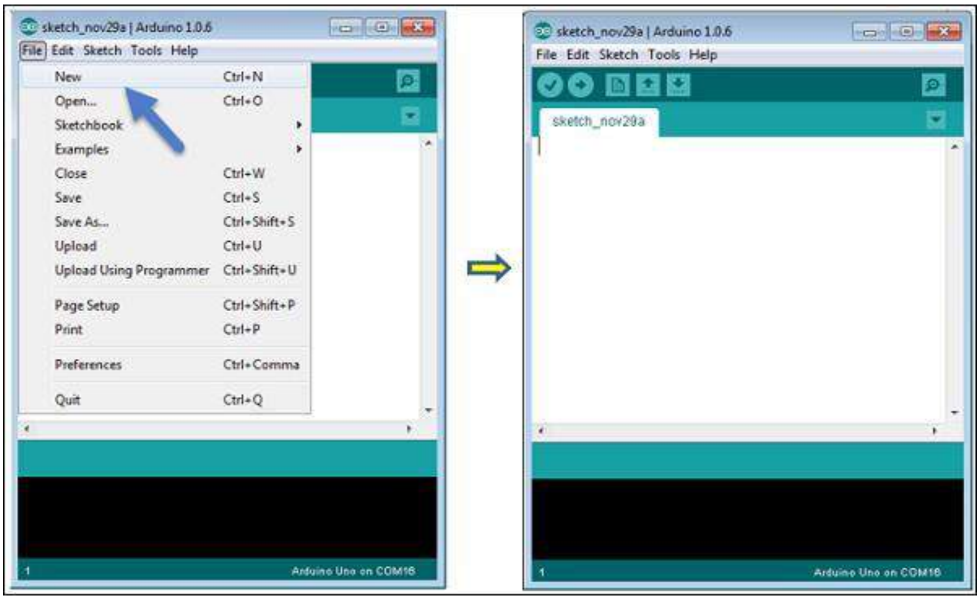


Step 5 − Open your first project.

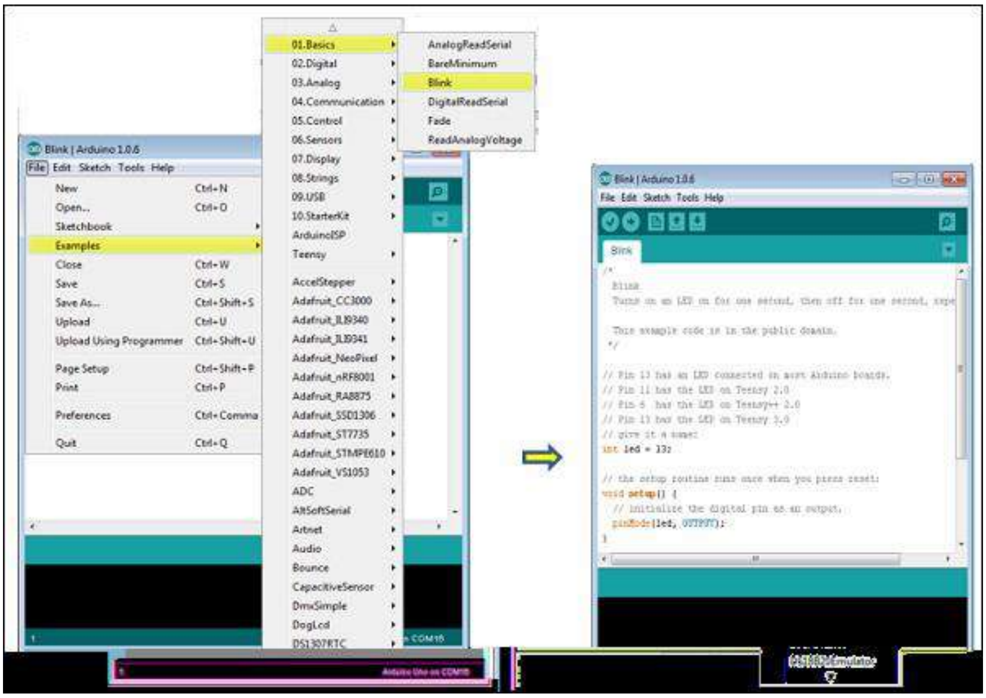
Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.



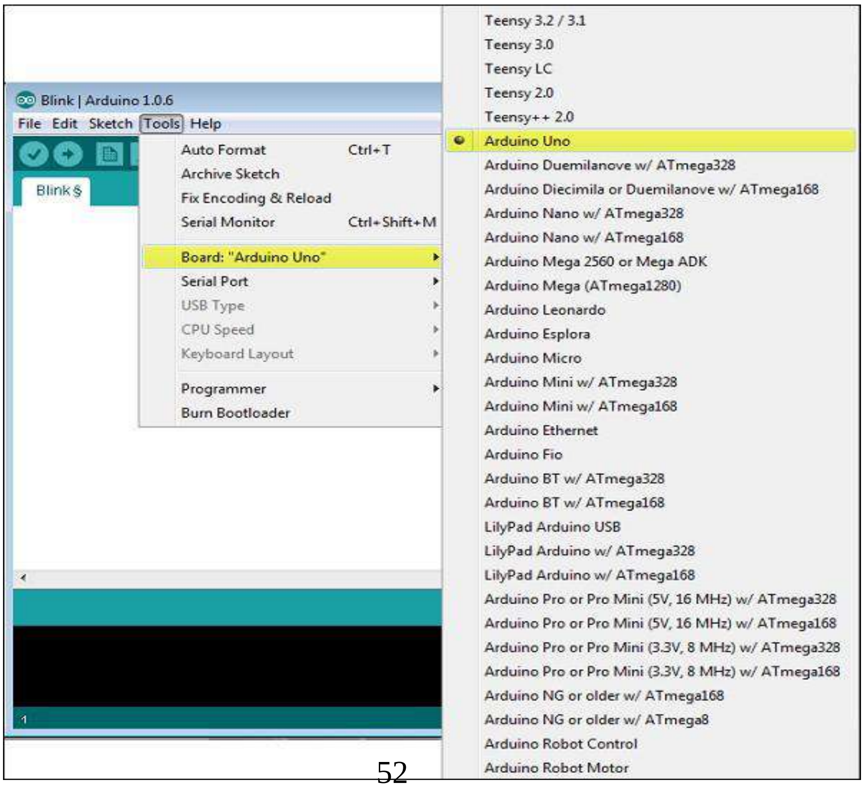
To open an existing project example, select File → Example → Basics → Blink.



Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

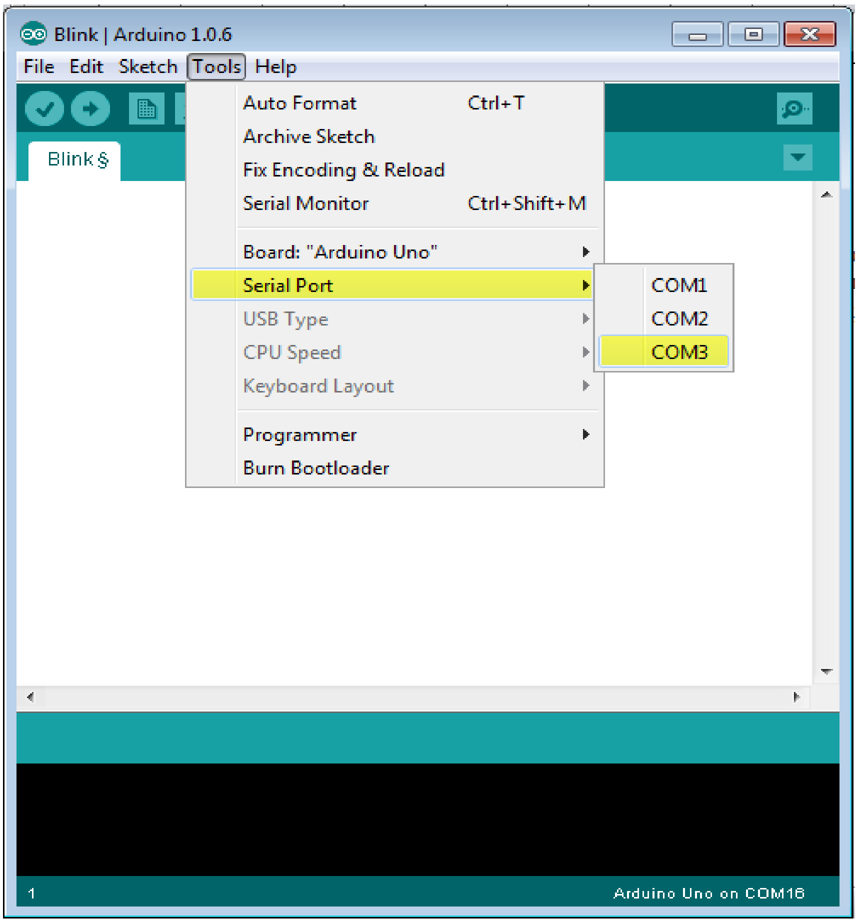
To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.Go to Tools → Board and select your board.



Here, we have selected the Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

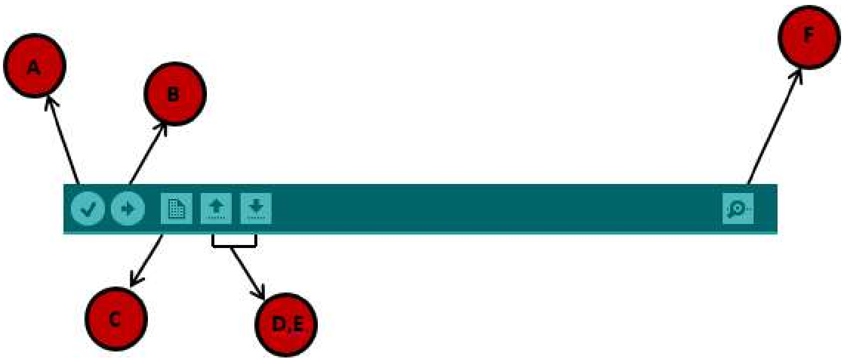
Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.



Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.



A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

CHAPTER-4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM OF PROJECT

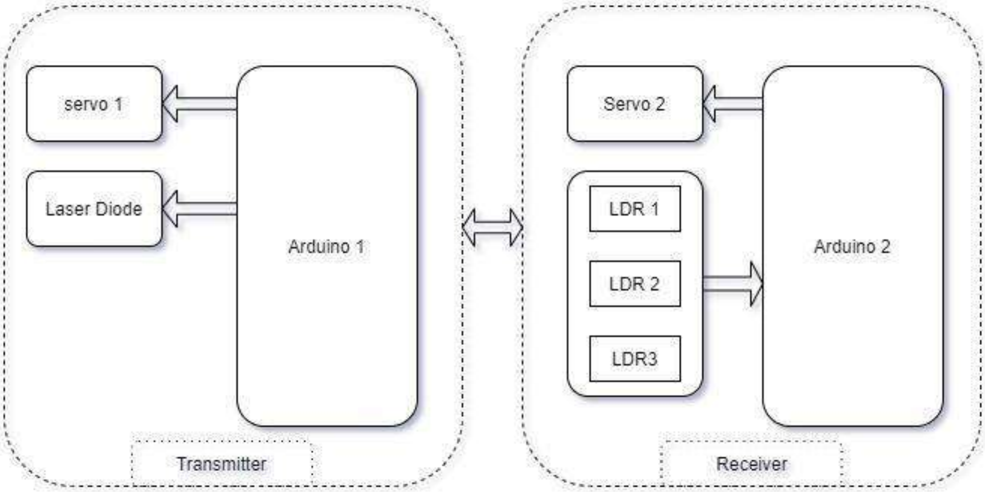
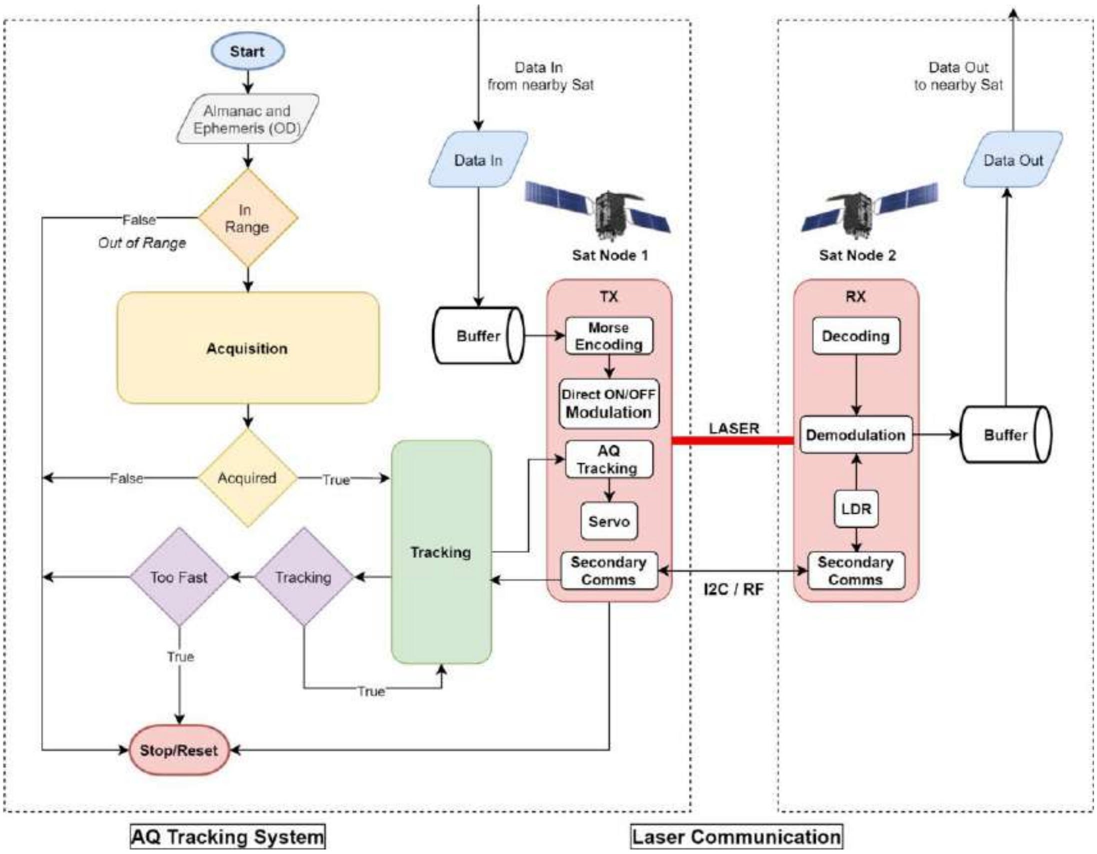


Fig 4.1 Block Diagram of Project

4.2 FLOW CHART



4.3 STATE FLOW DIAGRAM

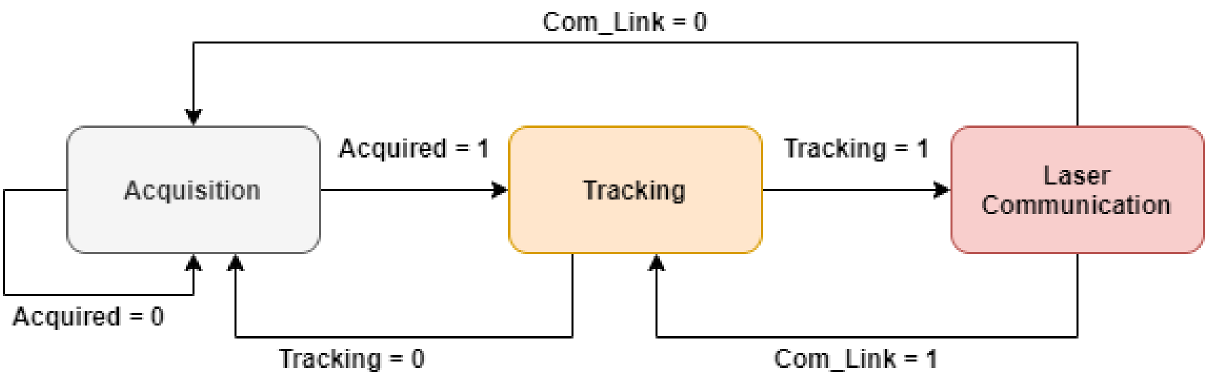


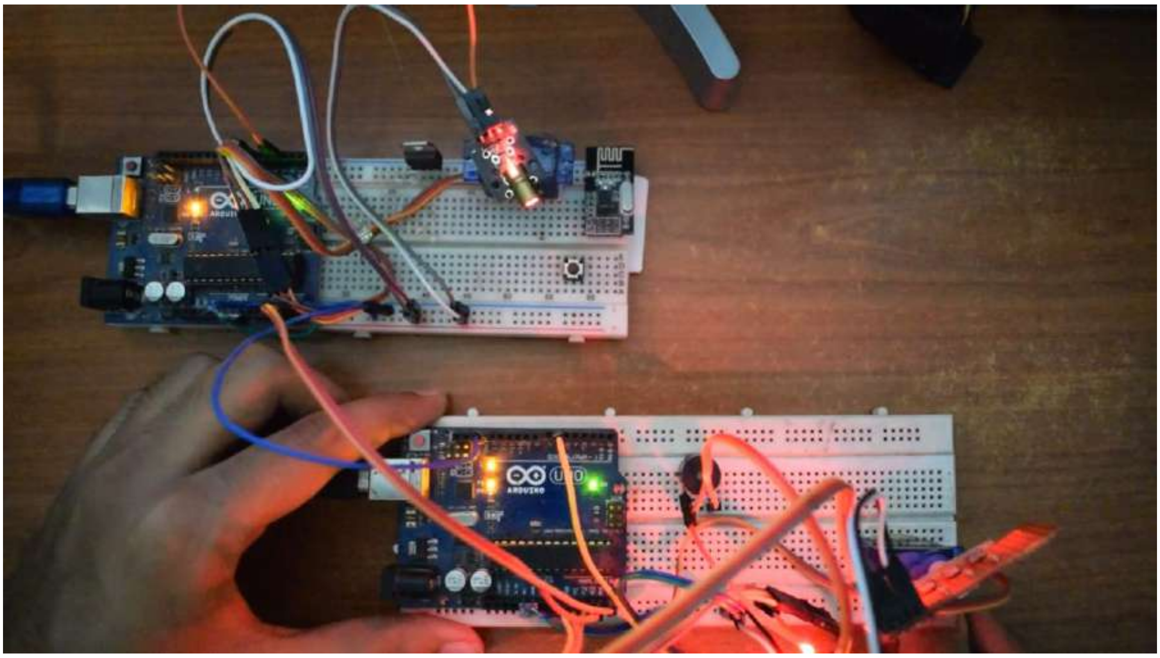
Fig 4.3 State Flow Diagram for Laser Com

State	Laser Comm	Track	Acq	Status
1	0	0	0	Hold
2	0	0	1	AQ
3	0	1	0	-
4	0	1	1	AQ & T
5	1	0	0	error
6	1	0	1	error
7	1	1	0	error
8	1	1	1	Comms Est

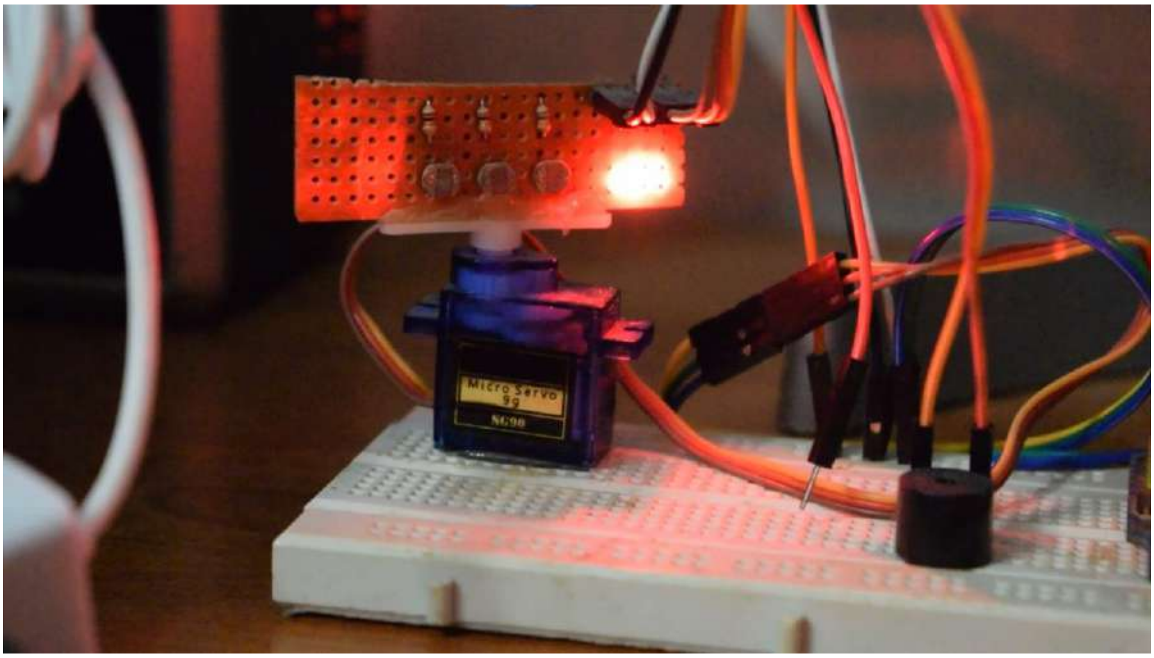
Table 7: State Flow Table

4.4 IMPLEMENTATION STAGES

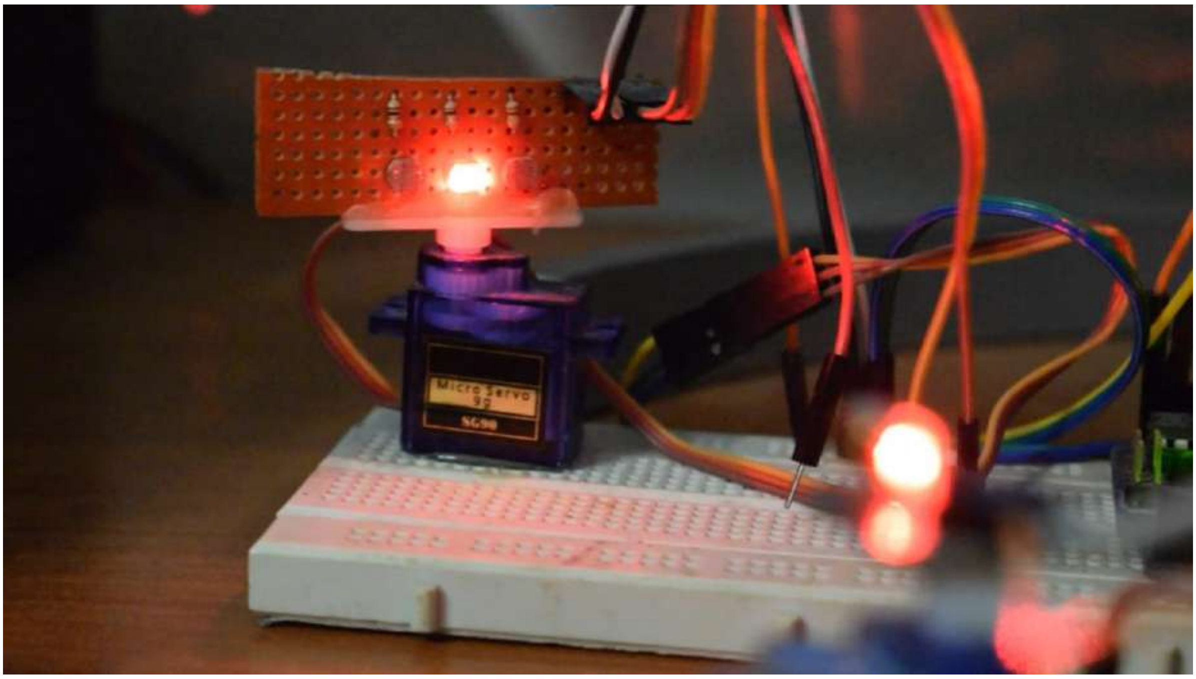
Stage 0: Pre-ARM Laser - This is the RESET stage.



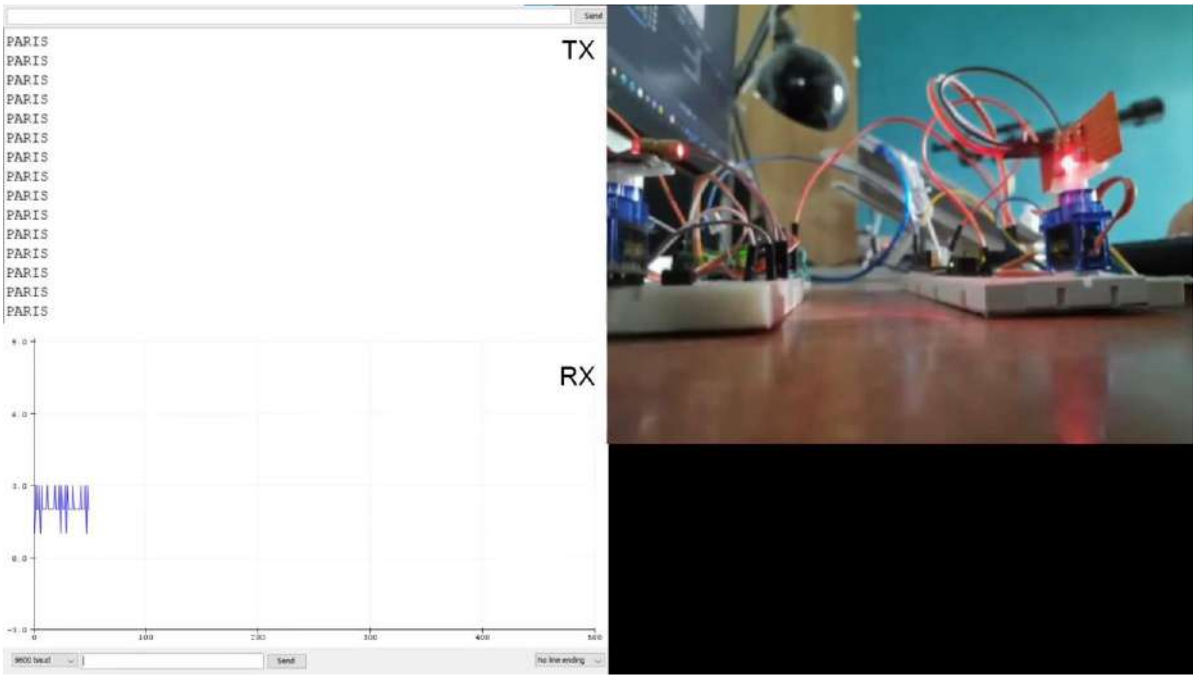
Stage 1: Acquisition- waiting for receiver to come into LOS and acquire it.



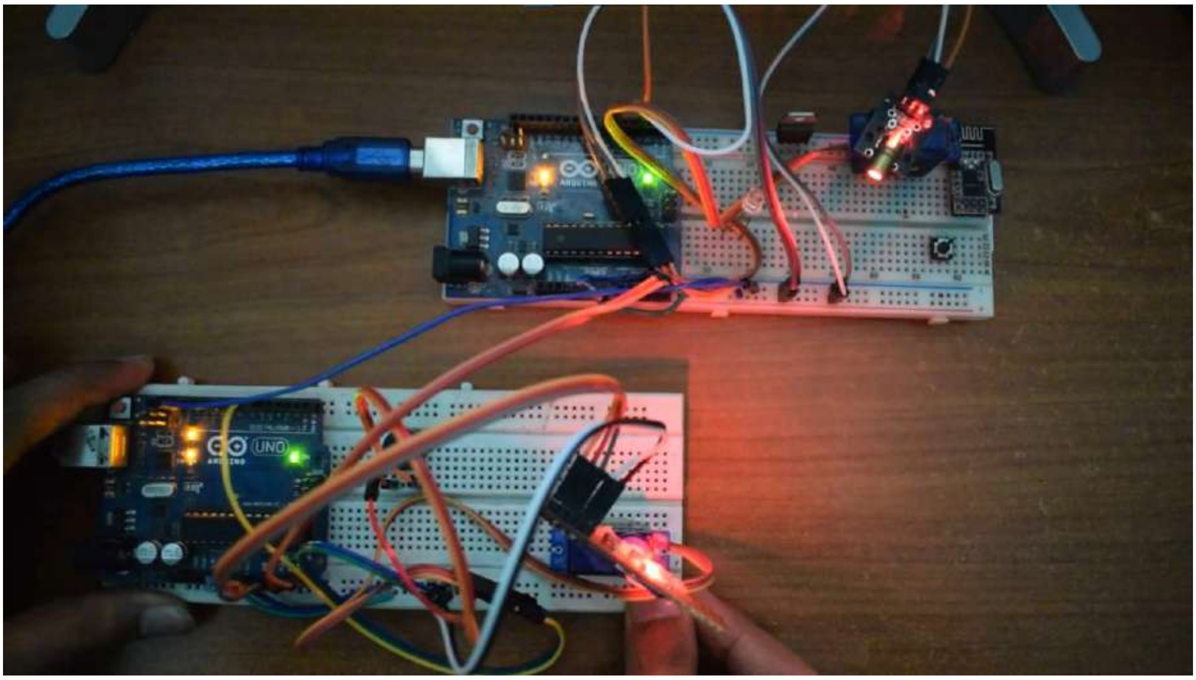
Stage 2: Tracking - The receiver is acquired and then tracking is done.



Stage 3: Laser Communication - Now laser communication is done between TX and RX using Morse code and Direct ON/OFF Modulation Method.



Stage 4: LOSS OF SIGNAL - when the TX and RX go out of sight the laser resets it's position and goes back to pre-arm stage



4.5 PROJECT CODE

Transmitter Code:

```
/**@Nikhil Mishra */
/**
 *   SLAVE      MASTER
    Base1      Base2
    TX         RX
*/
#include <Servo.h>
#include <Wire.h>
#include <EasyTransferI2C.h>

//Settings
bool demo = true;
bool testing = false;
bool com_test = false;
bool speedtest = false;
bool input_msg = false;
bool datalog = true;
bool nbc = false;

//Timers
unsigned long ct;
unsigned long time;
unsigned long pt = 0;
unsigned long pt1 = 0;
unsigned long pt2 = 0;
unsigned long pt3 = 0;
unsigned long pt4 = 0;
unsigned long pt5 = 0;

int ldr_val;
int laserpin = 3;
int avg_ldr = 0;
int live_pos = 0;
int pos = 0;
int aq_pos = 0;
int aq_tracking;
int poss = 0;
const int servo_pin = 6;
```

```
float output_angle = 0;
float tracking_pos;
float prev_tx_pos = 0;
float cm_ldr_prev = 0;
float tx_del_angle = 0;
float rx_vel;
bool tracking = false;
bool aquired = false;
bool local_scan_reset = false;
bool right_approach_state = false;
bool left_approach_state = false;
bool los = false;
double weights = 0.1;
double ldr_dist = 0.006; //metres

//Data - I2C
EasyTransferI2C ETout, ETin;
#define I2C_SLAVE_ADDRESS 4
#define I2C_MASTER_ADDRESS 1

struct RECEIVE_DATA_STRUCTURE {
  int ldr_1;
  int ldr_2;
  int ldr_3;
};

struct SEND_DATA_STRUCTURE {
  int tx_servo;
};

RECEIVE_DATA_STRUCTURE ldr_data;
SEND_DATA_STRUCTURE tx_data;

//PID
double Input;
double Output;
double Setpoint = 250;
double Kp = 0.45, Kd = 0.25, Ki = 0;

//Servo Instance
Servo myservo;
float tx_pos = 0;
```

```
//Laser Comms
char ch;
int test_i = 0;
int char_i = 0;
int intensity = 0;
int dotLength = 50;
int dashLength = dotLength*3;
int dash_high_count = 0;
int dash_low_count = 0;
int appraoch_left = 50;
int appraoch_right = 130;

int j = 0;
bool dash_state = LOW;
bool dot_state = LOW;
bool dot_finished = false;
bool dash_finished = false;
bool word_silence = false;
bool com_link = false;

const char* MorseTable[] = {
        //Morse Code Binary Tree
        NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
        NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
        NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
        NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
        // space, !, ", #, $, %, &, '
        NULL, "-.-.--", ".-..-.", NULL, NULL, NULL, NULL, ".----.",
        // ( ) * + , - . /
        "-.--.", "-.--.-", NULL, ".-.-.", "--..--", "-....-",
   ".-.-.-", "-..-.",
        // 0 1 2 3 4 5 6 7
        "-----", ".----", "..---", "...--", "....-", ".....",
   "-....", "--...",
        // 8 9 : ; < = > ?
        "---..", "----.", "---...", "-.-.-.", NULL, "-...-", NULL,
   "..--..",
        // @ A B C D E F G
        ".--.-.", ".-", "-...", "-.-.", "-..", ".", "..-.", "--.",
        // H I J K L M N O
        "....", "..", ".---", "-.-", ".-..", "--", "-.", "---",
        // P Q R S T U V W
        ".--.", "--.-", ".-.", "...", "-", "..-", "...-", ".--",
```

```
        // X Y Z [ \ ] ^ _
        "-..-", "-.--", "--..", NULL, NULL, NULL, NULL, "..--.-",
        // ' a b c d e f g
        NULL, ".-", "-...", "-.-.", "-..", ".", "..-.", "--.",
        // h i j k l m n o
        "....", "..", ".---", "-.-", ".-..", "--", "-.", "---",
        // p q r s t u v w
        ".--.", "--.-", ".-.", "...", "-", "..-", "...-", ".--",
        // x y z { | } ~ DEL
        "-..-", "-.--", "--..", NULL, NULL, NULL, NULL, NULL,
        };

void setup() {
  Serial.begin(9600);
  Serial.println('TX ON');

  //i2C
  Wire.begin(I2C_SLAVE_ADDRESS); // join i2c bus with address #4

  //define handler function on receiving data
  Wire.onReceive(receiveEvent);
  ETin.begin(details(ldr_data), &Wire);
  ETout.begin(details(tx_data), &Wire);

  //Laser
  pinMode(laserpin, OUTPUT);

  //Setup Servo
  myservo.attach(servo_pin);

  //PID
  myPID.SetMode(AUTOMATIC);
  myPID.SetTunings(Kp, Ki, Kd);
  //buzz
  tx_pos = 90;
}

void loop() {
  //Set Modes
  modes();

  //I2C IN
  if (ETin.receiveData()) {
```

```
    //Data Logging
    if (datalog){
      Serial.print(millis());
      Serial.print(",");
      Serial.print(ldr_data.ldr_1);
      Serial.print(",");
      Serial.print(ldr_data.ldr_2);
      Serial.print(",");
      Serial.print(ldr_data.ldr_3);
      Serial.print(",");
      Serial.print(tx_pos);
      Serial.print(",");
      Serial.print(tx_del_angle);
      Serial.print(",");
      Serial.print(int(tracking));
      Serial.print(",");
      Serial.print(int(aquired));
      Serial.print(",");
      Serial.println(los);
    }
  }
  else {
    // Serial.println("I2C:0");
  }

  //I2C OUT
  tx_data.tx_servo = tx_pos;
  ETout.sendData(I2C_MASTER_ADDRESS);

  /* AQUISITION */
  if (!aquired && !tracking) {
    //Pre-Arm
    tx_pos = appraoch_right;
    myservo.write(tx_pos);
    digitalWrite(laserpin, HIGH);

    if (ldr_data.ldr_1 >= Setpoint && !right_approach_state) {
      right_approach_state = true;
    }
    else if (ldr_data.ldr_3 >= Setpoint && !left_approach_state) {
      left_approach_state = true;
    }
```

```
    if (ldr_data.ldr_2 >= Setpoint-10) {
      aquired = true;
      los = true;
      aq_pos = tx_pos;
      unsigned long aq_time = millis();
    }
  }


  /* Tracking - Spotlight Method*/
  if (aquired && los && !testing){
    int tx_ldr_angle = tan(ldr_dist/0.1)*(180/3.14); //tan(d2/d1);
    float cm_ldr = cm(ldr_data.ldr_1,ldr_data.ldr_2,
   ldr_data.ldr_3);
    float cm_error = 1000 - cm_ldr*1000;
    float tol = 50;
    float Kp = 0.002;

    // Serial.print(tx_pos);
    // Serial.print(",");
    // Serial.print(tx_del_angle);
    // Serial.print(",");
    // Serial.print(ldr_data.ldr_2);
    // Serial.print(",");
    // Serial.println((Kp*cm_error));

    //Tracking Status
    if (ldr_data.ldr_2 >= 230){
      tracking = true;
    }
    else if (ldr_data.ldr_2 < 200) {
      //Wait and see if it happens again in 1 second.
      time = millis();
      if (time % 1000 == 0) {
        //Lost Track reset from last point (aq_pos)
        aquired = false;
        tracking = false;
      }
    }

    //Tracking Update
    if(cm_error > -1*tol || cm_error < tol) {
      // -150 <--> 0 <--> 150
```

```
      los = true;
      prev_tx_pos = tx_pos;
      if (cm_error > 0){
        //move servo right (ldr facing away) -->

        tx_pos = tx_pos + (Kp*cm_error);
        myservo.write(tx_pos);
      }
      else if (cm_error < 0) {
        //move servo left (ldr facing away) -->

        tx_pos = tx_pos + (Kp*cm_error);
        myservo.write(tx_pos);
      }
    }
    tx_del_angle = abs(prev_tx_pos - tx_pos);

    //Tracking Check
    if (tx_pos < appraoch_left && right_approach_state){
      los = false;
      myservo.write(90);
    }
  }

  if ((los && tracking)|| com_test) {
    // laser_com();
  }
}

//The ETin.receiveData() function in the loop() will check
//if data is available at each iteration
void receiveEvent(int numBytes) {}

void laser_com(){
  if (!nbc){
    StartLaserCom(MorseTable[ch]);
    delay(dotLength*2);
  }else{
    StartLaserCom_nbc(MorseTable[ch]);
  }
}

void StartLaserCom(const char * morseCode)
```

```
{
  int i = 0;
  while(morseCode[i] != 0)
  {
    if(morseCode[i] == '.'){
      dot();
    } else if (morseCode[i] == '-'){
      dash();
    }
    i++;
  }
  com_link = true;
}

void StartLaserCom_nbc(const char * morseCode)
{
  if(morseCode[char_i] != 0 && !word_silence)
  {
    //increment only when the first batch of dots/dashes are
   complete.
    if (dot_finished || dash_finished){
      char_i++;
      dot_finished  = false;
      dash_finished = false;
    }

    if(morseCode[char_i] == '.' ){
      nbc_dot();
    } else if (morseCode[char_i] == '-'){
      nbc_dash();
    }

  }else{
    word_silence = true;
    char_i = 0;
  }

  ct = millis();
  //Word Silence
  if ((ct - pt4 >= dotLength*2) && word_silence){
    pt4 = ct;
    word_silence = false;
  }
```

```
}

void perform_speedtest(){
  char *myString = "PARIS";

  if (myString[test_i] == 'S'){
    ch = myString[test_i];
    Serial.print(ch);
    Serial.println();

    test_i = 0;
  }
  else if(myString[test_i] != 0 && myString[test_i] != 'S')
  {
    ch = myString[test_i];
    Serial.print(ch);
    test_i++;
  }
}

void dot(){
  digitalWrite(laserpin, HIGH);
  delay(dotLength);
  digitalWrite(laserpin, LOW);
  delay(dotLength);
}

void dash(){
  digitalWrite(laserpin, HIGH);
  delay(dashLength);
  digitalWrite(laserpin, LOW);
  delay(dotLength);
}

void modes(){
  if (!demo && (speedtest && input_msg || !speedtest &&
   !input_msg)){
    Serial.println("Error: Too many inputs");
    while(1);
  }
  else if (speedtest){
    perform_speedtest();
  }
```

```
  else if (input_msg){
    if (Serial.available()){
      ch = Serial.read();
    }
  }else if (testing){
    sweep_laser();
  }
  else if (com_test){
    myservo.write(110);
    perform_speedtest();
  }
}

float cm(float data1, float data2, float data3) {

  return (data1*0 + data2*1 + data3*2)/(data1 + data2 + data3);
}

float AQTrack(float left_lim, float right_lim) {
  //AQTrack
  //Part2: PID Tracker

  myPID.Compute();
  output_angle = map(Output, 0, 255, right_lim, left_lim);
  return output_angle;
}

void sweep_laser() {
  time = millis();
  int nwe_period = 1;

  if (time % nwe_period == 0) {

    int new_pos = ++poss;
    tx_pos = new_pos;
    myservo.write(new_pos);
    poss = new_pos;
    if (poss == 180){
      poss = 0;
    }
  }
}
```

Receiver Code:

```
/**
 *  MASTER    SLAVE
    Base2     Base1
    RX        TX
*/

#include <Servo.h>
#include <Wire.h>
#include <EasyTransferI2C.h>

EasyTransferI2C ETout, ETin;

const int ldr1 = 0;
const int ldr2 = 1;
const int ldr3 = 2;
int ldr1_val;
int ldr2_val;
int ldr3_val;
int buzzer = 3;
const int servo_pin = 6;

//I2C Data
struct SEND_DATA_STRUCTURE {
  int ldr_1;
  int ldr_2;
  int ldr_3;
};
SEND_DATA_STRUCTURE ldr_data;

struct RECIEVE_DATA_STRUCTURE {
  int tx_servo;
};
RECIEVE_DATA_STRUCTURE tx_data;

#define I2C_SLAVE_ADDRESS 4
#define I2C_MASTER_ADDRESS 1

//NRF
//Servo Instance
```

```
Servo myservo;
int rxservo_pos;

const char MorseTree[] = {'\0','E', 'T', 'I', 'A', 'N', 'M', 'S',
                          'U', 'R', 'W', 'D', 'K', 'G', 'O', 'H',
                          'V', 'F', 'U', 'L', 'A', 'P', 'J', 'B',
                          'X', 'C', 'Y', 'Z', 'Q', '\0','\0','5',
                          '4', '\0','3', '\0','\0','\0','2', '\0',
                          '\0','+', '\0','\0','\0','\0','1', '6',
                          '=', '/', '\0','\0','\0','(', '\0','7',
                          '\0','\0','\0','8', '\0','9', '0', '\0',
                          '\0','\0','\0','\0','\0','\0','\0','\0',
                          '\0','\0','\0','?', '_', '\0','\0','\0',
                          '\0','"', '\0','\0','.', '\0','\0','\0',
                          '\0','@', '\0','\0','\0','\0','\0','\0',
                          '-', '\0','\0','\0','\0','\0','\0','\0',
                          '\0',';', '!', '\0',')', '\0','\0','\0',
                          '\0','\0',',', '\0','\0','\0','\0',':',
                          '\0','\0','\0','\0','\0','\0','\0'
                         };

int ctrHigh = 0;
int ctrLow = 0;
int codePtr = 0;
int dotLen = 400;

void receiveEvent(int numBytes) {}

void setup() {

  Serial.begin(9600);
  Wire.begin(I2C_MASTER_ADDRESS);

  ETout.begin(details(ldr_data), &Wire);
  ETin.begin(details(tx_data), &Wire);
  Wire.onReceive(receive);
  myservo.attach(servo_pin);
}

unsigned long time;
int loopcount = 0;
int avg_rx_pos = 0;
```

```
void loop() {
  //Radio TX Mode
  ldr1_val = map(analogRead(ldr1), 0, 1024, 0, 255);
  ldr2_val = map(analogRead(ldr2), 0, 1024, 0, 255);
  ldr3_val = map(analogRead(ldr3), 0, 1024, 0, 255);

  ldr_data.ldr_1 = ldr1_val;
  ldr_data.ldr_2 = ldr2_val;
  ldr_data.ldr_3 = ldr3_val;

  if (ETin.receiveData()) {
    rxservo_pos = tx_data.tx_servo;
    myservo.write(rxservo_pos);
  }

  ETout.sendData(I2C_SLAVE_ADDRESS);

   Serial.print(ldr1_val);
   Serial.print(",");
   Serial.println(ldr2_val);
   Serial.print(",");
   Serial.println(ldr3_val);

  // recieve_com(ldr2_val);
}

void receive(int numBytes){}

void sweep_laser() {
  int pos;

  for (pos = 0; pos <= 180; pos += 1) {
    myservo.write(pos);
    delay(15);
  }
  for (pos = 180; pos >= 0; pos -= 1) {
    myservo.write(pos);
    delay(15);
  }
}

void recieve_com(int val) {
  val = analogRead(4);
```

```
  if (val >= 150)
  {
    ctrHigh++;
    ctrLow = 0;
    digitalWrite(13, HIGH);
    tone(buzzer, 1000);
  } else {
    ctrLow++;
    if ((ctrHigh >= dotLen) && (ctrHigh < dotLen*2)) {
      Serial.print(".");
      codePtr = (2*codePtr) + 1;
    } else if (ctrHigh >= dotLen * 2) {
      Serial.print("-");
      codePtr = (2*codePtr) + 2;
    } else {
      if(ctrLow == dotLen*2){
        Serial.print(MorseTree[codePtr]);
        codePtr = 0;
      }
    }

    ctrHigh = 0;
    digitalWrite(13, LOW);
    noTone(buzzer);
  }
}
```

4.6 RESULTS

4.6.1 EXPERIMENTAL RESULTS

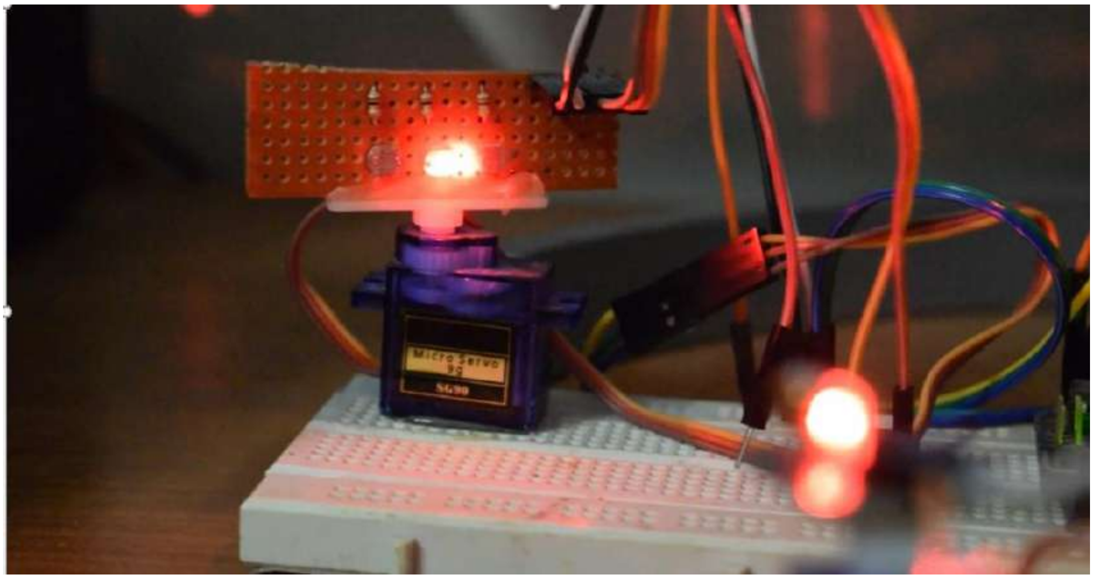


Fig 4.9 Receiver being Tracked by Transmitter

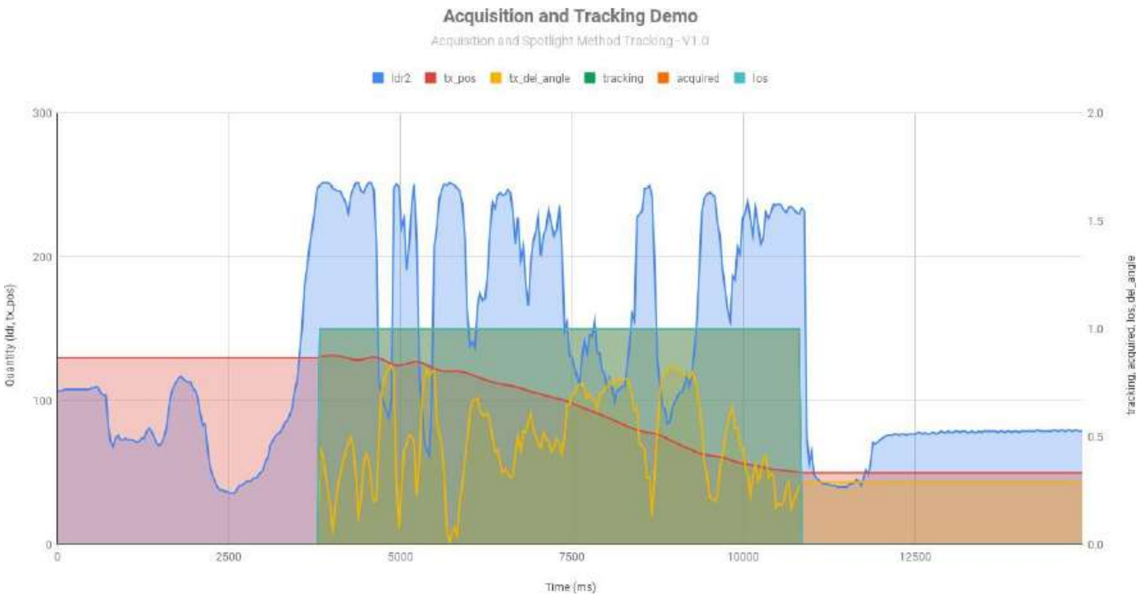


Fig 4.10 Acquisition and Tracking Demo Graph

4.6.2 PERFORMANCE

Parameter	Quantity
Acquisition and Tracking	
TX Angular Speed	1 - 4 deg/sec
Laser Communication	
Bit Rate	6 - 10 bps
Character Rate	3 - 4 cps
Word Rate	20 - 26 wpm

Table 8: Performance Metrics

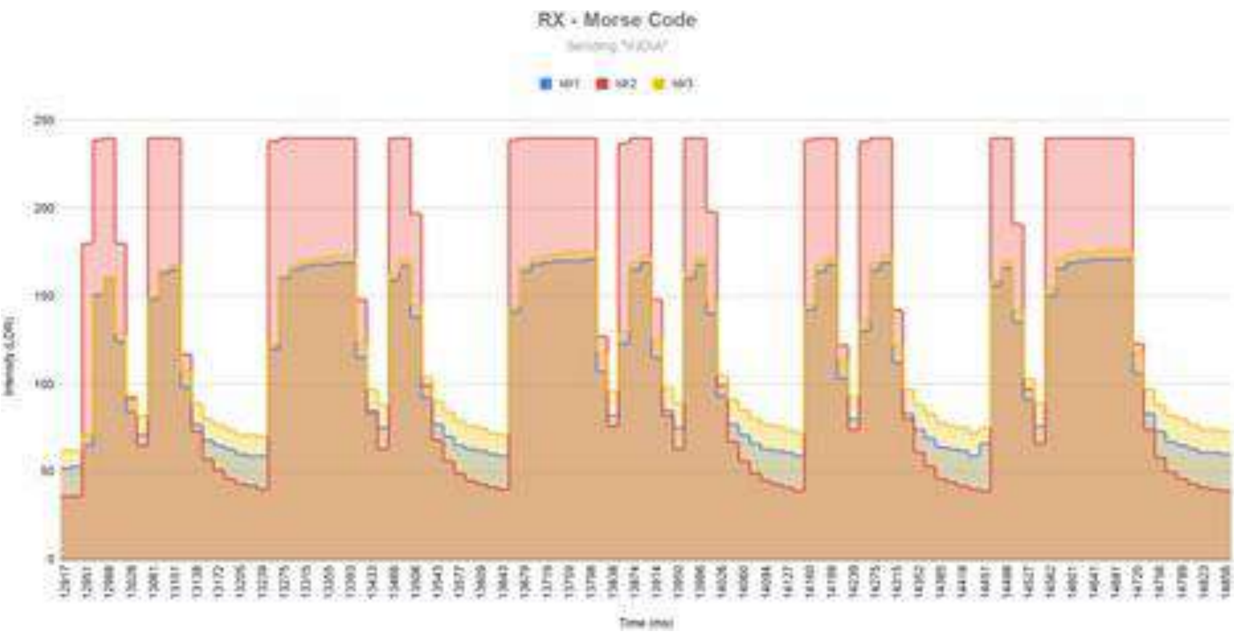


Fig 4.11 RX - Morse Coding

Fig 4.12 Data Rate - Laser Communication

CHAPTER 5

PROJECT TESTING

5.1 OVERVIEW OF TESTING METHODS

The following tests were conducted for the project:

1. Functional Test
2. Program Test
3. Speed Test
4. Debug Test
5. Error Testing

1. Functional Tests

Functional tests will test the functionality of various modules in the project, such as, LDR, Servos, Laser and others. The below graph represents the Intensity Characteristics of LASER Diode.

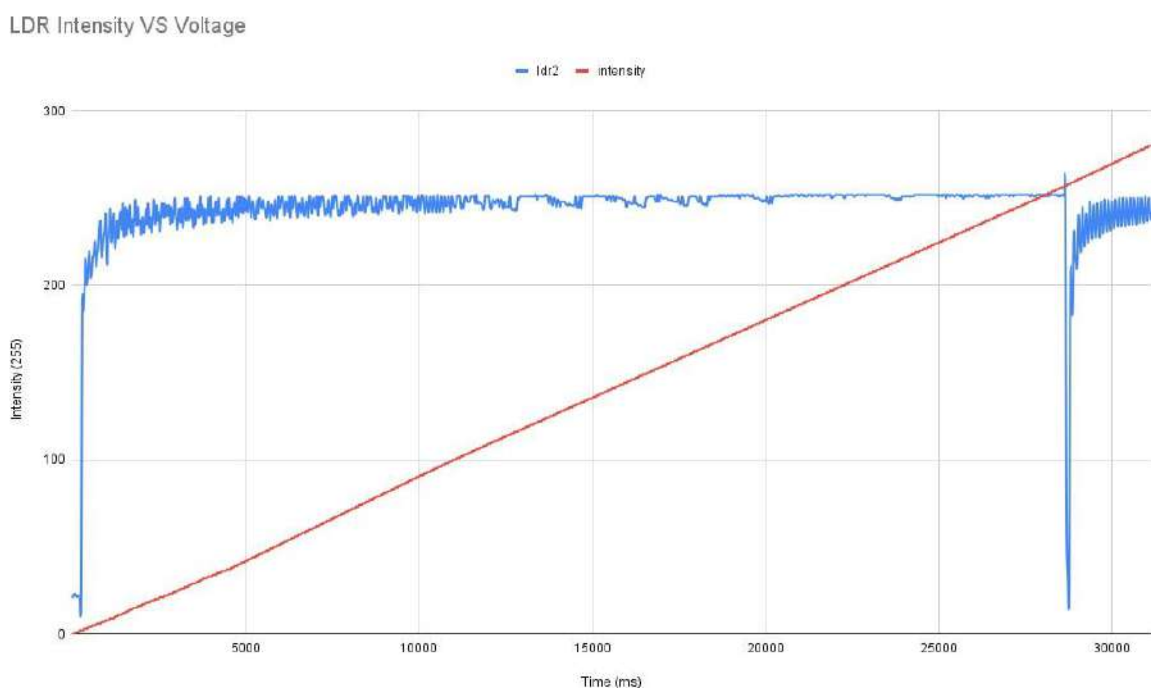


Fig 5.1 LDR Intensity graph

2. Program Tests

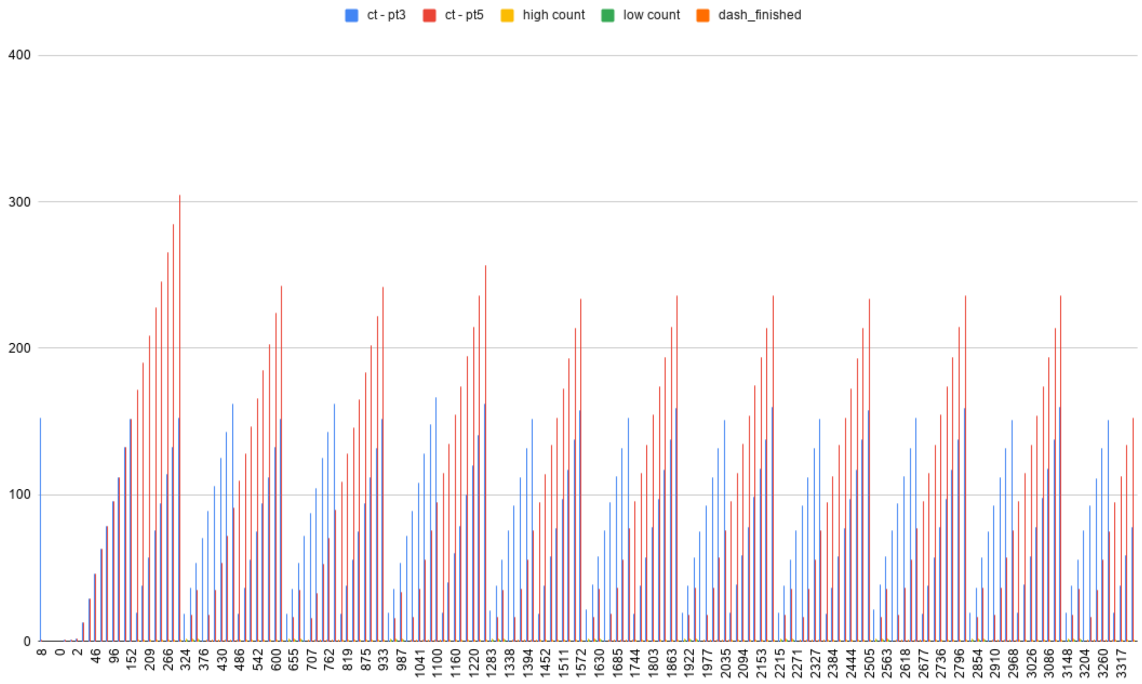


Fig 5.2 program test

3. Speed Test

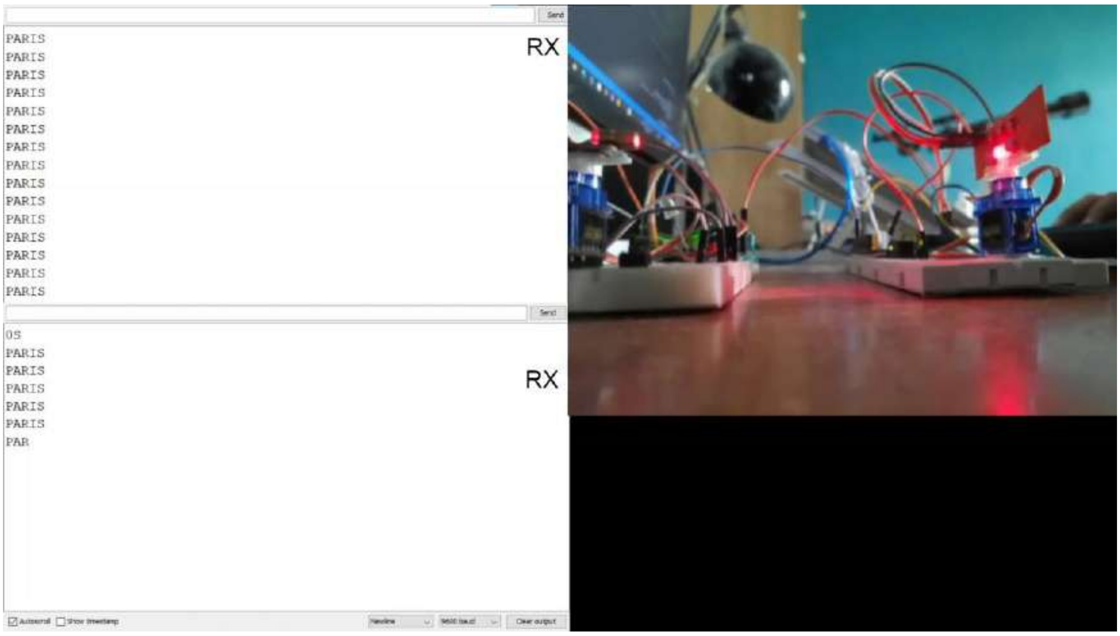


Fig 5.3 Speed test of morse code(WPM)

4. Debug Test:

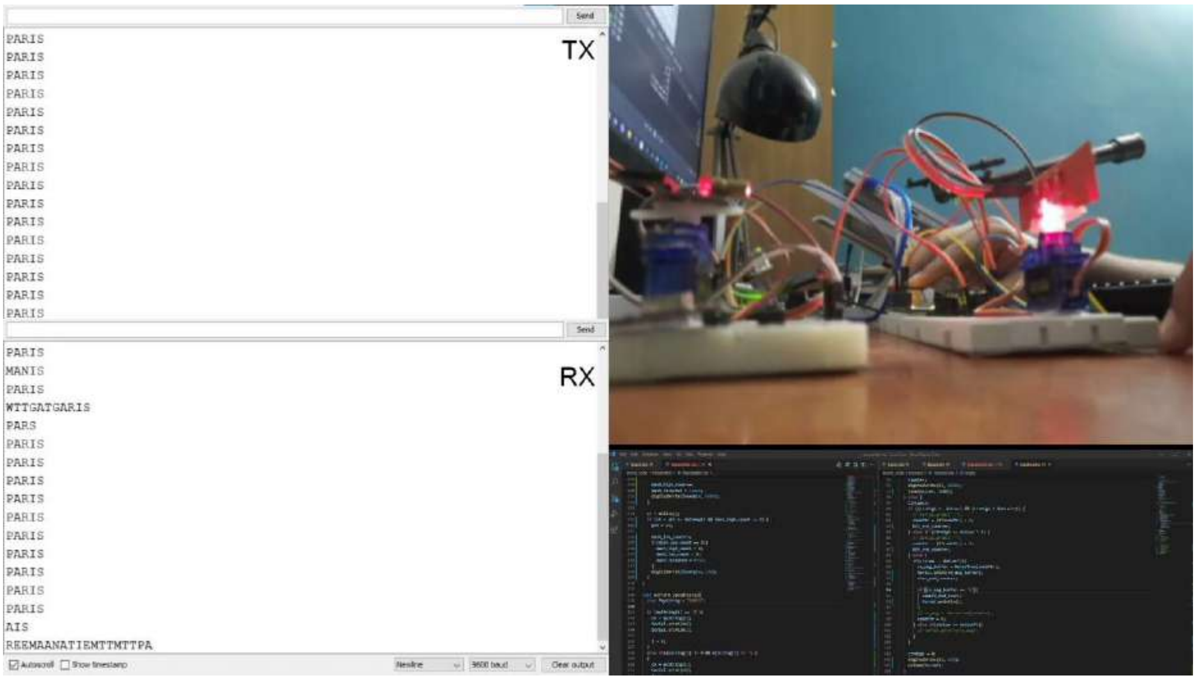


Fig 5.4 code debugging

5. Error Testing

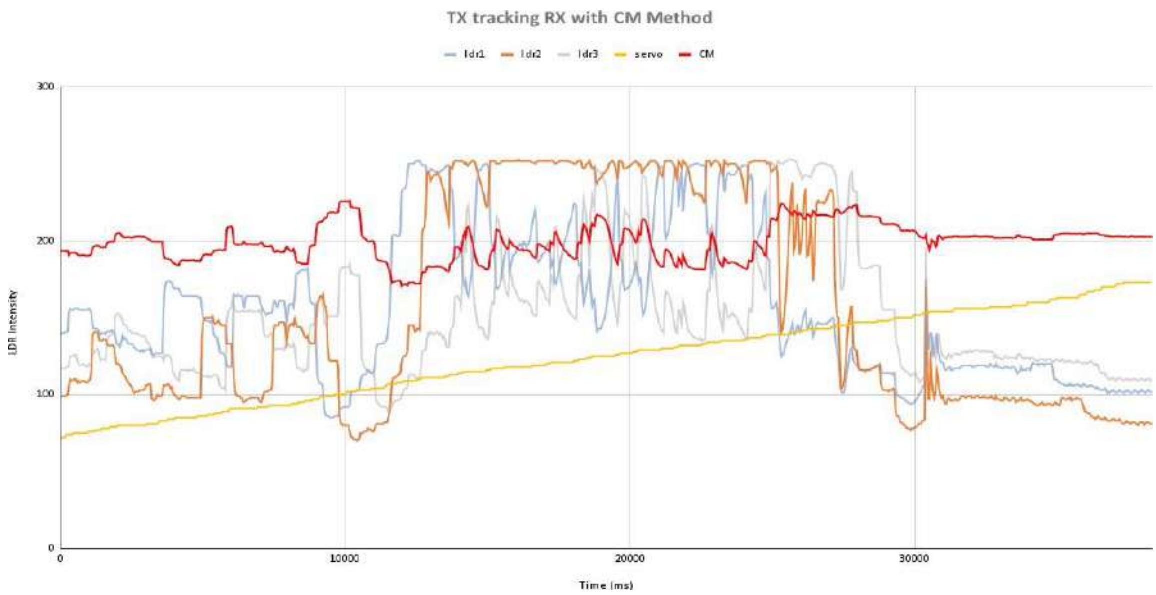


Fig 5.5 Tracking using CM method

The project testing was done in five parts:

1. **Acquisition** - Here how well the Tx is acquiring is tested by developing and testing multiple algorithms crossing down on which gives the better result.
2. **Tracking** - Here the ability of Tx and Rx to be in continuous LOS is tested by reading how uninterrupted the transmitted data is being received.
3. **Acquisition and Tracking** - Here the combination of both the methods is tested and how well they both work together for maintaining is LOS analysed.
4. **Laser Communication** - Here the implementation of laser comms using morse code is done and the data transmission and reception is tested by analysing the data error data speed in BPS and WPS.
5. **Complete Testing in 4 Stages** - a complete run of all the stages is done and performance is evaluated.

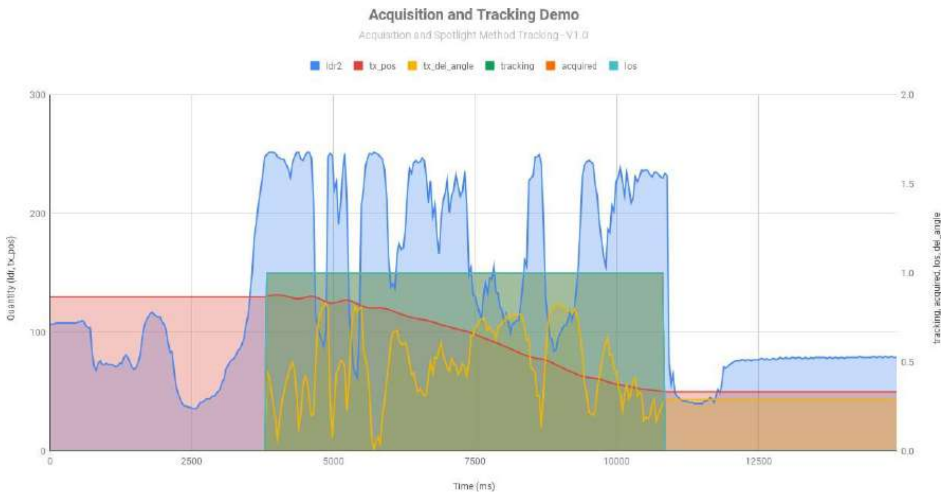


Fig 5.6 AQ and Tracking analysis

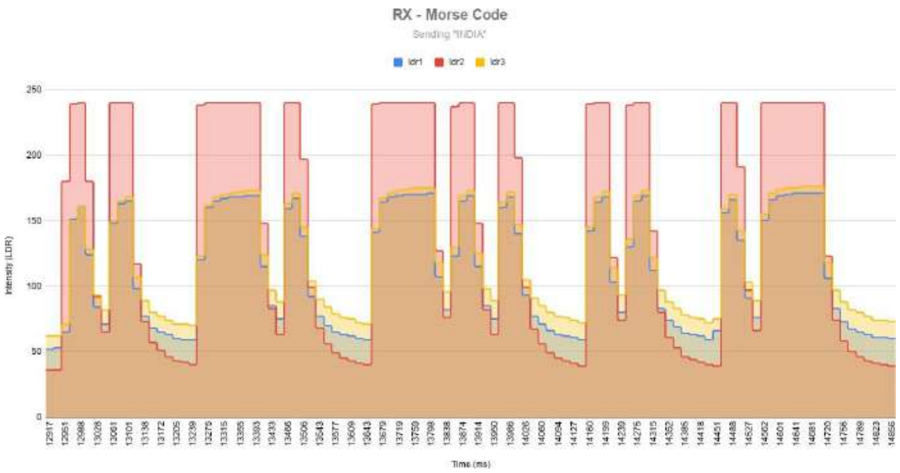


Fig 5.7 Morse code analysis

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

Laser crosslinks offer several benefits to current and near-future civil, commercial, and security sectors; the most obvious being highly secure, low error, high throughput data communications. Due to the protracted pace of laser crosslink development, melded with the lengthy lead-times for satcom development, compounded by the prolonged government procurement process, makes the time right for inclusive action now. Laser crosslinks technology is sufficiently mature to place payloads of small size, weight, and power on to-be-launched vehicles and follow in short order with the terrestrial legs—whether fixed, at sea, or in the air. The demand for data—raw facts—and its successful evolution to information—arranged in context—is growing, arguably at an exponential rate. The U.S. Department of Defense’s demand for data, information, and intelligence is exceeding its satellite communications capacity.

6.2 FUTURE ENHANCEMENT

- 3D Tracking Method for real time systems
- Develop Networking Topology
- Develop Routing Algorithm
- Build a Scalable Solution

PUBLICATION

Submitted Paper in the Conference ICISECT - 21 with Paper ID (ICISECT21-0062) and got Acceptance for the Paper.

REFERENCES

1. Klein, J.J. Space Warfare: Strategy. In Principles and Policy; Routledge: New York, NY, USA, 2006.

2. McKinney, M.M. Transformational Satellite (TSAT) Communications Systems: Falling Short on Delivering Advanced Capabilities and Bandwidth to Ground-Based Users Air University Press, Maxwell Air Force Base. 2007. Available online: https://www.semanticscholar.org/paper/Transformational-Satellite-(TSAT)-Communications-on-McKinney/9b46a7134f5 6c1605ed9aed842ed396b0453084b (accessed on 4 June 2020).

3. Baird, D. NASA Laser Communication Payload Undergoing Integration and Testing. Available online: https://www.nasa.gov/feature/ goddard/2017/nasa-laser-communication-payload-undergoing-integration-and-testing (accessed on 13 August 2019).

4. Guilhot, D.; Ribes-Pleguezuelo, P. Laser Technology in Photonic Applications for Space. Instruments 2019, 3, 50. [CrossRef]

5. Ribes-Pleguezuelo, P.; Guilhot, D.; Gilaberte Basset, M.; Beckert, E.; Eberhardt, R.; Tünnermann, A. Insights of the Qualified ExoMars Laser and Mechanical Considerations of Its Assembly Process. Instruments 2019, 3, 25. [CrossRef]

6. Clarke, A. Earthlight, Muller (UK)/Ballantine Books (US). 1955. Available online: https://www.goodreads.com/book/show/170574 8.Earthlight (accessed on 4 June 2020).

7. Kazemi, A.A.; Panahi, A. Space-Based Laser Systems for Inter-Satellite Communications. In Proceedings of the Photonic Applications for Aerospace, Transportation, and Harsh Environment III, Baltimore, MA, USA, 10 May 2012; SPIE: Bellingham, WA, USA, 2012.

8. Wilson, K.E. An Overview of the GOLD Experiment Between the ETS-VI Satellite and the Table Mountain Facility, TDA Progress Report. Available online: https://ipnpr.jpl.nasa.gov/progress_report/42-124/124I.pdf (accessed on 8 January 2020).

9. Strugarek, D.; So´snica, K.; Arnold, D.; Jäggi, A.; Zajdel, R.; Bury, G.; Dro ˙zd ˙zewski, M. Determination of Global Geodetic Parameters Using Satellite Laser Ranging Measurements to Sentinel-3 Satellites. Remote Sens. 2019, 11, 2282. [CrossRef]

10. Pearlman, M.; Arnold, D.; Davis, M.; Barlier, F.; Biancale, R.; Vasiliev, V.; Ciufolini, I.; Paolozzi, A.; Pavlis, E.; So´snica, K.; et al. Laser geodetic satellites: A high accuracy scientific tool. J. Geod. 2019, 93, 2181–2194. [CrossRef]

11. Drinkwater, M.R.; Haagmans, R.; Muzi, D.; Popescu, A.; Floberghagen, R.; Kern, M.; Fehringer, M. The GOCE gravity mission: ESA’s first core explorer. In Proceedings of the

3rd International GOCE User Workshop, Frascati, Italy, 6–8 November 2006; pp. 1–3. ISBN 92-9092-938-3.
12. Tapley, B.D.; Bettadpur, S.; Ries, J.C.; Thompson, P.F.; Watkins, M.M. GRACE Measurements of Mass Variability in the Earth System. Science 2004, 305, 503–505. [CrossRef].

A

MAJOR PROJECT REPORT

On

DRUNK DRIVING DETECTION USING CAR IGNITION LOCKING

Submitted By

1) Mr.M.Sharath Chandra(17K81A04M6) 2) Ms.A.Devi Sree (17K81A04J1)

3) Mr.P.Sri Datta(18K85A0442)

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Mrs.Ch.Swathi, M.Tech

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

AN UGC AUTONOMOUS INSTITUTE

Dhulapally, Secunderabad-500 100

NBA & NAAC A+ Accredited

Department Of Electronics & Communication Engineering

CERTIFICATE

JUNE 2021

This is to certify that the project entitled –DRUNK DRIVING DETECTION USING CAR IGNITION LOCKING‖, is being submitted by **1.Ms.A.DeviSree(17K81A04J1), 2.Mr.M.Sharath Chandra(17K81A04M6), 3.Mr.P.Sri Datta(18K85A0442)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

INTERNAL GUIDE	HEAD OF THE DEPARTMENT
Mrs. CH. SWATHI	Dr. B. HARI KRISHNA
Assistant Professor	Professor

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT M SHARATH CHANDRA WITH ROLL NO.17K81A04M6, A DEVI SREE WITH ROLL NO.17K81A04J1, P SRI DATTA WITH ROLL NO.18K85A0442, OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "DRUNK DRIVING DETECTION WITH CAR IGNITION LOCKING" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email :contact@lasyainfotech.com,ov@lasyainfotech.com
Website :www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of **Bachelor of Technology** in Department of ELECTRONICS AND COMMUNICATION ENGINEERING, session: <2017 – 2021>, St. Martin‘s Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled –DRUNK DRIVING DETECTION USING CAR IGNITION LOCKING‖ is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

1) Mr.M.Sharath Chandra(17K81A04M6)

2) Ms.A.Devi Sree(17K81A04J1)

3) Mr.Sri Datta(18K85A0442)

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr.B.HARI KRISHNA**, **Head & Professor**, Dept. of Electronics and Communication Engineering for his advice, supervision and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in our memory for years to come.

We would like to express our gratitude to our **Dept. project coordinator Dr.A.Anand Kumar, Professor** for his involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time to come.

We sincerely thank our project internal guide, **Mrs.CH.Swathi, Assistant Professor,** Department of ECE and our class project coordinator **Ms.P.Pushpa, Assistant Professor,** Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

1. M SHARATH CHANDRA (17K81A04M6)
2. A DEVI SREE (17K81A04J1)
3. P SRI DATTA (18K85A0442)

ABSTRACT

This study proposed an efficient technique for eradicating the upsurge in the number of cases of roads Accidents caused by excessive intake of alcohol. This study developed a prototype alcohol detection and Engine locking system by using an Arduino Uno microcontroller interfaced with an alcohol sensor along with an LCD screen and a DC motor to demonstrate the concept. The system uses MQ-3 alcohol sensor to Continuously monitor the blood alcohol content (BAC) to detect the existence of liquor in the exhalation of a driver. By placing the sensor on the steering wheel, our system has the capacity to continuously check alcohol level from the driver‘s breath. The ignition will fail to start if the sensors detects content of alcohol in the driver‘s breathe. In case the driver got drunk while driving, the sensor will still detect alcohol in his breath and stop the engine so that the car would not accelerate any further and the driver can park by the roadside. Results from testing the proposed system adequately matched the requirements for starting a car‘s engine once the level of alcohol detected in the breath of the driver is higher than the prescribed level permissible by law.

TABLEOFCONTENTS

LIST OF TABLES

LIST OF FIGURES

Glossary of terms

***Approximation of Blood Alcohol Concentration (BAC)* :**

There are far variables involved to accurately estimate or predict a person‘s BAC simply based on gender. The data is so important for us to build an accurate project design. Blood alcohol content (BAC), also called blood alcohol concentration or blood alcohol level, is a measurement of alcohol intoxication used for legal or medical purposes.[1] A BAC of 0.10 (0.10% or one tenth of one percent) means that there is 0.10 g of alcohol for every 100 ml of blood, which is the same as 21.7 ml.

MQ-3 Alcohol Sensor Unit :

Gas Sensor(MQ3) module is useful for gas leakage detection (in home and industry). It is suitable for detecting Alcohol, Benzine, CH4, Hexane, LPG, CO. Due to its high sensitivity and fast response time, measurements can be taken as soon as possible. The sensitivity of the sensor can be adjusted by using the potentiometer. The output voltage from the Gas sensor increases when the concentration of gas increases. Sensitivity can be adjusted by varying the potentiometer.

Potentiometer:

A potentiometer sensor measures the distance or displacement of an object in a linear or rotary motion and converts it into an electrical signal. Our potentiometer sensors are qualified for various applications and industries. Custom linear and rotary potentiometers are designed for harsh environment aerospace applications including fin actuation, flight controls, and valve position measurement.

LCD display:

A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in color or monochrome.

Liquid crystal display technology works by blocking light. Specifically, an LCD is made of two pieces of polarized glass (also called substrate) that contain a liquid crystal material between them. A backlight creates light that passes through the first substrate. At the same time, electrical currents cause the liquid crystal molecules to align to allow varying levels of light to pass through to the second substrate and create the colors and images that you see.

Buzzer:

A buzzer or beeper is an audio signalling device,[1] which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

CHAPTER-1
INTRODUCTION

1.1 OVERVIEW OF THE PROJECT:

Driving while drunk is hazardous and drivers with high Blood Alcohol Content (BAC) are at expanded danger of auto crashes, roadway wounds and vehicular passings. Anticipation measures assessed incorporate permit suspension or disavowal, appropriating or seizing vehicle plates, implementing open holder bans, expanding fine punishments, imprison, ordering instruction for youth and bringing down legitimate BAC's. Despite the fact that these much obstacles made by experts to drunken drive, it is as yet proceeding with like serial scenes. In that capacity there is no viable instrument to reduce this. Here, this process have intended to plan a Drunk driving detection, which is integrated with the directing wheel. This framework is meant for making vehicle driving more secure than previously and shield the mishaps from happening due to the liquor utilization of the driver. The individual when he is at vehicle, this is necessary to infer the driver's condition continuously and here this work proposed the detection of alcohol utilizing alcohol sensor associated with Arduino. Alcohol sensor is installed on the steering of the car, with the end goal that when the level of liquor crosses a reasonable farthest point, where the start of vehicle will kill and the motor will stop. The Arduino always uses the alcohol sensor information to check drunk driving and works a bolt on the vehicle motor to stop the engine. Key Words—Alcohol Detection, Car Ignition Locking, Alcohol Sensor, Arduino Uno, Lcd Display.

Drink & drive is a leading cause of road accidents. Detecting drunk driving requires stopping vehicles and manually scanning drivers by using breath analyzers. Well here we propose a system that allows to detect drunk driving in the vehicle itself. Our system uses alcohol sensor with raspberry pi along with a GSM modem for sms notification and LCD display and also a motor to demonstrate as vehicle motor. Now our system constantly checks for driver alcohol content. The system first allows the user to configure admin numbers into the system. And if the system detects driver is drunk above permissible limit, the sensor inputs trigger the processor about the issue by providing respective voltage. Now the system sends sms notifications to both the registered users/authorities to inform about the issue. Also the system stops the motor to demonstrate as engine locking of the vehicle. Thus the system detects and prevents drunk driving incidents automatically.

1.2 OBJECTIVES OF THE STUDY:

Drink and drive issue have become solemnly that needs immediate attention. This is due to drivers‘ ignorance towards road rules and regulations and their selfish attitude that caused loss of innocent lives. Although previously there is a drunk detecting mechanism using breathalyzer but it isn‘t suitable for current fast-paced lifestyle. Therefore, to overcome these issues, this system is proposed. This system is fixed on vehicle‘s steering to measure alcohol concentration reading using MQ-3 sensor from the driver‘s exhaled breath. If the driver found to be drunk beyond the threshold level of 400 ppm, then ignition lock is activated and the car engine does not start till alcohol concentration falls to a safe level. Or, if the driver consumes an alcoholic drink while driving, upon exceeding permissible limit, the car slows down till it stops. Then, the location of the vehicle is tracked and sent as Google Map integrated link via text message to authorized unit. Simultaneously, the car buzzer goes off while the car slows down so that surrounding road users are aware of the driver‘s condition and drives at a distance. The proposed detection system is highly potential to be implemented for reducing the drunk and drive accidents.

The target of this project is to give a idea and inventive method for avoiding drunken driving of a Motorcar by locking the car. Likewise to permit a man who is not alcoholic to drive a same Motorcar. To broaden this thought with more innovative headways and make it accessible in a financially effective way. We need to plan a sort of framework which can recognize the alcohol content in the cars to prevent the conduct of alcoholic driving. The framework comprises of these two sections: International Journal of Pure and Applied Mathematics Special Issue 3000 1. Sensor Part- - used to identify the centralization of alcohol all around and send the concentration as voltage signals to the accompanying part. 2. Display Part—used to get the prepared signal and demonstrate the information to users in LCD. The sensor will be fixed close to the driver's seat. The driver should breath to the system before the individual begins the car. On the off chance that the alcohol level identified is underneath the permissible standard, the car can be started regularly. In the event that the alcohol level is over the suitable standard, the framework will send caution to the driver through LCD display. The framework ought to be protected, delicate, exact, advantageous and cheap. This sort of framework can be fixed on each car to guarantee the driver's driving security. Drunk driving detection using car ignition locking project have primary targets which need to outline the framework with the ability that gives notification to drunken driver before driving the vehicles. The project is to set up intelligent innovations for vehicles to produce alert as the notice and message will be shown in LCD display as drunken driver. Another objective is to utilize alcohol sensor as the primary sensor to sense the presence of alcohol gas noticeable all around. Essentially, the alcohol gas was identified from drunken driver after the driver entered the vehicle.

1.3 SCOPE OF THE STUDY:

- This project is one of the important Sensor based project.
- It is designed for the safety of the people.
- Alcohol Detection with Vehicle Controlling project helps to control the vehicle in case the driver has consumed the alcohol. An alcohol breath analyzer (MQ3 sensor) should be fitted/ installed inside the vehicle.
- The main goal of the project is to avoid the road accidents by the people who drive the vehicles after intake of alcohol.
- Its also helps in detecting the people who were drunk and do not allowing them to drive the vehicle.
- Its helpful to the traffic police in monitoring drunken drive cases.

The target of this project is to give a idea and inventive method for avoiding drunken driving of a car by locking the ignition.

Here, is the framework comprises of these two sections:

1. Sensor Part- - used to identify the centralization of alcohol all around and send the concentration as voltage signals to the accompanying part.

2. Display Part—used to get the prepared

signal and demonstrate the information to users in LCD.

1.4 MATERIAL REQUIREMENT:

1.4.1 HARDWARE REQUIREMENT:

- Arduino uno
- LCD
- Power Supply
- MQ3 Sensor
- Buzzer
- Ignition System
- DC Motor

1.4.2 SOFTWARE REQUIREMENT:

- Proteus software
- Arduino software

1.5 PROCUMENT OF EQUIPMENT:

About hardware pursuing…………………

Assembling it…………….

Connecting to Arduino…………

Coding is done in ide…………

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH AREA:

To cope with helmet negligence and alcohol detection simultaneous the writer proposed a system which is very complicated and use of P89V57RD2 microcontroller makes it highly expensive also this system can only be equipped with 2 wheelers whereas ,Aurdino uno microcontroller is economical as well ascan be equipped with any class of vehicle making it more authentic and successful.

To prevent the mishap of drunken driving writer have used PIC16F877A microcontroller which is an outdated system and expensive one also which restrains its use to only certain class of society whereas we are using Arduino and Uno microcontroller which is advanced as well as economical.

Worrying about the drunken driving the writer suggests the system to overcome the issue but using mQ2 alcohol sensor has come flames .MQ2 alcohol sensor is not authentic and raises the chance of false alarm while we have used MQ3 which is highly authentic.

This study developed a prototype alcohol detection and engine locking system by using an Arduino Uno microcontroller interfaced with an alcohol sensor along with an LCD screen and a DC motor to demonstrate the concept. The system uses MQ-3 alcohol sensor to continuously monitor the blood alcohol content (BAC) to detect the existence of liquor in the exhalation of a driver. By placing the sensor on the steering wheel, our system has the capacity to continuously check alcohol level from the driver‘s breath. The ignition will fail to start if the sensors detects content of alcohol in the driver‘s breath. In case the driver got drunk while driving, the sensor will still detect alcohol in his breath and stop the engine so that the car would not accelerate any further and the driver can park by the roadside

.

2.2 REVIEW ON RELATED LITERATURE:

Currently reported by the World Health Organization (WHO), Malaysia lies at number 59 among 183 countries worldwide for road traffic accidents, stands with a percentage of 24.51

death rates by both male and female due to road accidents; stands medium high level. Apparently, death rates on Malaysian roads is third highest globally, more than China and India reads the headline of an article

[1]. This data is consistent with the data as provided by Malaysian Institute Road Safety Research (MIROS), whereby, among eight common causes of a vehicle accident in Malaysia, drunk and driving cases is the second in the list and second highest in the world with highest amount of lives lost. According to MIROS report in year 2012 [2], 23.3% contribute to drunk and driving whereby 13 times more likely to cause an accident. Based on these statistics motivates the needs to reduce and prevent such road accidents. The old-fashioned method used by officers to detect alcohol in the driver is by using Breathalyzer. Even though it has proved its function, however this method is not efficient. Nevertheless, it is impossible to do road block all the time and check driver of each car using Breathalyzer [3].

Breathalyzer is a device to check alcohol presence in the driver‘s breath by making them blow into it. This method is not as practical as it can cause congestion and traffic during peak days. In addition, there are chances of the driver to bribe officers just to avoid being summoned. If they escape this road block, and continue to drive in a drunken state, they are risking every other road users. This project has overcome previous paper limitations as a part and parcel of the system. As proposed in [4, 5], the system uses MQ-2 sensors to detect the presence of alcohol in the driver‘s breath. MQ-2 is designed to test for combustible gases such as LPG [6]. It can be tested for alcohol too, but MQ-3 sensor [7] is specially made to test for alcohol vapor in which it has a more accurate reading and higher sensitivity than MQ-2. Then, another method that was used to detect presence of alcohol is from driver‘s sweat using the IR sensor [8] whereby neither using the air-conditioner nor opening the window will crash the system functionality totally. And besides, every human‘s perspiration rate differs when consuming alcohol. A lack of accuracy will lead to poor or no result of the presence of alcohol in driver‘s sweat.

Other than that, one of the paper has used mobile phone [9] which has an accelerometer and an orientation sensor whereby the mobile phone computes accelerations based on sensor readings and compares it with typical drunk driving patterns extracted from real driving tests through acceleration of vehicles related to lane position maintenance problems such as weaving, drifting, swerving, turning abruptly or with a wide radius and speed control problems like accelerating or decelerating suddenly, braking erratically and stopping abruptly. The loophole in this system is that if kids‘ play with this phone or it gets damaged due to fall as it is portable, then the system fails. In spite of that, paper [10] made a system that has only alert system to prevent car theft by having password access, but it can be further enhanced by having an alcohol detection as well. Although the paper achieved its objective, but with alcohol detection the functionality of the system will be enhanced as well as reduce drunk and drive cases on the road.

Furthermore, in [11], the system has everything except for buzzer and led. It might seem that this paper lacks of something minor, but it does make some effect on the system. As such, while on the road, if the driver is consuming alcohol and it is within the range, any moment, it can go over the threshold level. Led light is the indicator for the driver‘s soberness in a car which can be seen by others road vehicle, and buzzer is necessary to alert any road user that the driver is not conscious enough to drive his vehicle. So, lack of this does make a little impact which can cost a life. Whilst in [12], there is no tracking system or any alert system except alcohol detection, which is not enough to take any further action. Although, the car does not start up upon detecting alcohol in the driver‘s breath, this system was not tested for the condition whereby a driver gets drunk while driving. And another important point would be the system was tested with butane from lighter than ethanol.

In [13], the paper only detects alcohol presence and locks the car upon hitting threshold level but no alert system to inform authorities. Not forgetting in paper [14], the system only detects alcohol presence and rings an alarm, but lacks GSM and GPS which is more important to alert the next of kin. Another paper had used GPS only, excluding GSM [15] which will only detect the location of the vehicle, but will not alert next of kin or police office for further action. Yet, a paper was vice versa whereby it has GSM instead of GPS [16] whereby the police cannot trace the car location and either help the driver or take legal action. Apart from that in [17-19], the system only has buzzer as an alarm system which is insufficient as it lacks of tracking system and presence of alarm inside the car can cause panic attacks and worsen the situation.

Next, in [20-22] the proposed system does not have a GPS tracker or a GSM module that can track the location of the vehicle and notify authorized people via SMS. This limitation can only detect alcohol presence in the driver‘s breath and stops the engine from running, but he will not be prosecuted for his irresponsible act of driving dangerously. Thus, the solution for this is by creating this system with engine lock using GSM, which is embedded into every four wheel vehicle. This system automatically detects the driver‘s intoxication level using an alcohol sensor (MQ-3 sensor). By using ignition, the engines lock if the driver is found to have exceeded the threshold limit of alcohol upon starting the car or slow down a moving car in case the driver consumes alcohol while driving.

This system analyses alcohol consumption by a vehicle driver and alerts authorized person if the driver is drunk by sending the vehicle‘s location using GPS module. Hence, this system can reduce drunken driving accidents and alert other vehicle users besides alerting officers so legal actions can be taken.

2.3 CONCLUSION ON REVIEWS:

Here we propose a framework where the individual is identified for liquor level in his body to stay away from accidents. Drivers will be detected before they begin their vehicle. Driver will be detected by a sensor once he seated on the driver seat by his breath. Alcohol Sensor is put in the steering to screen the breath level if the liquor content in breath is 0.08% then car motor won't start. In this framework if the driver isn't drunk he can drive else he cannot drive until the point that the liquor content decreases. Arduino uno is arranged and associated with the sensor additionally LCD display and one dc motor is associated. Once the association is given power supply to it so the engine will start running .Now liquor is sprayed in it where the liquor content is over 0.08% so the LCD display will demonstrate that alcohol is detected, dc engine will stop running and ignition also stopped. This procedure is executed same in all vehicle where the car motor will be associated with the sensor. Once the alcohol sensor is detected,its output will be sent to the engine by refering the range motor will stop its execution.While implementing this proposed framework, it can decrease the mischances by 75% and reduce the loss of property and lives. These are some more goals of this framework. 1. When driver starting car/vehicle then alcohol sensor begin detecting at condition vehicle speed equivalent to zero. 2. If alcoholic driver recognized then promptly ignition system will turn off and notification will be shown on LCD with alarm/buzzer. 3. A flag is set when first condition is passed without discovery of liquor. 4. If alcohol is recognized for this situation at that point signal is send to fuel blocker by Arduino for locking the start system.so driver feel's that vehicle is going to stop and after that he will place the car at proper location.

CHAPTER-3

PROJECT DESIGN

3.1 OVERVIEW OF THE DESIGN:

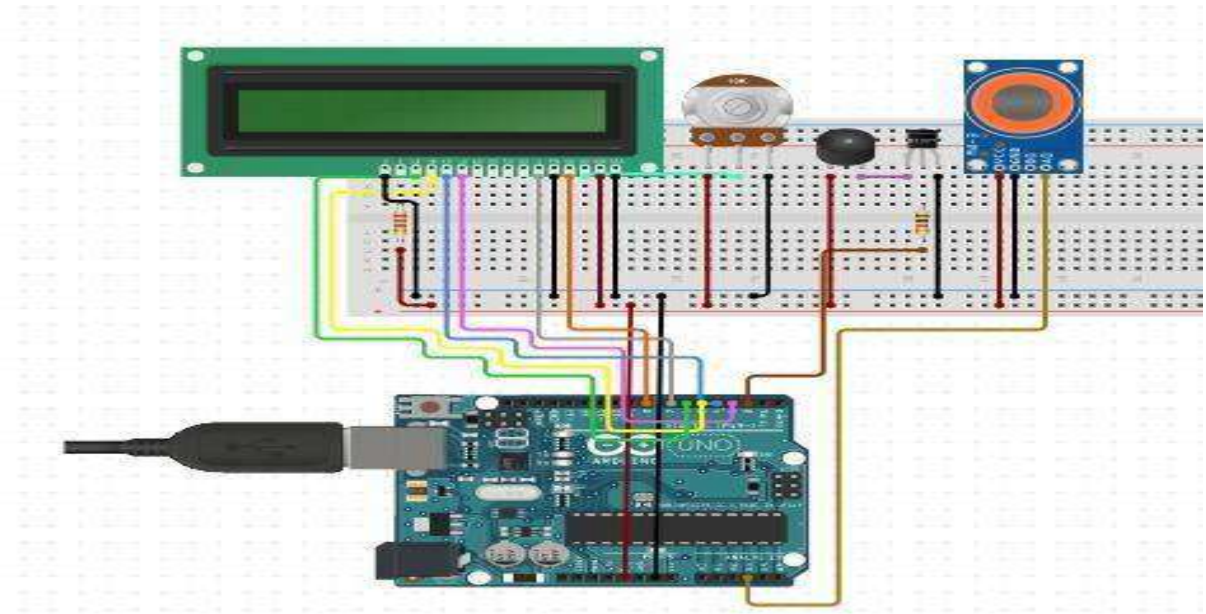


Fig.3.1 Overview of the design

3.2 EQUIPMENT ANALYSIS:

Accidents or Mishaps are happening calm regularly now days for some reasons. Among those reasons the calmest reason is because of the liquor dependent purchasers. A large number of the general population used to go to work places like production lines, ventures, medical clinics, workplaces and military by expending liquor. These reason hazardous mishaps in numerous spots over out of indiscretion.

A definitive purpose behind this proposed work is to lessen mishaps because of liquor utilization by identifying it.. This guarantees appropriate hard working attitudes are pursued. In this way, our proposed framework takes into account liquor checking in addition to announcing framework that screens this and reports it to concerned individual remotely over web. Our framework is made out of an IOT based circuit framework that utilizes Arduino board. The framework has MQ3 liquor sensor and to look at the liquor utilization of driver and to control vehicle start automatically. This data refresh to the cloud server alongside area and liquor content. This guarantees no marvel of mishaps because of liquor affect.

1. ARDUINO UNO

Arduino Uno Rev. 3 Microcontroller Board depends on the Atmel ATmega328 8-bit Microcontroller (MCU). Arduino Uno highlights 14 digital input/output pins,six analog inputs, and a 16MHz quartz crystal. Uno also includes a USB association, a power jack, serial port, and a reset button.

2. ALCOHOL SENSOR

This alcohol sensor is appropriate for recognizing alcohol concentration on your breath, much the same as your normal breathe analyzer. It has a high affectability and quick response time. Sensor gives a simple resistive output in view of alcohol concentration. The detection circuit is basic, all it needs is one resistor. The Alcohol sensor is shown in figure 3. A basic interface could be a 0-3.3V ADC.

3. LCD DISPLAY

LCD (Liquid Crystal Display) screen is an electronic display module and locate an extensive variety of uses. A 16x2 LCD display is extremely essential module and is regularly utilized as a part of different gadgets and circuits. These modules are favored more than seven segments and other multi section LEDs. The reasons being: LCDs are economic; easily programmable; have no confinement of showing uncommon and even custom characters. A 16x2 LCD implies it can show 16 characters for each line and there are 2 such lines. In this LCD each character is shown in figure 4 is a 5x7 pixel matrix.

4. BUZZER

A buzzer or beeper is a sound flagging device, which might be mechanical, electromechanical, or piezoelectric. Common uses of buzzers and beepers include caution devices, timers, and affirmation of client input. The buzzer gives an alarm when the alcohol is detected in the vehicle.

Figure 3.2 a) MQ3 alcohol sensor Character configuration Figure 3.2 b) LCD Display

Figure 3.2 c)Buzzer

3.2.1 ARDUINO:

3.2.1.1 INTRODUCTION:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

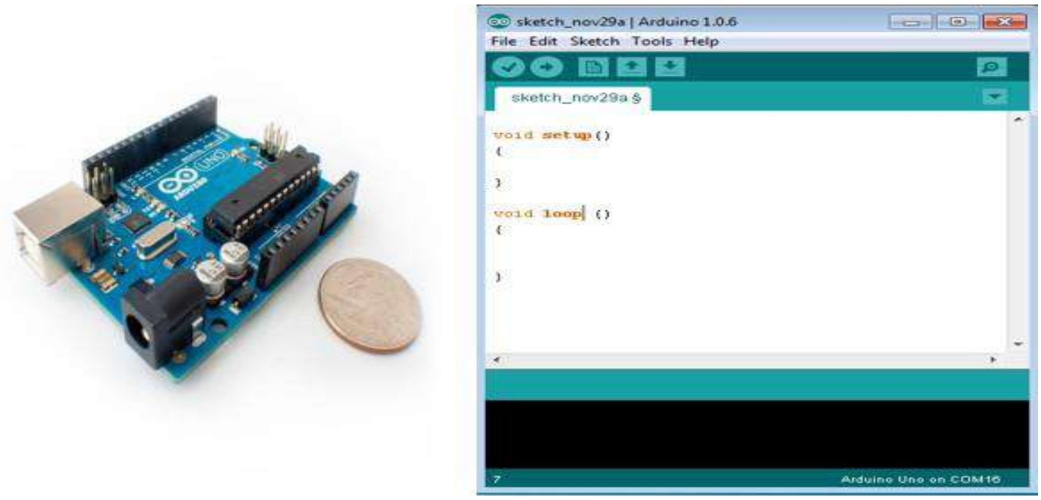


Fig 3.2.1.1 Arduino Uno

3.2.1.2 BOARD TYPES

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface

Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header

Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.2.1.2.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 3.2.1.2.2 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 3.2.1.2.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table 3.2.1.2.4Arduino boards based on AT91SAM3X8E microcontroller

3.2.1.3 BOARD DESCRIPTION:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

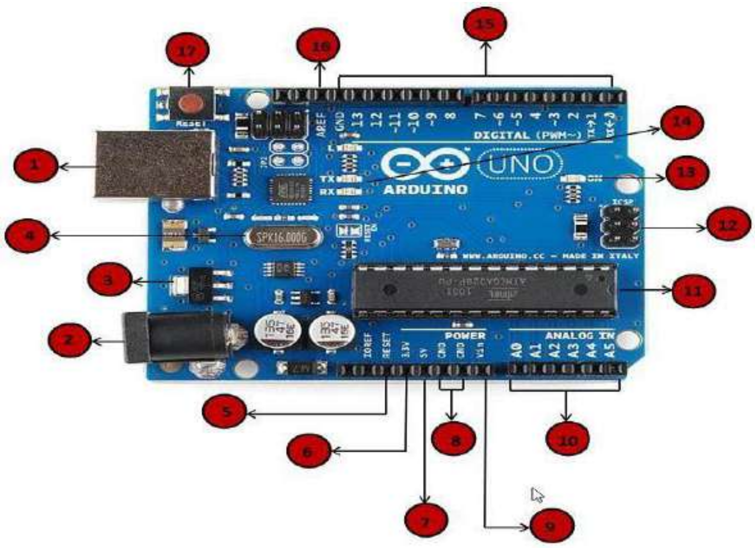


Fig.3.2.1.2 Board Description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can

	reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that

	your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled ―~‖ can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

3.2.1.4 ARDUINO FAMILY

Arduino makes several different boards, each with different capabilities. In addition, part of being open-source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

Fig.3.2.1.3: Arduino Family

3.2.1.5 SHIELDS:

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

Fig.3.2.1.4: Arduino Shields

3.2.1.6 PIN DESCRIPTION OF ATMEGA328:

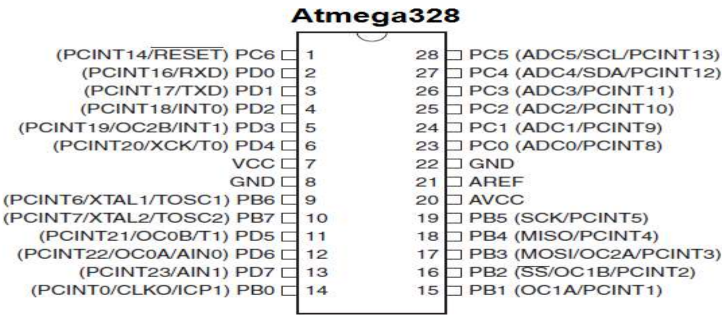


Fig.3.2.1.5 Pin description of ATMEGA328

3.2.1.7 APPLICATIONS:

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

3.2.2 16*2 LCD SCREEN DISPLAY

3.2.2.1 INTRODUCTION

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

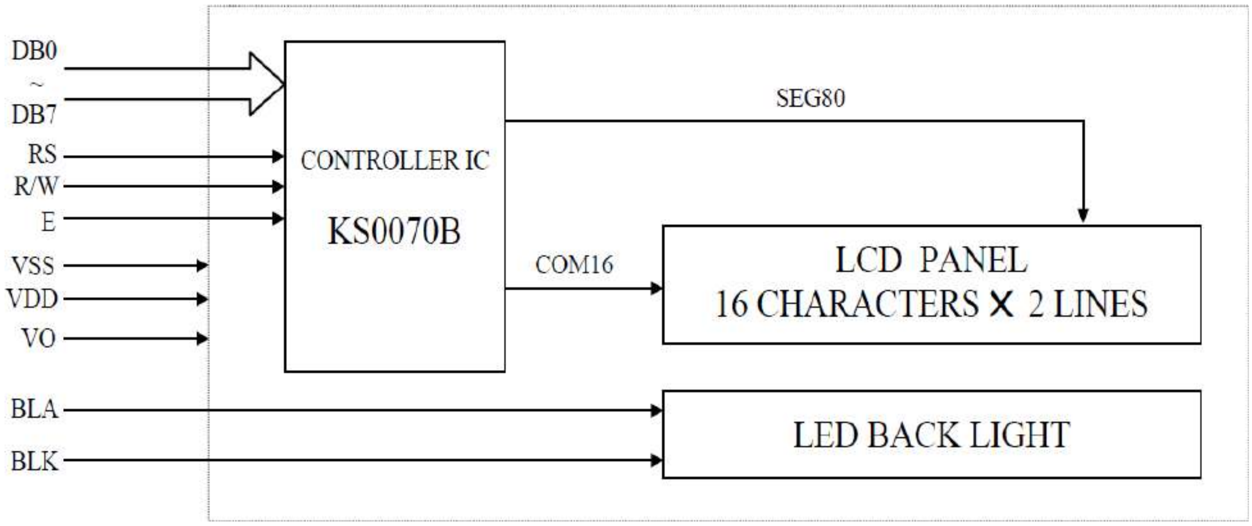


fig.3.2.2.1 LCD block diagram

3.2.2.2 DISPLAY DATA RAM:

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.

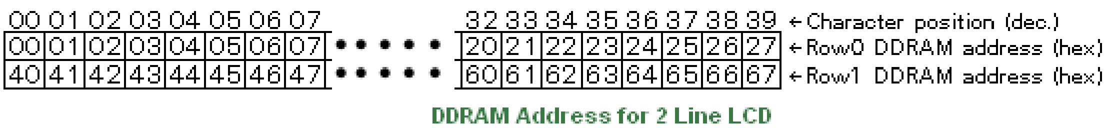


Fig.3.2.2.2 DDRAM addresses of 2 Line LCD

3.2.2.3 CHARACTER GENERATOR ROM:

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM.

3.2.2.4 BUS FLAG:

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e. BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

3.2.2.5 INSTRUCTION REGISTER (IR) AND DATA REGISTER (DR):

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g. LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD.
Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

3.2.2.6 16 X 2 ALPHANUMERIC LCD MODULE FEATURES:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

3.2.2.7 CIRCUIT DESCRIPTION:

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

3.2.3 BUZZER

3.2.3.1 MAGNETIC TRANSDUCER

Magnetic transducers contain a magnetic circuit consisting of a iron core with a wound coil and a yoke plate, a permanent magnet and a vibrating diaphragm with a movable iron piece. The diaphragm is slightly pulled towards the top of the core by the magnet's magnetic field. When a positive AC signal is applied, the current flowing through the excitation coil produces a fluctuating magnetic field, which causes the diaphragm to vibrate up and down, thus vibrating air. Resonance amplifies vibration through resonator consisting of sound hole(s) and cavity and produces a loud sound.

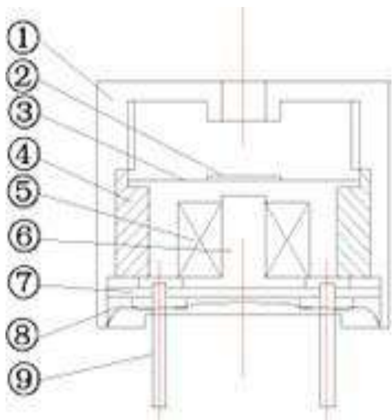


Figure 3.2.3.1 Magnetic transducer

3.2.3.2 MAGNETIC BUZZER(SOUNDER)

Buzzers like the TMB-series are magnetic audible signal devices with built-in oscillating circuits. The construction combines an oscillation circuit unit with a detection coil, a drive coil and a magnetic transducer. Transistors, resistors, diodes and other small devices act as circuit devices for driving sound generators. With the application of voltage, current flows to the drive coil on primary side and to the detection coil on the secondary side. The amplification circuit, including the transistor and the feedback circuit, causes vibration. The oscillation current excites the coil and the unit generates an AC magnetic field corresponding to an oscillation frequency. This AC magnetic field magnetizes the yoke comprising the magnetic circuit. The oscillation from the intermittent magnetization prompts the vibration diaphragm to vibrate up and down, generating buzzer sounds through the resonator.

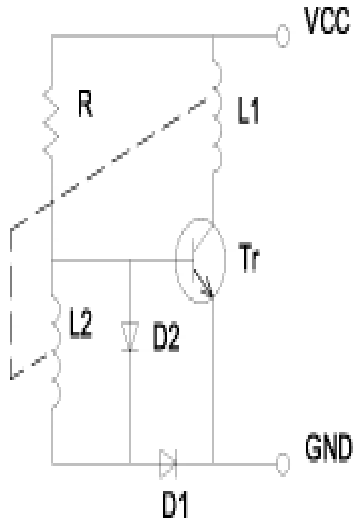


Figure 3.2.3.2 Magnetic buzzer

3.2.3.3 SPECIFICATIONS:

Rated Voltage: A magnetic buzzer is driven by 1/2 square waves (V o-p).
Operating Voltage: For normal operating. But it is not guaranteed to make the minimum Sound Pressure Level (SPL) under the rated voltage.
Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.
Direct Current Resistance: The direct current resistance is measured by ammeter directly.
Sound Output: The sound output is measured by decibel meter. Applying rated voltage and 1/2 square waves, and the distance of 10 cm.
Rated Frequency: A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.
Operating Temp. : Keep working well between -30℃ and +70℃.
How to choose:

Driving methods: AX series with built drive circuit will be the best choice when we cannot provide frequency signal to a buzzer, it only needs direct current.

Dimension: Dimension affects frequency, small size result in high frequency.

Voltage: Depend on V o-p (1/2 square waves)

Fixed methods: From the highest cost to the lowest- DIP, wires/ connector, SMD.

Soldering methods: AS series is soldered by hand, the frequency is lower because of the holes on the bottom. On the other hand, we suggest AC series for the reflow soldering, the reliability is better.

3.2.3.4HOW TO CHOOSE A BUZZER:

There are many different kinds of buzzer to choose, first we need to know a few parameters, such as voltage, current, drive method, dimension, mounting type, and the most important thing is how much SPL and frequency we want.

Operating voltage: Normally, the operating voltage for a magnetic buzzer is from 1.5V to 24V, for a piezo buzzer is from 3V to 220V. However, in order to get enough SPL, we suggest giving at least 9V to drive a piezo buzzer.

Consumption current: According to the different voltage, the consumption current of a magnetic buzzer is from dozens to hundreds of mill amperes; oppositely, the piezo type saves much more electricity, only needs a few mill amperes, and consumes three times current when the buzzer start to work.

Driving method: Both magnetic and piezo buzzer have self drive type to choose. Because of the internal set drive circuit, the self drive buzzer can emit sound as long as connecting with the direct current. Due to the different work principle, the magnetic buzzer need to be driven by 1/2 square waves, and the piezo buzzer need square waves to get better sound output.

Dimension: The dimension of the buzzer affects its SPL and the frequency, the dimension of the magnetic buzzer is from 7 mm to 25 mm; the piezo buzzer is from 12 mm to 50 mm, or even bigger.

Connecting way: Dip type, Wire type, SMD type, and screwed type for big piezo buzzer are usually seen.

Sound Pressure Level (SPL): Buzzer is usually tested the SPL at the distance of 10 cm, if distance double, the SPL will decay about 6 dB; oppositely, the SPL will increase 6 dB when the distance is shortened by one time. The SPL of the magnetic buzzer can reach to around 85 dB/ 10 cm; the piezo buzzer can be designed to emit very loud sound, for example, the common siren, are mostly made of piezo buzzer.

3.2.3.5 INTRODUCTION OF PIEZO BUZZER:

Piezo buzzers are simple devices that can generate basic beeps and tones. They work by using a piezo crystal, a special material that changes shape when voltage is applied to it. If the crystal pushes against a diaphragm, like a tiny speaker cone, it can generate a pressure wave which the human ear picks up as sound. Simple change the frequency of the voltage sent to the piezo and it will start generating sounds by changing shape very quickly!

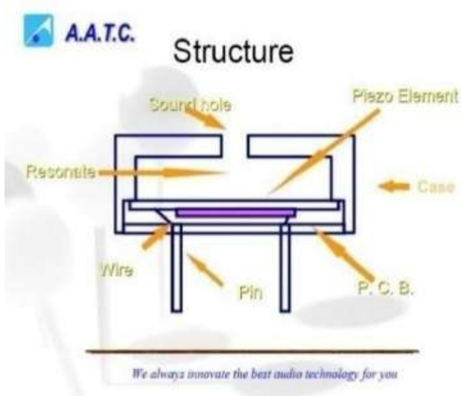


Figure 3.2.3.5 piezo buzzer

3.2.3.6 SPECIFICATIONS:

Rated Voltage: A piezo buzzer is driven by square waves(Vp-p).

Operating Voltage: For normal operating. But it is not guaranteed to make the minimum SPL under the rated voltage.

Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.

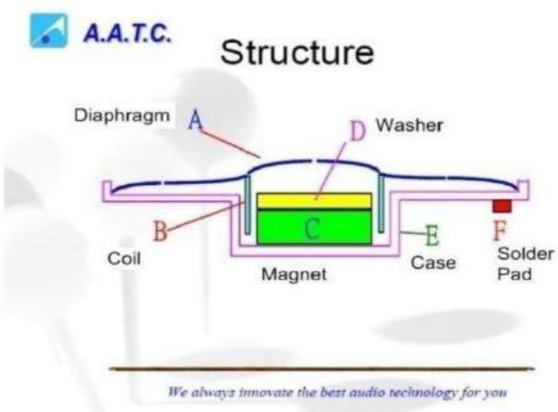


Figure 3.2.3.6 Structure of piezo buzzer

Capacitance: A piezo buzzer can make higher SPL with higher capacitance, but it consumes more electricity.

Sound Output: The sound output is measured by decibel meter.

Applying rated voltage and square waves, and the distance of 10 cm.

Rated Frequency: A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.

Operating Temp.: Keep working well between -30℃ and +70℃.

.

- ## 3.2.4 MOTOR

3.2.4.1 INTRODUCTION:

Motor is a device that creates motion, not an engine; it usually refers to either an electrical motor or an internal combustion engine.

It may also refer to:

- Electric motor, a machine that converts electricity into a mechanical motion
  - AC motor, an electric motor that is driven by alternating current
    - Synchronous motor, an alternating current motor distinguished by a rotor spinning with coils passing magnets at the same rate as the alternating current and resulting magnetic field which drives it
    - Induction motor, also called a squirrel-cage motor, a type of asynchronous alternating current motor where power is supplied to the rotating device by means of electromagnetic induction
  - DC motor, an electric motor that runs on direct current electricity
    - Brushed DC electric motor, an internally commutated electric motor designed to be run from a direct current power source
    - Brushless DC motor, a synchronous electric motor which is powered by direct current electricity and has an electronically controlled commutation system, instead of a mechanical commutation system based on brushes
  - Electrostatic motor, a type of electric motor based on the attraction and repulsion of electric charge

- Servo motor, an electric motor that operates a servo, commonly used in robotics
- Internal fan-cooled electric motor, an electric motor that is self-cooled by a fan, typically used for motors with a high energy density

Industrial motors come in a variety of basic types. These variations are suitable for many different applications. Naturally, some types of motors are more suited for certain applications than other motor types are. This document will hopefully give some guidance in selecting these motors.

3.2.4.1.1 DC Motors

The brushed DC motor is one of the earliest motor designs. Today, it is the motor of choice in the majority of variable speed and torque control applications.

Advantages

- Easy to understand design
- Easy to control speed
- Easy to control torque
- Simple, cheap drive design

Easy to understand design

The design of the brushed DC motor is quite simple. A permanent magnetic field is created in the stator by either of two means:

- Permanent magnets
- Electro-magnetic windings

If the field is created by permanent magnets, the motor is said to be a "permanent magnet DC motor" (PMDC). If created by electromagnetic windings, the motor is often said to be a "shunt wound DC motor" (SWDC). Today, because of cost-effectiveness and reliability, the PMDC motor is the motor of choice for applications involving fractional horsepower DC motors, as well as most applications up to about three horsepower.

At five horsepower and greater, various forms of the shunt wound DC motor are most commonly used. This is because the electromagnetic windings are more cost effective than permanent magnets in this power range.

Caution: If a DC motor suffers a loss of field (if for example, the field power connections are broken), the DC motor will immediately begin to accelerate to the top speed which the loading will allow. This can result in the motor flying apart if the motor is lightly loaded. The possible loss of field must be accounted for, particularly with shunt wound DC motors.

Opposing the stator field is the armature field, which is generated by a changing electromagnetic flux coming from windings located on the rotor. The magnetic poles of the armature field will attempt to line up with the opposite magnetic poles generated by the stator field. If we stopped the design at this point, the motor would spin until the poles were opposite one another, settle into place, and then stop -- which would make a pretty useless motor!

However, we are smarter than that. The section of the rotor where the electricity enters the rotor windings is called the commutator. The electricity is carried between the rotor and the stator by conductive graphite-copper brushes (mounted on the rotor) which contact rings on stator. Imagine power is supplied:

The motor rotates toward the pole alignment point. Just as the motor would get to this point, the brushes jump across a gap in the stator rings. Momentum carries the motor forward over this gap. When the brushes get to the other side of the gap, they contact the stator rings again and -- the polarity of the voltage is reversed in this set of rings! The motor begins accelerating again, this time trying to get to the opposite set of poles. (The momentum has carried the motor past the original pole alignment point.) This continues as the motor rotates.

In most DC motors, several sets of windings or permanent magnets are present to smooth out the motion.

Easy to control speed

Controlling the speed of a brushed DC motor is simple. The higher the armature voltage, the faster the rotation. This relationship is linear to the motor's maximum speed.

The maximum armature voltage which corresponds to a motor's rated speed (these motors are usually given a rated speed and a maximum speed, such as 1750/2000 rpm) are available in certain standard voltages, which roughly increase in conjuntion with horsepower. Thus, the smallest industrial motors are rated 90 VDC and 180 VDC. Larger units are rated at 250 VDC and sometimes higher.

Specialty motors for use in mobile applications are rated 12, 24, or 48 VDC. Other tiny motors may be rated 5 VDC.

Most industrial DC motors will operate reliably over a speed range of about 20:1 -- down to about 5-7% of base speed. This is much better performance than the comparible AC motor. This is partly due to the simplicity of control, but is also partly due to the fact that most industrial DC motors are designed with variable speed operation in mind, and have added heat dissipation features which allow lower operating speeds.

Easy to control torque

In a brushed DC motor, torque control is also simple, since output torque is proportional to current. If you limit the current, you have just limited the torque which the motor can achieve. This makes this motor ideal for delicate applications such as textile manufacturing.

Simple, cheap drive design

The result of this design is that variable speed or variable torque electronics are easy to design and manufacture. Varying the speed of a brushed DC motor requires little more than a large enough potentiometer. In practice, these have been replaced for all but sub-fractional horsepower applications by the SCR and PWM drives, which offer relatively precisely control voltage and current. Common DC drives are available at the low end (up to 2 horsepower) for under US$100 -- and sometimes under US$50 if precision is not important.

Large DC drives are available up to hundreds of horsepower. However, over about 10 horsepower careful consideration should be given to the price/performance tradeoffs with AC inverter systems, since the AC systems show a price advantage in the larger systems. (But they may not be capable of the application's performance requirments).

Disadvantages

- Expensive to produce
- Can't reliably control at lowest speeds
- Physically larger
- High maintenance
- Dust

3.2.4.2 WORKING OF DC MOTOR:

In any electric motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. As you are well aware of from playing with magnets as a kid, opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion.

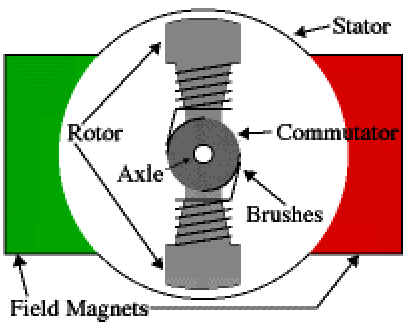


Figure 3.2.4.1 Working of dc motor

3.2.4.3 PRINCIPLE

When a rectangular coil carrying current is placed in a magnetic field, a torque acts on the coil which rotates it continuously.

When the coil rotates, the shaft attached to it also rotates and thus it is able to do mechanical work.

Every DC motor has six basic parts -- axle, rotor (a.k.a., armature), stator, commutator, field magnet(s), and brushes. In most common DC motors (and all that BEAMers will see), the

external magnetic field is produced by high-strength permanent magnets[1]. The stator is the stationary part of the motor -- this includes the motor casing, as well as two or more permanent magnet pole pieces. The rotor (together with the axle and attached commutator) rotate with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a common motor layout -- with the rotor inside the stator (field) magnets.

The geometry of the brushes, commentator contacts, and rotor windings are such that when power is applied, the polarities of the energized winding and the stator magnet(s) are misaligned, and the rotor will rotate until it is almost aligned with the stator's field magnets. As the rotor reaches alignment, the brushes move to the next commentator contacts, and energize the next winding. Given our example two-pole motor, the rotation reverses the direction of current through the rotor winding, leading to a "flip" of the rotor's magnetic field, driving it to continue rotating.

In real life, though, DC motors will always have more than two poles (three is a very common number). In particular, this avoids "dead spots" in the commutator. You can imagine how with our example two-pole motor, if the rotor is exactly at the middle of its rotation (perfectly aligned with the field magnets), it will get "stuck" there. Meanwhile, with a two-pole motor, there is a moment where the commutator shorts out the power supply (i.e., both brushes touch both commutator contacts simultaneously). This would be bad for the power supply, waste energy, and damage motor components as well. Yet another disadvantage of such a simple motor is that it would exhibit a high amount of torque "ripple" (the amount of torque it could produce is cyclic with the position of the rotor).

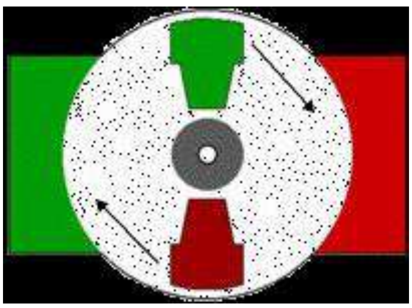


Figure 3.2.4.3 Principle

3.2.4.4 Construction and Working

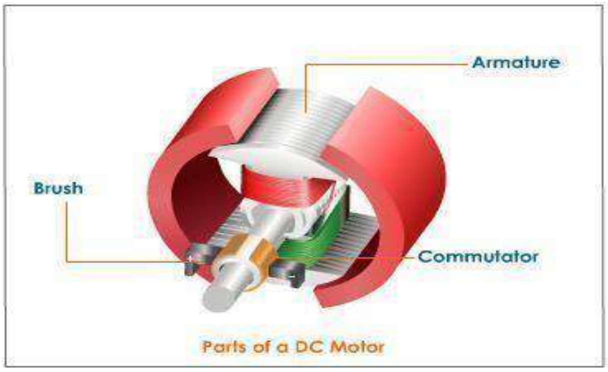


Figure 3.2.4.4 construction and working of dc motor

3.2.4.5 Parts of a DC Motor

3.2.4.6 Armature

A D.C. motor consists of a rectangular coil made of insulated copper wire wound on a soft iron core. This coil wound on the soft iron core forms the armature. The coil is mounted on an axle and is placed between the cylindrical concave poles of a magnet.

3.2.4.7 Commutator

A commutator is used to reverse the direction of flow of current. Commutator is a copper ring split into two parts C_1 and C_2. The split rings are insulated form each other and mounted on the axle of the motor. The two ends of the coil are soldered to these rings. They rotate along with the coil. Commutator rings are connected to a battery. The wires from the battery are not connected to the rings but to the brushes which are in contact with the rings.

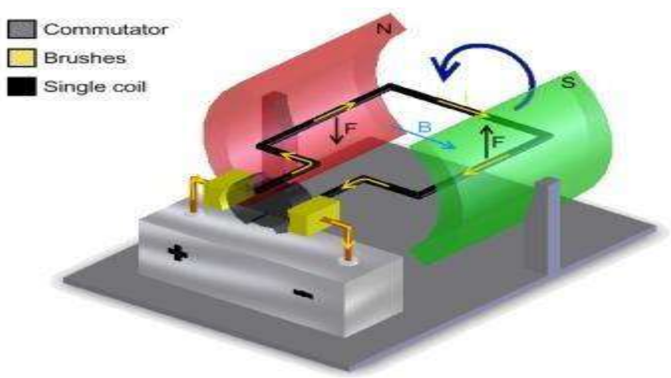


Figure 3.2.4.7 commutator

3.2.4.8 Brushes

Two small strips of carbon, known as brushes press slightly against the two split rings, and the split rings rotate between the brushes.

The carbon brushes are connected to a D.C. source.

3.2.4.9 Working of a DC Motor

When the coil is powered, a magnetic field is generated around the armature. The left side of the armature is pushed away from the left magnet and drawn towards the right, causing rotation.

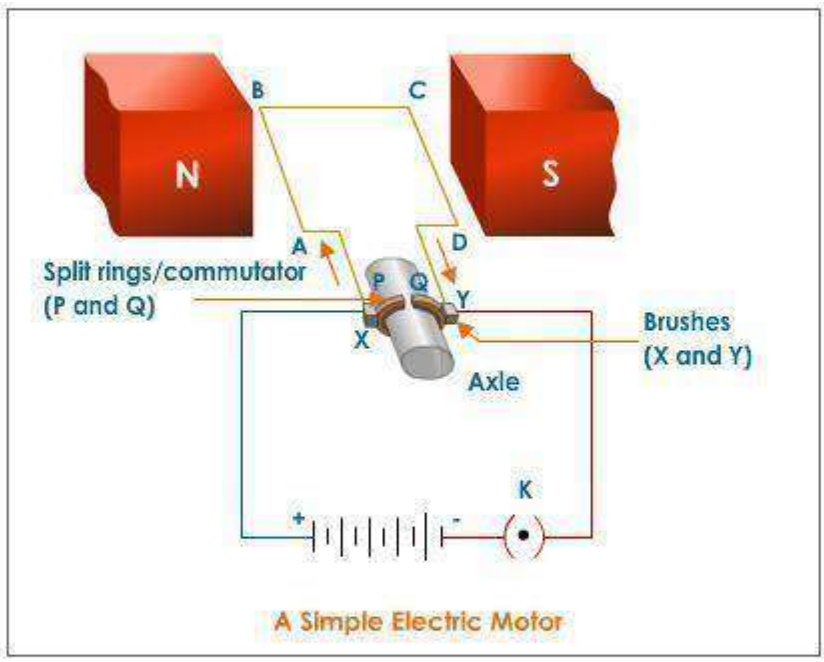
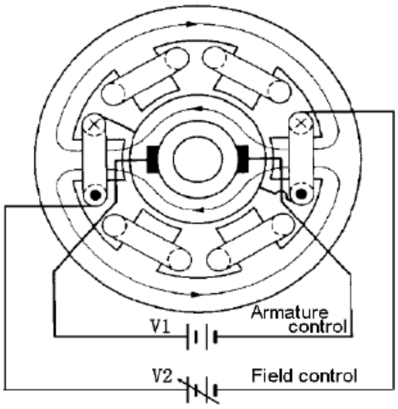


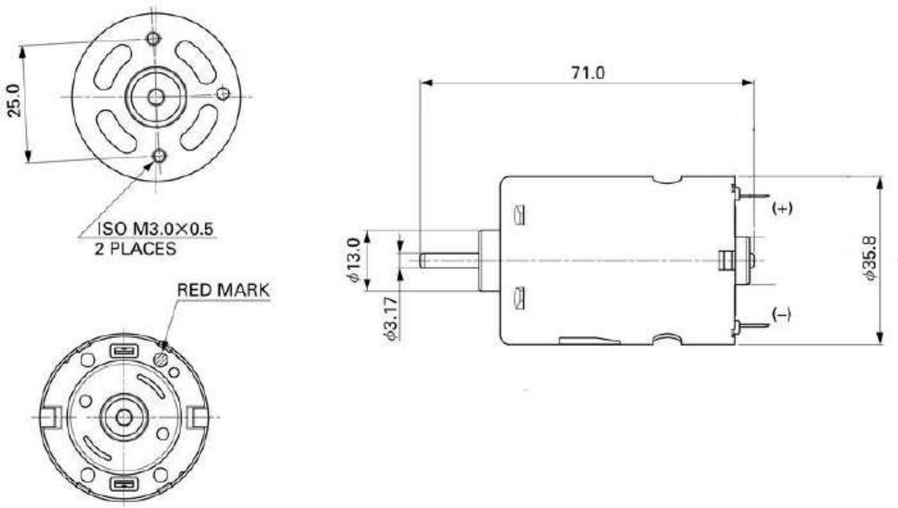
Figure 3.2.4.9 working of a dc motor

When the coil turns through 90^0, the brushes lose contact with the commutator and the current stops flowing through the coil.

However the coil keeps turning because of its own momentum.

Now when the coil turns through 180^0, the sides get interchanged. As a result the commutator ring C_1 is now in contact with brush B_2 and commutator ring C_2 is in contact with brush B_1. Therefore, the current continues to flow in the same direction.





3.2.5 SOFTWARE EXPLANATION:

- Arduino software
- Proteus simulation

3.2.5.1 ARDUINO SOFTWARE:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientificinstruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

3.2.5.2 WHY ARDUINO?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

● Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.

● Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.Wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

● Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.

● Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

● Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

● Open source and extensible software - The Arduino software is published as open sourcetools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

● Open source and extensible hardware - The plans of the Arduino boards are published under

a Creative Commons license, so experienced circuit designers c an make their own version Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money

● Getting Started with Arduino and Genuino products: -

Install the Arduino Software (IDE) on Windows PCs-

This document explains how to install the Arduino Software (IDE) on Windows machines.

- ✔ Download the Arduino Software (IDE)
- ✔ Proceed with board specific instructions.

3.2.5.3 HOW TO DOWNLOAD THE ARDUINO SOFTWARE (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

3.2.5.4 INSTALLATION:

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.3.2.5.4.1 Standard USB Cable (A plug to B plug)

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig.3.2.5.4.2Standard USB Cable (an A to Mini-B plug)

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

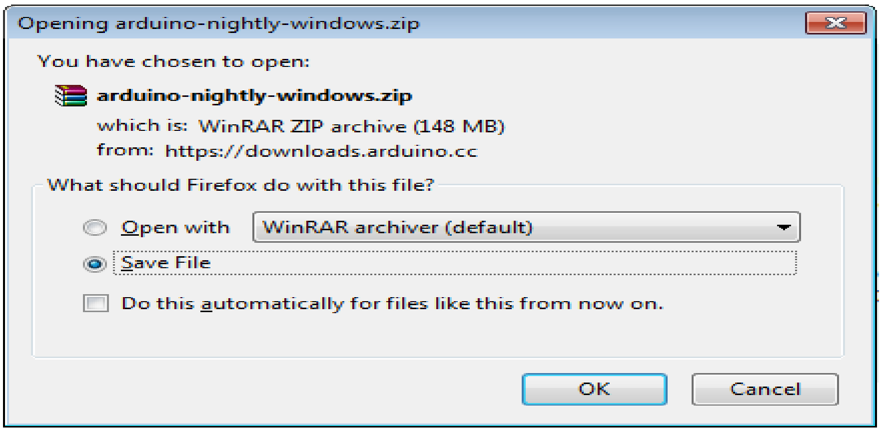


Fig.3.2.5.4.3Download Arduino IDE Software

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

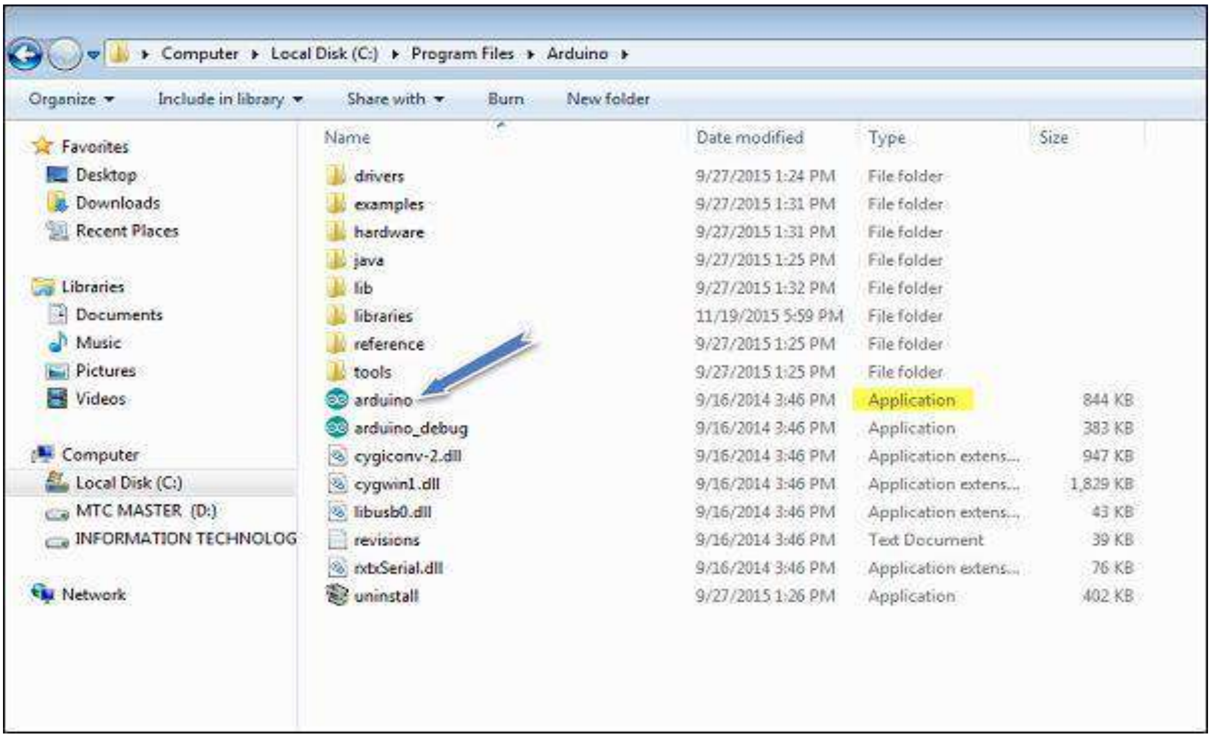


Fig.3.2.5.4.4Launch Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**

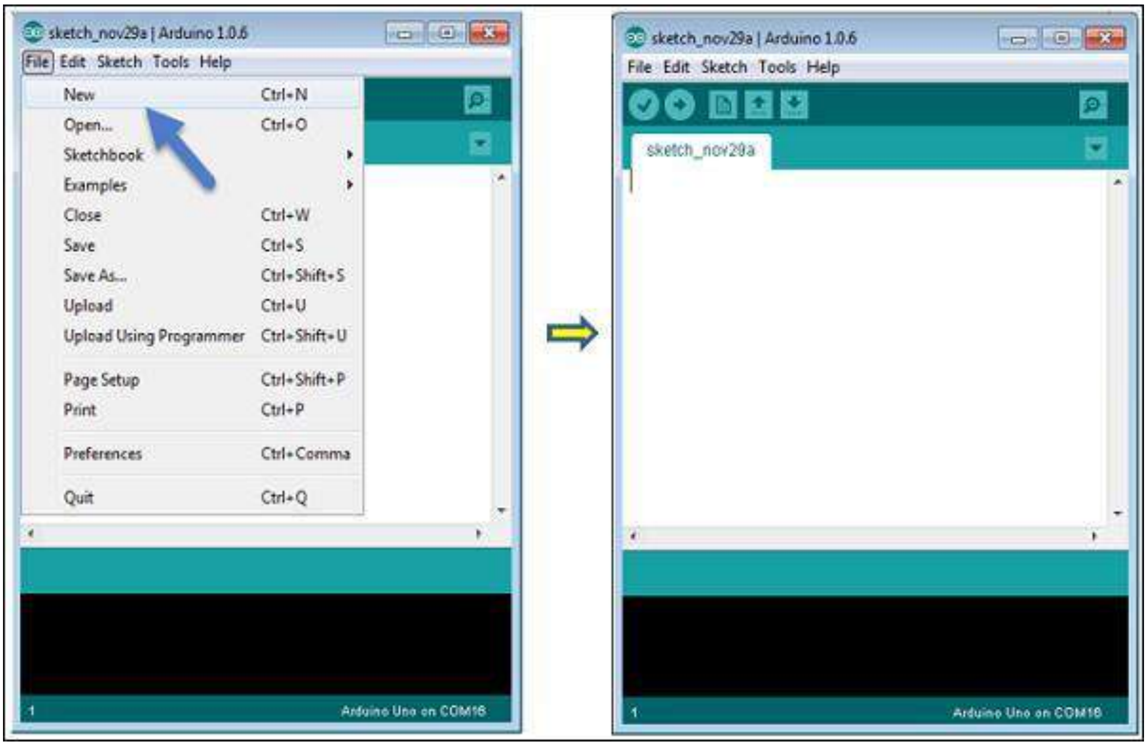


Fig.3.2.5.4.5Open your first project.

To open an existing project example, select File → Example → Basics → Blink.

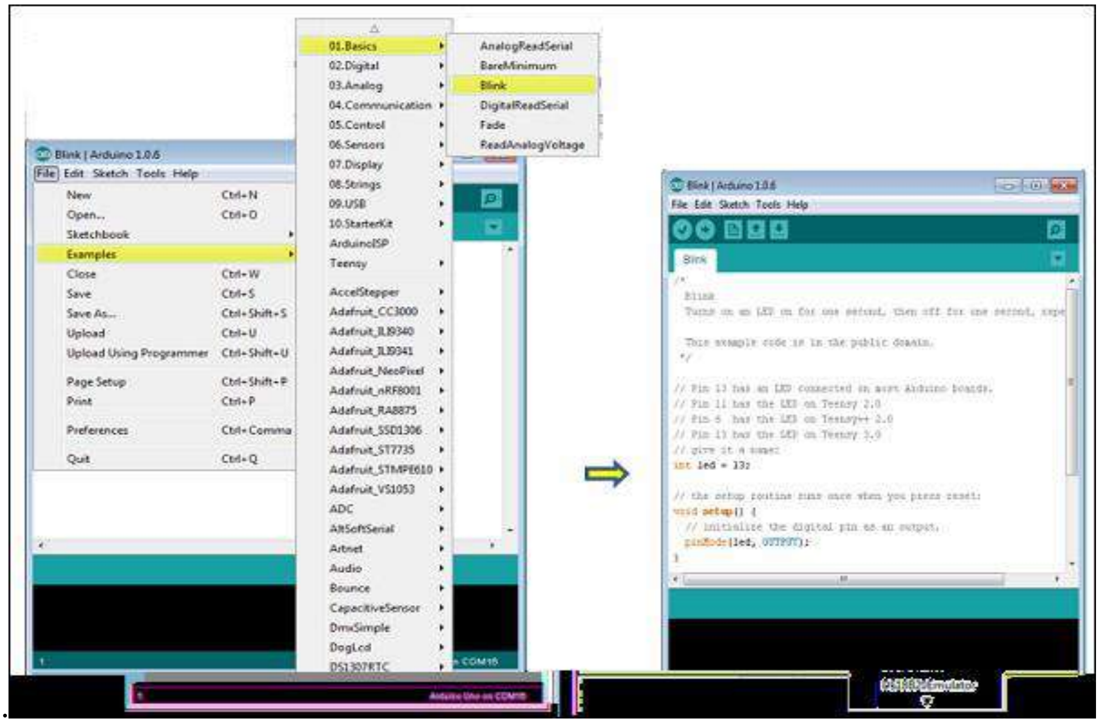


Fig.3.2.5.4.6Open your first program.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

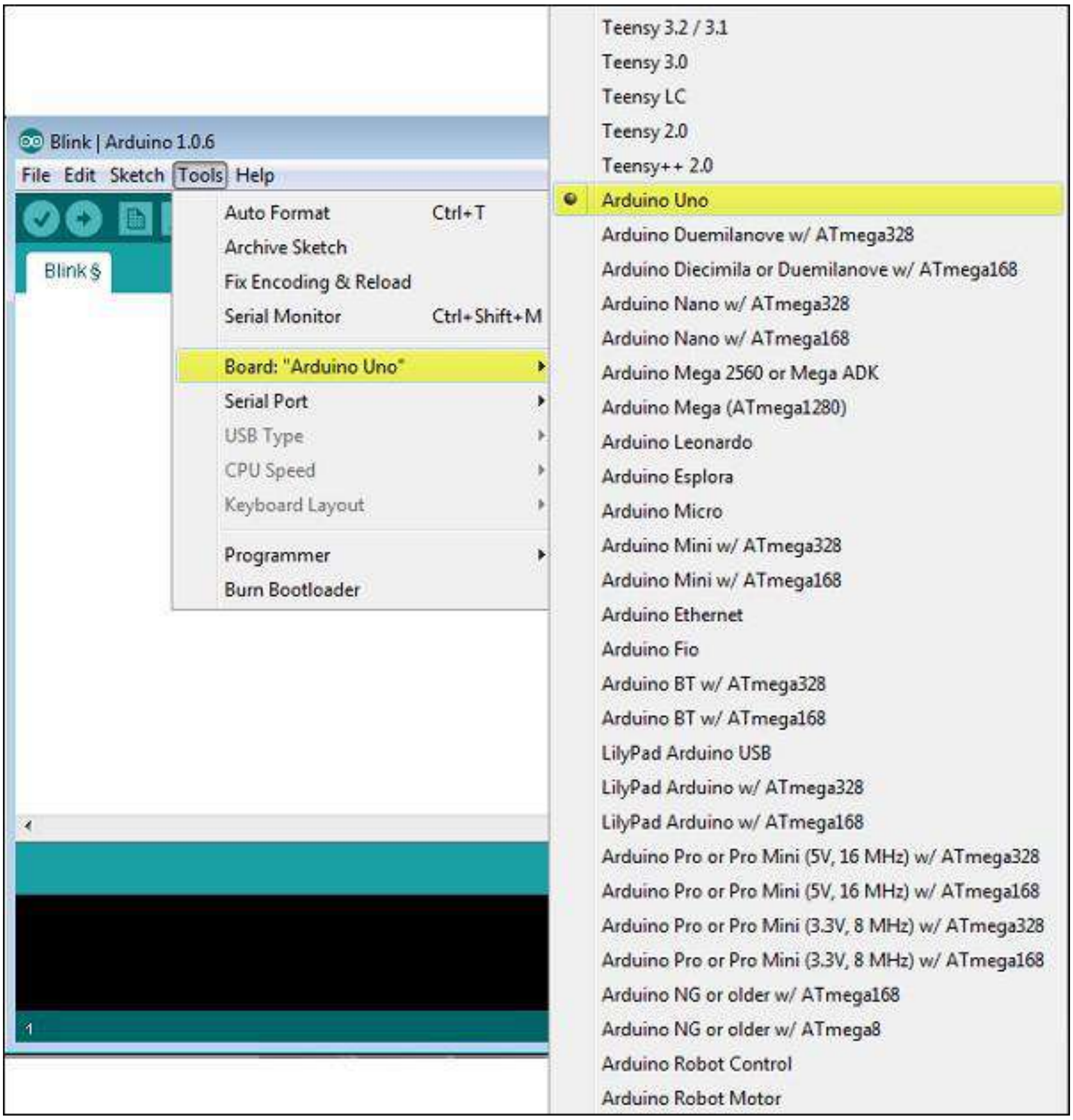


Fig.3.2.5.4.7Select your Arduino board.

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

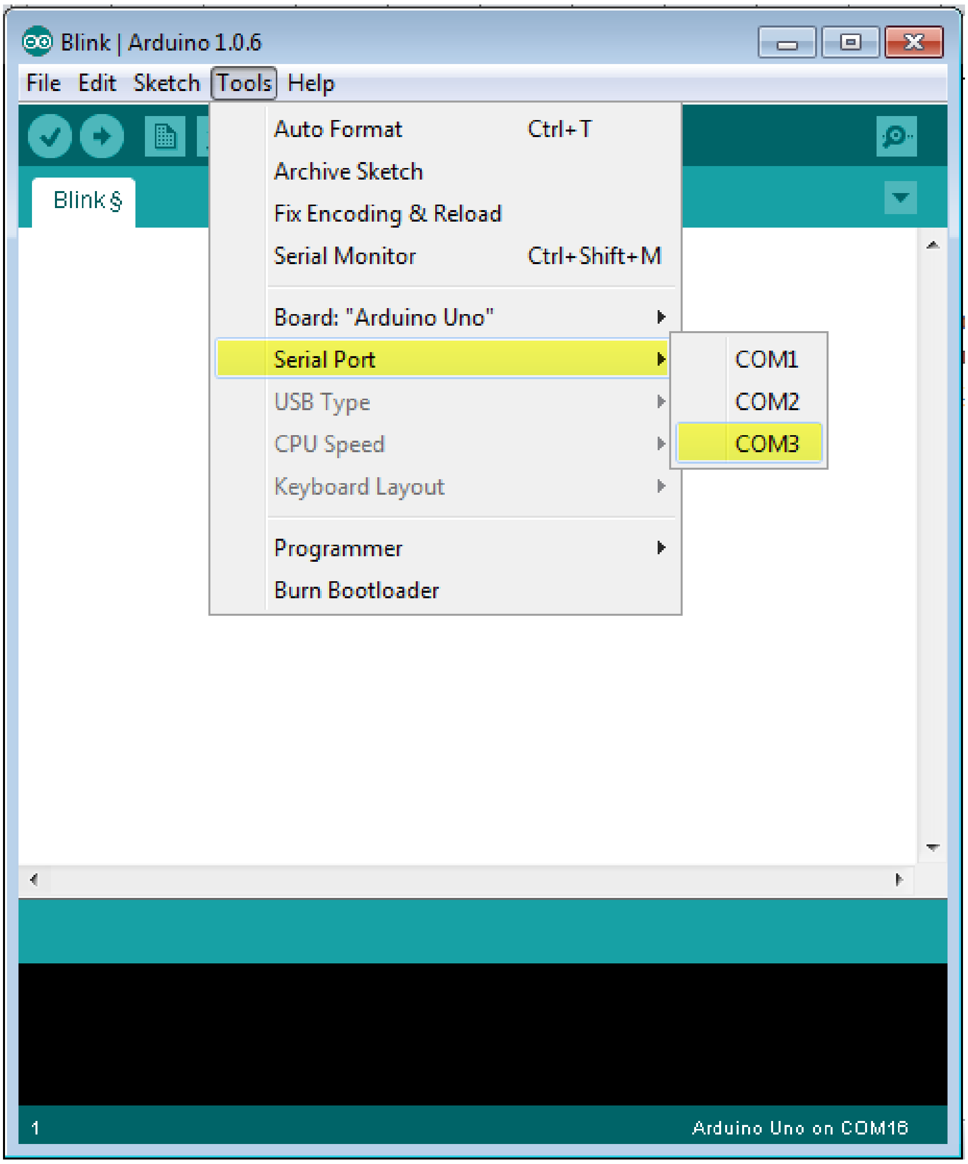


Fig.3.2.5.4.8Select your serial port.

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

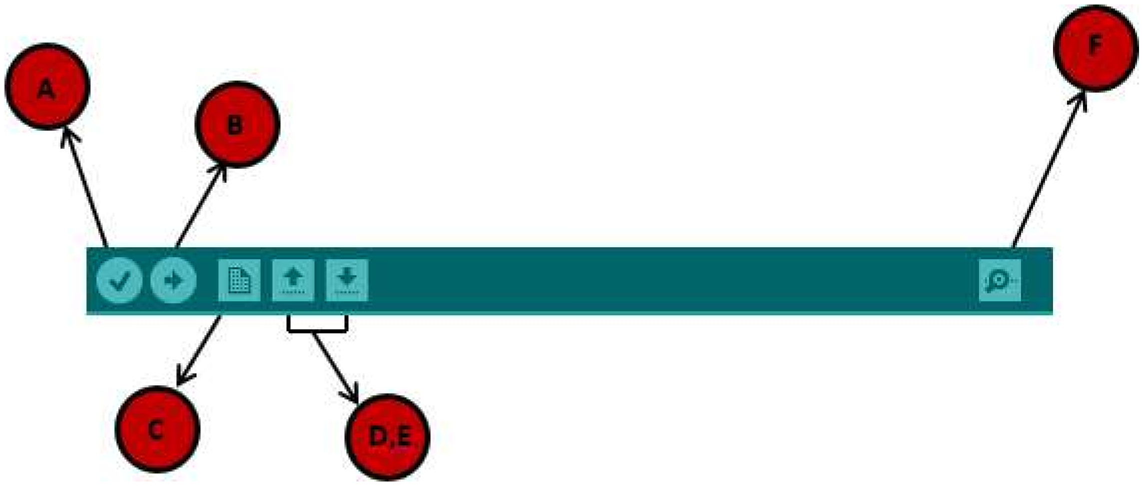


Fig.3.2.5.4.9 Upload the program to your board

A – Used to check if there is any compilation error.

B – Used to upload a program to the Arduino board.

C – Shortcut used to create a new sketch.

D – Used to directly open one of the example sketch.

E – Used to save your sketch.

F – Serial monitor used to receive serial data from board & send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note – *If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.*

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to

24V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

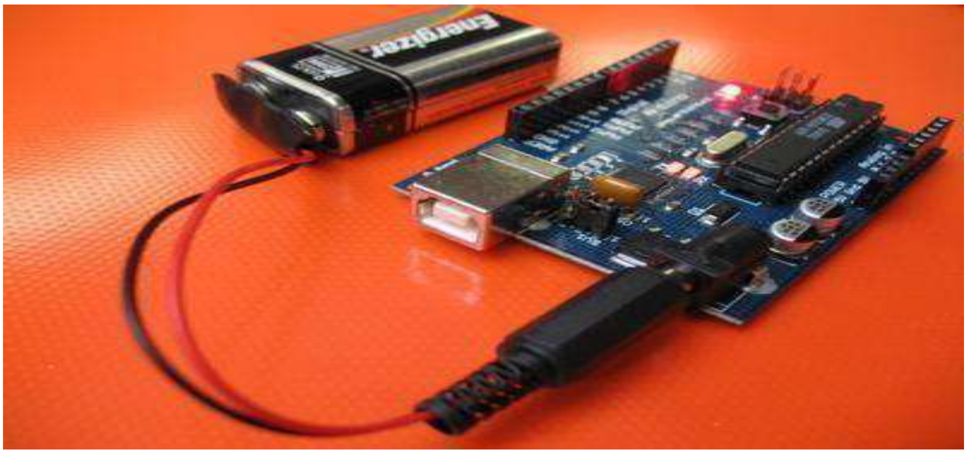


Fig.3.2.5.4.10Connecting a Battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called –sketches‖, but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

Fig.3.2.5.4.11Example Program

3.2.6 PROTEUS:

PROTEUS:

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electricaland Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline –From concept to completion‖.

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole

packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design:

Step 1: Open ISIS software and select New design in File menu

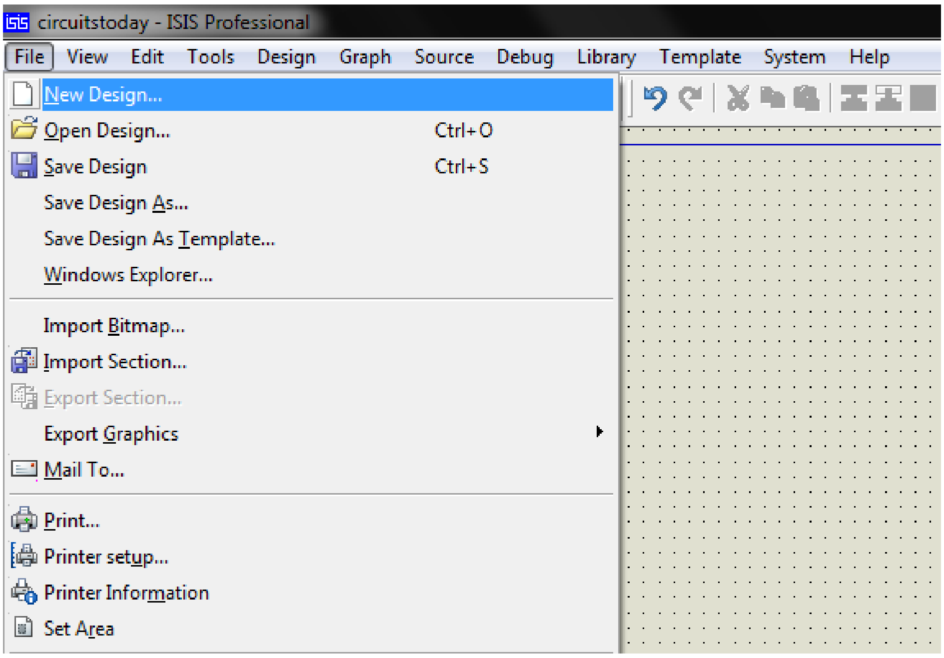


Fig.3.2.6.1Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

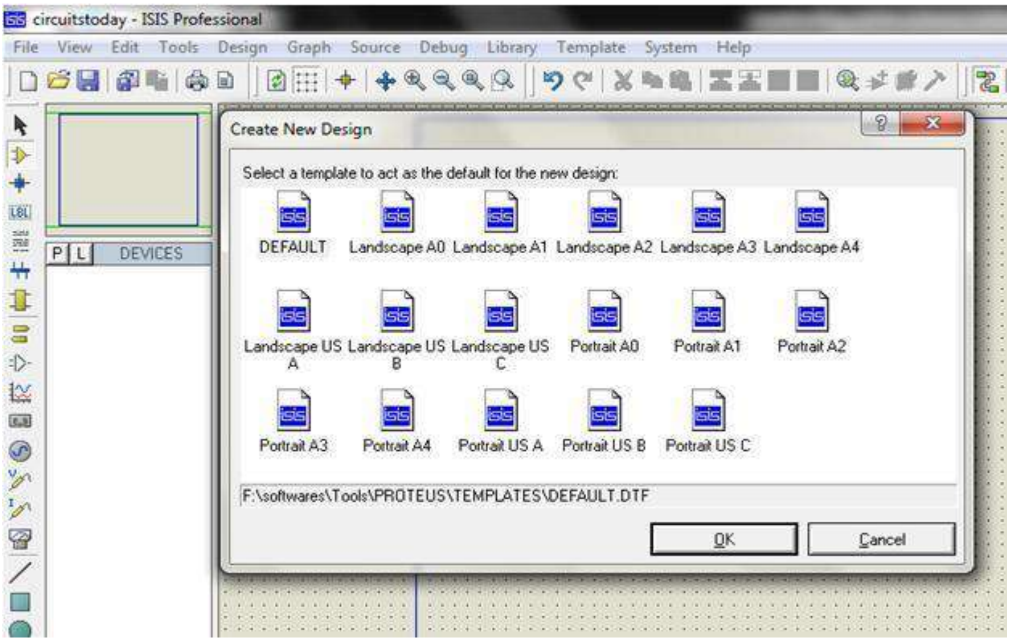


Fig.3.2.6.2Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

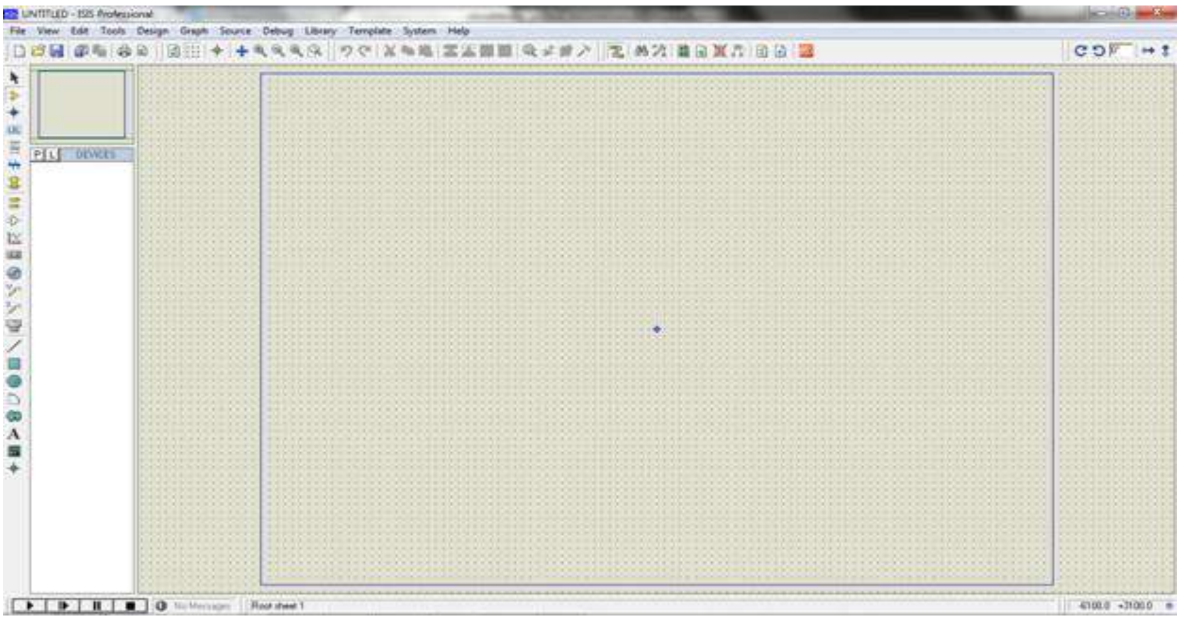


Fig.3.2.6.3Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

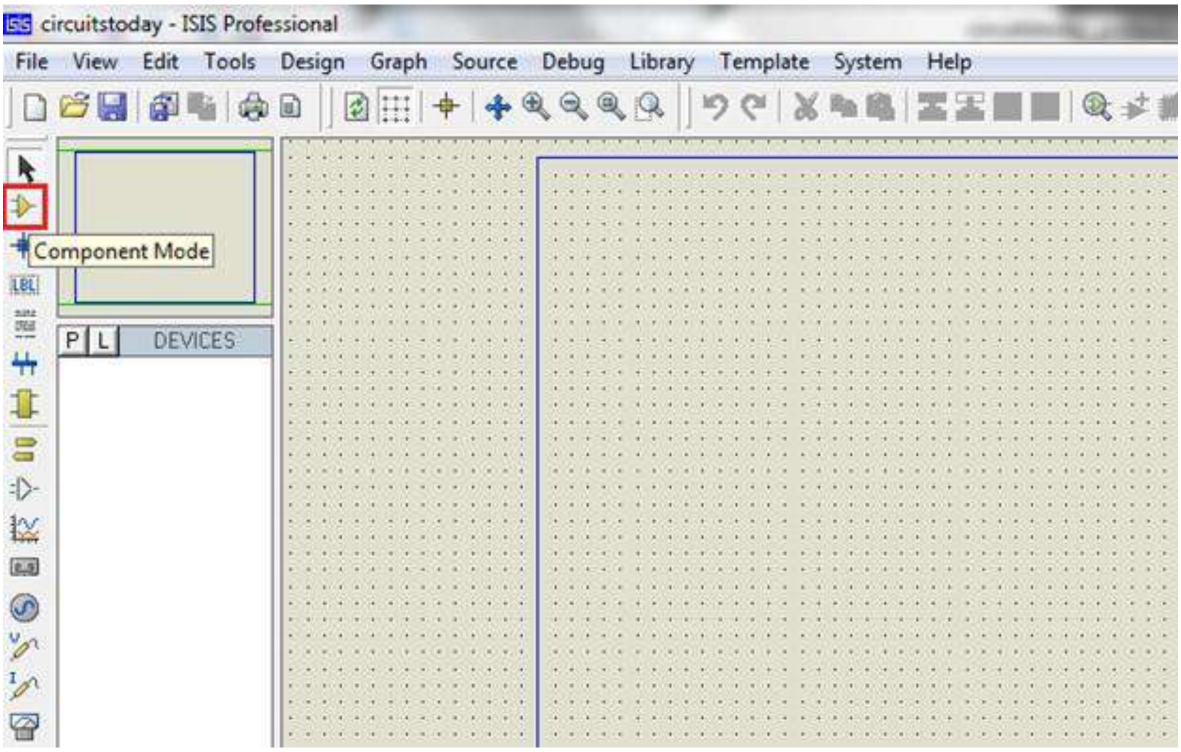


Fig.3.2.6.4Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

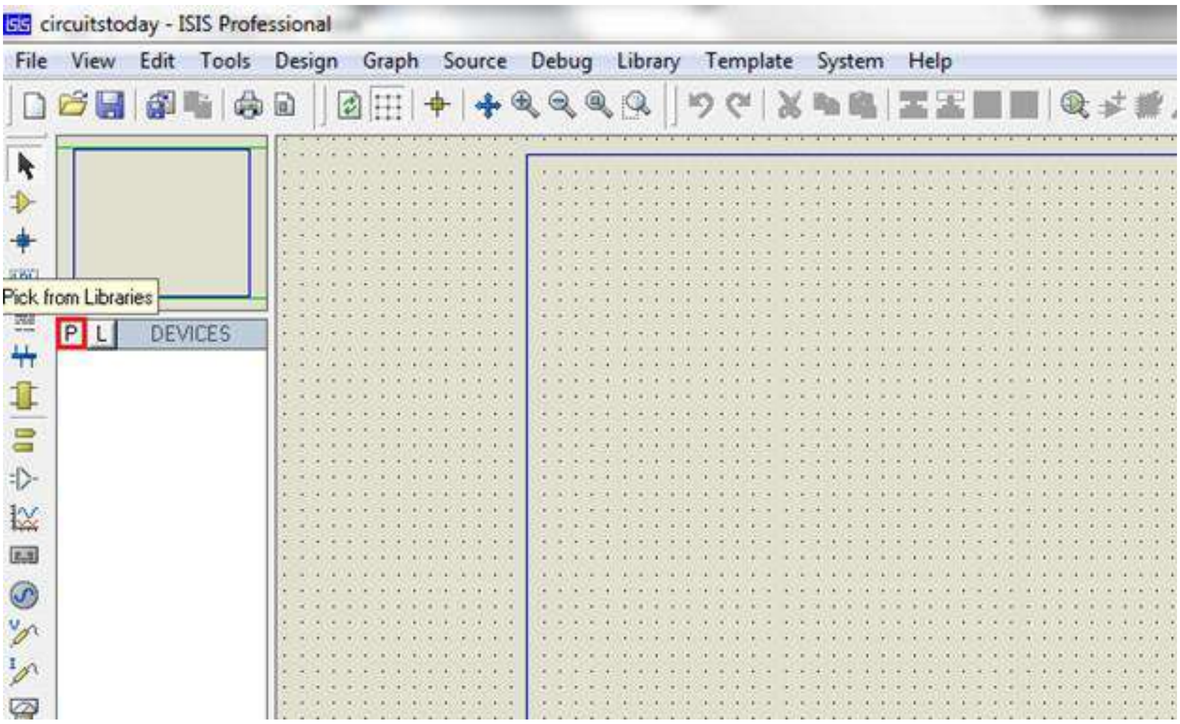


Fig.3.2.6.5Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

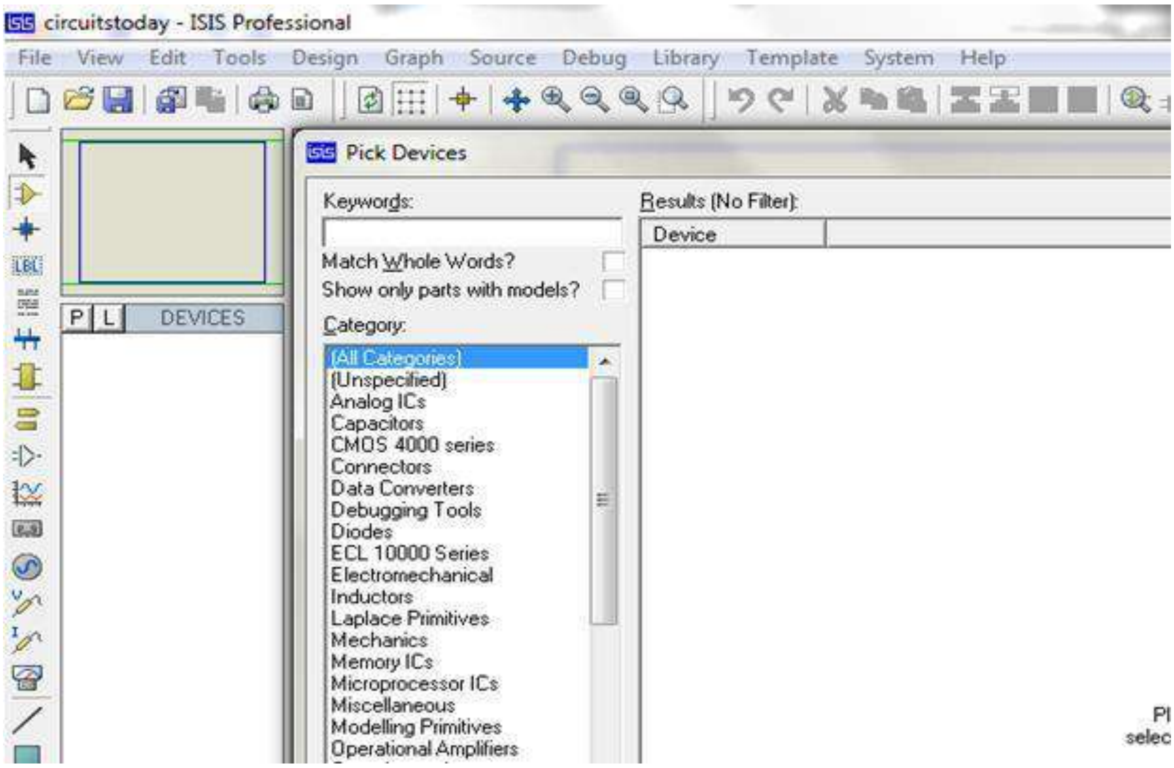


Fig.3.2.6.6Keywords Textbox

Example shows selection of push button. Select the components accordingly.

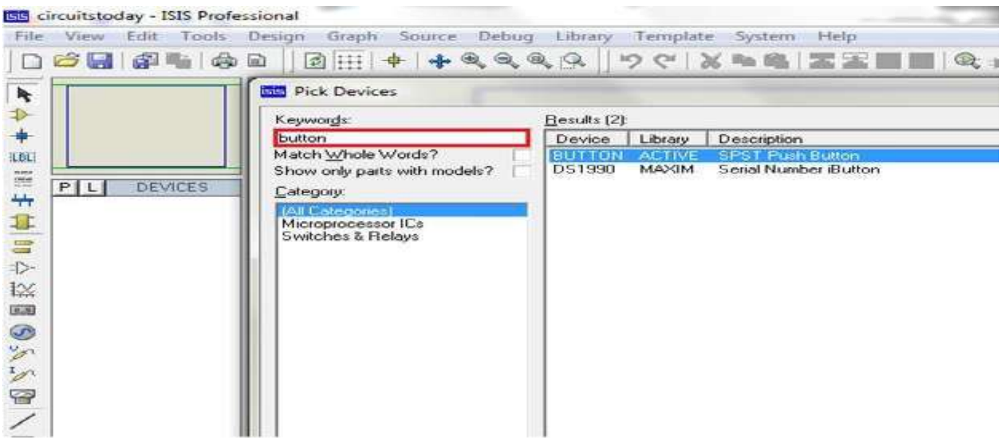


Fig.3.2.6.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

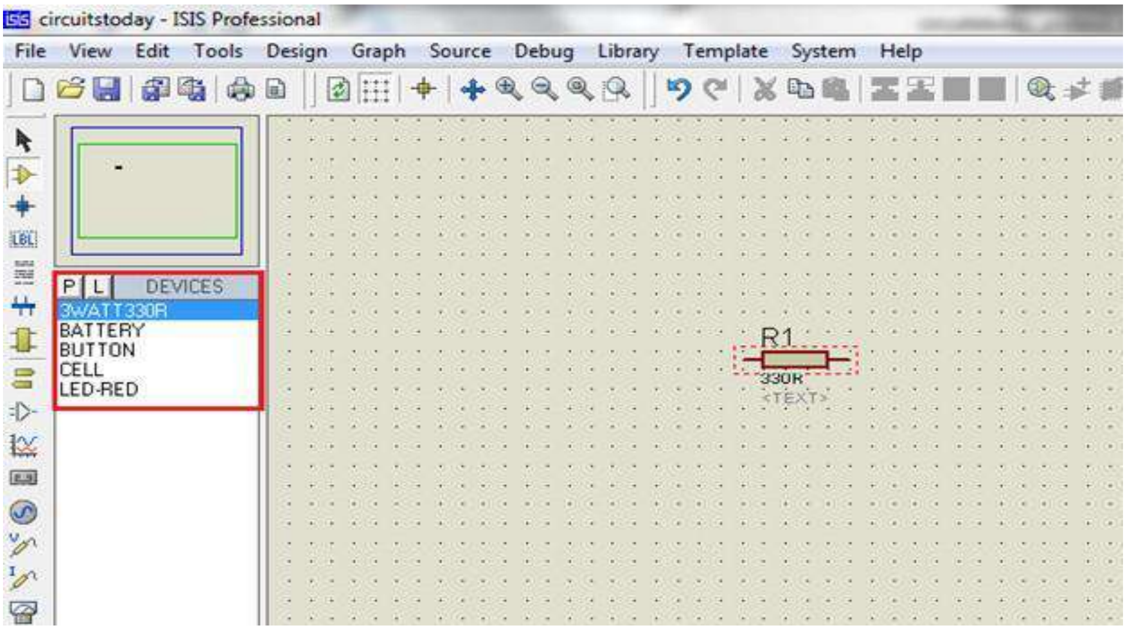


Fig.3.2.6.8Component Selection

Place all the required components and route the wires i.e., make connections. Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

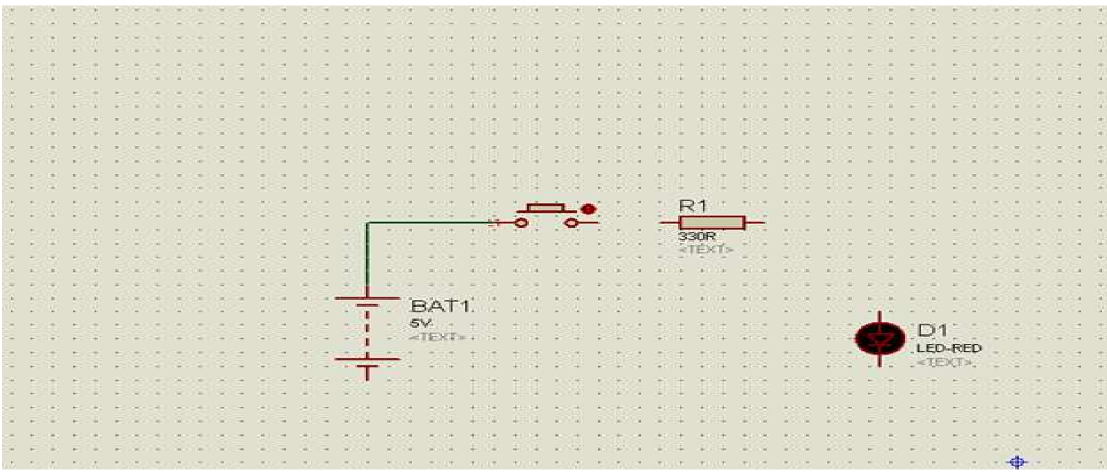


Fig.3.2.6.9Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

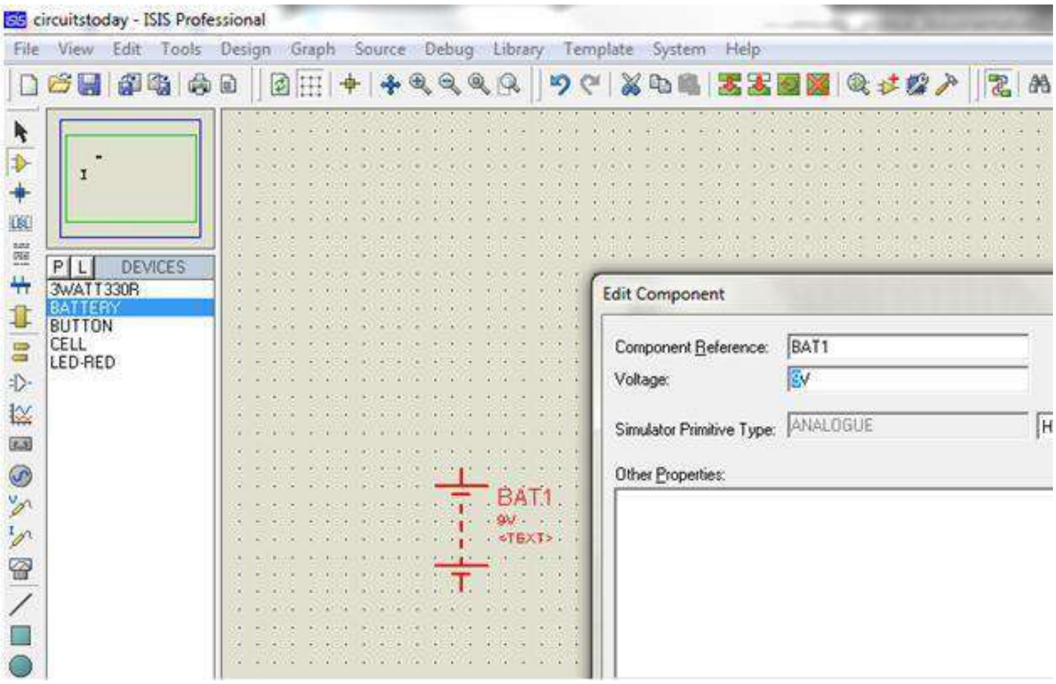


Fig.3.2.6.10Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

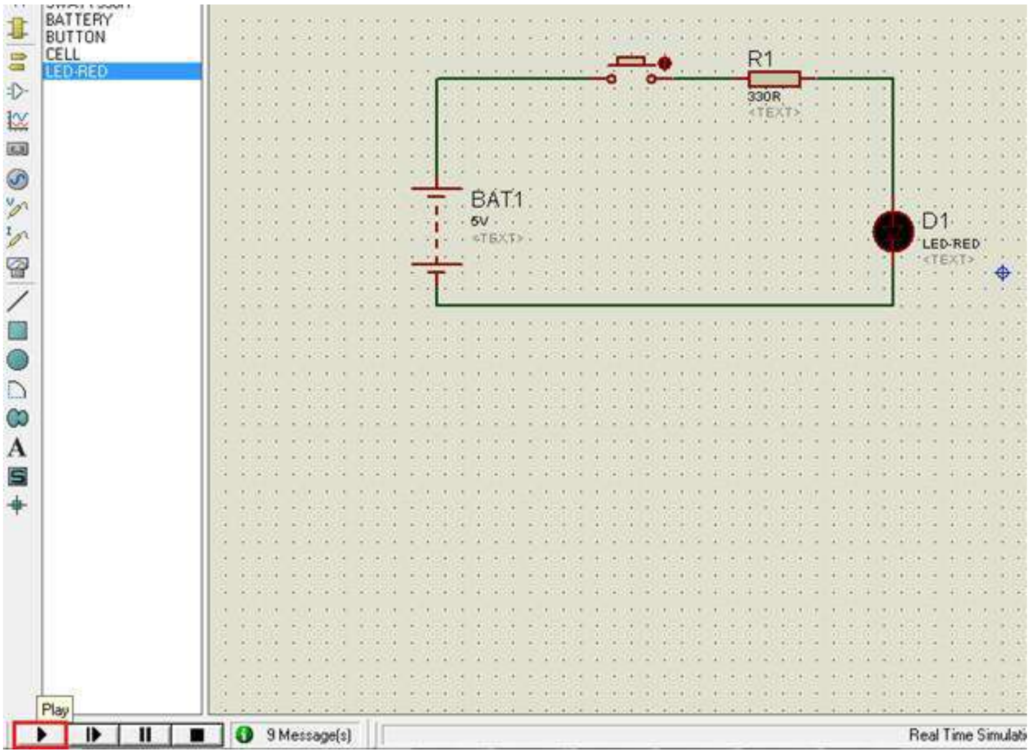


Fig.3.2.6.11Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

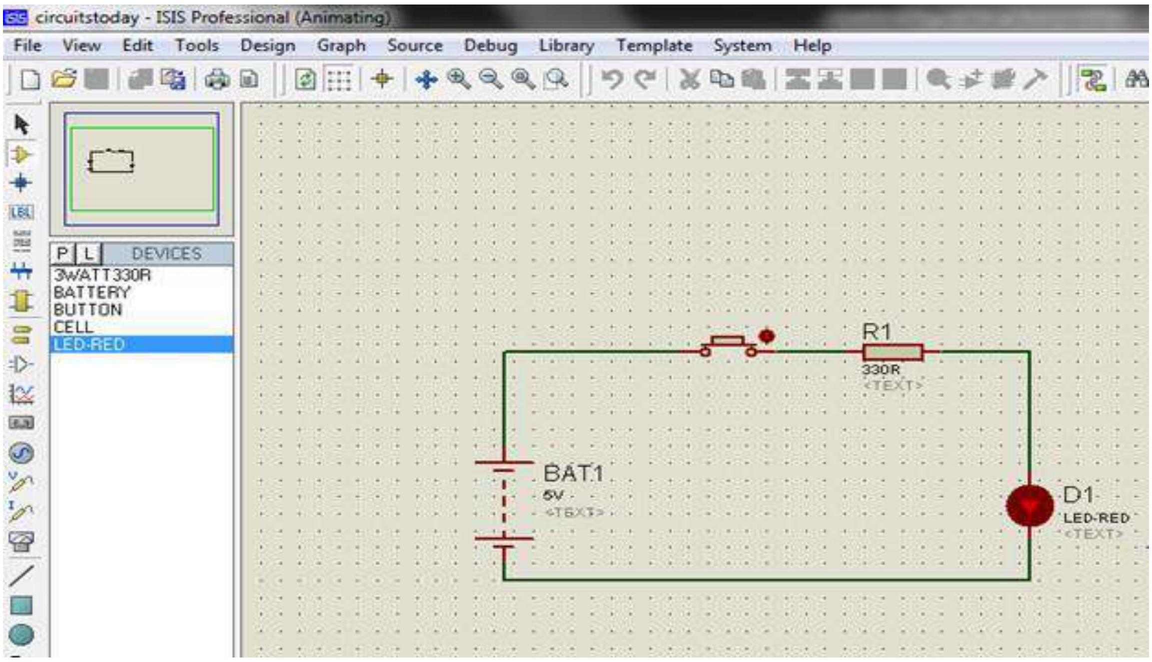


Fig. 3.2.6.12Simulation Animating

Simulation can be stepped, paused or stopped at any time.

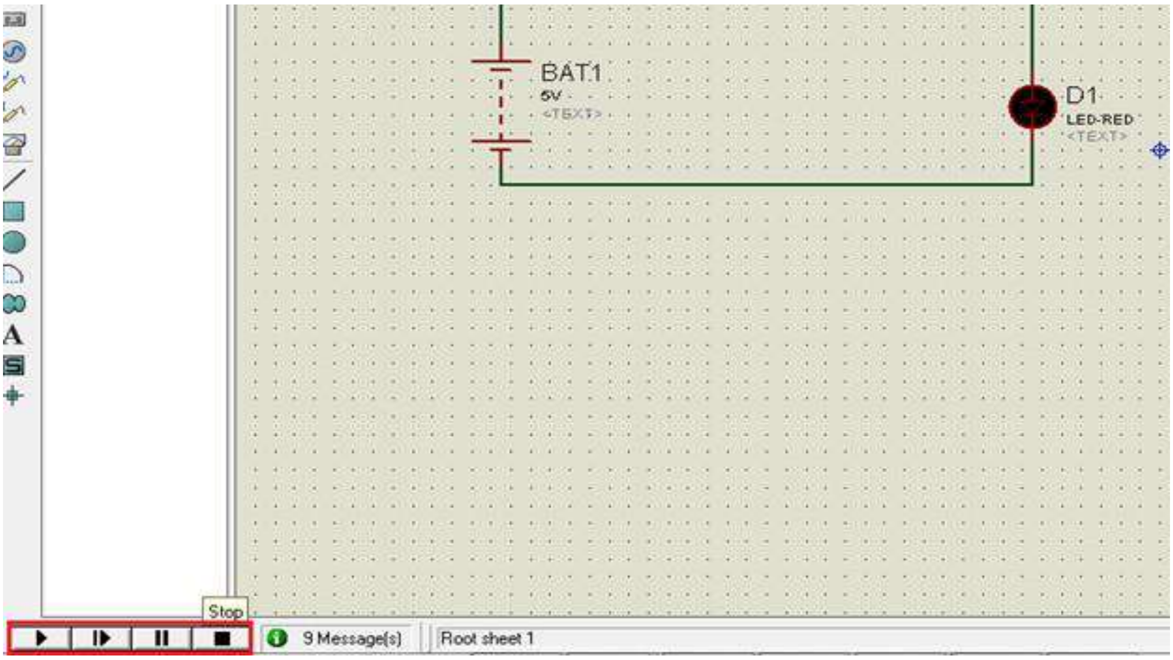


Fig.3.2.6.13Simulation Step-Pause-Stop Buttons

According to its developers, Things Speak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. Things Speak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates.

Things Speak was originally launched by iot Bridge in 2010 as a service in support of IoT applications.

Things Speak has integrated support from the numerical computing software MATLAB from MathWorks, allowing Things Speak users to analyse and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license from Math works.[citation needed].

Things Speak has a close relationship with Math works, Inc. In fact, all of the Things Speak documentation is incorporated into the Math works' Matlab documentation site and even enabling registered Math works user accounts as valid login credentials on the Things Speak website. The terms of service and privacy policy of ThingSpeak.com are between the agreeing user and Math works, Inc.

Setup Things Speak

Create a free MathWorks account or sign into Things Speak using an existing account.

Select the Things Speak channel you want your data to stream into. See Collect Data in a New Channel for help creating a new channel.

Record the following for the selected channel:

Channel ID, which is listed at the top of the channel view.

Write API key, which can be found on the API Keys tab of your channel view.

3.3.1 MQ3 SENSOR

3.3.1.1 INTRODUCTION

MQ3 is one of the most commonly used sensors in the MQ sensor series. It is a Metal Oxide Semiconductor (MOS) type of sensor. Metal oxide sensors are also known as Chemiresistors, because sensing is based on the change of resistance of the sensing material when exposed to alcohol. So by placing it in a simple voltage divider network, alcohol concentrations can be detected.

MQ3 alcohol sensor works on 5V DC and draws around 800mW. It can detect Alcohol concentrations anywhere from 25 to 500 ppm.

3.3.1.2 PIN CONFIGURATION

Pin name	**description**
VCC	This pin powers the module, typically the operating voltage is +5V
GND	Used to connect the module to system ground
DIGITAL OUT	You can also use this sensor to get digital output from this pin, by setting a threshold value using the potentiometer
ANALOG OUT	This pin outputs 0-5V analog voltage based on the intensity of the gas

3.3.1.3 SPECIFICATIONS OF MQ3 GAS SENSOR :

- Power requirements: 5 VDC @ ~165 mA (heater on) / ~60 mA (heater off)
- Current Consumption: 150mA
- DO output: TTL digital 0 and 1 (0.1 and 5V)
- AO output: 0.1- 0.3 V (relative to pollution), the maximum concentration of a voltage of about 4V
- Detecting Concentration: 0.05-10mg/L Alcohol
- Interface: 1 TTL compatible input (HSW), 1 TTL compatible output (ALR)
- Heater consumption: less than 750mW
- Operating temperature: 14 to 122 °F (-10 to 50°C)
- Load resistance: 200kΩ
- Sensitivity S: Rs(in air)/Rs(0.4mg/L Alcohol)≥5
- Sensing Resistance Rs: 2KΩ-20KΩ(in 0.4mg/l alcohol)
- Dimensions: 32 x 22 x 16 mm

.

Fig.3.3.1.3: MQ3 sensor

3.3.2 IGNITION SYSTEM

3.3.2.1 INTRODUCTION

The key ignition of an automobile serves two purposes. It ensures that only the person who carries the keys can start the car and drive it away.It not only starts the engine but activates all the other electronic and mechanical systems in the vehicle. It is connected to one of analog pic of arduino.

3.3.2.2 OBJECTIVE

We know that in case of Internal Combustion (IC) engines, combustion of air and fuel takes place inside the engine cylinder and the products of combustion expand to produce reciprocating motion of the piston. This reciprocating motion of the piston is in turn converted into rotary motion of the crank shaft through connecting rod and crank. This rotary motion of the crank shaft is in turn used to drive the generators for generating power. We also know that there are 4-cycles of operations viz.: suction; compression; power generation and exhaust. These operations are performed either during the 2-strokes of piston or during 4-strokes of the piston and accordingly they are called as 2-stroke cycle engines and 4-stroke cycle engines.

In case of petrol engines during suction operation, charge of air and petrol fuel will be taken in. During compression this charge is compressed by the upward moving piston. And just before the end of compression, the charge of air and petrol fuel will be ignited by means of the spark produced by means of for spark plug. And the ignition system does the function of producing the spark in case of spark ignition engines.

3.4 POWER SUPPLY

3.4.1 INTRODUCTION:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as –Regulated D.C Power Supply‖.

3.4.1.1 BLOCK DIAGRAM OF POWER SUPPLY:

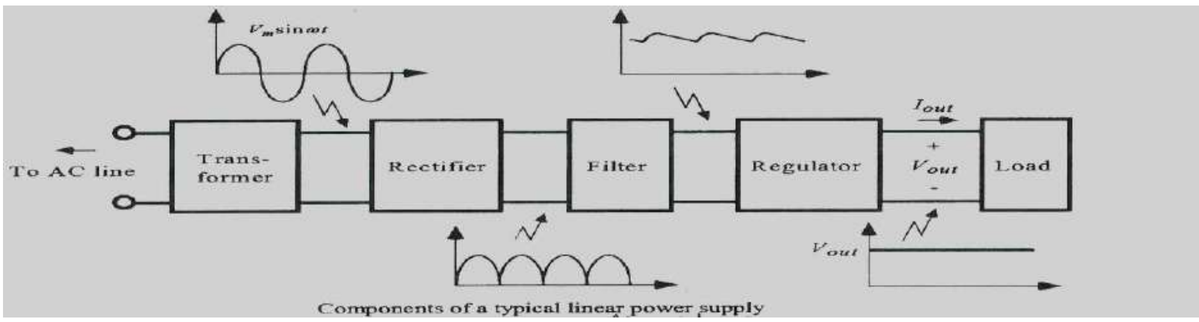


Fig.3.4.1.1.1: Block Diagram of Power Supply

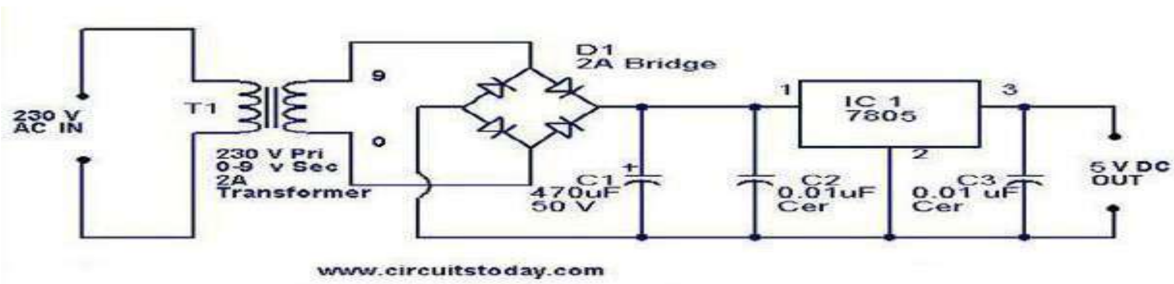


Fig.3.4.1.1.2(A): Schematic Diagram of Power Supply

3.4.2 DESCRIPTION OF POWER SUPPLY:

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a a different form of energy – such as solar, mechanical, or chemical - into electrical energy.

A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered. For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

3.4.3 TRANSFORMER:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

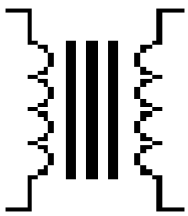


Fig.3.4.3.1: Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

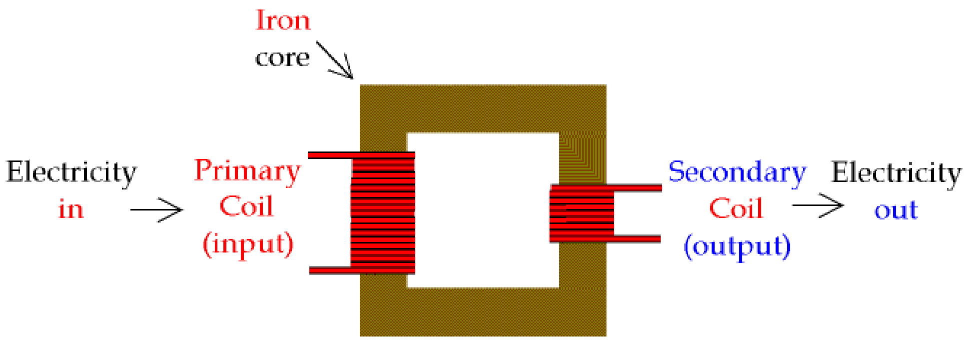


Fig.3.4.3.2: Transformer

3.4.3.1 Basic Principle of Transformer:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

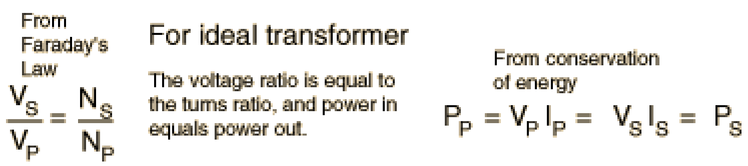


Figure 3.4.3.1.1 Basic principle of transformer

Working of Transformer:

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

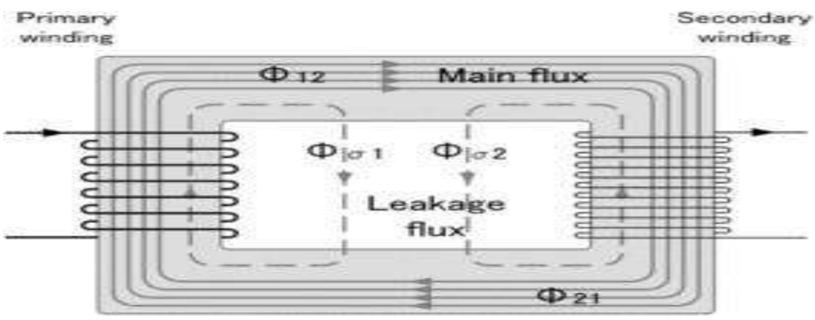


Fig.3.4.3.1.2: Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f. The iron core is laminated to prevent 'eddy currents' flowing in the core.

These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

3.4.3.2 Classification of Transformer:

- Step-Up Transformer
- Step-Down Transformer

Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn‘s ratio of the two sets of windings determines the amount of voltage transformation.

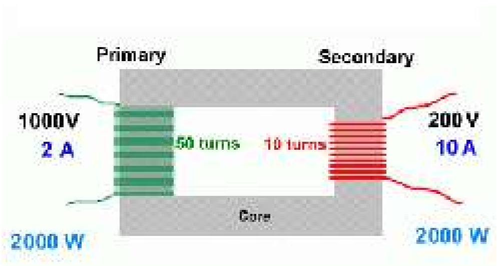


Fig.3.4.3.2.1: Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single phase steps down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase steps up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

Step-Up Transformer:

A step up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply. A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

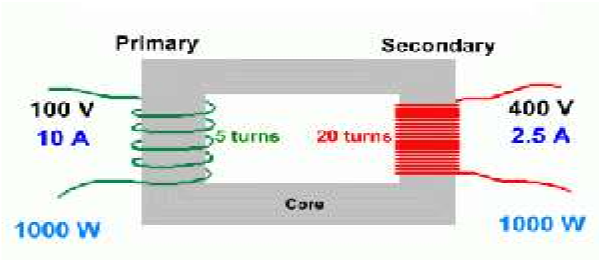


Fig. 3.4.3.2.2: Step-Up Transformer

3.4.4 DIODES:

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

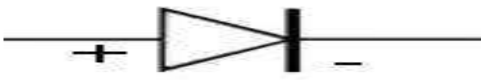


Fig.3.4.4.1: Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

3.4.5 Rectifier:

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

The Half-Wave Rectifier:

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure.

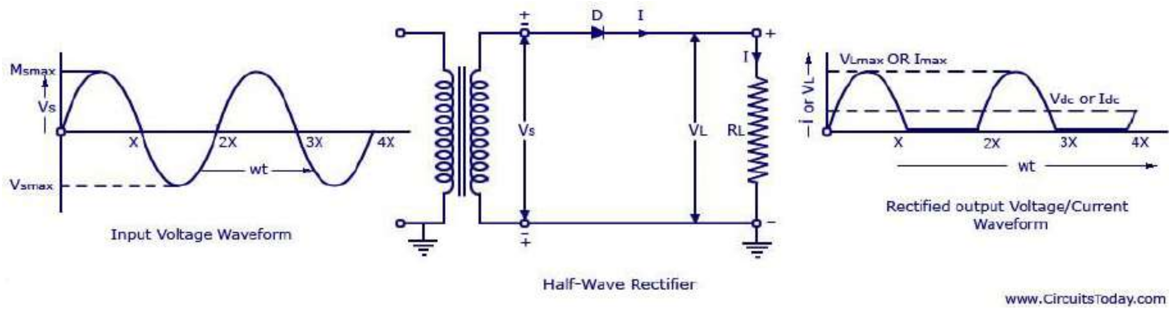


Fig.3.4.5.1: Half Wave Rectifier

Figure 2 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

The Full-wave Rectifier:

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC.

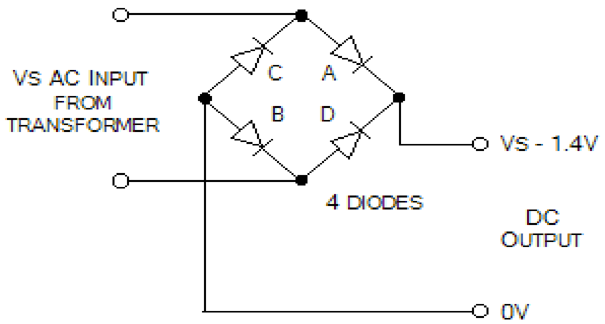


Fig.3.4.5.2: Full-Wave Rectifier

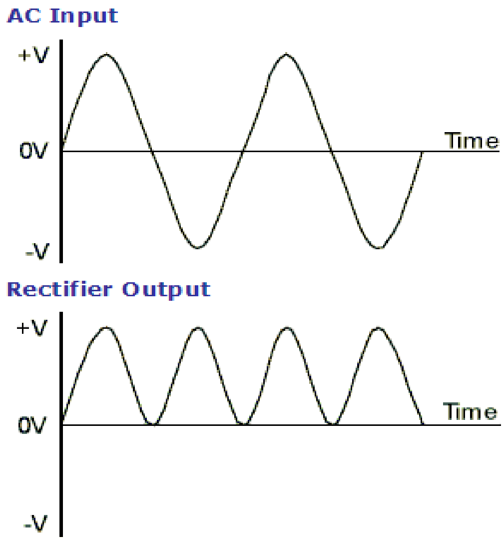


Fig.3.4.5.3(c): Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased. While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

3.4.6 CAPACITOR FILTER:

The capacitor-input filter, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

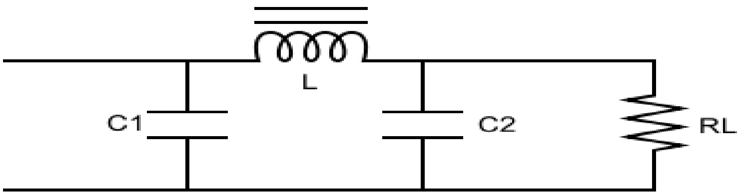


Fig.3.4.6.1: Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result, the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result, only the DC component appears across the load RL.

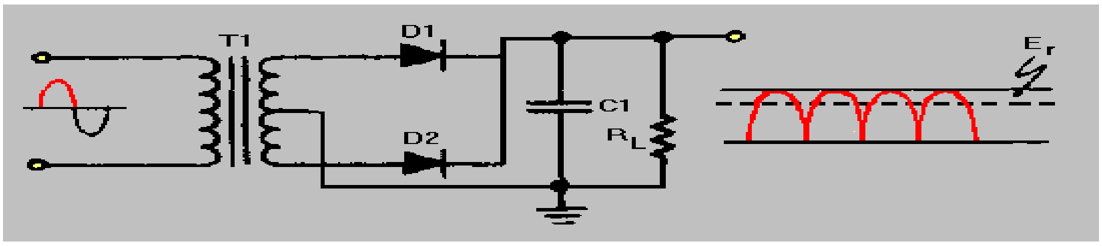


Fig. 3.4.6.2: Centered Tapped Full-Wave Rectifier with a Capacitor Filter

3.4.7 VOLTAGE REGULATOR:

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and _xx'indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:‘78‘ indicate the negative series and ‗xx‘indicates the voltage rating. Suppose 7905 produces the maximum -5V.‘05‘indicates the regulator output is -5V.

These regulators consist the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

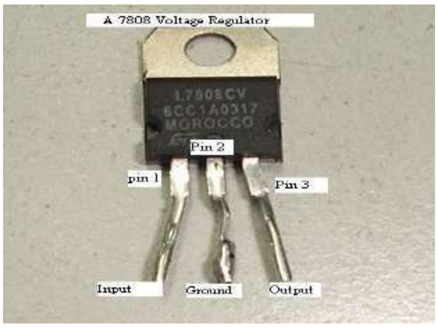


Fig.3.4.7: Regulator

CHAPTER-4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM OF PROJECT:

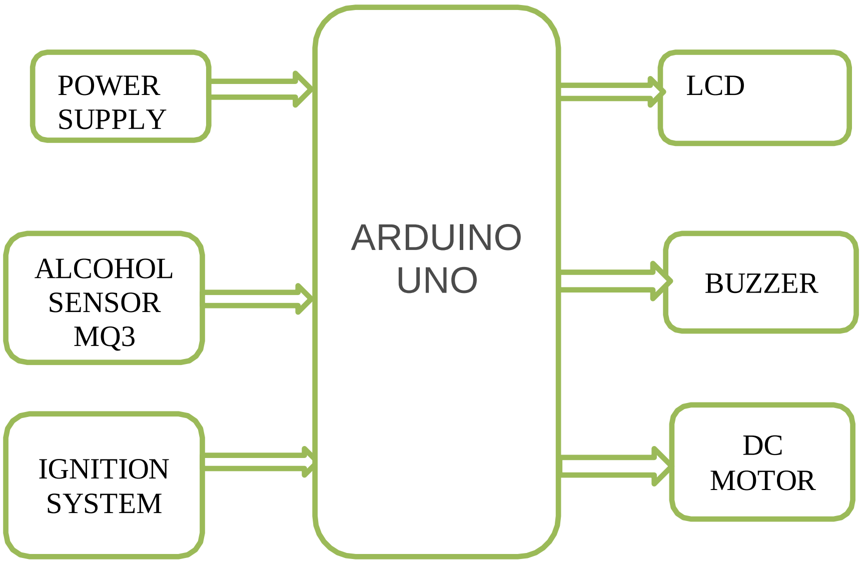


Fig.4.1: Block diagram of the project

4.2 Flow chart

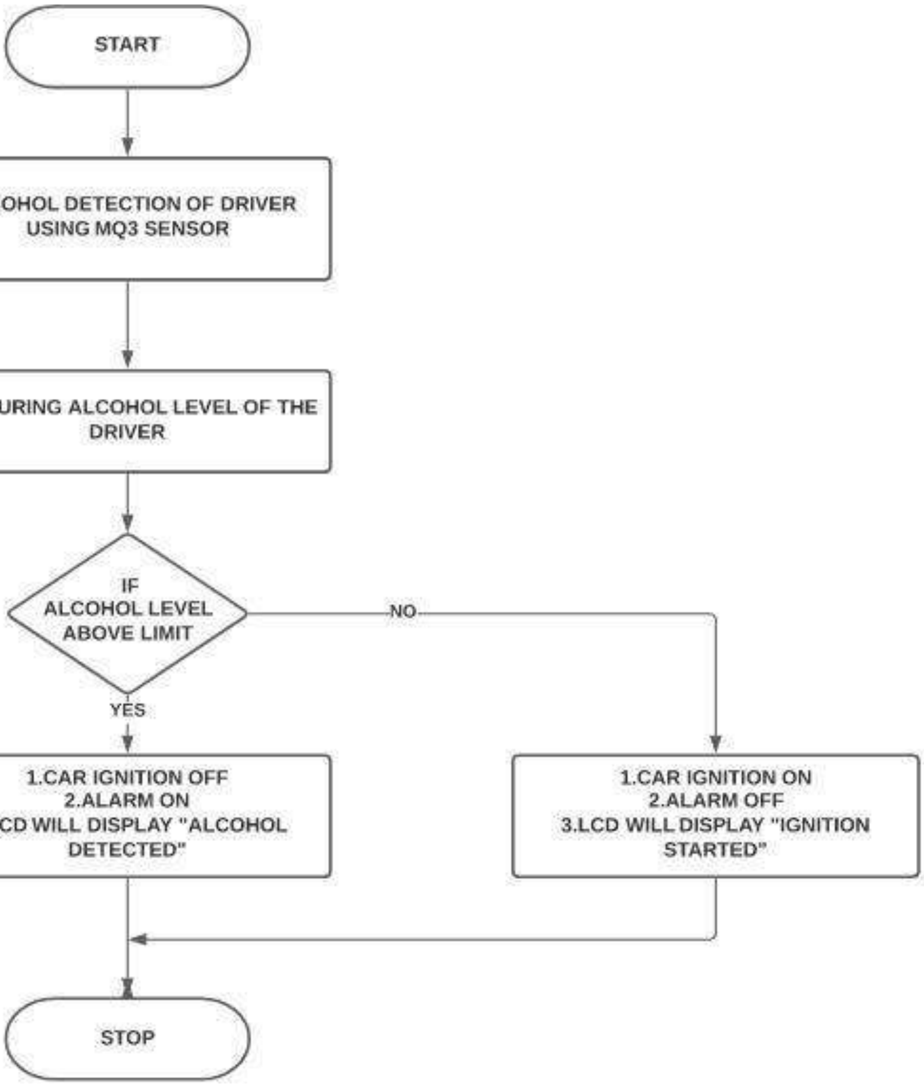


Fig.4.2 Flow Chart

4.3IMPLEMENTATION STAGES

Stage-1: Whenever,Alcohol sensor detects the alcohol it sends the data to the arduino and the LCD displays as :

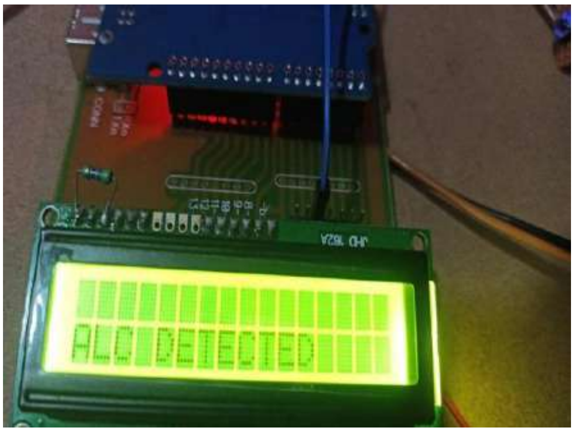


Fig.4.3.1 ALCOHOL DETECTED

Stage-2: Whenever,Alcohol is not detected by the alcohol sensor then the LCD displays as :

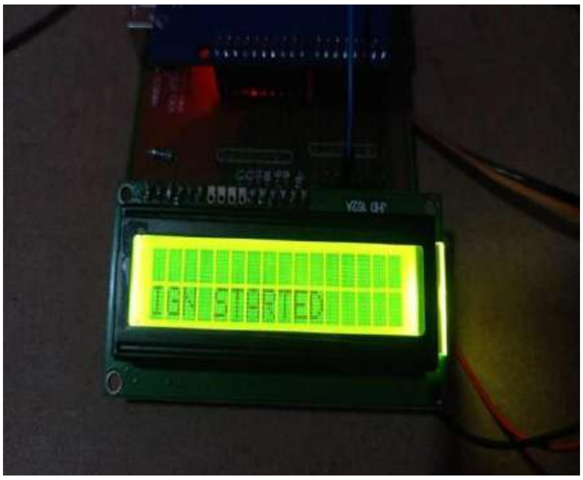


Fig.4.3.2 IGNITION STARTED

4.4 PROJECT CODE

```
#include <LiquidCrystal.h> //LCD

LiquidCrystal lcd(13,12,11,10,9,8);  //RS 13,EN 12, D4 11 ,D5 10,D6 9,D7 8
SoftwareSerial mySerial(2,3);//rx , tx  //GSM

int alc = A5;
int buz=7;
int relay=4;
int key=A4;

void setup(void)
{

  Serial.begin(9600);
  mySerial.begin(9600);
  lcd.begin(16,2);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("WELCOME");     // 1st romw 0th colom
  delay(1000);
  pinMode(alc, INPUT);
 pinMode(key, INPUT);
 pinMode(buz, OUTPUT);
 pinMode(relay, OUTPUT);
  digitalWrite(buz,HIGH);//OFF STATE
  digitalWrite(relay,LOW);

  delay(1000);

  lcd.clear();lcd.setCursor(0, 0);lcd.print("READY");

}
```

```
void loop()
{
   int alc_data = digitalRead(alc);
   int key_data = digitalRead(key);
   Serial.println(alc_data);
   lcd.clear();
   lcd.setCursor(0, 0); lcd.print("ALC DETECTION KEY ");
   delay(2000);
   lcd.clear();
   lcd.setCursor(0, 1); lcd.print("A:");lcd.print(alc_data);lcd.print("    ");
    lcd.setCursor(5, 1); lcd.print("K:");lcd.print(key_data);lcd.print("    ");
   delay(2000);

    if(alc_data == LOW)
    {
      lcd.clear();lcd.setCursor(0, 1);lcd.print("ALC DETECTED");
      mySerial.println("ALC DETECTED"); delay(500);
      digitalWrite(relay,LOW);
      digitalWrite(buz,LOW);
       delay(1000);
    }
    if(key_data == HIGH  )
    {
      lcd.clear();lcd.setCursor(0, 1);lcd.print("IGN STARTED");
      mySerial.println("IGN DETECTED"); delay(500);
      digitalWrite(relay,HIGH);
      digitalWrite(buz,HIGH);
       delay(1000);
   }
```

4.5 RESULTS

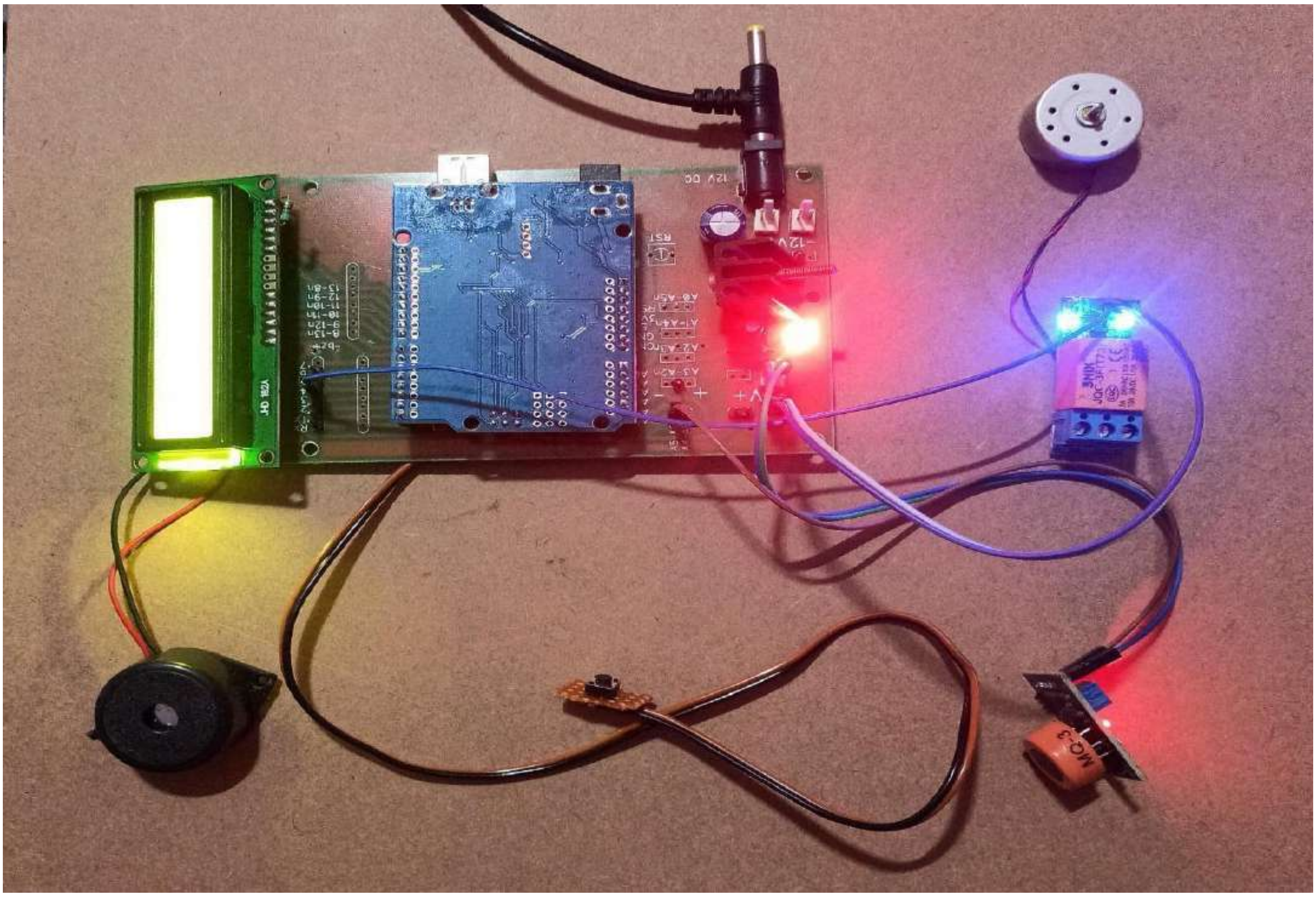


Fig.4.5 Final Output

CHAPTER-5

PROJECT TESTING

5.1 OVERVIEW OF TESTING METHODS

Here we propose a framework where the individual is identified for liquor level in his body to stay away from accidents. Drivers will be detected before they begin their vehicle. Driver will be detected by a sensor once he seated on the driver seat by his breath. Alcohol Sensor is put in the steering to screen the breath level if the liquor content in breath is 0.08% then car motor won't start. In this framework if the driver isn't drunk he can drive else he cannot drive until the point that the liquor content decreases. Arduino uno is arranged and associated with the sensor additionally LCD display and one dc motor is associated. Once the association is given power supply to it so the engine will start running .Now liquor is sprayed in it where the liquor content is over 0.08% so the LCD display will demonstrate that alcohol is detected, dc engine will stop International Journal of Pure and Applied Mathematics Special Issue 3002 running and ignition also stopped. This procedure is executed same in all vehicle where the car motor will be associated with the sensor. Once the alcohol sensor is detected,its output will be sent to the engine by refering the range motor will stop its execution.While implementing this proposed framework, it can decrease the mischances by 75% and reduce the loss of property and lives.

These are some more goals of this framework.

1. When driver starting car/vehicle then alcohol sensor begin detecting at condition vehicle speed equivalent to zero.

2. If alcoholic driver recognized then promptly ignition system will turn off and notification will be shown on LCD with alarm/buzzer.

3. A flag is set when first condition is passed without discovery of liquor.

4. If alcohol is recognized for this situation at that point signal is send to fuel blocker by Arduino for locking the start system.so driver feel's that vehicle is going to stop and after that he will place the car at proper location.

CHAPTER-6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

This device is useful in avoiding the accidents on roads to some extent. The detection of alcohol and making to know the levels of alcohol that is taken by the person will be useful stat to the driver so he can be cautious while taking the alcohol further during driving a vehicle. The location of the vehicle can be traced if a GPRS module is used along with the GSM module. This device can help in reducing the road accidents and also helps saving the lives of the people. This can play a major role in reducing the drunken driving cases. This can save the lives of innocent people who will get hit by the drunken drivers with their vehicles. Installing this device in every vehicle will reduce drunken driving case to large extent.

So, we can conclude that this type of devices and more advanced devices other than these can be very useful. These types of devices are cost efficient and can be maintained easily. These types of devices plays prominent role in avoiding accidents due to drunken driving. This will be one of the better ways in reducing the drunken driving cases. This is one of road safety measure that every person who drives a vehicle should practice for their safety and other peoples too. Life is important; we have to not waste it. –SAVING A INNOCENT PERSON IS SAVING HIS FAMILY‖

6.2 FUTURE ENHANCEMENT

In future this type of device can be developed with more accuracy and efficiently. This type of devices can be installed in the vehicles during the manufacturing of the vehicles itself. This can assure the safety for the customer and can versatile feature in the automobile industry for sure. We can connect the cameras to such devices and these cameras can be used to detect the drivers who are drunk and cameras can be used as theft alarms for the vehicles. We can protect the vehicles from getting stolen and even can avoid drunken driving accidents. We can connect the GPRS module to trace the location the vehicle and can be helpful in intimating the person about location of the vehicle which is drove by the drunk driver. We can find the vehicle if stolen by tracing the location of the vehicle when necessary precautions are taken and proper hidden GPRS modules are installed in the vehicle. We can modify this type of device by connect different modules like fire sensing alarms, sounds systems can be connected etc., can be added to device. Low power consumption devices will be efficient and working of device in any condition can be taken care.

So, this type of devices can be used as multiple purpose devices. Hence, there will be large scope in developing such kind of device and cost efficiency is must.

PUBLICATION

Submitted Paper in the Conference ICISECT - 21 with Paper ID (ICISECT21-0043) and got Acceptance for the Paper.

REFERENCES

- [1] Mandalkar Rahul B,Pandore Rahul N,Shinde Manoj B,Godse Valmik D, ‒Alcohol Detection and Accident Avoidance Using Locking With Tracking‖.

- [2] Abhi R. Varma, Seema V. Arote, Chetna Bharti, Kuldeep Singh, ‒Accident Prevention Using Eye Blinking and Head Movement‖. [3] Dhivya M and Kathiravan S, ‒Driver Authentication and Accident Avoidance System for Vehicles‖ [4] Road accidents in India [online] 2007 June 25. Available from: URL: http://www.easydriveforum.com/f44-share-yourroadexperience/road-accidents-in-india-834.html. [5] Articles base directory [online] 2011 Feb. 16 Available from:URL:http://www.dwworld.de/dw/article/0,,55 19345,00.html. [6] S.P. Bhumkar, V.V. Deotare, R.V. Babar, ‒Accident avoidance and detection on highways‖. [7] .Mandalkar Rahul B,Pandore Rahul N,Shinde Manoj B,Godse Valmik D., " Alcohol Detection and Accident Avoidance Using Locking With Tracking "Volume 3, Issue 9, September 2015 International Journal of Advance Research in Computer Science and Management Studies. [8] Abhi R. Varma, Seema V. Arote, Chetna Bharti, Kuldeep Singh, "Accident Prevention Using Eye Blinking and Head Movement" IJCA Proceedings on Emerging Trends in Computer Science and Information Technology - 2012(ETCSIT2012) etcsit1001 ETCSIT(4):31-35, April 2012. [9] Shan Zhu and Hanghui Zhu,Mi Zhou,‖A Design of Vehicle Collision Avoidance System Based On DSP‖,International Conference of Mechatronics andAutomation,IEEE-2009,August-2-12-, Changchun China

BIBILOGRAPHY

[1] Mandalkar Rahul B,Pandore Rahul N,Shinde Manoj B,Godse Valmik D, –Alcohol Detection and Accident Avoidance Using Locking With Tracking‖. [2] Abhi R. Varma, Seema V. Arote, Chetna Bharti, Kuldeep Singh, –Accident Prevention Using Eye Blinking and Head Movement‖. [3] Dhivya M and Kathiravan S, –Driver Authentication and Accident Avoidance System for Vehicles‖ [4] Road accidents in India [online] 2007 June 25. Available from: URL: http://www.easydriveforum.com/f44-share-yourroadexperience/road-accidents-in-india-834.html. [5] Articles base directory [online] 2011 Feb. 16 Available from:URL:http://www.dwworld.de/dw/article/0,,55 19345,00.html. [6] S.P. Bhumkar, V.V. Deotare, R.V. Babar, –Accident avoidance and detection on highways‖. [7] .Mandalkar Rahul B,Pandore Rahul N,Shinde Manoj B,Godse Valmik D., " Alcohol Detection and Accident Avoidance Using Locking With Tracking "Volume 3, Issue 9, September 2015 International Journal of Advance Research in Computer Science and Management Studies. [8] Abhi R. Varma, Seema V. Arote, Chetna Bharti, Kuldeep Singh, "Accident Prevention Using Eye Blinking and Head Movement" IJCA Proceedings on Emerging Trends in Computer Science and Information Technology -2012(ETCSIT2012) etcsit1001 ETCSIT(4):31-35, April 2012. [9] Shan Zhu and Hanghui Zhu,Mi Zhou,‖A Design of Vehicle Collision Avoidance System Based On DSP‖,International Conference of Mechatronics and

APPENDICES

Driving while drunk is hazardous and drivers with high Blood Alcohol Content (BAC) are at expanded danger of auto crashes, roadway wounds and vehicular passings. Anticipation measures assessed incorporate permit suspension or disavowal, appropriating or seizing vehicle plates, implementing open holder bans, expanding fine punishments, imprison, ordering instruction for youth and bringing down legitimate BAC's. Despite the fact that these much obstacles made by experts to drunken drive, it is as yet proceeding with like serial scenes. In that capacity there is no viable instrument to reduce this.

Here, this process have intended to plan a Drunk driving detection, which is integrated with the directing wheel. This framework is meant for making vehicle driving more secure than previously and shield the mishaps from happening due to the liquor utilization of the driver. The individual when he is at vehicle, this is necessary to infer the driver's condition continuously and here this work proposed the detection of alcohol utilizing alcohol sensor associated with Arduino. Alcohol sensor is installed on the steering of the car, with the end goal that when the level of liquor crosses a reasonable farthest point, where the start of vehicle will kill and the motor will stop. The Arduino always uses the alcohol sensor information to check drunk driving and works bolt on the vehicle motor to stop the engine.

A

MAJOR PROJECT REPORT ON

AN EXPERIMENTAL COMPARISION OF DIFFERENT OBJECT TRACKING ALGORTHMS

Submitted by

1) Mr. P. Rahul (17K81A04N9) **2) Mr. K. Jagadeeshwar (17K81A04L7)**

3) Mr. A. Anil Kumar (17K81A04J2)

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Esteemed Guidance of

Mr. M. Thirupathi

Associate Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENIGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 10

June 2021

BONAFIDE CERTIFICATE

This is to certify that the project entitled "AN EXPERIMENTAL COMPARISION OF DIFFERENT OBJECT TRACING ALGORITHS", is being submitted by 1.**Ms.P.Rahul (17K81A04N9), 2.Mr.K. Jagadeeshwar (17K81A04L7), 3.Mr. Anil Kumar (17K81A04J2)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OFTECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

	Head of the Department
Mr. M. THIRUPATHI	**Dr. B. HARI KRISHNA**
Department of ECE	Department of ECE

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT P RAHUL WITH ROLL NO.17K81A04N9, K JAGADEESHWAR WITH ROLL NO.17K81A04L7, A. ANIL KUMAR WITH ROLL NO.17K81A04J2, OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "IOT BASED TOLL BOOTH MANAGER SYSTEM" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email :contact@lasyainfotech.com,ov@lasyainfotech.com
Website :www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of **Bachelor of Technology** in Department of ELECTRONICS AND COMMUNICATION ENGINEERING, session: <2017 – 2021>, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled "AN EXPERIMENTAL COMPARISON OF DIFFERENT OBJECT TRACKING ALGORITHMS" is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

K. Jagadeeshwar (17K81A04L7)

A. Anil Kumar (17K81A04J2)

P. Rahul (17K81A04N9)

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr.P.SANTOSH KUMAR PATRA, our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr. B.HARI KRISHNA, Head &Professor,** Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in our memory for years to come.

We would like to express our gratitude to our dept. project coordinator **Dr. A.ANAND, Professor** for his involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time to come.

Finally, we are highly grateful and obliged to our project guide **Mr. M.Thirupathi, Associate Professor,** for his advice, supervision, and the vital contribution as and when required during this research work and for their support and cooperation that is difficult to express in words.

K. Jagadeeshwar (17K81A04L7)

A. Anil Kumar (17K81A04J2)

P. Rahul (17K81A04N9)

ABSTRACT

- This paper reviews and evaluates several state of the art object tracking algorithms. It compares the various techniques on different parameters. These algorithms are used to recognize, detect and track objects under the influence of different problems like occluded object, background clutter, illumination variations, and deformed objects or in shadowed regions.

- The target object is simultaneously tracked and learned which will help make the algorithm robust against failures. It works by working across every pixel of a given new frame but only surrounds a certain area of the previous location. Then the score is recorded of this classification algorithm for each comparison

- We will be discussing all the overall results of the different algorithms employing tables and plots. The overall performance can be a combined output of the graphs measuring precision and success. The future scope of this research includes a comprehensive examination of the application of these algorithms, along with the calculation of their efficiencies and the use of different techniques.

TABLE OF CONTENTS

List of Figures

CHAPTER-1
INTRODUCTION

OBJECT tracking in video streams and object detection and classification in still images, as well as videos, is an active and prominent research area in the field of computer vision. It has a plethora of applications in the real world which include surveillance, traffic control and analysis, face detection, medical image processing and many more

[1]. While many advances and breakthroughs in this field over recent years have leaded to the development of a huge set of unique algorithms it still remains to be a very challenging problem. Due to the enormous set of environmental factors and other variables, it is next to impossible to develop a universal tracking algorithm. Also, the selection of the most suited algorithm doesn't depend only on the concept of the algorithm but also on its implementation. Therefore, we compare some of the readily available algorithms on several challenging video sequences with different evaluation criteria using the Open CV library

[2]. The algorithms compared are easily found in the OpenCV library of python. It is a widely used library, with over 3000 algorithms embedded in it with all the needful structures and tools for object tracking algorithms. On one hand, detection failures are inevitable, which include the miss detection where a target is misclassified as the background, the false detection where a background region is incorrectly recognized as a target, and the occlusion where an object is partially or fully invisible because of the limited camera view. Therefore, there exists misalignment between trajectories and detections during the data association where not every trajectory or detection can find its correspondence.

This paper will help in observing the real-world implementations of the many algorithms found in this broadly used library. This evaluation and comparison discussed in this paper will not only be useful in selecting an appropriate method for specific applications but also be beneficial for researching and developing new tracking algorithms. The remaining paper proceeds as follows: Section II gives a brief explanation of the different algorithms used. Followed by Section III which presents the various evaluation metrics against which the algorithms where compared. Section IV displays the results and the success and precision plots. Section V has the conclusion of the paper.

On the other hand, in real-world scenarios, targets may appear and disappear anytime and anywhere in the scene. We need to automatically tackle the initializations and terminations of trajectories to accommodate dynamic target changes. All these complex scenarios make the data association challenging. To deal with such complex scenarios, various algorithms have been proposed in the past decade. Detection failures are addressed by the continuous confidence output along with Particle Filter and the explicit occlusion reasoning.

Alternatively, they can be addressed in a global temporal window using the network flow , high-order energy minimization, and hierarchical data association. Furthermore, it is often assumed that new targets may enter or move out of the scene from some certain areas, e.g., the border of the camera view, in order to deal with trajectory initializations and terminations.

However, these data association models are either difficult to estimate (e.g., explicit occlusion reasoning) from a single camera view or subject to high computational burdens, restricting their adaptations to time-critical scenarios (e.g., surveillance). Additional reliance on the scene prior knowledge may also limit their adaptability to wider application areas. In this paper, we propose an online multi-target tracking method with unified handling of aforementioned complex scenarios.

1.2 Object Representation

In a tracking scenario, an object can be defined as anything that is of interest for further analysis. For instance, boats on the sea, fish inside an aquarium, vehicles on a road, planes in the air, people walking on a road, or bubbles in the water are a set of objects that may be important to track in a specific domain. Objects can be represented by their shapes and appearances. In this section, we will first describe the object shape representations commonly employed for tracking and then address the joint shape and appearance representations.

Points - The object is represented by a point, that is, the centroid (Figure 1(a)) or by a set of points (Figure 1(b)). In general, the point representation is suitable for tracking objects that occupy small regions in an image.

1.3 PRIMITIVE GEOMETRIC SHAPES

Object shape is represented by a rectangle, ellipse (Figure 1(c), (d), etc. Object motion for such representations is usually modeled by translation, affine, or projective (homography) transformation (see Section 5.2 for details). Though primitive geometric shapes are more suitable for representing simple rigid objects, they are also used for tracking non rigid objects.

1.3.1 Object silhouette and contour

Contour representation defines the boundary of an object (Figure 1(g), (h). The region inside the contour is called the silhouette of the object (see Figure 1(i)). Silhouette and contour representations are suitable for tracking complex non rigid shapes.

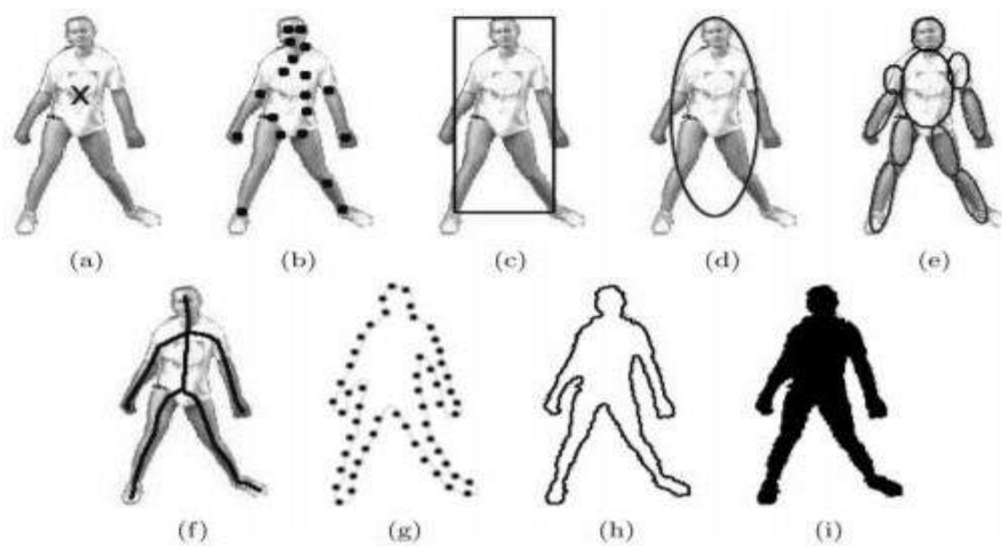


Fig. 1.1. Object representations. (a) Centroid, (b) multiple points, (c) rectangular patch, (d) elliptical patch, (e) part-based multiple patches, (f) object skeleton, (g) complete object contour, (h) control points on object contour, (i) object silhouette.

1.3.2 Articulated shape models

Articulated objects are composed of body parts that are held together with joints. For example, the human body is an articulated object with torso, legs, hands, head, and feet connected by joints. The relationship between the parts is governed by kinematic motion models, for example, joint angle, etc. In order to represent an articulated object, one can model the constituent parts using cylinders or ellipses as shown in Figure 1(e).

1.3.2 Skeletal models

Object skeleton can be extracted by applying medial axis transform to the object silhouette. This model is commonly used as a shape representation for recognizing objects. Skeleton representation can be used to model both articulated and rigid objects (see Figure 1(f)).

There are a number of ways to represent the appearance features of objects. Note that shape representations can also be combined with the appearance representations for tracking. Some common appearance representations in the context of object tracking are:

1.3.3 Probability densities of object appearance

The probability density estimates of the object appearance can either be parametric, such as Gaussian and a mixture of Gaussians or nonparametric, such as Parzen windows and histograms. The probability densities of object appearance features (color, texture) can be computed from the image regions specified by the shape models (interior region of an ellipse or a contour).

1.3.4 Templates

Templates are formed using simple geometric shapes or silhouettes. An advantage of a template is that it carries both spatial and appearance information. Templates, however, only encode the object appearance generated from a single view. Thus, they are only suitable for tracking objects whose poses do not vary considerably during the course of tracking.

1.3.5 Active appearance models

Active appearance models are generated by simultaneously modeling the object shape and appearance. In general, the object shape is defined by a set of landmarks. Similar to the contour-based representation, the landmarks can reside on the object boundary or, alternatively, they can reside inside the object region. For each landmark, an appearance vector is stored which is in the form of color, texture, or gradient magnitude. Active appearance models require a training phase where both the shape and its associated appearance is learned from a set of samples using, for instance, the principal component analysis.

1.3.5 Multiview appearance models

These models encode different views of an object. One approach to represent the different object views is to generate a subspace from the given views. Subspace approaches, for example, Principal Component Analysis (PCA) and Independent Component Analysis (ICA), have been used for both shape and appearance representation.

Another approach to learn the different views of an object is by training a set of classifiers, for example, the support vector machines or Bayesian networks. One limitation of multiview appearance models is that the appearances in all views are required ahead of time.

In general, there is a strong relationship between the object representations and the tracking algorithms. Object representations are usually chosen according to the application domain. For tracking objects, which appear very small in an image, point representation is usually appropriate. For instance, use the point representation to track the seeds in a moving dish sequence. Similarly, Shafique and Shah [2003] use the point representation to track distant birds. For the objects whose shapes can be approximated by rectangles or ellipses, primitive geometric shape representations are more appropriate use an elliptical shape representation and employ a color histogram computed from the elliptical region for modeling the appearance. In 1998, Black and Jepson used eigenvectors to represent the appearance. The eigenvectors were generated from rectangular object templates. For tracking objects with complex shapes, for example, humans, a contour or a silhouette based representation is appropriate use silhouettes for object tracking in a surveillance application.

1.4 FEATURE SELECTION FOR TRACKING

Selecting the right features plays a critical role in tracking. In general, the most desirable property of a visual feature is its uniqueness so that the objects can be easily distinguished in the feature space. Feature selection is closely related to the object representation. For example, color is used as a feature for histogram-based appearance representations, while for contour-based representation, object edges are usually used as features. In general, many tracking algorithms use a combination of these features. The details of common visual features are as follows.

—Color. The apparent color of an object is influenced primarily by two physical factors,

1) The spectral power distribution of the illuminant and

2) The surface reflectance properties of the object. In image processing, the RGB (red, green, blue) color space is usually used to represent color. However, the RGB space is not a perceptually uniform color space, that is, the differences between the colors in the RGB space do not correspond to the color differences perceived by humans. Additionally, the RGB dimensions

are highly correlated. In contrast, L∗u∗v∗ and L∗a∗b∗ are perceptually uniform color spaces, while HSV (Hue, Saturation and Value) is an approximately uniform color space. However, these color spaces are sensitive to noise. In summary, there is no last word on which color space is more efficient, therefore a variety of color spaces have been used in tracking.

—Edges. Object boundaries usually generate strong changes in image intensities. Edge detection is used to identify these changes. An important property of edges is that they are less sensitive to illumination changes compared to color features. Algorithms that track the boundary of the objects usually use edges as the representative feature. Because of its simplicity and accuracy, the most popular edge detection approach is the Canny Edge detector. An evaluation of the edge detection algorithms is provided by Bowyer et al.

—Optical Flow. Optical flow is a dense field of displacement vectors which defines the translation of each pixel in a region. It is computed using the brightness constraint, which assumes brightness constancy of corresponding pixels in consecutive frames. Optical flow is commonly used as a feature in motion-based segmentation and tracking applications. Popular techniques for computing dense optical flow include methods by Horn and Schunck, Lucas and Kanade, Black and Anandan, and Szeliski and Couglan. For the performance evaluation of the optical flow methods, we refer the interested reader to the survey by Barron et al.

—Texture. Texture is a measure of the intensity variation of a surface which quantifies properties such as smoothness and regularity. Compared to color, texture requires a processing step to generate the descriptors. There are various texture descriptors: Gray-Level Cooccurrence Matrices (GLCM's) (a 2D histogram which shows the cooccurrences of intensities in a specified direction and distance), Law's texture measures (twenty-five 2D filters generated from five 1D filters corresponding to level, edge, spot, wave, and ripple), wavelets (orthogonal bank of filters), and steerable pyramids. Similar to edge features, the texture features are less sensitive to illumination changes compared to color.

CHAPTER-2

MATLAB

2.1 What is MATLAB?

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth- generation programming language. A proprietary programming language developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python. Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

The name MATLAB stands for matrix laboratory. It was invented in the late 1970s by Cleve Moler, then chairman of the computer science department at the University of New Mexico. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development and analysis. MATLAB was first adopted by control design engineers, little's specialty, but quickly spread to many other domains. It is now also used in education, in particular the teaching of linear algebra and numerical analysis, and is popular amongst scientists involved with image processing.

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.

Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran.

2.2 Introduction of MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building.

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrixlaboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

2.3 The MATLAB System

The MATLAB system consists of five main parts:

a. Development Environment:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

b. The MATLAB Mathematical Function:

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigen values, Bessel functions, and fast Fourier transforms.

c. The MATLAB Language:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

d. Graphics:

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

e. The MATLAB Application Program Interface (API):

This is a library that allows you to write C and Fortran programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files

2.4 MATLAB Working Environment

Matlab Desktop is the main Mat lab application window. The desktop contains five sub windows, the command window, the workspace browser, the current directory window, the command history window, and one or more figure windows, which are shown only when the user displays a graphic.

The command window is where the user types MATLAB commands and expressions at the prompt (>>) and where the output of those commands is displayed. MATLAB defines the workspace as the set of variables that the user creates in a work session. The workspace browser shows these variables and some information about them. Double clicking on a variable in the workspace browser launches the Array Editor, which can be used to obtain information and income instances edit certain properties of the variable.

The current Directory tab above the workspace tab shows the contents of the current directory, whose path is shown in the current directory window. For example, in the windows operating system the path might be as follows: C:\MATLAB\Work, indicating that directory “work” is a subdirectory of the main directory “MATLAB”; WHICH IS INSTALLED IN DRIVE C. clicking on the arrow in the current directory window shows a list of recently used paths. Clicking on the button to the right of the window allows the user to change the current directory.

MATLAB uses a search path to find M-files and other MATLAB related files, which are organize in directories in the computer file system. Any file run in MATLAB must reside in the current directory or in a directory that is on search path. By default, the files supplied with MATLAB and math works toolboxes are included in the search path. The easiest way to see which directories are on the search path. The easiest way to see which directories are soon the search path, or to add or modify a search path, is to select set path from the File menu the desktop, and then use the set path dialog box. It is good practice to add any commonly used directories to the search path to avoid repeatedly having the change the current directory.

The Command History Window contains a record of the commands a user has entered in the command window, including both current and previous MATLAB sessions. Previously entered MATLAB commands can be selected and re-executed from the command history window by right clicking on a command or sequence of commands. This action launches a menu from which to select various options in addition to executing the commands. This is useful to select various options in addition to executing the commands. This is a useful feature when experimenting with various commands in a work session.

2.5 Using the MATLAB Editor to create M-Files:

The MATLAB editor is both a text editor specialized for creating M-files and a graphical MATLAB debugger. The editor can appear in a window by itself, or it can be a sub window in the desktop. M-files are denoted by the extension .m, as in pixelup.m. The MATLAB editor window has numerous pull-down menus for tasks such as saving, viewing, and debugging files. Because it performs some simple checks and also uses color to differentiate between various elements of code, this text editor is recommended as the tool of choice for writing and editing M-functions. To open the editor , type edit at the prompt opens the M-file filename.m in an editor window, ready for editing. As noted earlier, the file must be in the current directory, or in a directory in the search path.

2.5.1 Getting Help

The principal way to get help online is to use the MATLAB help browser, opened as a separate window either by clicking on the question mark symbol (?) on the desktop toolbar, or by typing help browser at the prompt in the command window. The help Browser is a web browser integrated into the MATLAB desktop that displays a Hypertext Markup Language(HTML) documents. The Help Browser consists of two panes, the help navigator pane, used to find information, and the display pane, used to view the information. Self-explanatory tabs other than navigator pane are used to perform a search.

2.6 History of MATLAB

Cleve Moler, the chairman of the computer science department at the University of New Mexico, started developing MATLAB in the late 1970s.He designed it to give his students access to LINPACK and EISPACK without them having to learn Fortran. It soon spread to other universities and found a strong audience within the applied mathematics community.

Jack Little, an engineer, was exposed to it during a visit Moler made to Stanford University in 1983. Recognizing its commercial potential, he joined with Moler and Steve Bangert. They rewrote MATLAB in C and founded MathWorks in 1984 to continue its development. These rewritten libraries were known as JACKPAC. In 2000, MATLAB was rewritten to use a newer set of libraries for matrix manipulation, LAPACK

MATLAB was first adopted by researchers and practitioners in control engineering, Little's specialty, but quickly spread to many other domains. It is now also used in education, in particular the teaching of linear algebra, numerical analysis, and is popular amongst scientists involved in image processing. With 2004 marking the 20th anniversary of The MathWorks, it’s a good time to look back at the origins of MATLAB.MATLAB is now a full-featured technical computing environment, but it started as a simple “Matrix Laboratory.”

Three men, J. H. Wilkinson, George Forsythe, and John Todd, played important roles in the origins of MATLAB. Our account begins more than 50 years ago.

Wilkinson was a British mathematician who spent his entire career at the National Physical Laboratory (NPL) in Teddington, outside London.Working on a simplified version of a sophisticated design by Alan Turing, Wilkinson and colleagues at NPL built the Pilot Automatic Computing Engine (ACE), one of Britain's first stored-program digital computers. The Pilot ACE ran its first program in May 1950. Wilkinson did matrix computations on the machine and went on to become the world's leading authority on numerical linear algebra.

2.7 Strengths of MATLAB ?

- MATLAB is relatively easy to learn
- MATLAB code is optimized to be relatively quick when performing matrix operations
- MATLAB may behave like a calculator or as a programming language
- MATLAB is interpreted, errors are easier to fix

Although primarily procedural (for example: *C*), MATLAB does have some object-oriented elements(for example: *C++*)

2.8 Key Features

High-level language for numerical computation, visualization, and application development

- Interactive environment for iterative exploration, design, and problem solving
- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration, and solving ordinary differential equations
- Built-in graphics for visualizing data and tools for creating custom plots
- Development tools for improving code quality and maintainability and maximizing performance
- Tools for building applications with custom graphical interfaces

- Functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET, and Microsoft Excel

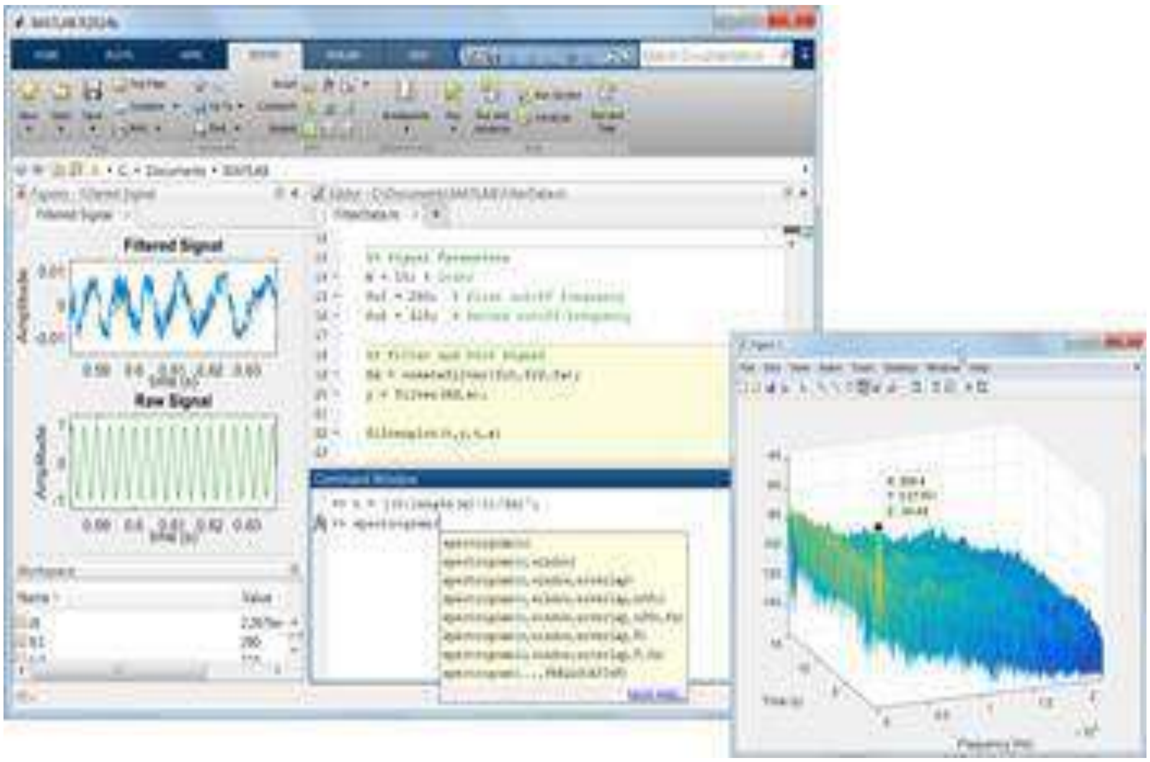


Figure: 2.1

Analyzing and visualizing data using the MATLAB desktop. The MATLAB environment also lets you write programs and develop algorithms and applications.

2.9 Numeric Computation

MATLAB provides a range of numerical computation methods for analyzing data, developing algorithms, and creating models. The MATLAB language includes mathematical functions that support common engineering andscience operations. Core math functions use processor-optimized libraries to provide fast execution of vector and matrix calculations.

Available methods include:

- Interpolation and regression

- Differentiation and integration
- Linear systems of equations
- Fourier analysis
- Eigenvalues and singular values
- Ordinary differential equations (ODEs)
- Sparse matrices

MATLAB add-on products provide functions in specialized areas such as statistics, optimization, signal analysis, and machine learning.

CHAPTER-3

INTRODUCTION TO IMAGE PROCESSING

3.1 What is an image?

An image is an array, or a matrix, of square pixels (picture elements) arranged in columns and rows.

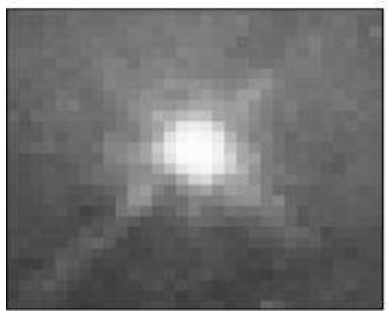


Figure 3.1: An image — an array or a matrix of pixels arranged in columns and rows.

In a (8-bit) greyscale image each picture element has an assigned intensity that ranges from 0 to 255. A grey scale image is what people normally call a black and white image, but the name emphasizes that such an image will also include many shades of grey.

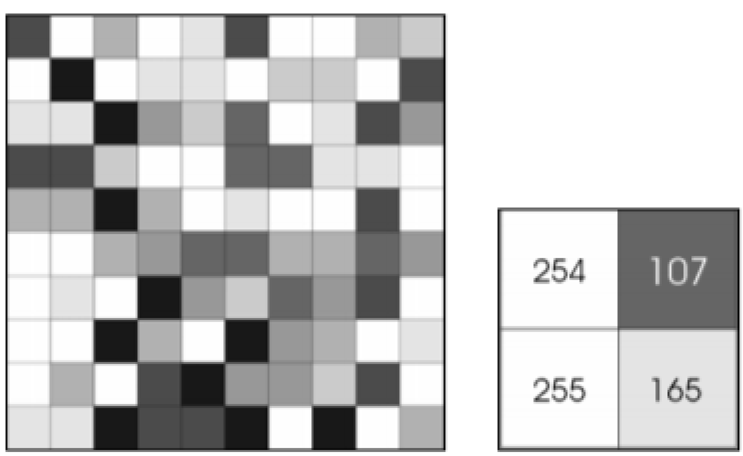


Figure 3.2: Each pixel has a value from 0 (black) to 255 (white). The possible range of the pixel values depend on the colour depth of the image, here 8 bit = 256 tones or greyscales.

A normal greyscale image has 8 bitcolour depth = 256 greyscales. A “true colour” image has 24 bitcolour depth = 8 x 8 x 8 bits = 256 x 256 x 256 colours = ~16 million colours.

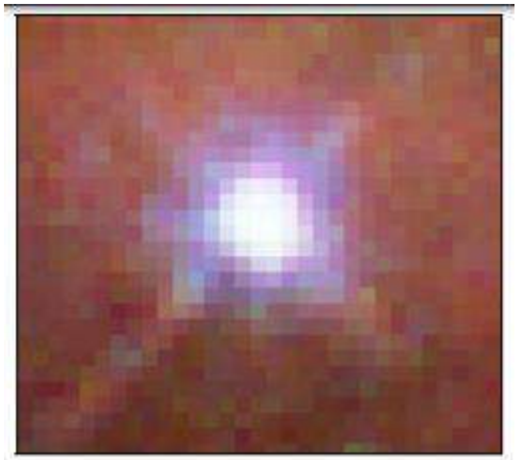


Figure 3.3: A true-colour image assembled from three greyscale images coloured red, green and blue. Such an image may contain up to 16 million different colours.

Some greyscale images have more greyscales, for instance 16 bit = 65536 greyscales. In principle three greyscale images can be combined to form an image with 281,474,976,710,656 greyscales.

There are two general groups of ‘images’: vector graphics (or line art) and bitmaps (pixel-based or ‘images’). Some of the most common file formats are:

GIF — an 8-bit (256 colour), non-destructively compressed bitmap format. Mostly used for web. Has several sub-standards one of which is the animated GIF.

JPEG — a very efficient (i.e. much information per byte) destructively compressed 24 bit (16 million colours) bitmap format. Widely used, especially for web and Internet (bandwidth-limited).

TIFF — the standard 24 bit publication bitmap format. Compresses nondestructively with, for instance, Lempel-Ziv-Welch (LZW) compression.

PS — Postscript, a standard vector format. Has numerous sub-standards and can be difficult to transport across platforms and operating systems.

PSD – a dedicated Photoshop format that keeps all the information in an image including all the layers.

Colours

For science communication, the two main colour spaces are RGB and CMYK.

RGB

The RGB colour model relates very closely to the way we perceive colour with the r, g and b receptors in our retinas. RGB uses additive colour mixing and is the basic colour model used in television or any other medium that projects colour with light. It is the basic colour model used in computers and for web graphics, but it cannot be used for print production. The secondary colours of RGB – cyan, magenta, and yellow – are formed by mixing two of the primary colours (red, green or blue) and excluding the third colour. Red and green combine to make yellow, green and blue to make cyan, and blue and red form magenta. The combination of red, green, and blue in full intensity makes white.

In Photoshop using the "screen" mode for the different layers in an image will make the intensities mix together according to the additive colour mixing model. This is analogous to stacking slide images on top of each other and shining light through them.

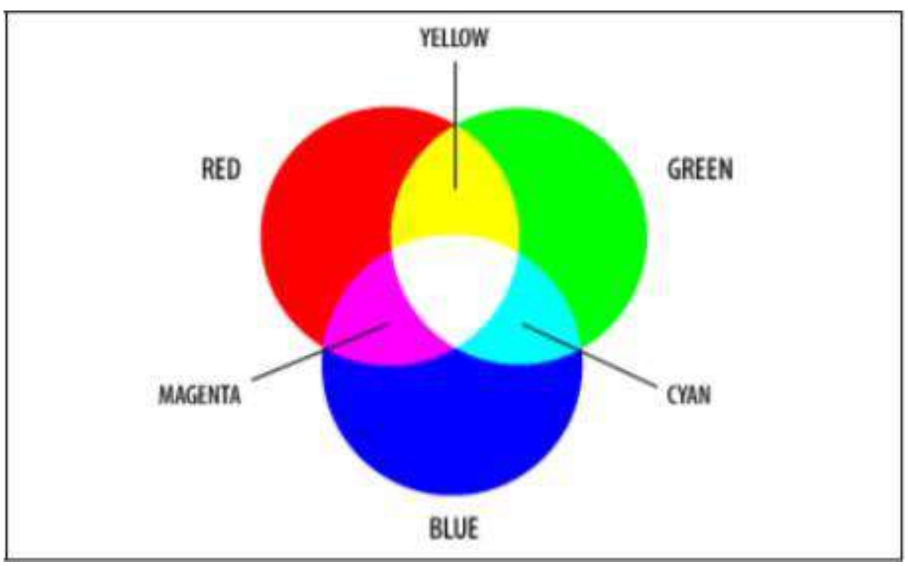


Figure 3.4: The additive model of RGB. Red, green, and blue are the primary stimuli for human colour perception and are the primary additive colours. Courtesy of adobe.com.

CMYK

The 4-colour CMYK model used in printing lays down overlapping layers of varying percentages of transparent cyan (C), magenta (M) and yellow (Y) inks. In addition a layer of black (K) ink can be added. The CMYK model uses the subtractive colour model.

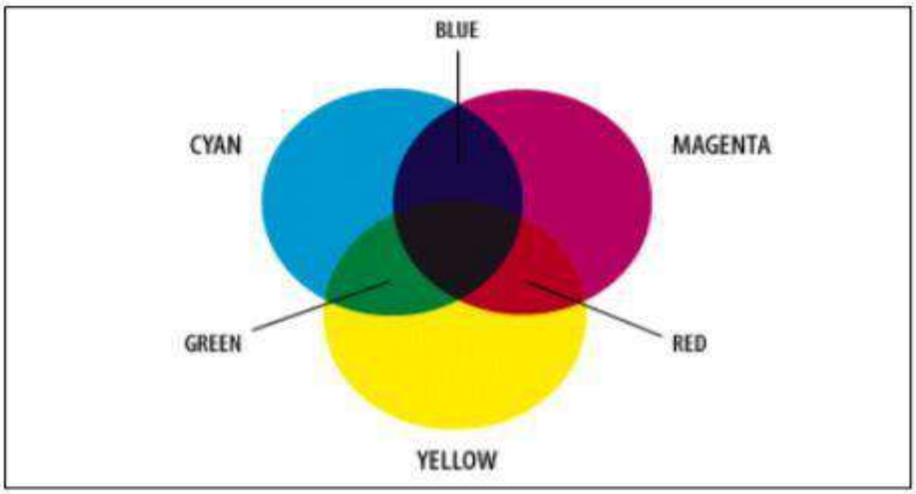


Figure 3.5: The colours created by the subtractive model of CMYK don't look exactly like the colours created in the additive model of RGB Most importantly, CMYK cannot reproduce the brightness of RGB colours. In addition, the CMYK gamut is much smaller than the RGB gamut. Courtesy of adobe.com.

3.2 Gamut

The range, or gamut, of human colour perception is quite large. The two colour spaces discussed here span only a fraction of the colours we can see. Furthermore the two spaces do not have the same gamut, meaning that converting from one colour space to the other may cause problems for colours in the outer regions of the gamuts.

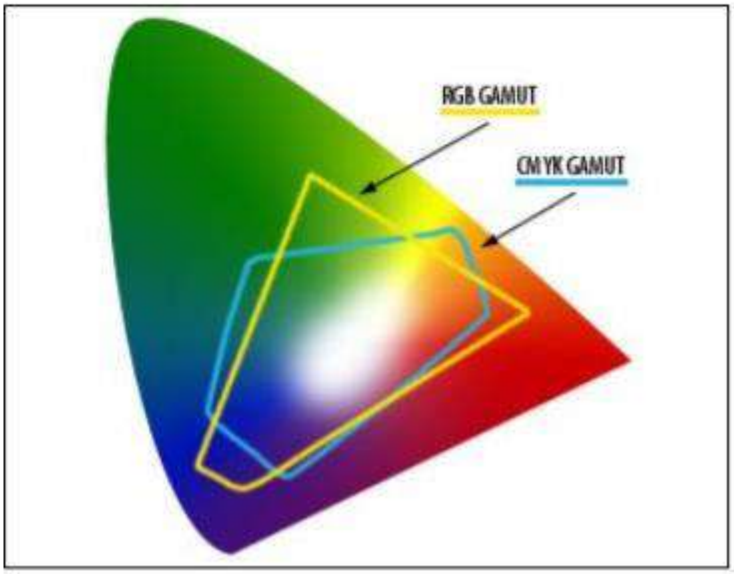


Figure 3.6: This illustration clearly shows the different gamuts of the RGB and CMYK colour spaces. The background is the CIE Chromaticity Diagram (representing the whole gamut of human colour perception). Courtesy adobe.com.

3.3 Astronomical images

Images of astronomical objects are usually taken with electronic detectors such as a CCD (Charge Coupled Device). Similar detectors are found in normal digital cameras. Telescope images are nearly always greyscale, but nevertheless contain some colour information. An astronomical image may be taken through a colour filter. Different detectors and telescopes also usually have different sensitivities to different colours (wavelengths).

3.4 Filters

A telescope such as the NASA/ESA Hubble Space Telescope typically has a fixed number of well-defined filters. A filter list for Hubble’s WFPC2 (Wide Field and Planetary Camera 2) camera is seen below.

Name	Type	Wheel	Slot	Notes	In WF/PC-1?	$\bar{\lambda}$(Å)	$\Delta\bar{\lambda}$(Å)	Peak T (%)	Peak λ (Å)
F122M	A	1	4	H Ly α - Red Leak	Y	1259	224.4	19.3	1240
F130LP	B	2	1	CaF2 Blocker (zero focus)	N	2681	5568.3	94.5	8852
F160AW	A	1	3	Woods A - redleak from pinholes	N	1471	457.2	10.1	1403
F160BW	A	1	2	Woods B	N	1446	457.1	12.1	1400
F165LP	B	2	2	Suprasil Blocker (zero focus)	N	3301	5533.2	95.4	5796
F170W	A	8	1	-	N	1666	434.6	30.7	1655
F185W	A	8	2	-	N	1899	297.4	25.9	1849
F218W	A	8	3	Interstellar feature	N	2117	367.9	21.1	2092
F255W	A	8	4	-	N	2545	408.2	14.8	2489
F300W	A	9	4	Wide U	N	2892	727.6	50.8	2760
F336W	A	3	1	WFPC2 U, Strömgren *u*	Y	3317	370.5	82.6	3447
F343N	A	5	1	Ne V	N	3427	23.5	9.3	3432
F375N	A	5	2	[OII] 3727 RS	Y	3732	24.4	19.5	3736
F380W	A	9	1	-	N	3912	694.8	65.0	3980
F390N	A	5	3	CN	N	3888	45.0	36.5	3886
F410M	A	3	2	Strömgren *v*	N	4086	147.0	70.4	4097
F437N	A	5	4	[OIII]	Y	4369	25.2	52.0	4368
F439W	A	4	4	WFPC2 B	Y	4283	464.4	68.2	4176
F450W	A	10	4	Wide B	N	4410	925.1	91.4	5060
F467M	A	3	3	Strömgren *b*	N	4663	166.4	75.3	4728
F469N	A	6	1	He II	Y	4694	25.0	52.4	4697
F487N	A	6	2	H β	Y	4865	25.9	58.6	4862
F502N	A	6	3	[OIII]	Y	5012	26.9	63.7	5008
F547M	A	3	4	Strömgren y (but wider)	Y	5446	486.6	91.3	5360
F555W	A	9	2	WFPC2 V	Y	5202	1222.6	94.6	5148
F569W	A	4	2	F555W generally preferred[a]	Y	5524	965.7	94.2	5310
F588N	A	6	4	He I & Na I (NaD)	Y	5893	49.0	91.4	5894
F606W	A	10	2	Wide V	Y	5767	1579.0	96.7	6186
F622W	A	9	3	-	Y	6131	935.4	95.6	6034
F631N	A	7	1	[OI]	Y	6306	30.9	85.7	6301
F656N	A	7	2	Hα	Y	6564	21.5	77.8	6562
F658N	A	7	3	[NII]	Y	6591	28.5	79.7	6591
F673N	A	7	4	[SII]	Y	6732	47.2	87.0	6732
F675W	A	4	3	WFPC2 R	Y	6714	889.5	97.3	6780
F702W	A	10	3	Wide R	Y	6940	1480.6	97.1	6538
F785LP	A	2	3	F814W generally preferred[a]	Y	9283	2096.1	91.7	9959
F791W	A	4	1	F814W generally preferred[a]	Y	7969	1304.6	95.9	8082
F814W	A	10	1	WFPC2 I	Y	8203	1758.0	94.8	8387
F850LP	A	2	4	-	Y	9650	1672.4	89.2	10028
F953N	A	1	1	[SIII]	N	9546	52.5	95.6	9528
F1042M	A	11	2	-	Y	10437	611.0	81.6	10139

Filter list for Hubble's WFPC2 camera (Wide Field and Planetary Camera 2). Filter names are to the left (names include approximate wavelength in nm) in column 1. Column 5 contains the physical property of the radiation the filter lets through. Column 7 is the central wavelength. The N's and W's are short for Narrow and Wide.

Filters can either be broad-band (Wide) or narrow-band (Narrow). A broad-band filter lets a wide range of colours through, for instance the entire green or red area of the spectrum. A narrow-band filter typically only lets a small wavelength span through, thus effectively restricting the transmitted radiation to that coming from a given atomic transition, allowing astronomers to investigate individual atomic processes in the object.

A filename such as 502nmos.fits indicates that the filter used has a peak at 502 nm. In the table below, you can see that this filter is a narrow bandwidth filter, i.e. it only lets radiation with wavelengths within a few nm of 502 nm through. Below is an example of an image composed from narrow-band exposures. This results in very sharply defined wisps of nebulosity since each exposure separates light from only some very specific physical processes and locations in the nebula.

Figure 3.7: Example of an image constructed from narrow-band exposures. Since the narrowband exposures probe individual atomic transitions the result is an image that has very ‘sharp’ features.

Galaxies are often studied through broad-band filters as they allow more light to get through. Also the processes in a galaxy are more ‘mixed’ or complicated, result from the outputs of billions of stars and so narrow-band filters give less ‘specific’ information about the processes there.

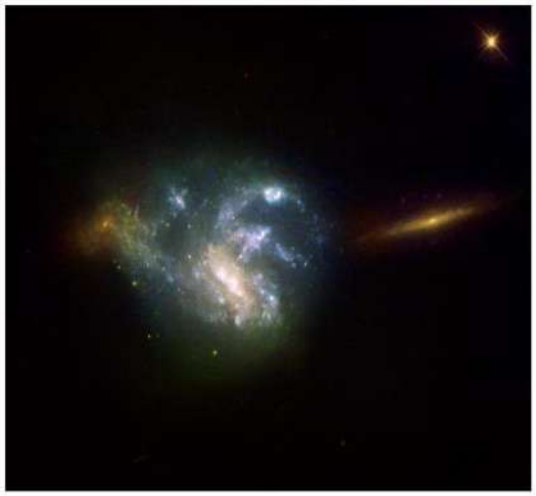


Figure 3.8: A broad-band image of the “Hyperactive galaxy NGC 7673”.

A visual example of the different filters available onboard Hubble is seen in the following figure.

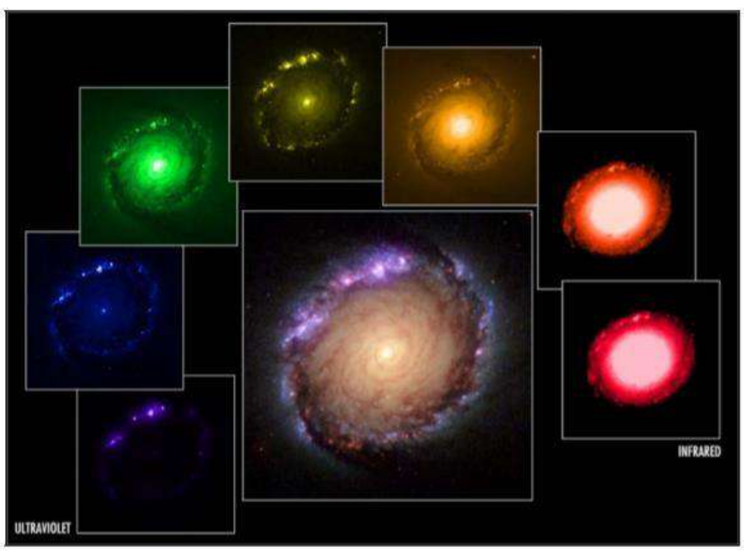


Figure 3.9: An example of an image constructed from 7 broad-band filters all the way from ultraviolet (left) to infrared (right).

A figure illustrating the process of stacking together different colour exposures is seen below.

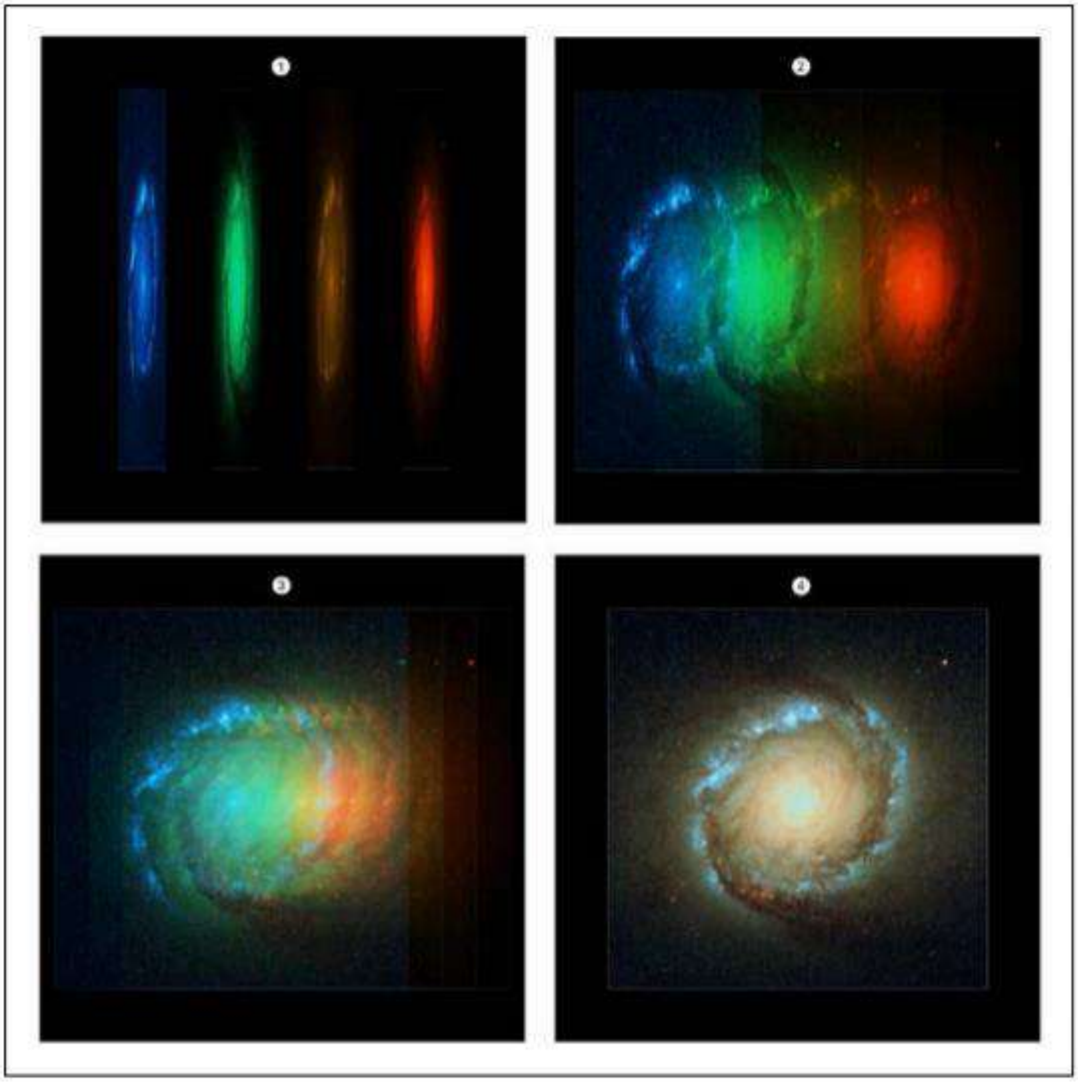


Figure 3.10: An example of how a colour image is constructed from four broad-band filters (seen from the side in 1.): blue, green yellow and red. When the images are overlaid (2. and 3.) the resulting image (4.) is a colour composite.

3.5 Assigning colours to different filter exposures

The astronomical images we see on the web and in the media are usually ‘refined’ or ‘processed’ as compared to the raw data that the astronomers work on with their computers. In ‘pretty pictures’ all artefacts coming from the telescope or the detectors are for instance removed as they do not say anything about the objects themselves. It is very rare that images are taken with the sole intention of producing a ‘pretty’ colour picture. Most ‘pretty pictures’ are constructed from data that was acquired to study some physical process, and the astronomer herself probably never bothered to assemble the greyscale images to a colour image.

3.5 Natural colour images

It is possible to create colour images that are close to “true-colour” if three wide band exposures exist, and if the filters are close to the r, g and b receptors in our eyes. Images that approximate what a fictitious space traveller would see if he or she actually travelled to the object are called “natural colour” images. To make a natural colour image the order of the colours assigned to the different exposures should be in “chromatic order”, i.e. the lowest wavelength should be given a blue hue, the middle wavelength a green hue and the highest wavelength should be red.

3.6 Representative colour images

If one or more of the images in a data set is taken through a filter that allows radiation that lies outside the human vision span to pass – i.e. it records radiation invisible to us - it is of course not possible to make a natural colour image. But it is still possible to make a colour image that shows important information about the object. This type of image is called a representative colour image. Normally one would assign colours to these exposures in chromatic order with blue assigned to the shortest wavelength, and red to the longest. In this way it is possible to make colour images from electromagnetic radiation far from the human vision area, for example x-rays. Most often it is either infrared or ultraviolet radiation that is used.

3.7 Enhanced colour images

Sometimes there are reasons to not use a chromatic order for an image. Often these reasons are purely aesthetic, as is seen in the example below. This type of colour image is called an enhanced colour image.

Figure 3.11: An example of an enhanced colour image (not in chromatic order): Sometimes it is necessary to break the 'rules' for image processing. Here the Hydrogen-alpha filter is coloured blue instead of the red colour it is in nature. This is an example of a so-called false-colour image, where the blue was chosen for aesthetic reasons.

3.8 You are the judge

When processing raw science images one of the biggest problems is that, to a large degree, you are 'creating' the image and this means a colossal freedom within a huge parameter space. There are literally thousands of sliders, numbers, dials, curves etc. to twist and turn. Speaking of right and wrong, there really are no wrong or right images. There are some fundamental scientific principles that should normally be observed, but the rest is a matter of aesthetics — taste. Chromatic ordering of the exposures is one of the important scientific principles.

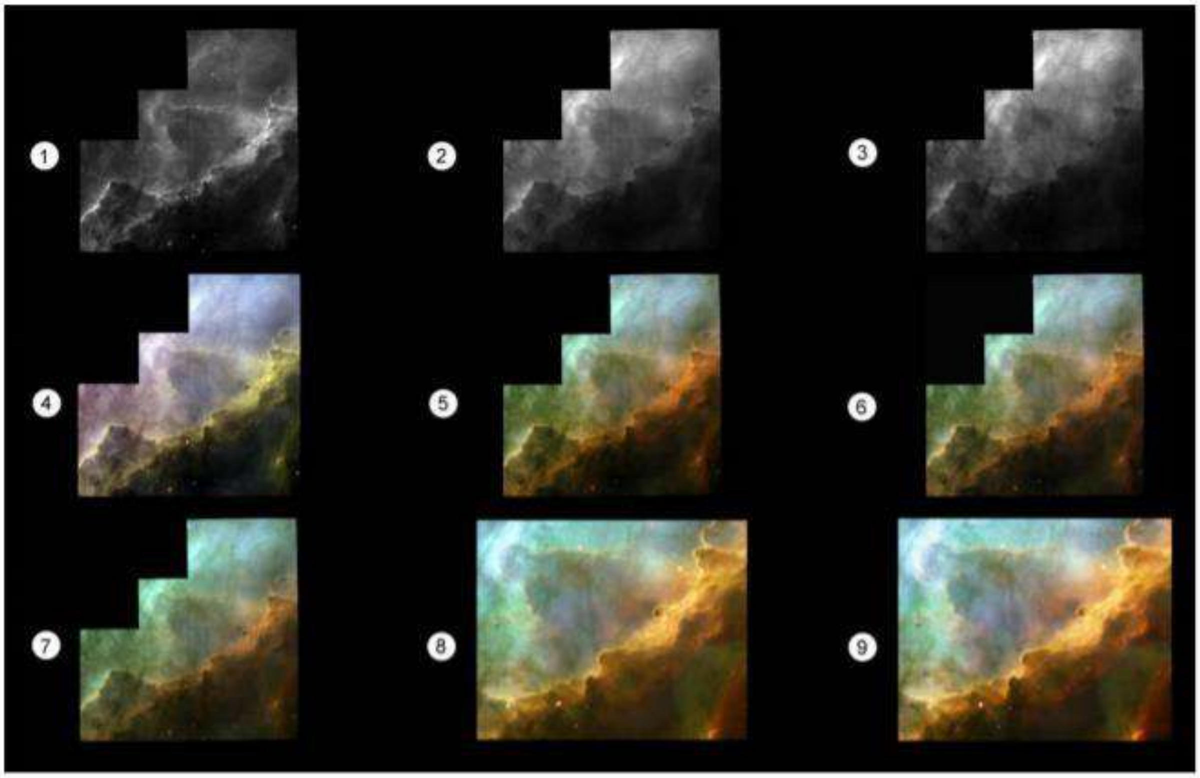


Figure 3.12: Sequences in the production of a Hubble Space Telescope image of Messier 17. First the individual exposure (taken through three different filters): 1. 673n (Sulphur) shown in red in the final image), 2. 656n (hydrogen, green), 3. 502n (oxygen, blue), 4. First colour composite attempt, 5. Improving, 6. Improving, 7. Improving, 8. Adjusting the composition and then 9. Final colour and contrast adjustments for the final image.

3.9 Stretch function

One particularly important aspect of image processing is the choice of the best stretch function. You choose which "stretch function" or representation to use in the Fits Liberator window. A logarithmic representation of the pixel values tends to suppress the bright parts of the image, i.e. the stars, and to enhance the fainter part, e.g. nebulosity. This can be desirable if the 'faint stuff' needs 'a boost', but a logarithmic stretch function can also reduce the contrast in an image, producing a lower dynamic range as is seen in the example below.

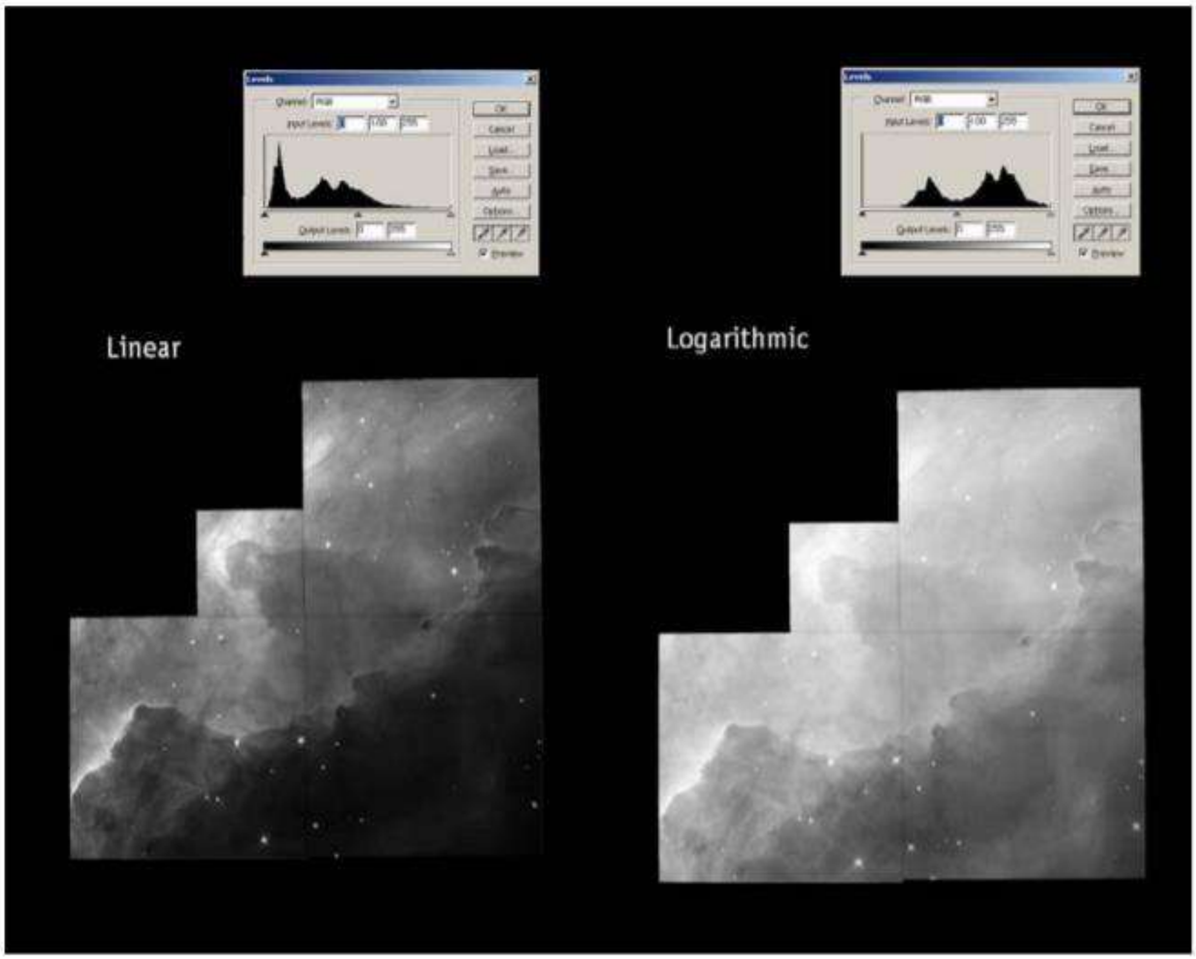


Figure 3.13: The difference between two stretch functions. To the left is a linear representation of the pixels and to the right a logarithmic. It is seen that the log lowers the contrast too much and therefore is not the aesthetically desirable function to choose here.

Various techniques have been developed in Image Processing during the last four to five decades. Most of the techniques are developed for enhancing images obtained from unmanned spacecrafts, space probes and military reconnaissance flights. Image Processing systems are becoming popular due to easy availability of powerful personnel computers, large size memory devices, graphics softwares etc.

Image Processing is used in various applications such as:

- Remote Sensing
- Medical Imaging

- Non-destructive Evaluation
- Forensic Studies
- Textiles
- Material Science.
- Military
- Film industry
- Document processing
- Graphic arts
- Printing Industry

The common steps in image processing are image scanning, storing, enhancing and interpretation. The schematic diagram of image scanner-digitizer diagram is shown in figure.

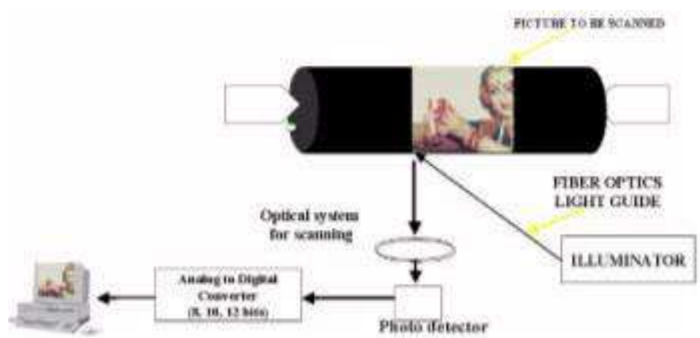


Figure 3.14

Methods of Image Processing

There are two methods available in Image Processing.

3.10 Analog Image Processing

Analog Image Processing refers to the alteration of image through electrical means. The most common example is the television image. The television signal is a voltage level which varies in amplitude to represent brightness through the image. By electrically varying the signal, the displayed image appearance is altered. The brightness and contrast controls on a TV set serve to adjust the amplitude and reference of the video signal, resulting in the brightening, darkening and alteration of the brightness range of the displayed image.

3.11 Digital Image Processing

In this case, digital computers are used to process the image. The image will be converted to digital form using a scanner – digitizer (as shown in Figure 1) and then process it. It is defined as the subjecting numerical representations of objects to a series of operations in order to obtain a desired result. It starts with one image and produces a modified version of the same. It is therefore a process that takes an image into another. The term digital image processing generally refers to processing of a two-dimensional picture by a digital computer. In a broader context, it implies digital processing of any two-dimensional data. A digital image is an array of real numbers represented by a finite number of bits. The principle advantage of Digital Image Processing methods is its versatility, repeatability and the preservation of original data precision.

The various Image Processing techniques are:

- Image representation
- Image preprocessing
- Image enhancement
- Image restoration
- Image analysis
- Image reconstruction
- Image data compression

3.12 Image Representation

An image defined in the "real world" is considered to be a function of two real variables, for example, f(x,y) with f as the amplitude (e.g. brightness) of the image at the real coordinate position (x,y). The effect of digitization is shown in Figure.

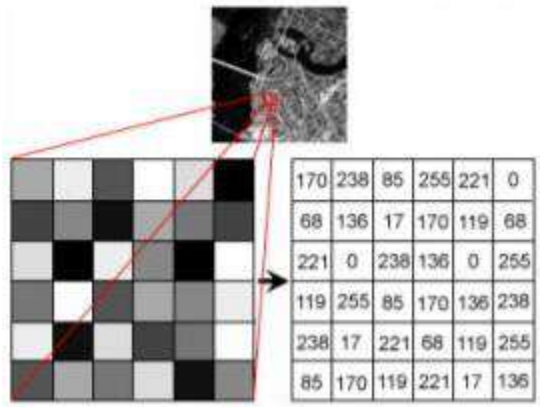


Figure: 3.15

The 2D continuous image f(x,y) is divided into N rows and M columns. The intersection of a row and a column is called as pixel. The value assigned to the integer coordinates [m,n] with {m=0,1, 2,...,M-1} and {n=0,1,2,...,N-1} is f[m,n]. In fact, in most cases f(x,y)--which we might consider to be the physical signal that impinges on the face of a sensor. Typically an image file such as BMP, JPEG, TIFF etc., has some header and picture information. A header usually includes details like format identifier (typically first information), resolution, number of bits/pixel, compression type, etc.

3.13 Image Preprocessing

3.13.1 Scaling

The theme of the technique of magnification is to have a closer view by magnifying or zooming the interested part in the imagery. By reduction, we can bring the unmanageable size of data to a manageable limit. For resampling an image Nearest Neighborhood, Linear, or cubic convolution techniques are used.

3.13.2 Magnification

This is usually done to improve the scale of display for visual interpretation or sometimes to match the scale of one image to another. To magnify an image by a factor of 2, each pixel of the original image is replaced by a block of 2x2 pixels, all with the same brightness value as the original pixel.

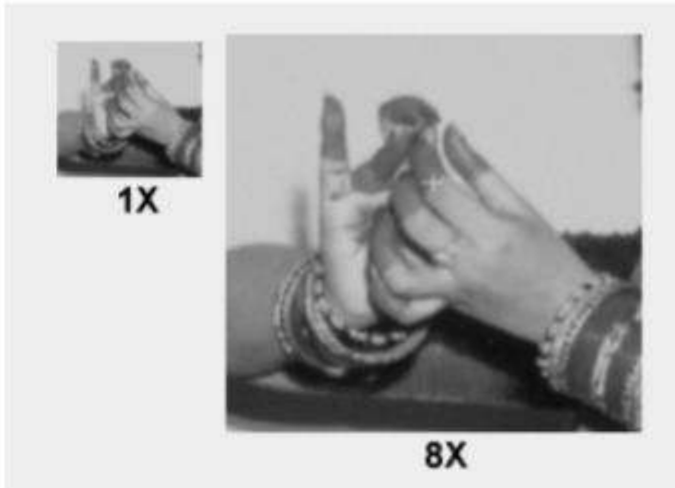


Figure 3.16 Figure Image Magnification

3.13.3 Reduction

To reduce a digital image to the original data, every mth row and mth column of the original imagery is selected and displayed. Another way of accomplishing the same is by taking the average in 'm x m' block and displaying this average after proper rounding of the resultant value.

Figure 3.17 Image Reduction Rotation

Rotation is used in image mosaic, image registration etc. One of the techniques of rotation is 3-pass shear rotation, where rotation matrix can be decomposed into three separable matrices.

3-pass shear rotation

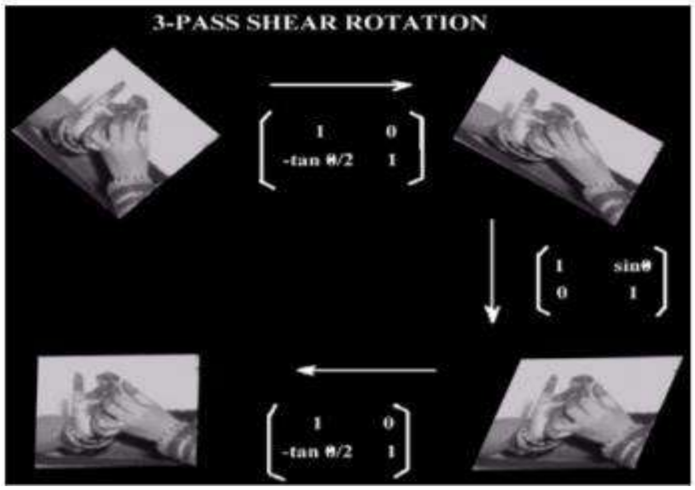


Figure 3.18 Pass Shear Rotation

3.14 Advantages

1. No scaling – no associated resampling degradations.

2. Shear can be implemented very efficiently.

3.15 Mosaic

Mosaic is a process of combining two or more images to form a single large image without radiometric imbalance. Mosaic is required to get the synoptic view of the entire area, otherwise capture as small images.

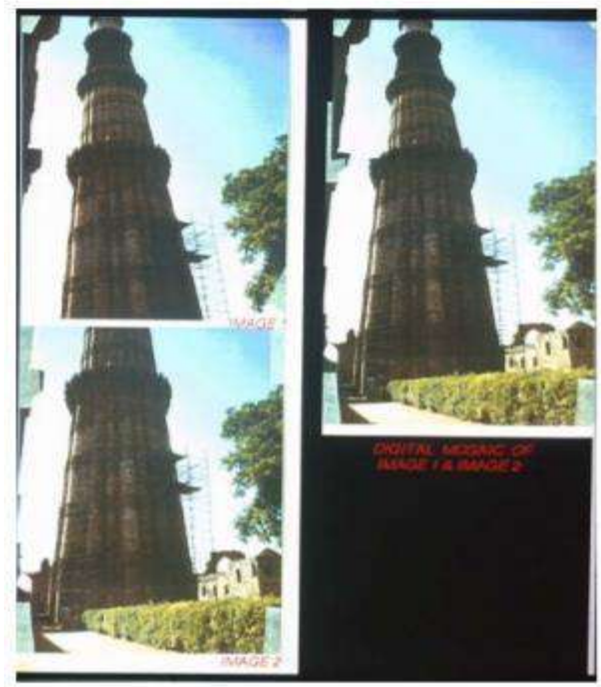


Figure 3.19 Image Mosaicking

3.16 Image Enhancement Techniques

Some times images obtained from satellites and conventional and digital cameras lack in contrast and brightness because of the limitations of imaging sub systems and illumination conditions while capturing image. Images may have different types of noise. In image enhancement, the goal is to accentuate certain image features for subsequent analysis or for image display. Examples include contrast and edge enhancement, pseudo-coloring, noise filtering, sharpening, and magnifying. Image enhancement is useful in feature extraction, image analysis and an image display. The enhancement process itself does not increase the inherent information content in the data. It simply emphasizes certain specified image characteristics. Enhancement algorithms are generally interactive and application dependent.

Some of the enhancement techniques are:

- Contrast Stretching
- Noise Filtering
- Histogram modification

3.17 Contrast Stretching:

Some images (eg. over water bodies, deserts, dense forests, snow, clouds and under hazy conditions over heterogeneous regions) are homogeneous i.e., they do not have much change in their levels. In terms of histogram representation, they are characterized as the occurrence of very narrow peaks. The homogeneity can also be due to the incorrect illumination of the scene.

Ultimately the images hence obtained are not easily interpretable due to poor human perceptibility. This is because there exists only a narrow range of gray-levels in the image having provision for wider range of gray-levels. The contrast stretching methods are designed exclusively for frequently encountered situations. Different stretching techniques have been developed to stretch the narrow range to the whole of the available dynamic range.

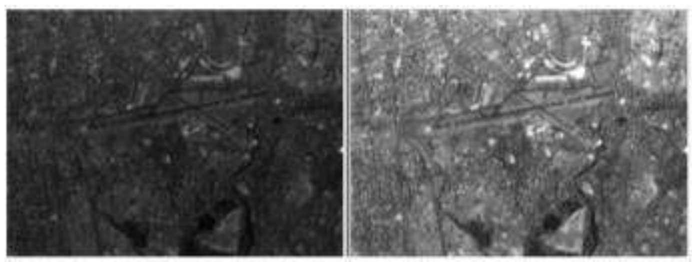


Figure contrast stretching

Noise filtering is used to filter the unnecessary Noise Filtering N information from an image. It is also used to remove various types of noises from the images. Mostly this feature is interactive. Various filters like low pass, high pass, mean, median etc., are available.

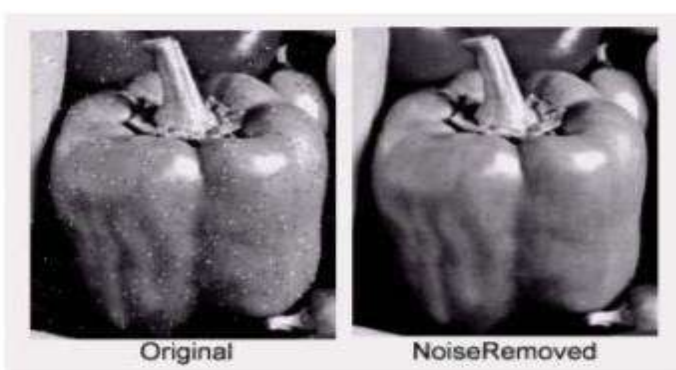


Figure 3.20 Noise Removal

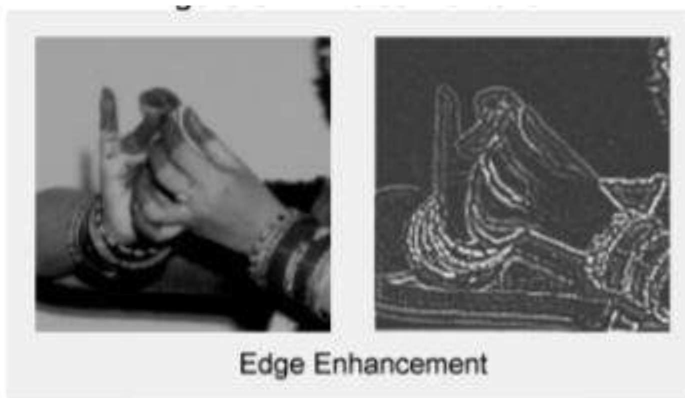


Figure 3.21: Edge Enhancemen

3.18 Histogram Modification

Histogram has a lot of importance in image enhancement. It reflects the characteristics of image. By modifying the histogram, image characteristics can be modified. One such example is Histogram Equalization. Histogram equalization is a nonlinear stretch that redistributes pixel values so that there is approximately the same number of pixels with each value within a range. The result approximates a flat histogram. Therefore, contrast is increased at the peaks and lessened at the tails.

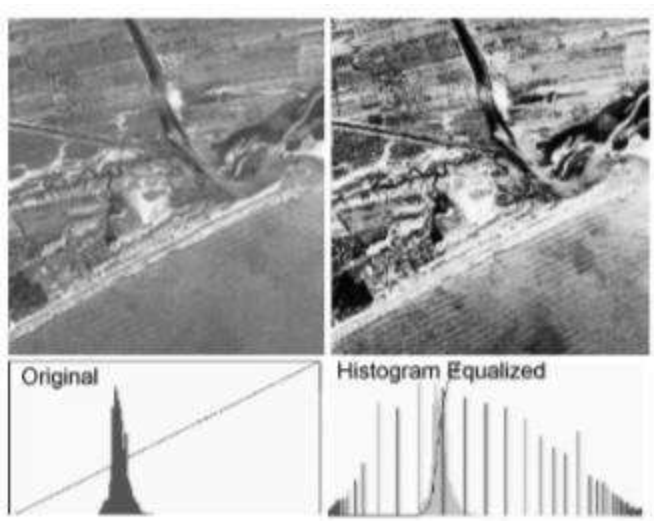


Figure 3.22 : Histogram equalized output

3.19 Image Analysis

Image analysis is concerned with making quantitative measurements from an image to produce a description. In the simplest form, this task could be reading a label on a grocery item, sorting different parts on an assembly line, or measuring the size and orientation of blood cells in a medical image. More advanced image analysis systems measure quantitative information and use it to make a sophisticated decision, such as controlling the arm of a robot to move an object after identifying it or navigating an aircraft with the aid of images acquired along its trajectory. Image analysis techniques require extraction of certain features that aid in the identification of the object. Segmentation techniques are used to isolate the desired object from the scene so that measurements can be made on it subsequently. Quantitative measurements of object features allow classification and description of the image.

3.20 Image Segmentation

Image segmentation is the process that subdivides an image into its constituent parts or objects. The level to which this subdivision is carried out depends on the problem being solved, i.e., the segmentation should stop when the objects of interest in an application have been isolated e.g., in autonomous air-toground target acquisition, suppose our interest lies in identifying vehicles on a road, the first step is to segment the road from the image and then to segment the contents of the road down to potential vehicles. Image thresholding techniques are used for image segmentation.

3.20.1 Classification

Classification is the labeling of a pixel or a group of pixels based on its grey value. Classification is one of the most often used methods of information extraction. In Classification, usually multiple features are used for a set of pixels i.e., many images of a particular object are needed. In Remote Sensing area, this procedure assumes that the imagery of a specific geographic area is collected in multiple regions of the electromagnetic spectrum and that the images are in good registration. Most of the information extraction techniques rely on analysis of the spectral reflectance properties of such imagery and employ special algorithms designed to perform various types of 'spectral analysis'. The process of multispectral classification can be performed using either of the two methods: Supervised or Unsupervised. In Supervised classification, the identity and location of some of the land cover types such as urban, wetland, forest etc., are known as priori through a combination of field works and toposheets. The analyst attempts to locate specific sites in the remotely sensed data that represents homogeneous examples of these land cover types. These areas are commonly referred as TRAINING SITES because the spectral characteristics of these known areas are used to 'train' the classification algorithm for eventual land cover mapping of reminder of the image. Multivariate statistical parameters are calculated for each training site. Every pixel both within and outside these training sites is then evaluated and assigned to a class of which it has the highest likelihood of being a member.

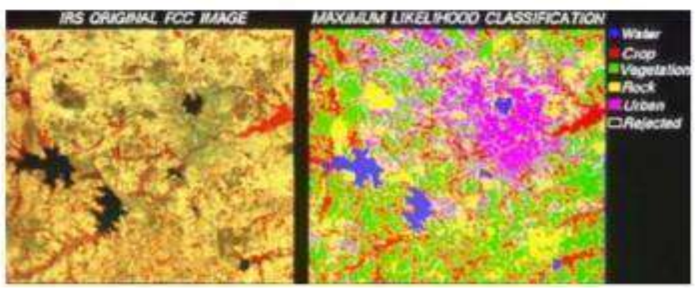


Figure 3.23: Image Classification

In an Unsupervised classification, the identities of land cover types has to be specified as classes within a scene are not generally known as priori because ground truth is lacking or surface features within the scene are not well defined. The computer is required to group pixel data into different spectral classes according to some statistically determined criteria. The comparison in medical area is the labeling of cells based on their shape, size, color and texture, which act as features. This method is also useful for MRI images.

3.21 Image Restoration

Image restoration refers to removal or minimization of degradations in an image. This includes de-blurring of images degraded by the limitations of a sensor or its environment, noise filtering, and correction of geometric distortion or non-linearity due to sensors. Image is restored to its original quality by inverting the physical degradation phenomenon such as defocus, linear motion, atmospheric degradation and additive noise.

Figure 3.24 Weiner – Image Restoration

3.22 Image Reconstruction from Projections

Image reconstruction from projections is a special class of image restoration problems where a two- (or higher) dimensional object is reconstructed from several one-dimensional projections. Each projection is obtained by projecting a parallel X-ray (or other penetrating radiation) beam through the object. Planar projections are thus obtained by viewing the object from many different angles. Reconstruction algorithms derive an image of a thin axial slice of the object, giving an inside view otherwise unobtainable without performing extensive surgery. Such techniques are important in medical imaging (CT scanners), astronomy, radar imaging, geological exploration, and nondestructive testing of assemblies.

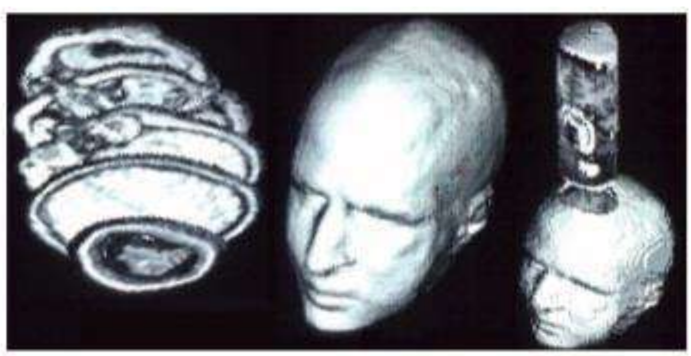


Figure 3.26 MRI Slices

3.23 Image Compression

Compression is a very essential tool for archiving image data, image data transfer on the network etc. They are various techniques available for lossy and lossless compressions. One of most popular compression techniques, JPEG (Joint Photographic Experts Group) uses Discrete Cosine Transformation (DCT) based compression technique. Currently wavelet based compression techniques are used for higher compression ratios with minimal loss of data.

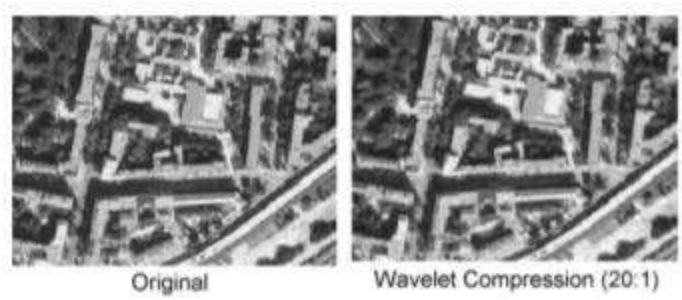


Figure 3.27. Wavelet Image Compression

CHAPTER-4

DIGITAL IMAGE PROCESSING

4.1 Digital image processing

Background

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An important characteristic underlying the design of image processing systems is the significant level of testing & experimentation that normally is required before arriving at an acceptable solution. This characteristic implies that the ability to formulate approaches &quickly prototype candidate solutions generally plays a major role in reducing the cost & time required to arrive at a viable system implementation.

4.2 What is DIP

An image may be defined as a two-dimensional function f(x, y), where x & y are spatial coordinates, & the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital image by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensor, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis& computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting & somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low-level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low- level process is characterized by the fact that both its inputs & outputs are images.

Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images. Finally higher- level processing involves "Making sense" of an ensemble of recognized objects, as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

4.3 What is an image?

An image is represented as a two dimensional function f(x, y) where x and y are spatial co-ordinates and the amplitude of 'f' at any pair of coordinates (x, y) is called the intensity of the image at that point.

4.3.1 Gray scale image:

A grayscale image is a function I (xylem) of the two spatial coordinates of the image plane.

I(x, y) is the intensity of the image at the point (x, y) on the image plane.

I (xylem) takes non-negative values assume the image is bounded by a rectangle [0, a] ×[0, b]I: [0, a] × [0, b] → [0, info)

4.3.2 Color image:

It can be represented by three functions, R (xylem) for red, G (xylem) for green andB (xylem) for blue. An image may be continuous with respect to the x and y coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate's values is calledsampling. Digitizing the amplitude values is called quantization.

4.4 Coordinate convention:

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image f(x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size M X N. The values of the coordinates (xylem) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates.

In many image processing books, the image origin is defined to be at (xylem)= (0,0).The next coordinate values along the first row of the image are (xylem)=(0,1).It is important to keep in mind that the notation (0,1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Following figure shows the coordinate convention. Note that x ranges from 0 to M-1 and y from 0 to N-1 in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (xylem) the toolbox uses the notation (race) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate topples, (alb), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentation refers to the coordinates. Less frequently the toolbox also employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refers to rows. This is the opposite of our use of variables x and y.

4.5 Image as Matrices:

The preceding discussion leads to the following representation for a digitized image function:

f (0,0) f(0,1) ……….. f(0,N-1)

f (1,0) f(1,1) ………… f(1,N-1)

f (xylem) = f (M-1,0) f(M-1,1) .... f(M-1,N-1)

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements.

A digital image can be represented naturally as a MATLAB matrix:

f (1,1) f(1,2) .... f(1,N)

f (2,1) f(2,2) …….. f (2,N) . . .

f = . . .

f (M,1) f(M,2) .... f(M,N)

Where f (1,1) = f(0,0) (note the use of a monoscope font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation f(p ,q) denotes the element located in row p and the column q. For example f(6,2) is the element in the sixth row and second column of the matrix f. Typically we use the letters M and N respectively to denote the number of rows and columns in a matrix. A 1xN matrix is called a row vector whereas an Mx1 matrix is called a column vector. A 1x1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using mono-scope characters. We use conventional Roman, italic notation such as f(x ,y), for mathematical expressions

Reading Images:

Images are read into the MATLAB environment using function imread whose syntax is

Imread (‘filename’)

Format name	Description	recognized extension
TIFF	Tagged Image File Format	.tif, .tiff
JPEG	Joint Photograph Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	.xwd

Here filename is a spring containing the complete of the image file(including any applicable extension).For example the command line

>> f = imread (‘8. jpg’);

Reads the JPEG (above table) image chestxray into image array f. Note the use of single quotes (‘) to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output If a semicolon is not included. MATLAB displays the results of the operation(s) specified in that line. The prompt symbol (>>) designates the beginning of a command line, as it appears in the MATLAB command window.

4.6 Data Classes:

Although we work with integers coordinates the values of pixels themselves are not restricted to be integers in MATLAB. Table above list various data classes supported by MATLAB and IPT are representing pixels values. The first eight entries in the table are refers to as numeric data classes. The ninth entry is the char class and, as shown, the last entry is referred to as logical data class.

All numeric computations in MATLAB are done in double quantities, so this is also a frequent data class encounter in image processing applications. Class unit 8 also is encountered frequently, especially when reading data from storages devices, as 8 bit images are most common representations found in practice. These two data classes, classes logical, and, to a lesser degree, class unit 16 constitute the primary data classes on which we focus. Many ipt functions however support all the data classes listed in table. Data class double requires 8 bytes to represent a number uint8 and int 8 require one byte each, uint16 and int16 requires 2bytes and unit 32.

Name	Description
Double	Double _ precision, floating_ point numbers the Approximate.
Uint8	unsigned 8_bit integers in the range [0,255] (1byte per Element).
Uint16	unsigned 16_bit integers in the range [0, 65535] (2byte per element).
Uint 32	unsigned 32_bit integers in the range [0, 4294967295](4 bytes per element).
Int8	signed 8_bit integers in the range [-128,127] 1 byte per element)
Int 16	signed 16_byte integers in the range [32768, 32767] (2 bytes per element).
Int 32	Signed 32_byte integers in the range [-2147483648, 21474833647] (4 byte per element).

Single single _precision floating _point numbers with values

In the approximate range (4 bytes per elements)

Char characters (2 bytes per elements).

Logical values are 0 to 1 (1byte per element).

Int 32 and single required 4 bytes each. The char data class holds characters in Unicode representation. A character string is merely a 1*n array of characters logical array contains only the values 0 to 1,with each element being stored in memory using function logical or by using relational operators.

4.7 Image Types:

The toolbox supports four types of images:

1 .Intensity images;

2. Binary images;

3. Indexed images;

4. R G B images.

Most monochrome image processing operations are carried out using binary or intensity images, so our initial focus is on these two image types. Indexed and RGB colour images.

4.8 Intensity Images:

An intensity image is a data matrix whose values have been scaled to represent intentions. When the elements of an intensity image are of class unit8, or class unit 16, they have integer values in the range [0,255] and [0, 65535], respectively. If the image is of class double, the values are floating point numbers. Values of scaled, double intensity images are in the range [0, 1] by convention.

4.9 Binary Images:

Binary images have a very specific meaning in MATLAB.A binary image is a logical array 0s and1s.Thus, an array of 0s and 1s whose values are of data class, say unit8, is not considered as a binary image in MATLAB .A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, we create an array B using the statement.

B=logical (A)

If A contains elements other than 0s and 1s.Use of the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.

Using relational and logical operators also creates logical arrays.

To test if an array is logical we use the I logical function: islogical(c).

If c is a logical array, this function returns a 1.Otherwise returns a 0. Logical array can be converted to numeric arrays using the data class conversion functions.

4.10 Indexed Images

An indexed image has two components:

A data matrix integer, x

A color map matrix, map

Matrix map is an m*3 arrays of class double containing floating point values in the range [0, 1].The length m of the map are equal to the number of colors it defines. Each row of map specifies the red, green and blue components of a single color. An indexed images uses “direct mapping” of pixel intensity values color map values. The color of each pixel is determined by using the corresponding value the integer matrix x as a pointer in to map. If x is of class double ,then all of its components with values less than or equal to 1 point to the first row in map, all components with value 2 point to the second row and so on. If x is of class units or unit 16, then all components value 0 point to the first row in map, all components with value 1 point to the second and so on.

4.11 RGB Image

An RGB color image is an M*N*3 array of color pixels where each color pixel is triplet corresponding to the red, green and blue components of an RGB image, at a specific spatial location. An RGB image may be viewed as “stack” of three gray scale images that when fed in to the red, green and blue inputs of a color monitor

Produce a color image on the screen. Convention the three images forming an RGB color image are referred to as the red, green and blue components images. The data class of the components images determines their range of values. If an RGB image is of class double the range of values is [0, 1].

Similarly the range of values is [0,255] or [0, 65535].For RGB images of class units or unit 16 respectively. The number of bits use to represents the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8bit image, the corresponding RGB image is said to be 24 bits deep.

Generally, the number of bits in all component images is the same. In this case the number of possible color in an RGB image is (2^b) ^3, where b is a number of bits in each component image. For the 8bit case the number is 16,777,216 colors.

CHAPTER-5

PROPOSED METHOD

Describing the different algorithms implemented in this paper in brief to highlight the working mechanism along with a few important merits and demerits and Table I described the list of problems.

5.1 Tracking-Learning-Detection (TLD)

TLD is a framework used for tracking an arbitrary object in a video for a long time . Since the three different subtasks of tracking, detection and learning are undertaken by different parts of the same process, it is described as a framework and not as a tracking method. The target object is simultaneously tracked and learned which will help make the algorithm robust against failures . The tracker estimates the target's motion and predicts its position in the next frame according to the frame of reference. The Detector scans the image to restrict all appearances that have been observed before to a location. The Learning component estimates the detector's errors and produces numerous training examples to make sure that such errors are avoided in the future. The framework aims to achieve improved tracking by preventing learning from disinformation. It achieves this by shutting down the learning task in case of occlusions or when the object is no longer in the frame. The detection component can detect the object again if it appears in the video afterward.

5.2 Boosting

Boosting is one of the older algorithms which is usually used for comparison to show an improvement in the latest ones. It's based on an online version of AdaBoost, which also powers the machine learning supporting Haarcascades.This algorithm has to be trained rigorously with numerous positive examples (initial or starting bounding box provided by the user or the previous detection algorithms) and with an equal number of negative examples (patches around the actual object in the image). It works by working across every pixel of a given new frame but only surrounds a certain area of the previous location. Then the score is recorded of this classification algorithm for each comparison. The highest scoring part is labeled as the new location. As more new frames are added to this classifier it keeps getting updated.

5.3 Multiple Instance Learning

(MIL) With large datasets being used to solve complex issues today, research and interest in multiple instance learning has grown rapidly. Another advantage is that it allows the use of data with weak labeling. As a result, it has been used for various applications such as drug activity prediction, text categorization and content based image retrieval and classification. It is a supervised learning algorithm used for cases in which the training data is not labeled completely. In supervised learning, all training instances are labeled individually. But, only a bag of instances is labeled in MIL. In the case of binary classification, if at least one training instance in a bag is positive, the bag is positively labeled. Only if every training instance in a bag is negative, the bag is negatively labeled. The objective of the algorithm is to classify instances or bags by using labeled bags as training data.

5.4 Kernelized Correlation Filter

(KCF) The KCF algorithm used the shift invariance property of Fourier transform. This use not only simplifies the calculation of the correlation parameter but also makes it exceedingly fast. KCF is also able to bypass the issue of a huge amount of training data available from each frame of input due to its lightweight implementation. The large overlapping regions between various positive samples in Multiple Instance Learning are the property exploited by this tracker. KCF algorithm adopts and implements FFT and IFT which are fast and inverse Fourier Transform. By doing so the computation efficiency is increased of the algorithm. The accuracy and speed of this algorithm can be easily be compared on accounts of performance to various different and latest algorithms in OpenCV library. Although it’s reporting of tracking failure is an improvement over the previous versions it is unable to recover from full occlusions.

5.5 Median Flow

While using the median flow algorithm, an object is represented by a bounding box and then the object’s movement is estimated between successive frames using sparse optical flow. The algorithm assumes that the object is a combination of independent points which move in a synchronous manner and thus constitute the motion of the entire object. The algorithm is initialized by setting up the bounding box around the target object. A grid is then generated within the box. The points on the grid are then tracked between two consecutive frames. Next, the quality of the point predictions is estimated and each point is assigned an error. The best half of the predictions are used to estimate the displacement of the entire bounding box

CHAPTER 6

EVALUATION METRICS

In this work, the success and precision rate of the algorithms are used for quantitative analysis and comparison. Additionally, the robustness of the algorithms has been evaluated using the TRE parameter. TRE stands for Temporal Robustness Evaluation. TRE is a parameter to measure the tracker's robustness to initialization by varying the initialization by starting at different frames. Some tracker's performance may depend on the initialization, changing if the initialization is done at another start frame. The success parameter is used to define the intersections over union of the ground truth and the predicted bounding boxes while the precision metric defines the percentage of frame locations within a certain threshold distance from those of the ground truth

6.1 Precision

This precision value is measured and tracked using a very common and well known metric called the Center Location error. Its definition is the average Euclidean distance between the manually labeled ground truths and the center locations of the tracked target [14]. Hence for precision evaluation, we utilized the scale of rectangle obtained from the tracker considering the ground truth.

$$P = \frac{|r_t|}{|r_a|} \qquad (1)$$

The overall performance for that sequence is measured using the average center location error over all the frames of one sequence. However, the output location could be arbitrary and the average center location error might not properly estimate the tracking performance if the tracker loses the target.

6.2 Success

Bounding box overlap is another evaluation metric. For evaluation of the success parameter we used the following formula for each frame is the video sequence:

$$S = \frac{r_t \cap r_a}{r_t \cup r_a} \qquad (2)$$

Here art is the bounding rectangle obtained from the tracker and ra is the bounding rectangle to represent the ground truth. The value thus obtained from the above formula is known as the overlap score which can be use to define success of the algorithm [2, 15-17]. The overlap score is essentially a ratio of the area of interception and area of union. Tracking algorithms which have a number of frames with an S value of greater than a given threshold are said to be successful. Although instead of basing it all on one threshold value we have also found out the values of Area Under Curve (AUC) for each algorithm for a better evaluation.

CHAPTER 7

RESULTS

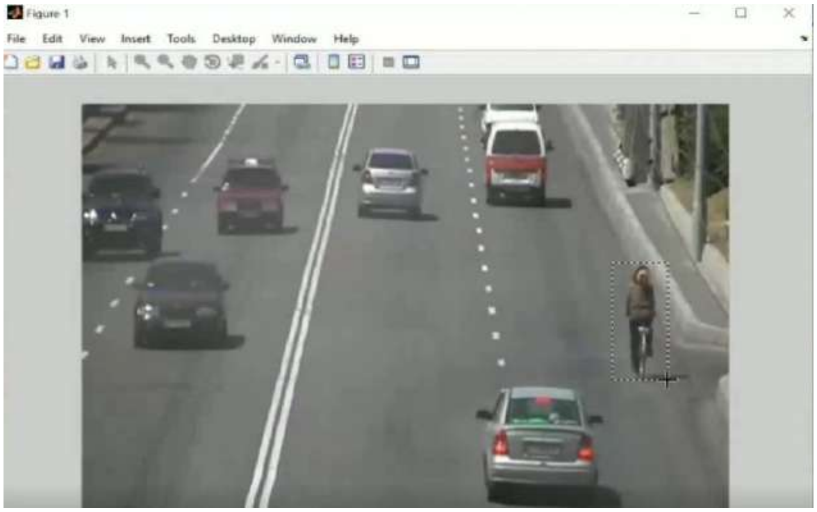


Figure: 7.1

- The Above figure shows the input in figure 7.1
- The Below given shows the output figures 7.2, figures 7.3

Figure: 7.2

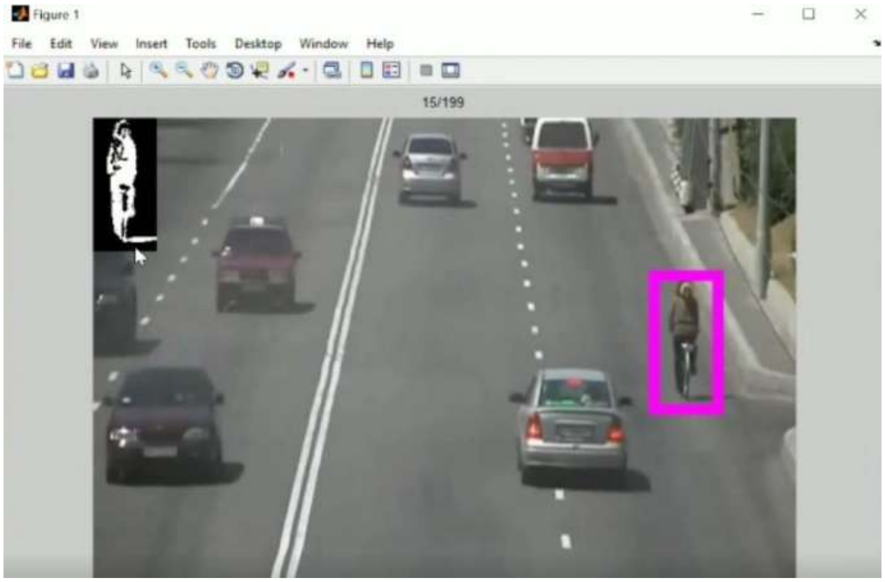


Figure 7.3 Output Image

CHAPTER-8
CONCLUSION

When all the algorithms were tested under the same environment, it can be concluded that BOOSTING Tracker is quite slow and doesn't work very well. It should be noted that it has only been included for legacy reasons. The MIL Tracker is found to be wanting against other algorithms especially in reporting failure. It was interesting to observe the TLD tracker as it was incredibly prone to false-positives, which was not initially expected, thus not making a good choice when using the OpenCV object trackers. Median-Flow Tracker was the fastest tracker to be tested in our environment which did a considerably good job when it came to reporting failures. However, if the object changes their appearance quickly enough, or there is a too large jump in object motion, the object changes their appearance quickly enough, or there is a too large jump in object motion, the tracker fails to perform. The tracker should not be used for fast-moving objects. The Kernelized Correlation Filters (KCF) tracker was one of the quicker trackers, faster than BOOSTING, MIL, and TLD tracker. It was noted that similar to MIL Tracker it was prone to occlusion but was much more reliable in its performance, making it one of the most favorable trackers that can be used. The future scope of this research includes a comprehensive examination of the application of these algorithms, along with the calculation of their efficiencies and the use of different techniques.

8.1 REFERENCES

[1] Q. Wang, F. Chen, W. Xu, and M.-H. Yang, “An experimental comparison of online object-tracking algorithms,” Wavelets and Sparsity XIV, Aug. 2011.

[2] P.Janku, K.Koplik, T.Dulik, and I.Szabo, “Comparison of tracking algorithms implemented in OpenCV,” MATEC Web of Conferences, vol. 76, p. 04031, 2016.

[3] Z. Kalal, K. Mikolajczyk and J. Matas, "Tracking-Learning-Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 7, pp. 1409-1422, July 2012.

[4] J. Hu, S. Hu and Z. Sun, "A real time dual-camera surveillance system based on tracking-learning-detection algorithm," 2013 25th Chinese Control and Decision Conference (CCDC), Guiyang, 2013, pp. 886-891

[5] Jig et al., "A Tracking-Learning-Detection (TLD) method with local binary pattern improved," 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, 2015, pp. 1625-1630.

[6] M.-A. Carbonneau, V. Cheplygina, E. Granger, and G. Gagnon, “Multiple instance learning: A survey of problem characteristics and applications,” Pattern Recognition, vol. 77, pp. 329–353, 2018.

[7] O. Maron, and T. Lozano-Pérez. "A framework for multiple-instance learning." In Advances in neural information processing systems, 1998, pp. 570-576.

[8] B. Babenko, M. Yang and S. Belongie, "Visual tracking with online Multiple Instance Learning," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, 2009, pp. 983-990.

[9] V.lehtola, H.Huttunen, F.Christophe, and T.Mikkonen “Evaluation of Visual Tracking Algorithms for Embedded Devices,” Image Analysis Lecture Notes in Computer Science, pp. 88-97, 2017.

[10] C. Zhang et al., "Output Constraint Transfer for Kernelized Correlation Filter in Tracking," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 4, pp. 693-703, April 2017.

[11] Varfolomieiev and O. Lysenko, “An improved algorithm of median flow for visual object tracking and its implementation on ARM platform,” Journal of Real-Time Image Processing, vol. 11, no. 3, pp. 527–534, 2013.

[12] Z. Kalal, K. Mikolajczyk and J. Matas, "Forward-Backward Error: Automatic Detection of Tracking Failures," 2010 20th International Conference on Pattern Recognition, Istanbul, 2010, pp. 2756-2759.

[13] X. Dong, J. Shen, W. Wang, Y. Liu, L. Shao and F. Porikli, "Hyperparameter Optimization for Tracking with Continuous Deep QLearning," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 518-527.

[14] Y. Wu, J. Lim and M. Yang, "Online Object Tracking: A Benchmark," 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, 2013, pp. 2411-2418.

[15] S. Moon, J. Lee, D. Nam, H. Kim and W. Kim, "A comparative study on multi-object tracking methods for sports events," 2017 19th International Conference on Advanced Communication Technology (ICACT), Bongpyeong, 2017, pp. 883-885.

[16] Bibi and B. Ghanem, "Multi-template Scale-Adaptive Kernelized Correlation Filters," 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), Santiago, 2015, pp. 613-620.

[17] W.Chen, K.Zhang, and Q.Liu, “Robust visual tracking via patch based kernel correlation filters with adaptive multiple feature ensemble,” Neurocomputing vol.214, pp.607-617, 2016.

A

MAJOR PROJECT REPORT

On

BABY CRADLE AND HEALTH MONITORING SYSTEM USING IOT

Submitted by

1) Ms. P. Venkata Sai Kumar (17K81A04N7)
2) Mr. K. Nagaraju (18K85A0438)
3) Mr. J. Ram Reddy (18K85A0440)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Esteemed Guidance of

Mr. S. Ravi Kumar

Associate Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE
An Autonomous Institute

Dhulapally, Secunderabad – 500 100

BONAFIDE CERTIFICATE

This is to certify that the project entitled "BABY CRADLE AND HEALTH MONITORING SYSTEM USING IOT", is being submitted by **1. Mr. P. Venkata Sai Kumar (17K81A04N7), 2. Mr. K. Nagaraju (18K85A0438), 3. Mr. J. Ram Reddy (18K85A0440)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

PROJECT GUIDE	**HEAD OF THE DEPARTMENT**
Mr. S. RAVI KUMAR	**Dr. B. HARI KRISHNA**
Associate Professor	**Professor**
Department of ECE	**Department of ECE**

Internal Examiner External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **POTHRALA VENKATA SAI KUMAR** WITH ROLL NO.**17K81A04N7**, **KANCHIPATI NAGARAJU** WITH ROLL NO.**18K85A0438**, **JANGAMGARI RAM REDDY** WITH ROLL NO.**18K85A0440**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERIN**G **COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**BABY CRADLE AND HEALTH MONITORING SYSTEM USING IOT**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

DECLARATION

We, the students of **Bachelor of Technology** in Department of ELECTRONICS AND COMMUNICATION ENGINEERING, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled "BABY CRADLE AND HEALTH MONITORING SYSTEM USING IOT" is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

1) Ms. P. Venkata Sai Kumar (17K81A04N7)

2) Mr. K. Nagaraju (18K85A0438)

3) Mr. J. Ram Reddy (18K85A0440)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtednessto **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr. P. Santosh Kumar Patra**, **our beloved Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr. B. Hari Krishna**, **Head &Professor**, the Dept. of Electronics and Communications for his/her advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We sincerely thank our project internal guide, Mr. S. Ravi Kumar, Associate Professor, Department of ECE and our class project coordinator Ms. P. pushpa Assistant Professor, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE**for their advice, supervision, and the vital contribution as and when required during this project work and for their support and co- operation that is difficult to express in words.

P. VENKATA SAI KUMAR	K. NAGARAJU	J. RAM REDDY
(17K81A04N7)	(18K85A0438)	(18K85A0440)

ABSTRACT

Juggling the responsibilities of a job and new born baby may seem challenging for working parents. Infant health safety and security is always a matter of concern for kith and kin by leaving their baby in care centers. At present, female participation in the work force in the industrialized nations has greatly increased, thereby affecting infant care in many families. Both parents are required to work due to the high cost of living. However, they still need to look after their babies, thereby increasing workload and stress, especially of the mother. Working parents cannot always care of their babies. They either send their babies to their parents or hire a baby caregiver while they are working. Some parents worry about the safety of their babies in the care of others. Thus, they go home to check on their babies during their free time. In this paper author has designed an advance cradle system which monitor baby health conditions such as temperature, baby cry and wetness. Parents not only monitor but also can control cradle via mobile. The experimental result shows that the designed system works successfully for the infant healthcare and thus can be implemented practically.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT:

At present, female participation in the work force in the industrialized nations has greatly increased, thereby affecting infant care in many families. Both parents are required to work due to the high cost of living. During early stages infant needs proper rest and sleep for growth and development. Hence, it is a responsibility of parents by guardian to provide necessary care and attention to the infant. But with the modern lifestyle, parents are busy and have a lot of work with little time to provide for the little ones. When a baby is born in a family there has to be someone to look after the baby. Some parents have to do a double task of keeping a check on the baby as well as do the household work.

However, they still need to look after their babies, thereby increasing workload and stress, especially of the mother. Working parents cannot always care for their babies. They either send their babies to their parents or hire a baby caregiver while they are working. It is always difficult to relay on some strangers to look after their baby. Some parents worry about the safety of their babies in the care of others. Thus, they go home to check on their babies during their free time, such as lunch or tea break. So to help such parents we have decided to come up with a smart cradle which will help a mother or a father have a track of their child and do their work simultaneously.

A baby monitoring system that can monitor the babies' condition real time is proposed to solve these problems. A baby monitoring system consisting of different sensors and Wi-Fi modules. This proposed system will give the parents relaxing time and it can send data and immediately notify the parents about urgent situations, thereby shortening the time needed to handle such scenarios. Generally, babies cry because they are hungry, tired, unwell, or need their diaper changed. If a system could understand the uneasiness of the baby or why the baby might be crying it means that for both working and non-working parents could save their time and provide the necessary care and attention to the infant. This baby cradle health monitoring system is used in various hospitals and maternity homes for infants to sleep in and take care of them.

1.2 OBJECTIVES OF THE STUDY:

Under fast-paced life conditions, everyone is busy in their professional life including parents. They leave the house early in the morning and come back before dinner time. Even the mothers are working. Thus, they do not have sufficient time to take care of their babies. Not all parents could afford a nanny to help them with their children. Then, after working for long hours, the mothers still have to manage the house and take care of their babies simultaneously. Parents might not have the time to take care of their baby.

Studies about the monitoring of a baby have been carried out and found that baby health conditions better while being monitored by the parents. Most available automated cradles are designed to check the infant health conditions non-stop. However, the existing systems not accurate and they doesn’t give absolute values. Thus, allowing the automated cradle to take care of the baby is also a problem. Several types of baby cradles are available in stores, but they are expensive, and not everyone can afford them. In addition, the existing automatic cradles in the literature have many limitations in terms of functionality, cost, and communication technology support.

To the best of our knowledge, no previous studies have developed a smart cradle with IOT support from scratch, similar to that in the present study. To overcome this problem, a new automatic IOT-based baby monitoring system is designed, allowing the parents to access an account to monitor the baby’s condition anywhere and anytime.

This proposed system will give the parents relaxing time and allow them to take less stress about wellbeing if the baby when they are away as they can get update about the status of infant inside the cradle. It can send data and immediately notify the parents about urgent situations, thereby shortening the time needed to handle such scenarios. Generally, babies cry because they are hungry, tired, unwell, or need their diaper changed. If a system could understand the uneasiness of the baby or why the baby might be crying it means that for both working and non-working parents could save their time and provide the necessary care and attention to the infant. This baby cradle health monitoring system is used in various hospitals and maternity homes for infants to sleep in and take care of them.

1.3 SCOPE OF THE STUDY:

IOT (**internet of things**) based baby cradle and health monitoring system is such a type of system that is used to monitor and check the health conditions of the baby present in the cradle with the help of internet resources and Wi-Fi module. At present scenario, the number of working women had been increased. They can't take care of their baby due to long working hours. So, they need to send their babies to their grandparent's home or some baby care centers. On the other hand it is difficult to afford the baby care taker due to financial instabilities and also requires man power as well. Even if the parents sends their baby to grandparents houses still, they worry about their baby and they can't see the real time conditions of their baby. If they want to check the conditions of the baby they have to come home from working place which is quite complicate and waste of time.

To overcome and to facilitate all the above situations, so many baby care health monitoring systems are available in the market but, they can't give accurate and absolute values about baby health conditions. Here we proposed a system that is called baby cradle and health monitoring system using IOT which is efficient, reliable, and friendly to use. By using this system the parents can monitor and check all the health conditions of the baby from their work place or else any place which is far away from the baby. This IOT based baby cradle and health monitoring system is very much useful to working parents as well as nurses in maternity units of hospital.

1.4 MATERIAL REQUIREMENT:

1.4.1 HARDWARE REQUIREMENT:

- Arduino Uno
- LCD
- Power Supply
- Wi-Fi Module
- Wet Sensor
- Heart Beat Sensor
- Lm35
- Sound Sensor

1.4.2 SOFTWARE REQUIREMENT:

- Proteus Software
- Arduino Software

1.5 PROCUMENT OF EQUIPMENT:

We brought all the required hardware components in an electronic manufacturing shop andbefore assembling and interfacing hardware components to Arduino UNO micro-controller, we connected individual components to Arduino in Proteus Software and understood working of every individual component in Proteus Software.

Now, we assembled all components and interfaced to Arduino UNO micro-controller using connecting wires through bread board and we wrote program to work all the hardware components in Arduino IDE we understood the working of all components in a practical way.

We were able to monitor the toll booth system with better efficiency, with less time delay and the results were also satisfactory and we successfully got the output in mobile telnet app and LCD screen. We would like to improve the project in future for further developments.

CHAPTER-2
LITERATURE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH AREA:

To address these challenges, we designed and fabricated a baby monitoring system for a smart cradle using Node MCU as the microcontroller while the system was developed using Arduino IDE. This system consists of a cradle that can swing whenever the sound sensor detects crying. A mini fan is attached on top of the cradle to provide ventilation. The mini fan and the swinging of the cradle can be switched on either by the sensors or through remote control from the MQTT server. An external Wi-Fi camera has been installed on the cradle to enable real-time vision monitoring. The parents can see the baby's condition and talk to the baby using the ready-made mobile application of the Wi-Fi camera. An Internet of Things-based baby monitoring system for smart cradle is proposed in this paper. The novelty of this work lies in the proposed IOT-BBMS automation system by adopting the following methodology and contributions:

(i) A smart baby cradle prototype is designed and fabricated with auto-swinging support, web camera and musical toy to test the proposed system.

(ii) A new Algorithm is proposed and implemented in Node MCU controller to perform the required monitoring and control tasks.

(iii) Utilizing the Node MCU as the microcontroller and adafruit MQTT as IOT server to retrieve data from sensors and send commands to actuators.

(iv) The system could gather accurate real time data and response by actuating the proper relay to switching fan, toys and swinging motor.

2.2 REVIEW ON RELATED LITERATURE:

Few studies have investigated the possibilities of automated baby cradle using different perspectives. A baby monitoring system has been proposed in [1], in which an enhanced noise cancelling system that monitors the baby and reduces sound pollution has been suggested. The main function of the system is to reduce the noise that might disturb the baby by playing relaxing songs.

This system can also adjust the room's light intensity with the aid of a light sensor. However, our system has more advanced features, such as supporting real-time monitoring over the IOT network and vision monitoring using web camera.

The authors Goyal and Kumar [2] introduced an E-baby cradle that can swing automatically when it detects crying and stops swinging when the crying stops. The speed for the swinging cradle can be controlled based on the user's need. First, the alarm goes off when the mattress is wet, indicating that the mattress should be changed. Second, when the baby does not stop crying for a certain time, the alarm alerts the parents to attend to their baby. However, it is only applicable when parents are near the cradle, because it only uses a buzzer alarm, the sound of which might frighten the baby. Parents cannot monitor their baby when they are away from home, for example when at work or when traveling to other places. A similar automatic baby monitoring system was proposed in [3].

The authors developed a low-budget system that swings the cradle when the crying sound is detected, and the cradle stops when the baby stops crying. The built-in alarm goes off under either one of the following conditions: the mattress is wet or the baby does not stop crying after a certain period.

2.3 CONCLUSION ON REVIEW:

The overall methodology adopted in this research had been discussed earlier. Issues in the existing systems were identified by conducting a comprehensive literature review on studies related to baby monitoring systems. Then, we introduced a smart cradle that combines the concept of IOT with baby monitoring system. Subsequently, the selection of material for the smart cradle was carried out. All the hardware and materials used in building this system, which were suitable for a baby, were selected. The priority is to ensure the safety of the baby. The modelling phase is followed by the system design, determining the GUI of applications, and prototype phase.

The system design is separated into two phases, namely, the cradle design and control system design. A cradle prototype for the baby monitoring system was designed. In the control system design, the types of electronic components were determined and purchased for implementation in the system. Then, coding was performed according to how the system was proposed. After the modelling phase, the designed baby monitoring system was then enhanced and optimized through several tests to achieve the expected outcome. Subsequently, the system was installed on the cradle prototype for the testing phase before finalizing the smart cradle.

When the testing failed due to some coding errors or other problems, the testing phase was repeated until the cradle achieved the expected outcome that satisfied the research object.

CHAPTER-3

EMBEDDED SYSTEMS

3.1 INTRODUCTION TO EMBEDDED SYSTEMS:

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer. The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes. The applications software on such processors is sometimes referred to as firmware. The simplest devices consist of a single microprocessor (often called a "chip”), which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine. As the embedded system is the combination of both software and hardware.

3.2 APPLICATION OF EMBEDDED SYSTEMS:

- Manufacturing and process control
- Construction industry
- Transport

- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

3.3 MICRO PROCESSOR (µp):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Processors are classified into four types like:

- Micro Processor (µp)
- Micro controller (µc)
- Digital Signal Processo1r (DSP)
- **Application Specific Integrated Circuits (ASIC)**

3.4.1 THREE BASIC ELEMRNTS OF MICROPROCESSOR:

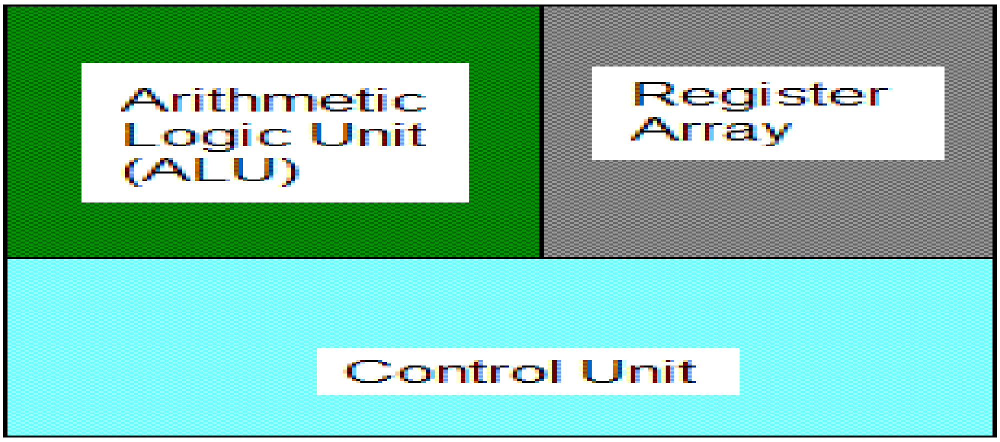


Fig.3.3.1: Three Basic Elements Of Microprocessor

3.4 HARVARD ARCHITECTURE:

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

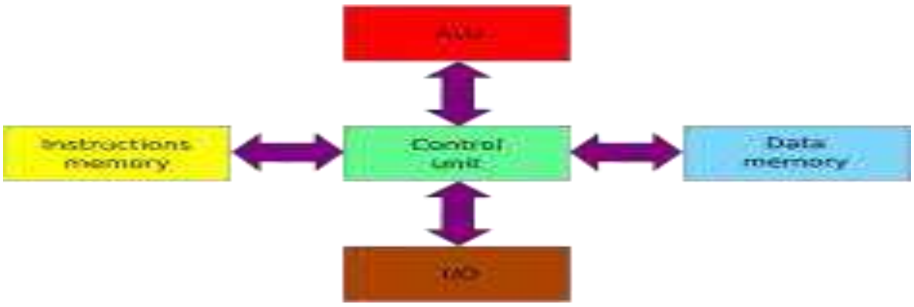


Fig.3.4.1: Harvard Architecture

3.4.1 USES OF THE HARVARD ARCHITECTURE:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces. Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behaviour must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.

Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip

Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture

3.4.2 VON- NEUMANN ARCHITECTURE:

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. They cannot be performed at the same time. The von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A stored-program digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

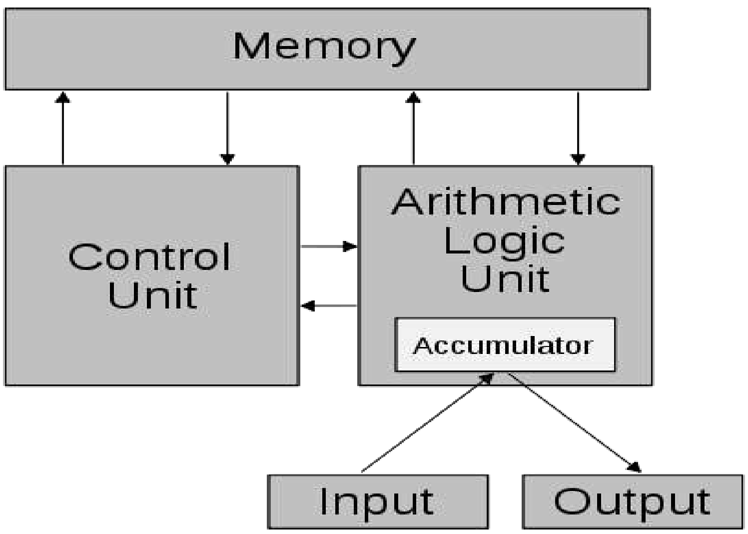


Fig.3.4.2: Schematic of the Von-Neumann Architecture.

CHAPTER-4

PROJECT DESIGN

4.1 OVERVIEW OF DESIGN:

The cradle is designed to connect to the all sensors to it. It swings when the switch is turned on by the user or when the sound sensor detected the baby's cry. The control circuit components are installed into the attached electrical box after the circuit testing phase. Figure 4.1 shows the circuit diagram of the virtual connection of electrical components using Fritzing software. The components are installed on the breadboard to ensure that the system works successfully before transferring the components onto the solder circuit panel. The shaft of the DC motor is connected to the cradle to allow swinging whenever the baby cry is detected or initiated by the user. The LED in Figure 6 replaces the lullaby toy given that the circuit design software does not have a library of lullaby toys. The control system of the smart baby cradle is equipped with 5 V USB power source, sound sensor module, temperature and humidity sensor, relays, ESP 8266 Wi-Fi module, and geared motor. Figure 4.1 shows the stimulation of temperature and wet sensors in the Proteus software. After the electronics stimulation in Proteus Simulation software, all the electrical and electronic components were assembled and connected to the micro-controller and programmed using Arduino software (IDE).

However, the Proteus software was not fully utilized, because some of the component libraries were not found in the software, as well as on the Internet. Therefore, the sensors and components had to be tested the Proteus software. In the practical implementation of the designed system, the original 9 V DC battery source was changed to AC to DC 12 V output converter, because the 9 V DC battery was unable to supply the current efficiently to the entire circuit. Hence, an AC to DC power supply was used. A 5 V 3 A USB module was used to convert the 12 V output to 5 V output for the activation of the mini fan. It was also used to supply electricity to the NodeMCU microcontroller. The USB module has a built-in step-down converter that can reduce an input of 6–24 V to the output of 5 V, with a maximum output current of 3 A at full load. The connection of the NodeMCU is coded to connect to a certain Wi-Fi network with the given password written in the coding, because the Internet of UMP has certain restrictions, which would cause the connection of the system to fail. The process is then retested using the programmer mobile phone’s data hotspot.

The connection was successful, and no time delay was encountered during the performance of the sending and fetching of data to and from the MQTT server. The DHT22 temperature and humidity sensor was mounted on the side of the electrical box by drilling a hole to enable the sensor to fit in the hole, as shown in Figure 8 (a), to measure the ambient temperature. Thus, the DHT22 sensor can be exposed to the surrounding temperature and then measured for uploading to the MQTT server. A sound sensor is placed at the top of the baby’s cradle to enable sound detection. Figure 8(b) shows the MD10-POT 10 Amp DC motor driver used to vary the speed of the 12 V DC motor that swings the cradle. The motor driver supports the motor voltage that ranges from 7 V to 30 V and can withstand the maximum current up to 10 A continuous and 30 A peak (10 seconds). The speed, round per minute (RPM), of the connected DC motor can be varied by tuning the potentiometer and the rotating direction with a two-direction switch. The black knob under the DHT22 in Figure 8 (a) is the potentiometer of the DC motor driver that allows the user to manually adjust the speed of the baby cradle.

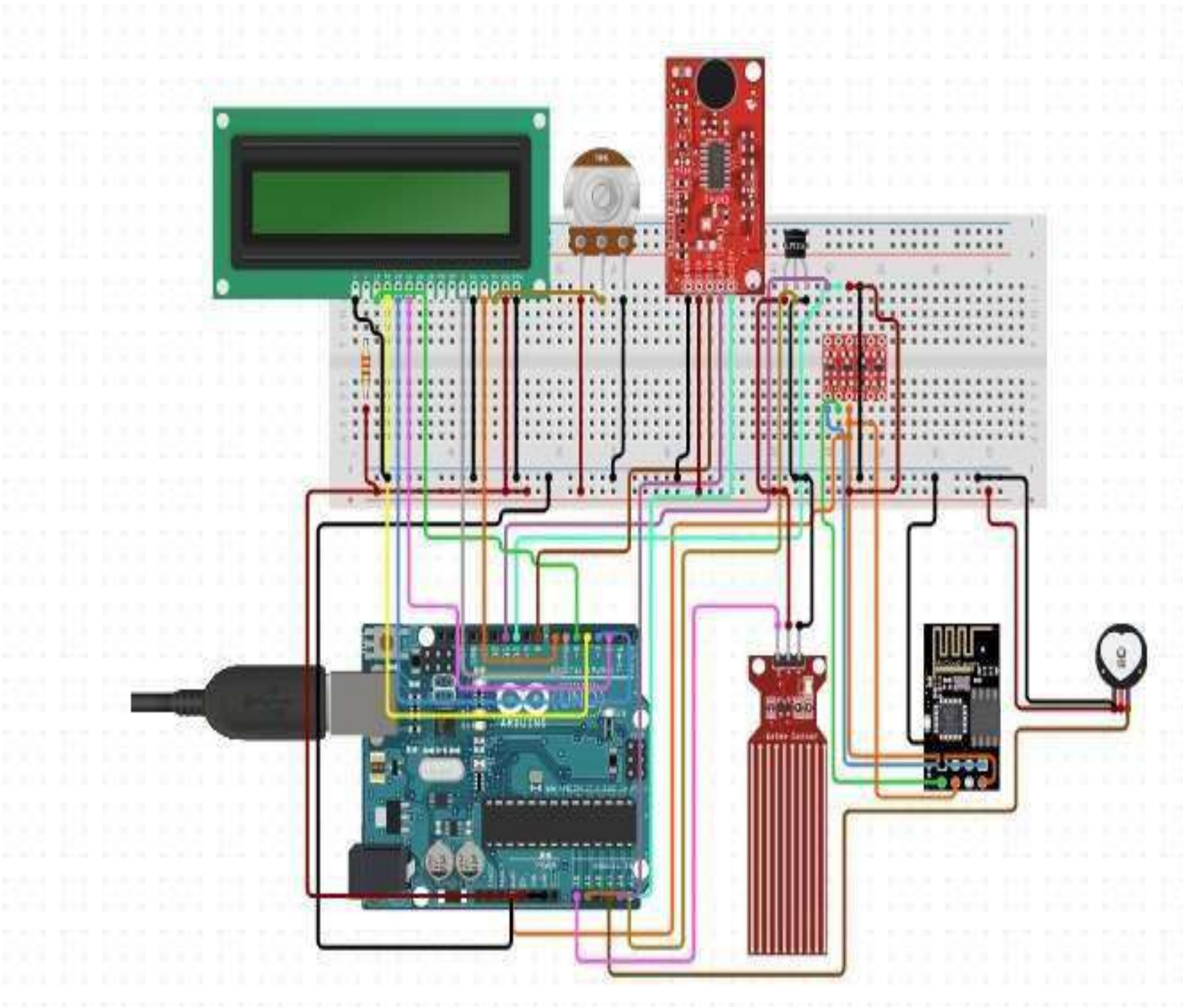


Fig.4.1 Overview Of The Design

4.2 EQUIPMENT ANALYSIS:

4.2.1 ARDUINO UNO:

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The ATmega8 microcontroller contains 32 general purpose working registers. As shown in the below figure these registers are directly connected to ALU. Two registers can carry one single instruction consequently in one clock cycle. Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions. You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software). Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

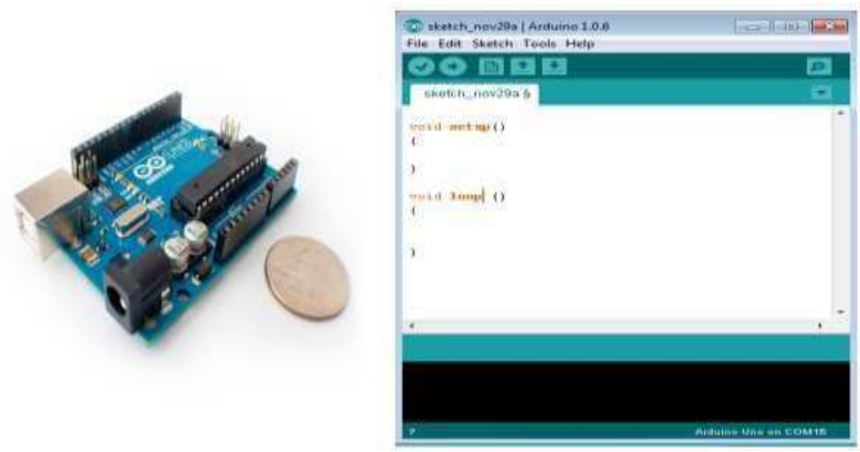


Fig 4.2.1 Arduino Uno

4.2.2 BOARD TYPES:

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE. The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega 16U2

Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega 16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header

Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 4.2.2 Arduino boards based on ATMEGA328 microcontroller

Board Name	**Operating Volt**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 4.2.3 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI Compatible Header

Table 4.2.4 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table 4.2.5 Arduino boards based on AT91SAM3X8E microcontroller

4.2.3 BOARD DESCRIPTION:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

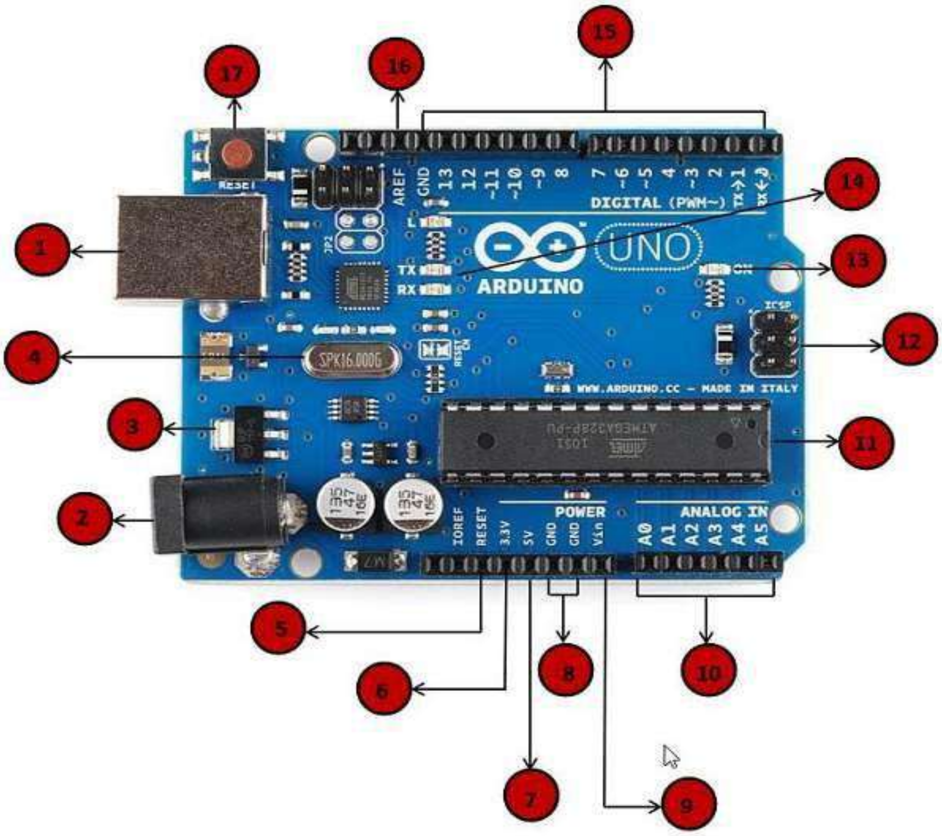


Fig 4.2.2 Board Description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. The Arduino calculates time by using the crystal oscillator. The number printed on top of the Arduino crystal is

	16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works with 3.3 volt and 5 volt. • GND (8) (Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator**

	This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labelled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

4.2.4 ARDUINO FAMILY:

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino. Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

Fig.4.2.3 Arduino Family

4.2.5 ADVANTAGES OF ARDUINO:

- It is cheap
- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
- For beginners, it is very simple to use.

4.2.6 APLLICATIONS

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model.

4.3 POWER SUPPLY UNIT:

4.3.1 INTRODUCTION:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as "Regulated D.C Power Supply".

4.3.2 BLOCK DIAGRAM OF POWER SUPPLY:

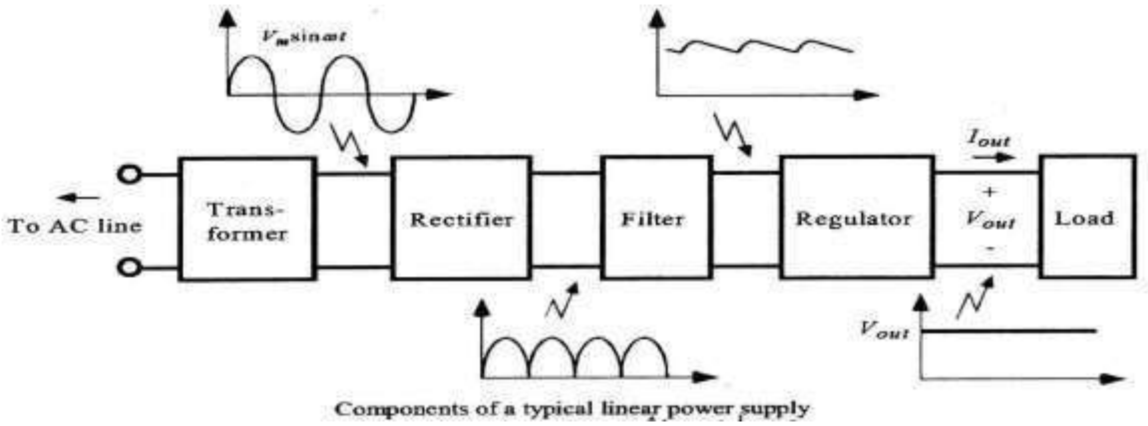


Fig.4.3.1 Block Diagram of Power Supply

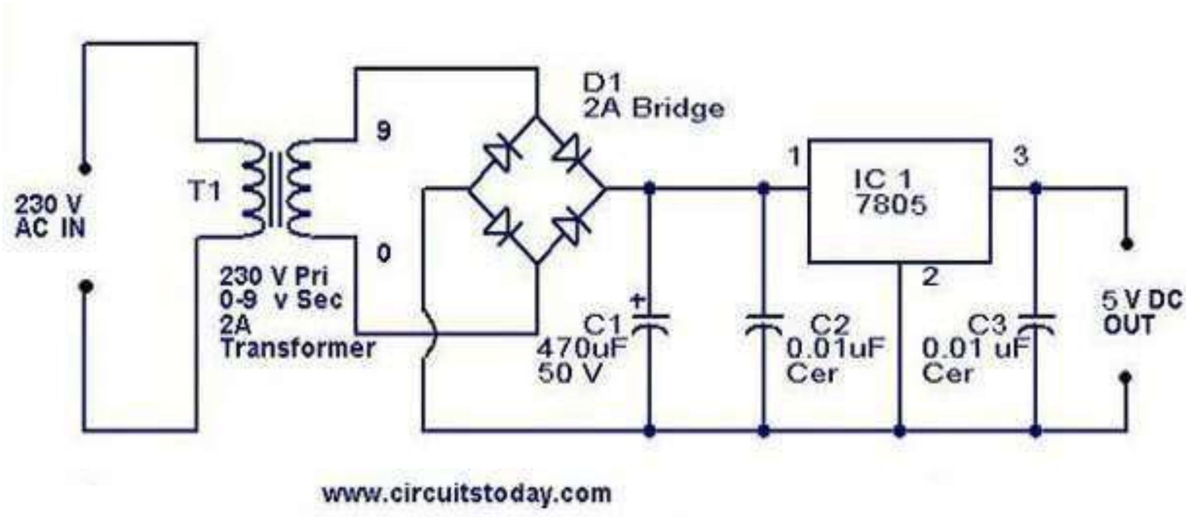


Fig.4.3.2 Schematic Diagram of Power Supply

4.3.3 DESCRIPTION OF POWER SUPPLY:

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a different form of energy – such as solar, mechanical, or chemical - into electrical energy. A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered. For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

4.3.4 TRANSFORMER:

A transformer is a device that transfers electrical energy transfers from one circuit to another through inductively coupled conductors to the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction (or) Transformer. A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa. Transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core. Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core.

Transformers can increase voltage (step-up) as well as reduce voltage (step-down). Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow.

The iron core is laminated to prevent 'eddy currents' flowing in the core. These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

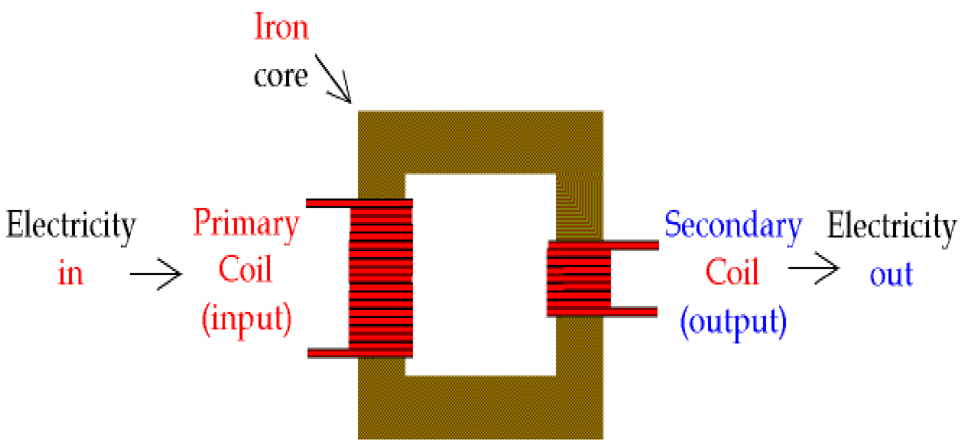


Fig.4.3.3 Transformer

4.3.5 DIODES:

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

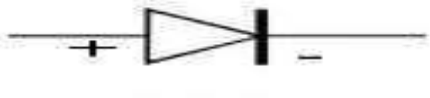


Fig.4.3.4 Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

4.3.6 RECTIFIER:

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

The Half-Wave Rectifier: The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure

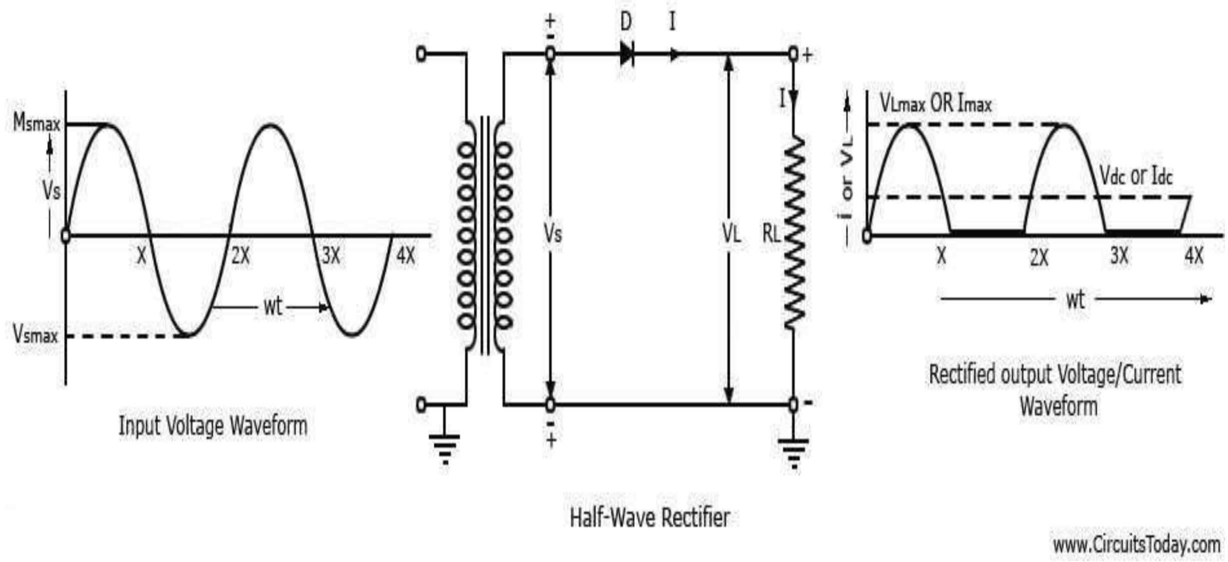


Fig.4.3.5 Half Wave Rectifier

As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs. While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

The Full-wave Rectifier:

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC.

.

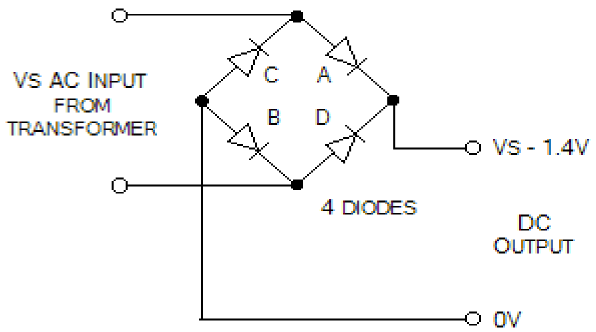


Fig.4.3.6 Full-Wave Rectifier

4.3.7 CAPACITOR FILTER:

The capacitor-input filter, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

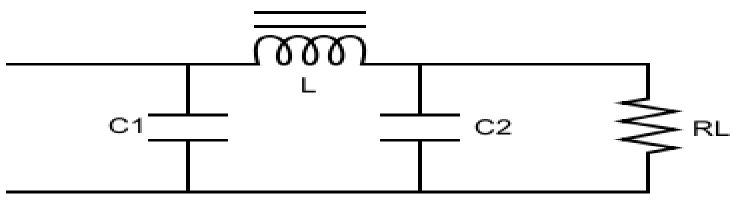


Fig.4.3.7 Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result, the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result, only the DC component appears across the load RL.

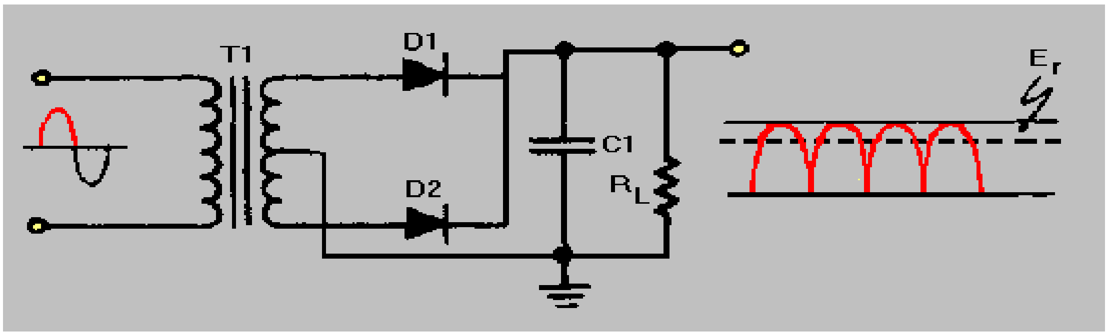


Fig.4.3.8 Centered Tapped Full-Wave Rectifier with a Capacitor Filter

4.3.8 VOLTAGE REGULATOR:

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx' indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx' indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consist the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

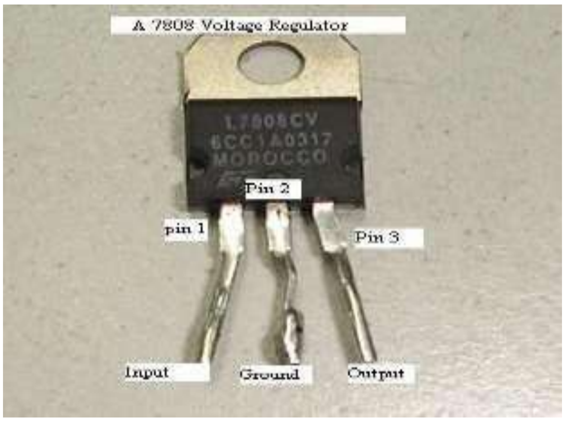


Fig.4.3.9 Regulator

4.4 16*2 ALPHANUMERIC LCD:

4.4.1 LCD SCREEN DISPLAY:

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the LCD has different memories to display data, those are discussed below.

4.4.2 BLOCK DIAGRAM OF LCD:

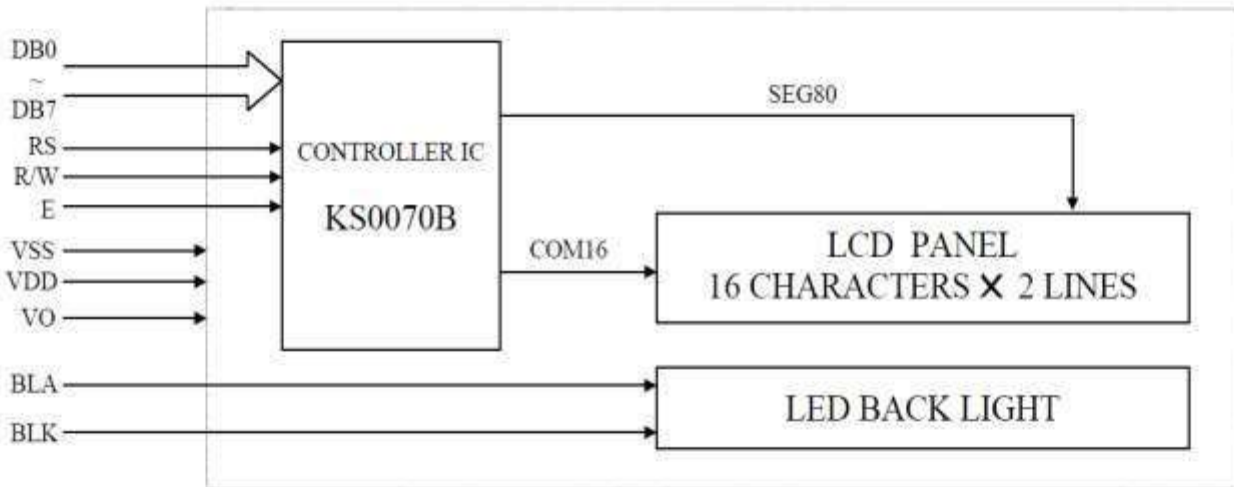


Fig.4.4.1 LCD Block Diagram

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Fig.4.4.2 DDRAM Addresses For 2 Line LCD

4.4.3 CHARACTER GENERATOR ROM:

Now you might be thinking that when you send an ASCII value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM.

4.4.4 BUSY FLAG:

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e. BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing. To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

4.4.5 INSTRUCTION REGISTER (IR) AND DATA REGISTER (DR):

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g. LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD. Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

4.4.6 16 X 2 ALPHANUMERIC LCD MODULE FEATURES:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing.
- 61 x 15.8 mm viewing area and 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line.
- Can display 224 different symbols
- Low power consumption (1 mA typical)

4.4.7 SCHEMATIC DIAGRAM:

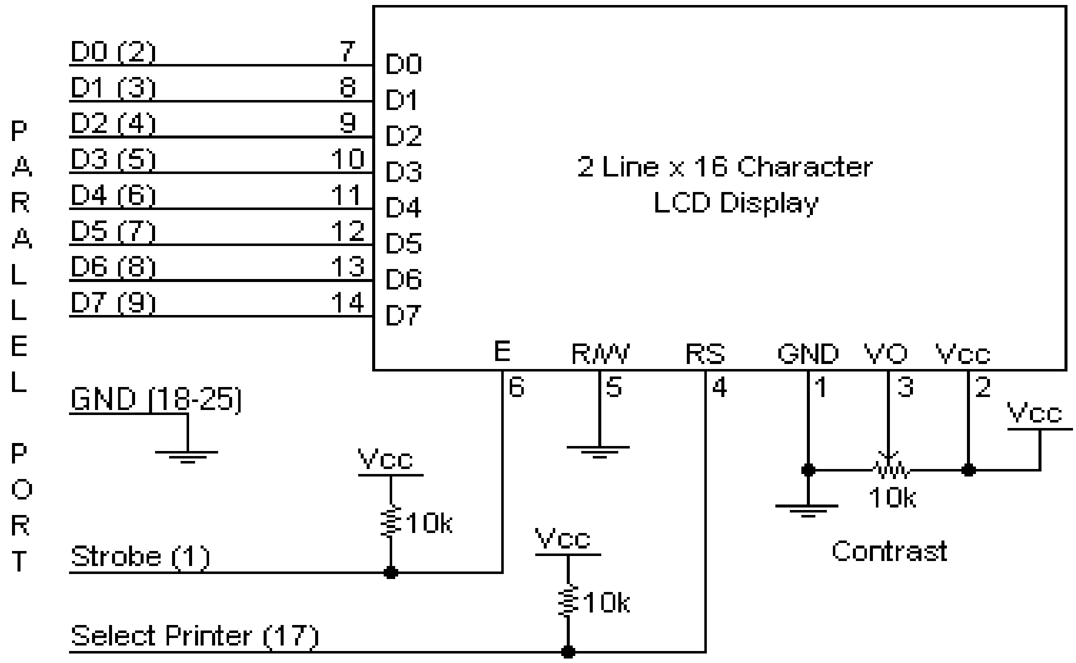


Fig.4.4.3 Schematic Diagram Of LCD

4.4.8 CIRCUIT DESCRIPTION:

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors. We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program. The 10k Potentiometer controls the contrast of the LCD panel.

You can use a bench power supply set to 5v or use a on board +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly. The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

4.5 LM35 SENSOR:

Lm35 is a Temperature Sensor which converts temperature value into electrical signals. We used IC called LM 35 as a temperature sensor. LM35 series sensors are precision integrated-circuit temperature sensors whose output voltage is linearly proportional to the Celsius temperature. The LM35 requires no external calibration since it is internally calibrated. . The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°C over a full −55 to +150°C temperature range.

The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies.

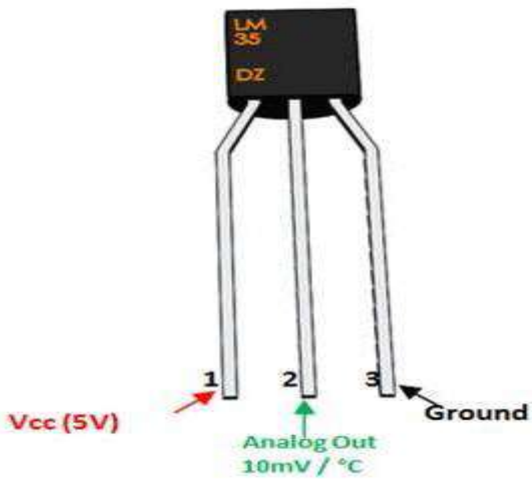


Fig.4.5 Temperature Sensor

4.5.1 THE CHARACTERISTICS OF LM35 SENSOR IS:

For each degree of centigrade temperature it outputs 10milli volts. ADC accepts the output from LM35 and converts that data into digital form which is sent to microcontroller for further processing. The methods of temperature measurement may be divided into two main classes according as the exchange of heat between the testing body and the hot system takes place by contact, or by radiation across a space. In the contact methods, thermometers or thermocouples are used and they are immersed in solids or liquids.

The thermodynamic equilibrium between the hot body and the testing body is established by material contact. In the non-contact methods, the thermodynamic equilibrium is established by the radiation emitted as excited atom.

4.5.2 PIN CONFIGURATION:

Pin Number	Pin Name	Description
1	Vcc	Input voltage is +5V for typical applications
2	Analog Out	There will be increase in 10mV for raise of every 1°C. Can range from -1V(-55°C) to 6V(150°C)
3	Ground	Connected to ground of circuit

Table 4.5 Pin Configuration Of Temperature Sensor

4.5.3 FEATURES AND SPECIFICATIONS:

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteed (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±1⁄4°C typical
- Low impedance output, 0.1 W for 1 mA load

4.5.4 APPLICATIONS:

- Measuring temperature of a particular environment
- Providing thermal shut down for a circuit/component
- Monitoring Battery Temperature and Measuring Temperatures for HVAC applications.

4.6 HEART BEAT SENSOR:

4.6.1 ABOUT HEART BEAT SENSOR?

Heartbeat Sensor is an electronic device that is used to measure the heart rate i.e. speed of the heartbeat. Monitoring body temperature, heart rate and blood pressure are the basic things that we do in order to keep us healthy. The working of the Pulse/Heart beat sensor is very simple. The sensor has two sides, on one side the LED is placed along with an ambient light sensor and on the other side we have some circuitry. This circuitry is responsible for the amplification and noise cancellation work. The LED on the front side of the sensor is placed over a vein in our human body. This can either be your Finger tip or you ear tips, but it should be placed directly on top of a vein. Now the LED emits light which will fall on the vein directly. The veins will have blood flow inside them only when the heart is pumping, so if we monitor the flow of blood we can monitor the heart beats as well. If the flow of blood is detected then the ambient light sensor will pick up more light since they will be reflect ted by the blood, this minor change in received light is analyzed over time to determine our heart beats.

Fig.4.6 Heart Beat Sensor Diagram

4.6.2 PIN CONFIGURATION:

Pin Number	Pin Name	Wire Colour	Description
1	Ground	Black	Connected to the ground of the system
2	Vcc	Red	Connect to +5V or +3.3V supply voltage
3	Signal	Purple	Pulsating output signal.

Table 4.6 Pin Configuration

4.6.3 WORKING OF HEART BEAT SENSOR:

4.6.4 FEATURES AND SPECIFICATIONS:

- Biometric Pulse Rate or Heart Rate detecting sensor
- Plug and Play type sensor
- Operating Voltage: +5V or +3.3V
- Current Consumption: 4mA
- Inbuilt Amplification and Noise cancellation circuit
- .Diameter: 0.625”
- Thickness: 0.125” Thick

4.6.4 APPLICATIONS:

- Sleep Tracking
- Anxiety monitoring
- Remote patient monitoring/alarm system
- Health bands
- Advanced gaming consoles

4.7 SOUND SENSOR:

Sound detection sensor module detects the intensity of sound where sound is detected via a microphone and fed into an LM393 op-amp. It comprises an onboard potentiometer to adjust the set point for sound level.

Fig.4.7 Sound Sensor Diagram

4.7.1 ABOUT SOUND SENSOR MODULE:

Sound Detection Sensor Module consists of four pins i.e. VCC, GND, DO, AO. Digital out pin is connected to the output pin of LM393 comparator IC while the Analog pin is connected to Microphone. The internal Circuit diagram of the Sound Detection Sensor Module is given below. Using Sound Detection Sensor Module with a microcontroller is very easy. Connect the Analog/Digital Output pin of the module to the Analog/Digital pin of Microcontroller. Connect VCC and GND pins to 5V and GND pins of Microcontroller. When the sound level exceeds the setpoint, an LED on the module is illuminated and the output is set low.

This Sound Detection Sensor Module consists of a Microphone, resistors, capacitor, potentiometer, comparator LM393 IC, Power, and status LED in an integrated circuit. LM393 IC LM393 Comparator IC is used as a voltage comparator in this Sound Detection Sensor Module. Pin 2 of LM393 is connected to Preset (10KΩ Pot) while pin 3 is connected to Microphone. The comparator IC will compare the threshold voltage set using the preset (pin2) and the Microphone pin (pin3).

Microphone: The microphone in the Sound sensor module detects the sound. This sound is fed into the LM393 IC.

Preset (Trimmer pot): Using the onboard preset, you can adjust the threshold (sensitivity) of the digital output.

4.7.2 PIN CONFIGURATION:

Pin Name	Description
VCC	The Vcc pin powers the module, typically with +5V
GND	Power Supply Ground
DO	Digital Output Pin. Directly connected to digital pin of Microcontroller
AO	Analog Output Pin. Directly connected to an analog pin of Microcontroller

Table 4.7 Pin Configuration

4.7.3 SPECIFICATIONS:

- Operating voltage: 3.3V to 5V DC
- LM393 comparator with threshold preset
- PCB Size: 3.4 * 1.6cm
- Induction distance: 0.5 Meter
- Operating current; 4~5mA
- Microphone sensitivity (1kHz): 52 to 48 dB

4.7.4 APPLICATIONS:

- Hearing aids
- Telephones
- Tape recorders and karaoke
- Live and recorded audio engineering
- Radio and television broadcasting
- Speech recognition technology

4.8 WET SENSOR:

4.8.1 ABOUT WET SENSOR

The moisture sensor is used to measure the wet condition of the baby's bed. It is fixed in the bed of the baby and measure the values. The probe can be powered with an DC supply or batteries in the range of 3.5 to 20 volts. The output is a voltage in the range of 0 to 3V, so any multimeter can be used to measure the moisture level.

Fig.4.8 Wet Sensor Diagram

4.8.2 PIN CONFIGURATION:

S.No:	Name	Function
1	VCC	Connects supply voltage- 5V
2	GND	Connected to ground
3	D0	Digital pin to get digital output
4	A0	Analog pin to get analog output

Table 4.8 Pin Configuration

4.8.3 SPECIFICATIONS:

- Measuring ranges: permittivity 1 - 80, bulk conductivity 0 - 200 mS.m-1, temperature -5 to 50°C, volumetric soil moisture 0 -1 m3.m-3.
- Accuracy: permittivity ±.2.5, bulk conductivity ± 10 mS.m-1, temperature ± 0.7, volumetric soil moisture ± 0.03 m3.m-3.
- Response time: ~5 seconds.
- Calibration: suitable for a variety of standard soil types.
- Environmental sealed to IP67, connector to IP65.
- Operating temperature: 0 to 50 °C.

4.8.4 APLPLICATIONS:

- Salinity monitoring
- Plant nutrient status and fertigation
- Pollution and land reclamation
- Soil moisture mapping Specifications

4.9 DEFINE MODULES:

4.9.1 ESP8266 WIFI MODULE:

The **ESP8266 Wi-Fi Module** is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your **Wi-Fi** network. The **ESP8266** is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

4.9.2 CHARACTER CONFIGURATION:

- 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2),
- general-purpose input/output (16GPIO),
- Inter-Integrated Circuit (I^2C) serial communication protocol,
- analog-to-digital conversion (10-bitADC)
- Serial Peripheral Interface (SPI) serial communication protocol,
- I^2S (Inter-IC Sound) interfaces with DMA(Direct Memory Access) (sharing pins with GPIO),
- UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2),and
- Pulse-width modulation (PWM).

4.9.3 APPLICATIONS:

- Smart Home Appliances
- Improved Home safety and security
- Home air quality and water quality monitoring
- Natural Language-based voice assistants.

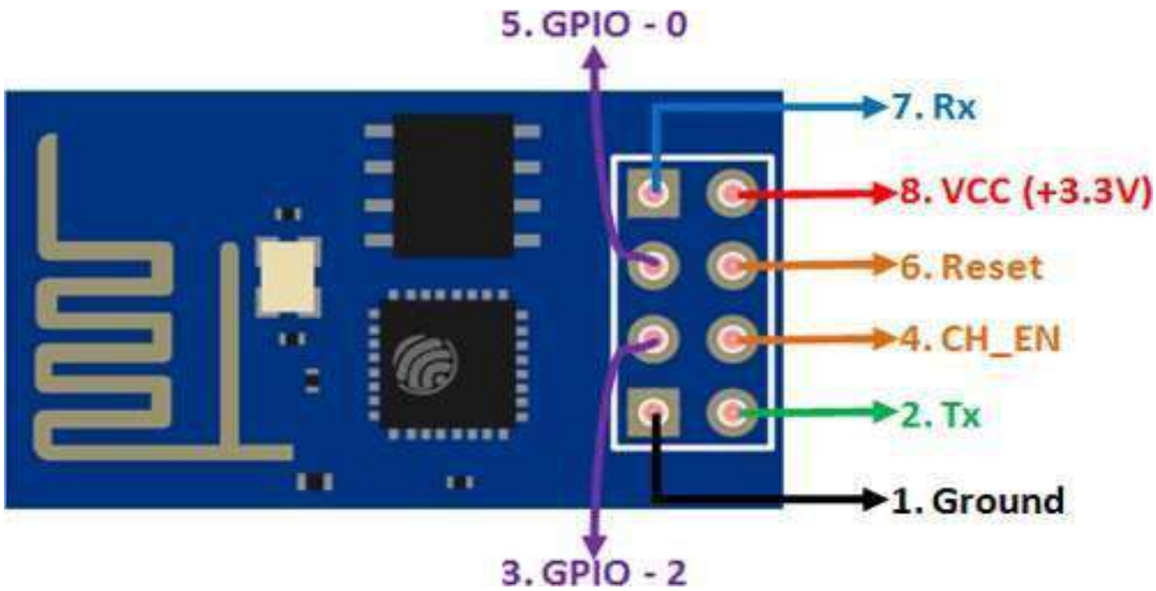


Fig.4.9.1 ESP8266 Wi-Fi Module Pin Out

4.9.4 STRUCTURE AND CONFIGURATION:

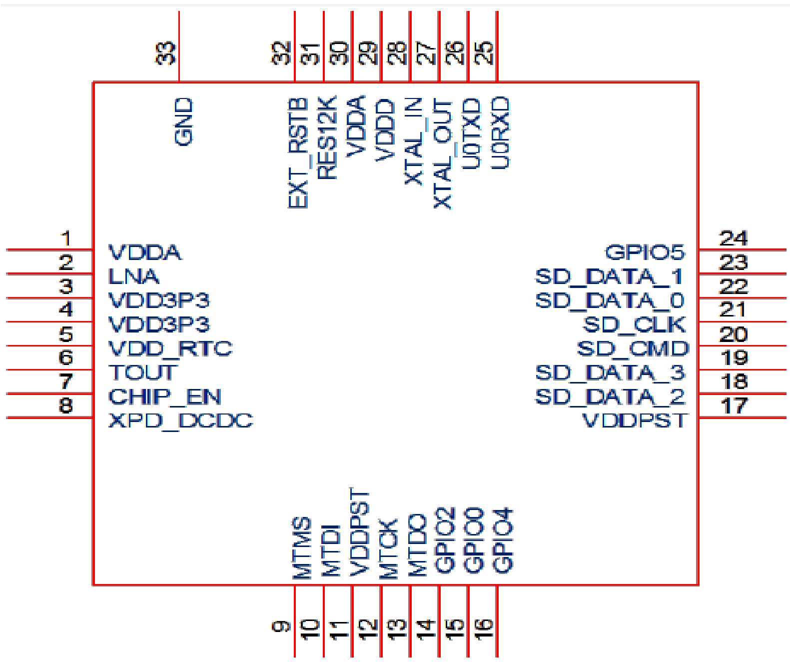


Fig.4.9.2 Structure and configuration

4.10 MODULE FUNCTIONALITIES:

4.10.1 ESP8266 WIFI MODULE

The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much Wi-Fi ability as a Wi-Fi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost-effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area.

CHAPTER-5
SOFTWARE DESCRIPTION

5.1 SOFTWARE REQUIREMENTS:

- Arduino software
- Programming language
- Proteus simulation

5.1.1 ARDUINO SOFTWARE:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments.

A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming.

As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IOT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

5.1.2 WHY ARDUINO?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics.

Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers.
- The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version. Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money

- Getting Started with Arduino and Genuino products:-
  Install the Arduino Software (IDE) on Windows PCs-
  This document explains how to install the Arduino Software (IDE) on Windows machines.

✓ Download the Arduino Software(IDE)

✓ Proceed with board specific instructions.

5.1.3 HOW TO DOWNLOAD THE ARDUINO SOFTWARE (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.
When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

5.1.4 INSTALLATION:

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.5.1.1 Standard USB Cable (A plug to B plug)

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig.5.1.2 Standard USB Cable (an A to Mini-B plug)

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

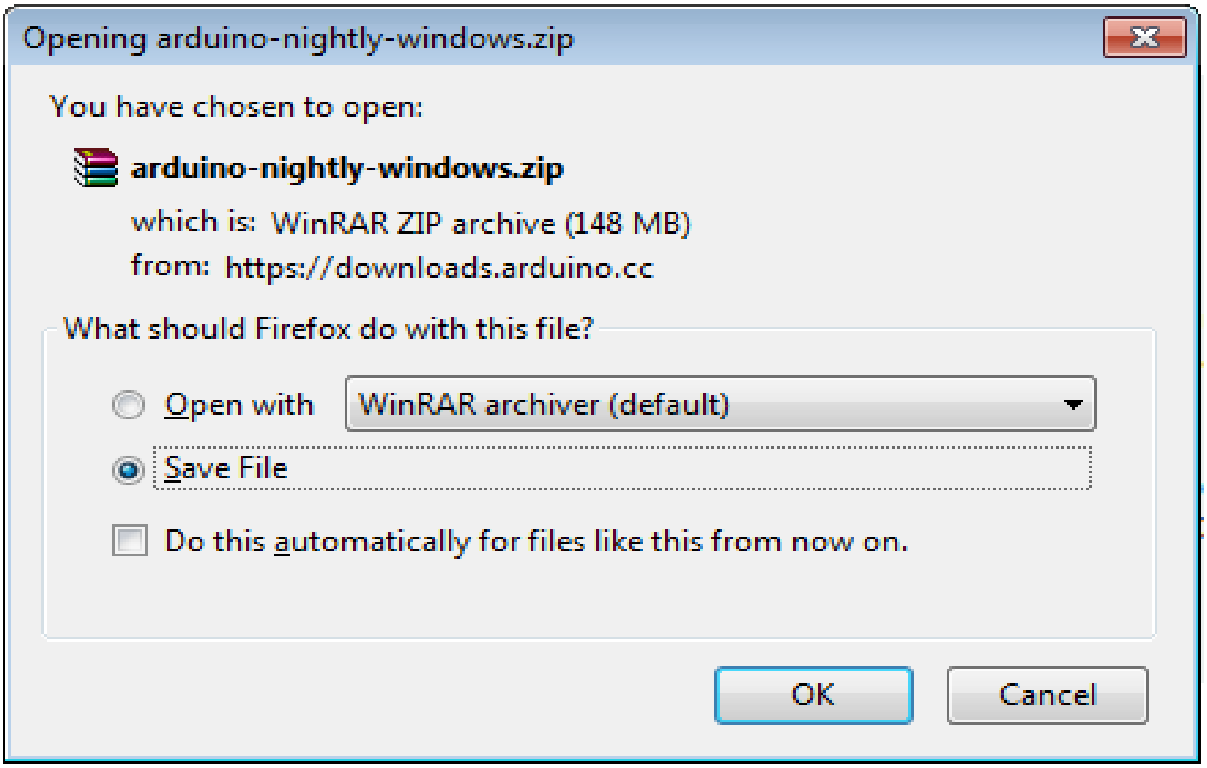


Fig.5.1.3 Download Arduino IDE Software

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

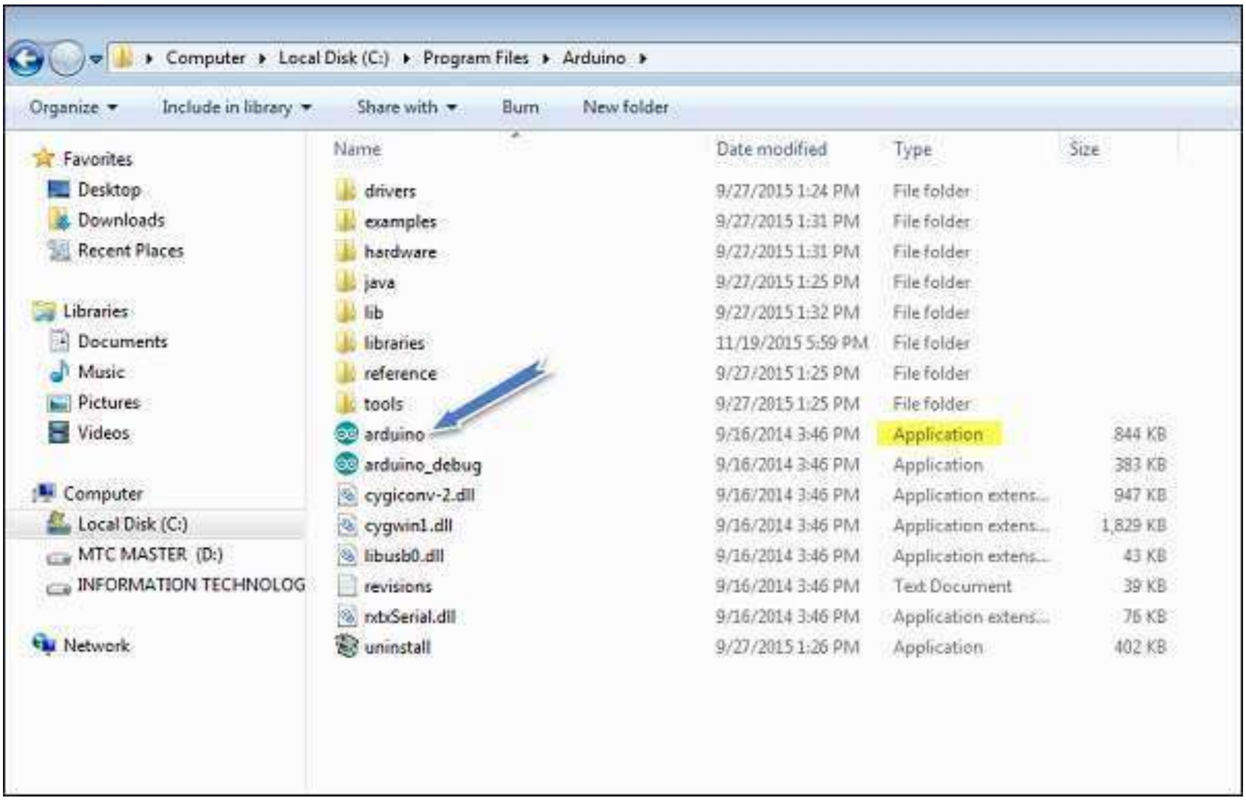


Fig.5.1.4 Launch Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.

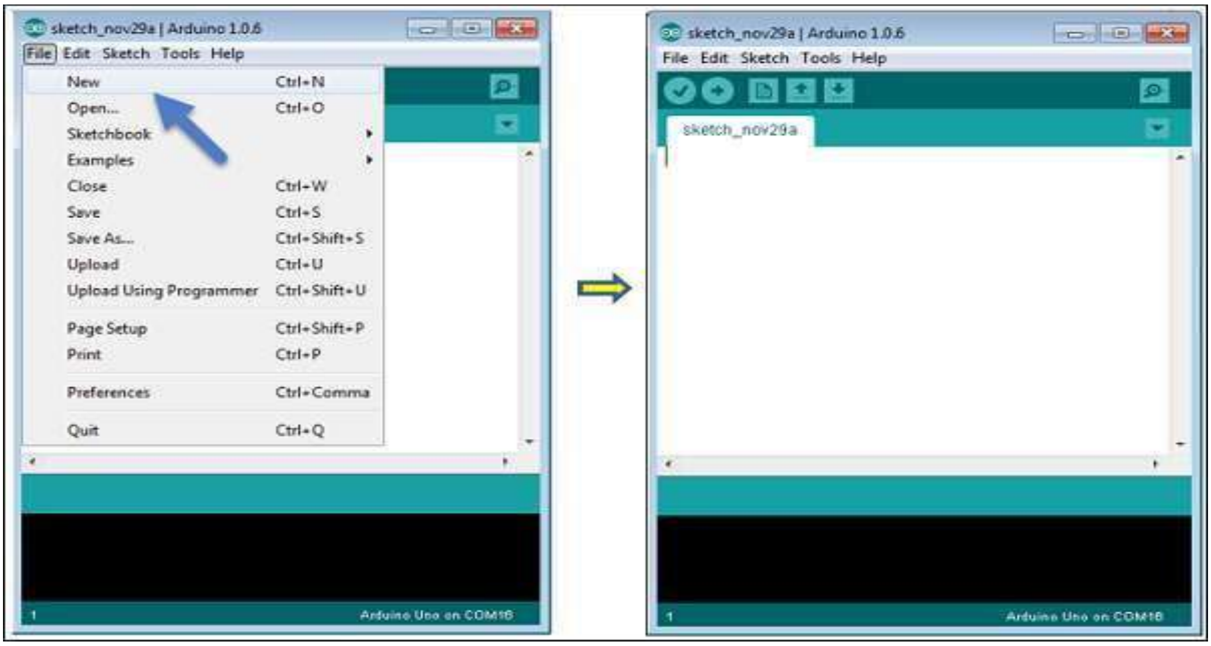


Fig.5.1.5 Open Your First Project

To open an existing project example, select File → Example → Basics → Blink.

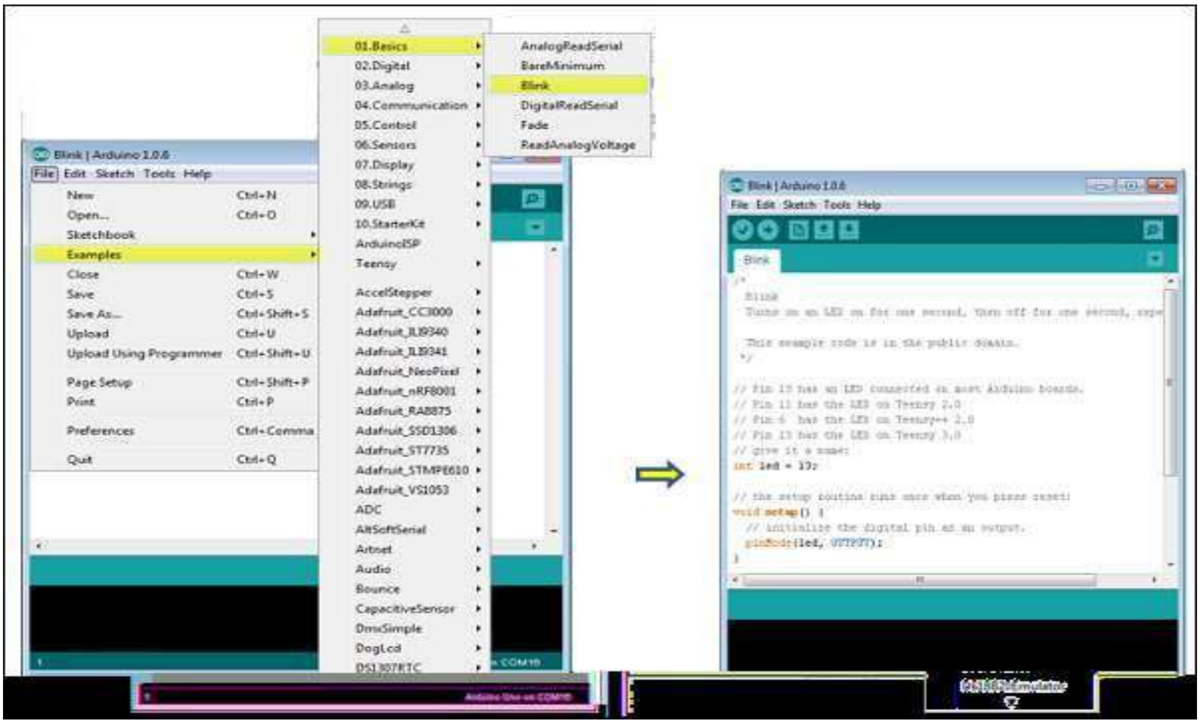


Fig.5.1.6 Open Your First Program

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

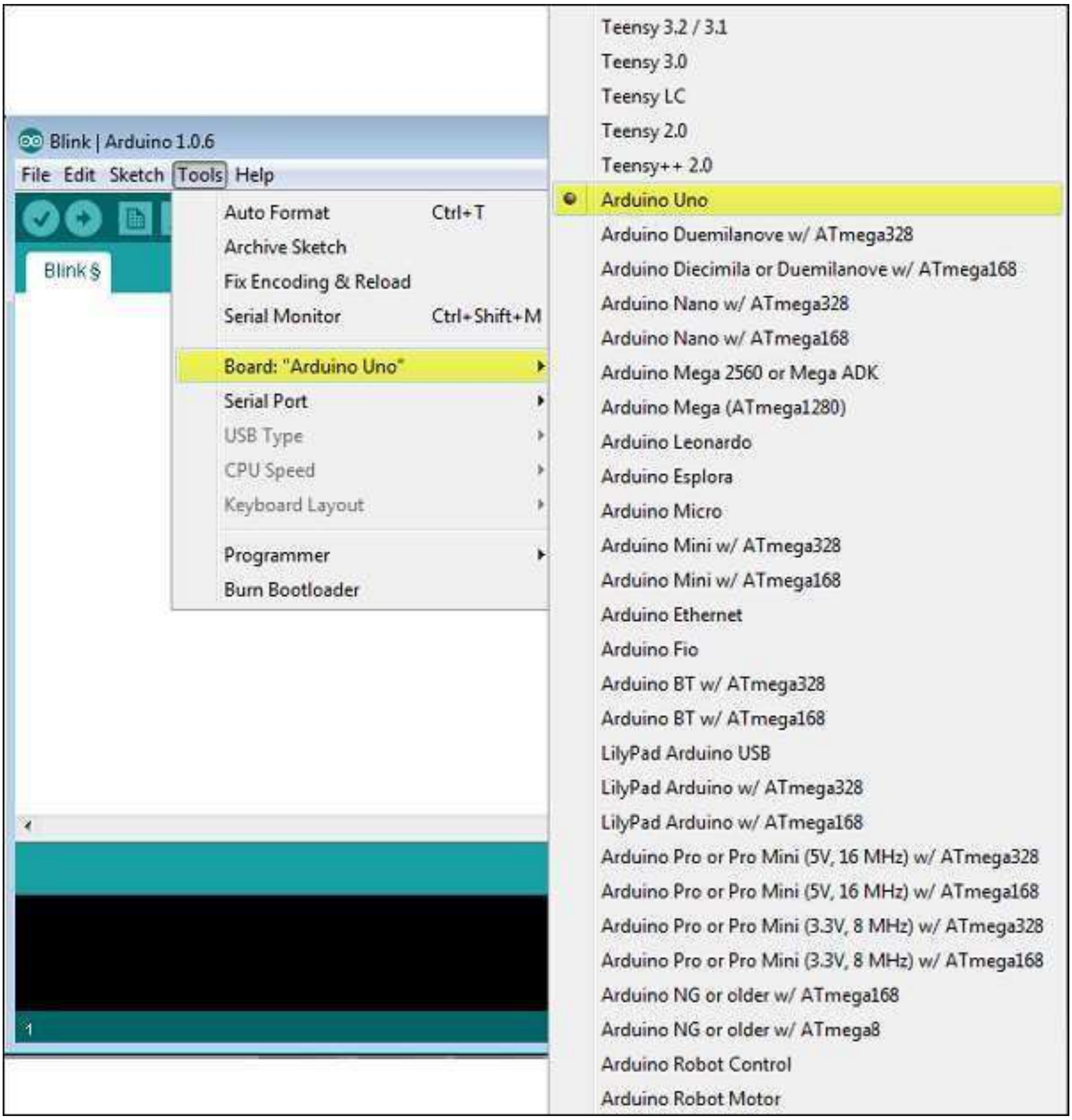


Fig.5.1.7 Select Your Arduino Board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

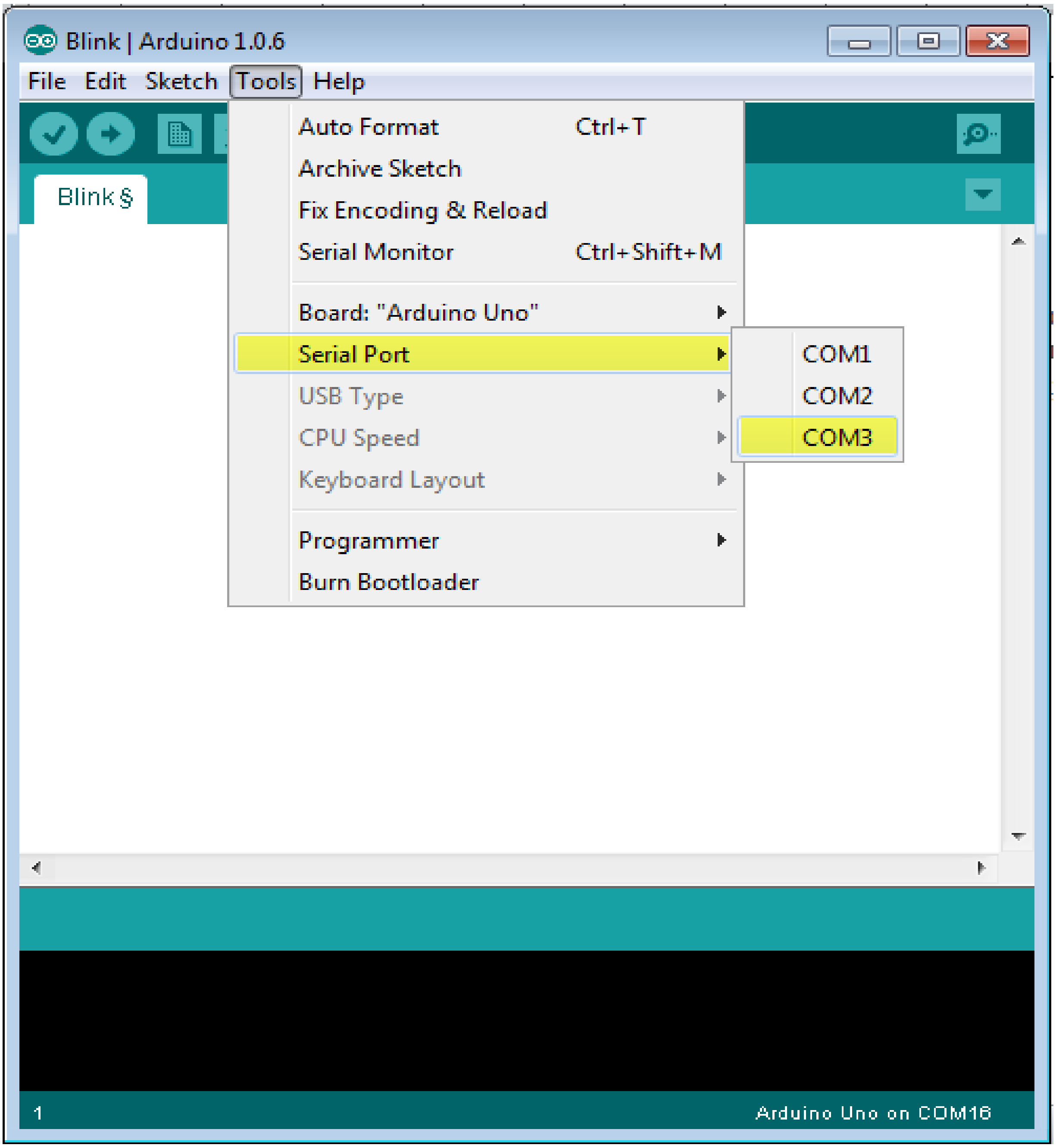


Fig.5.1.8 Select Your Serial Port

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

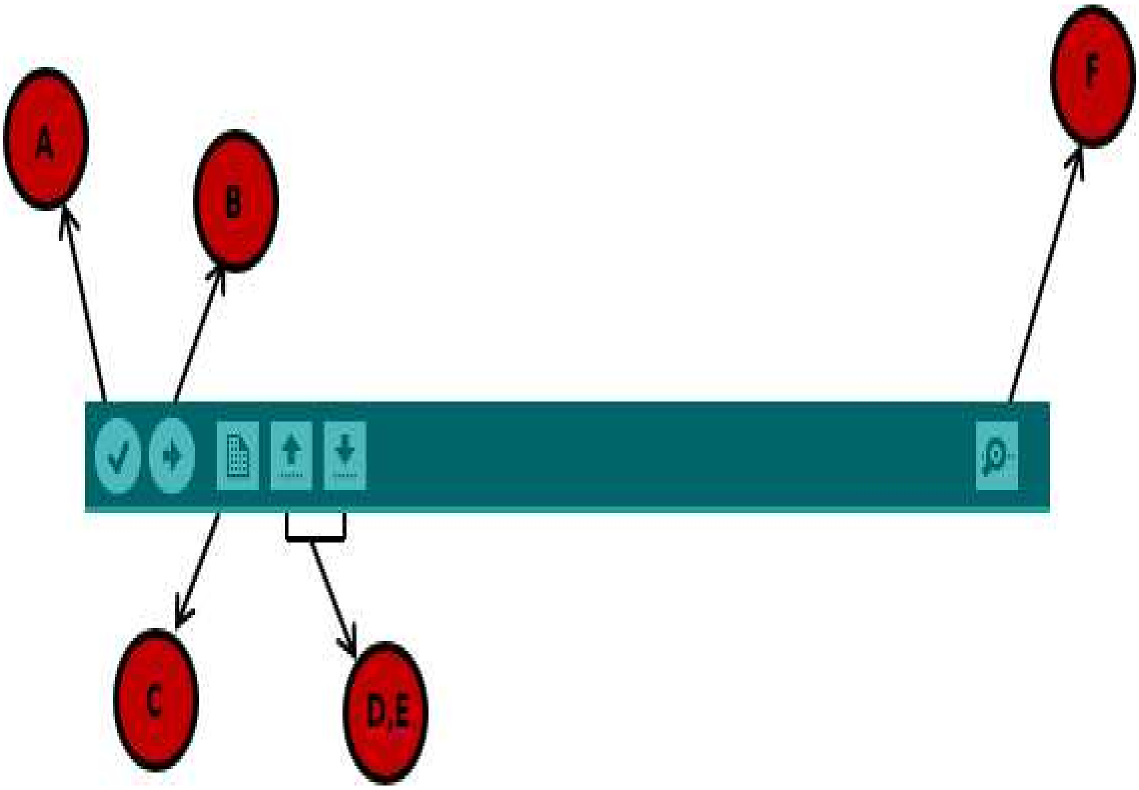


Fig.5.1.9 Upload the Program to Your Board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24 V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

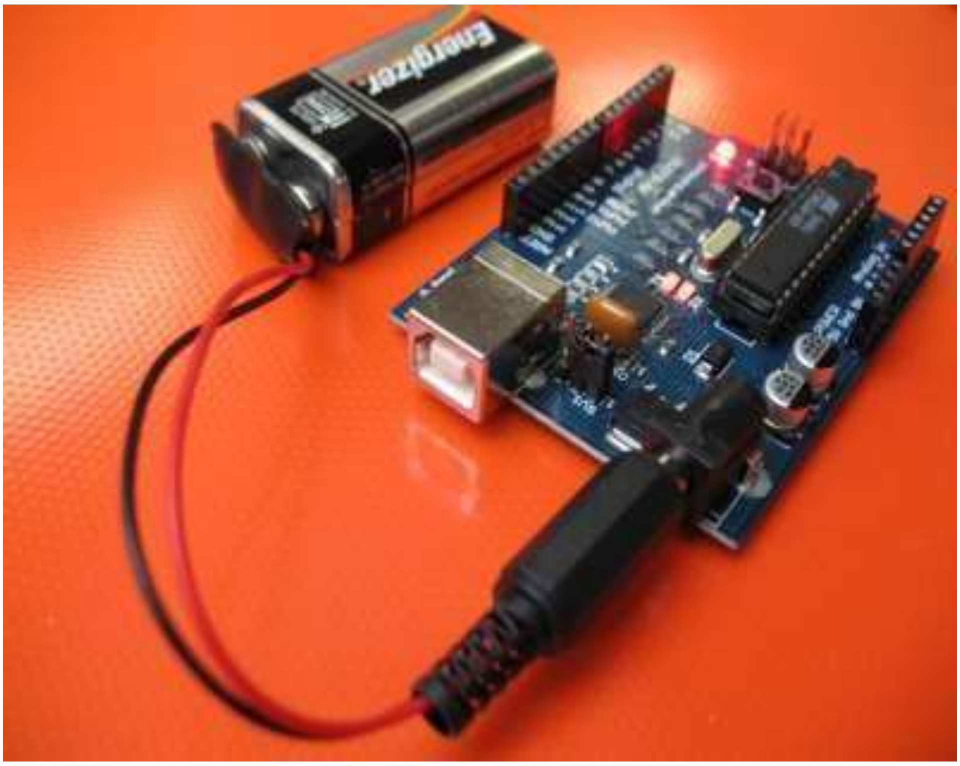


Fig.5.1.10 Connecting A battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run. Start the Arduino development environment. In Arduino-speak, programs are called "sketches", but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

Fig.5.1.11 Example Program

5.1.5 MC PROGRAMMING LANGUAGE: EMBEDDED C:

This is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements. Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

5.2 PROTEUS SOFTWARE:

PROTEUS:

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.

The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto manual routing options to the PCB Designer.

Starting New Design

Step 1: Open ISIS software and select New design in File menu

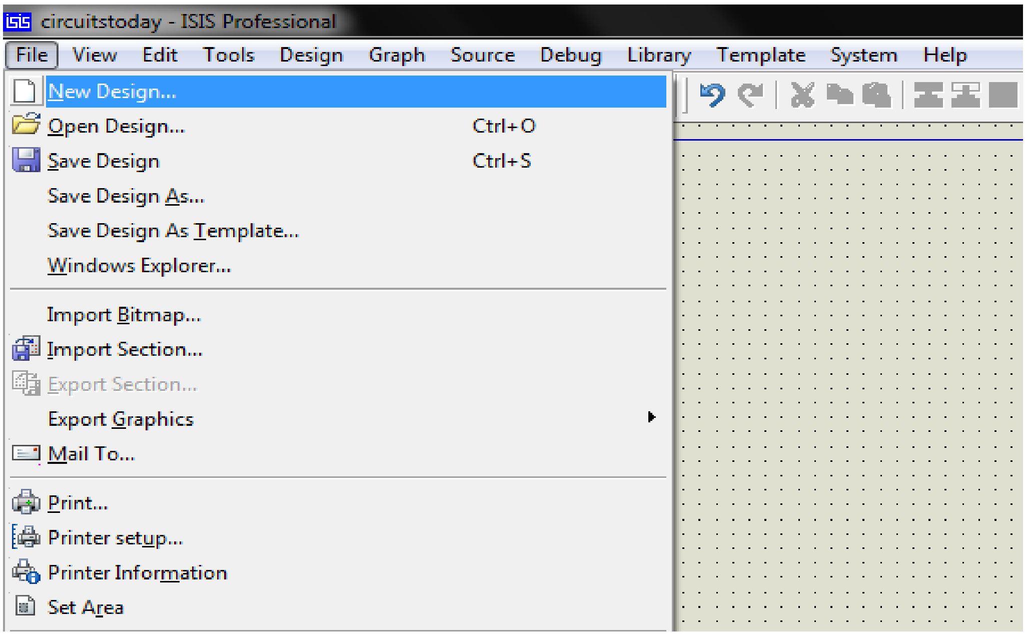


Fig.5.2.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

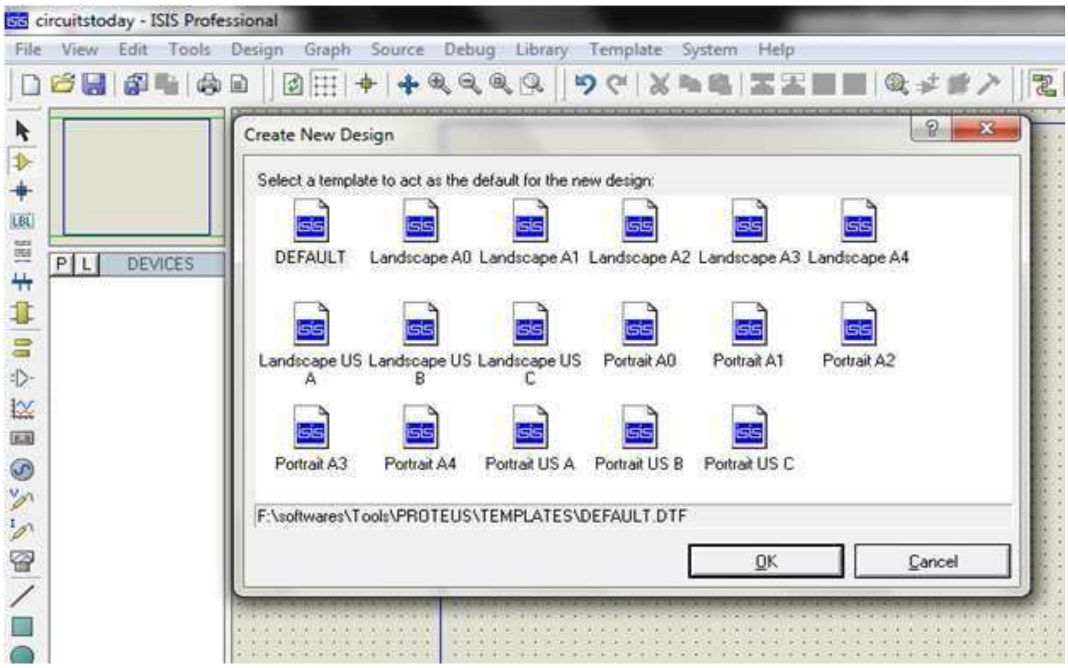


Fig.5.2.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

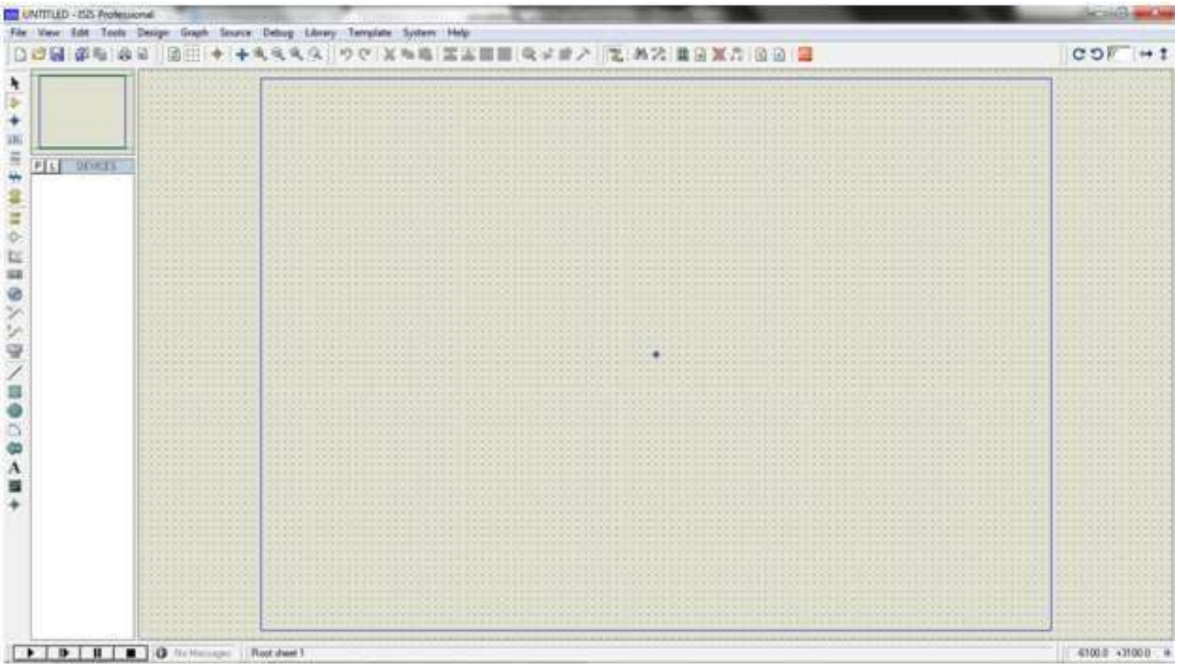


Fig.5.2.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

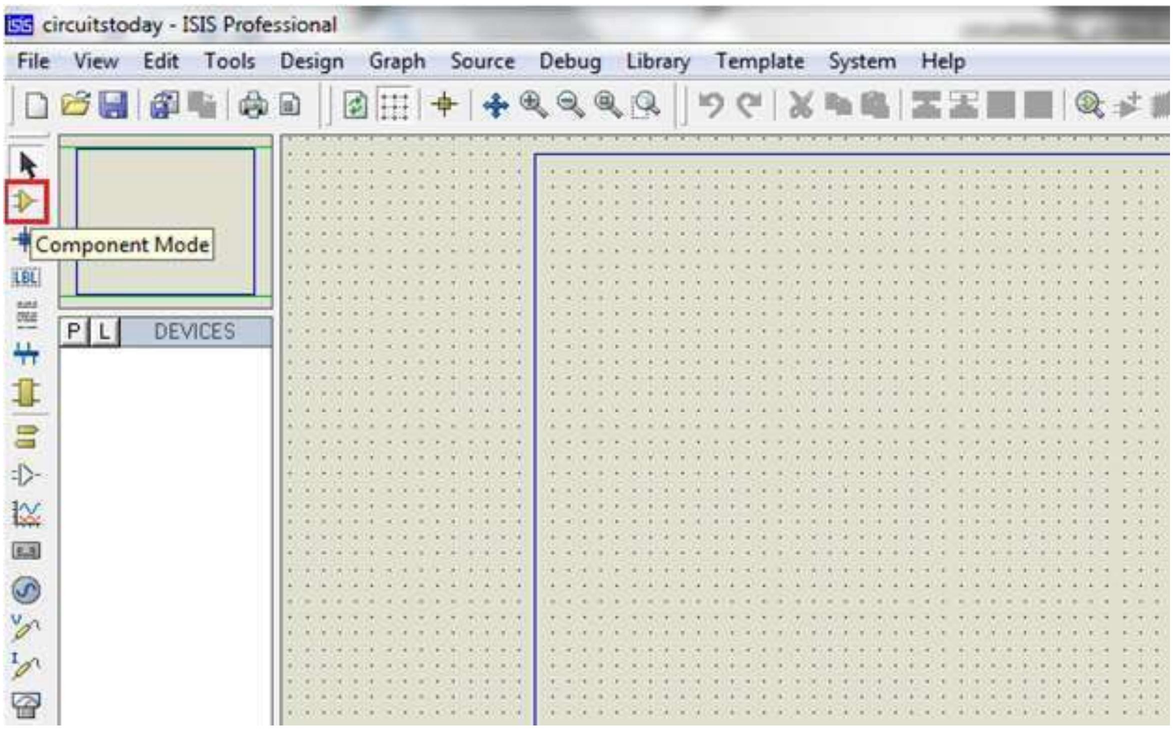


Fig.5.2.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

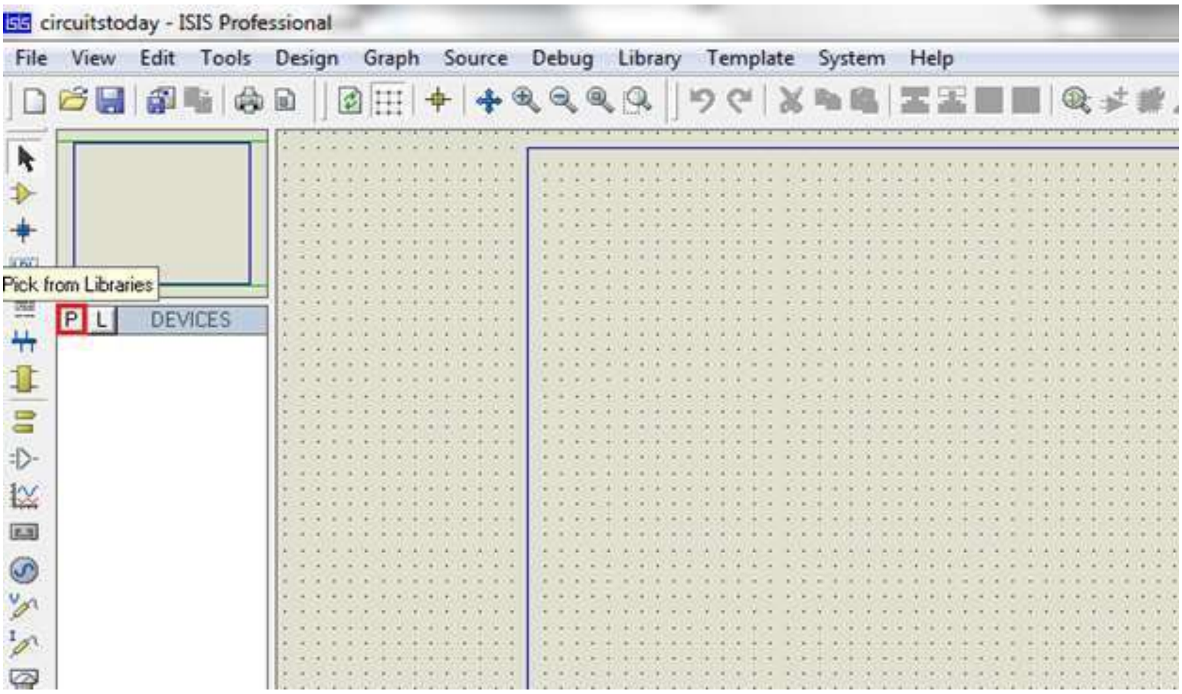


Fig.5.2.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

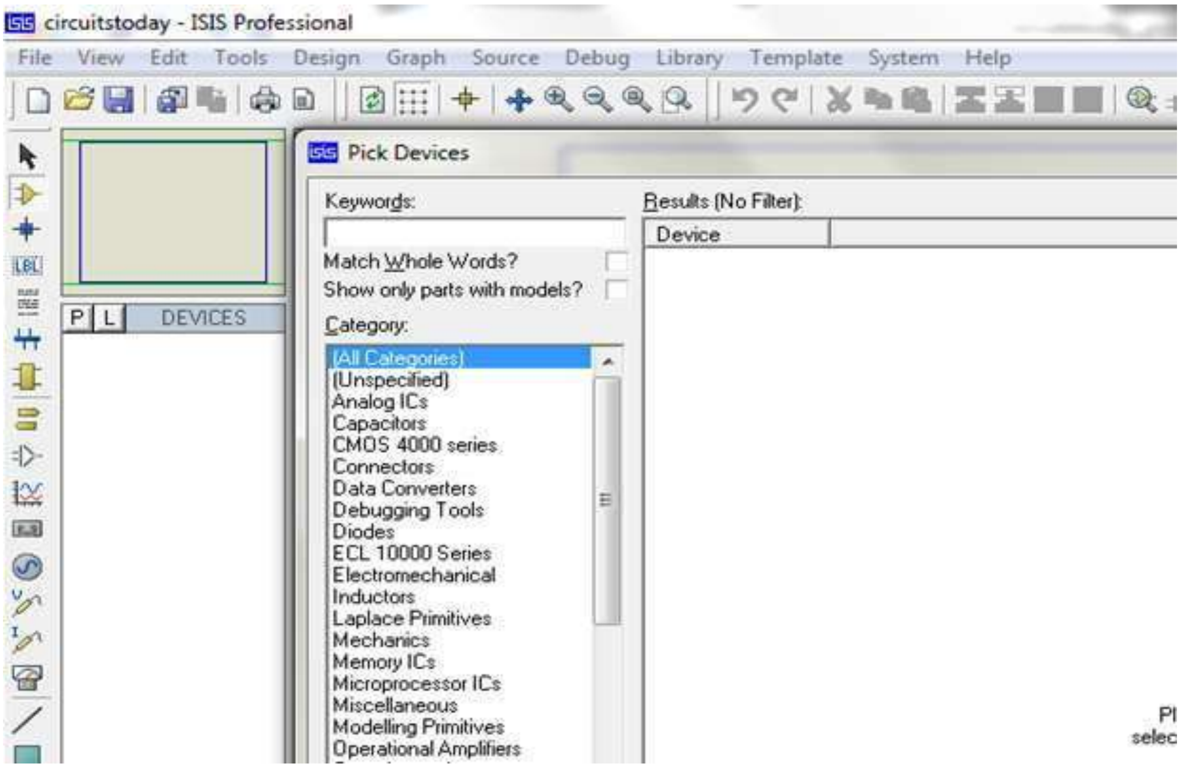


Fig.5.2.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

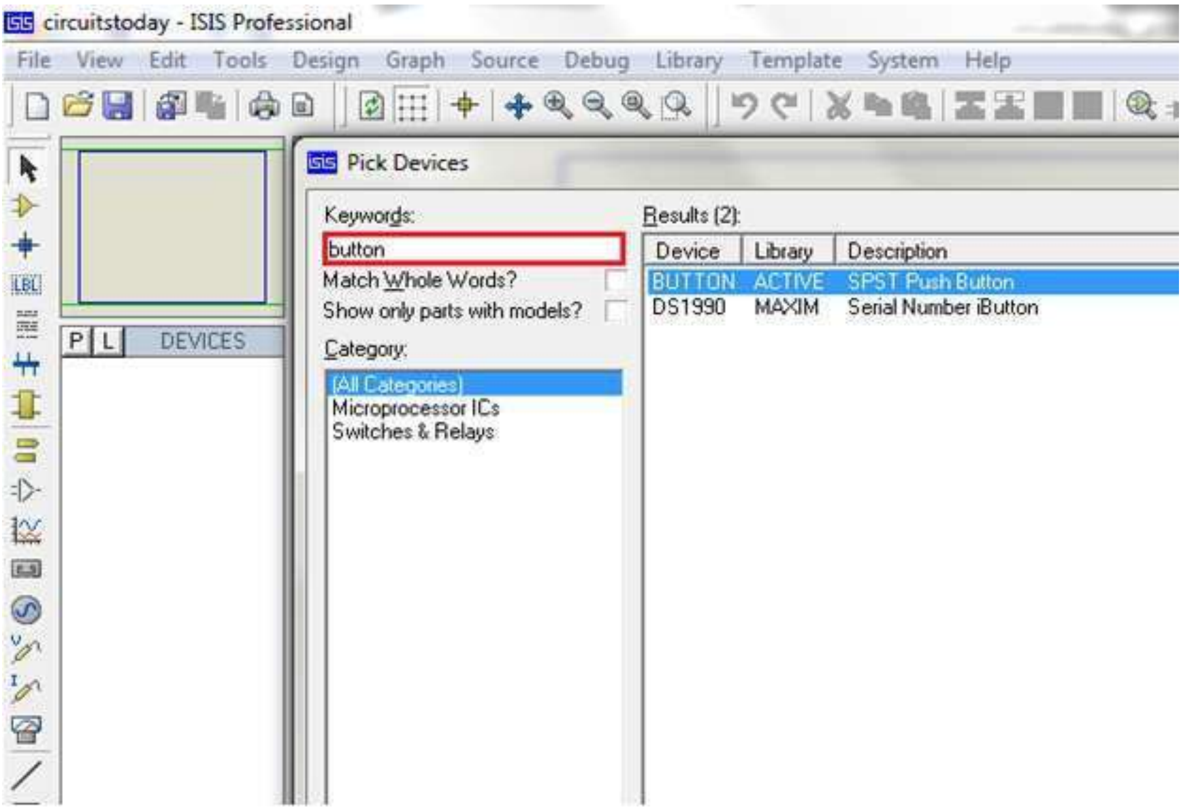


Fig.5.2.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

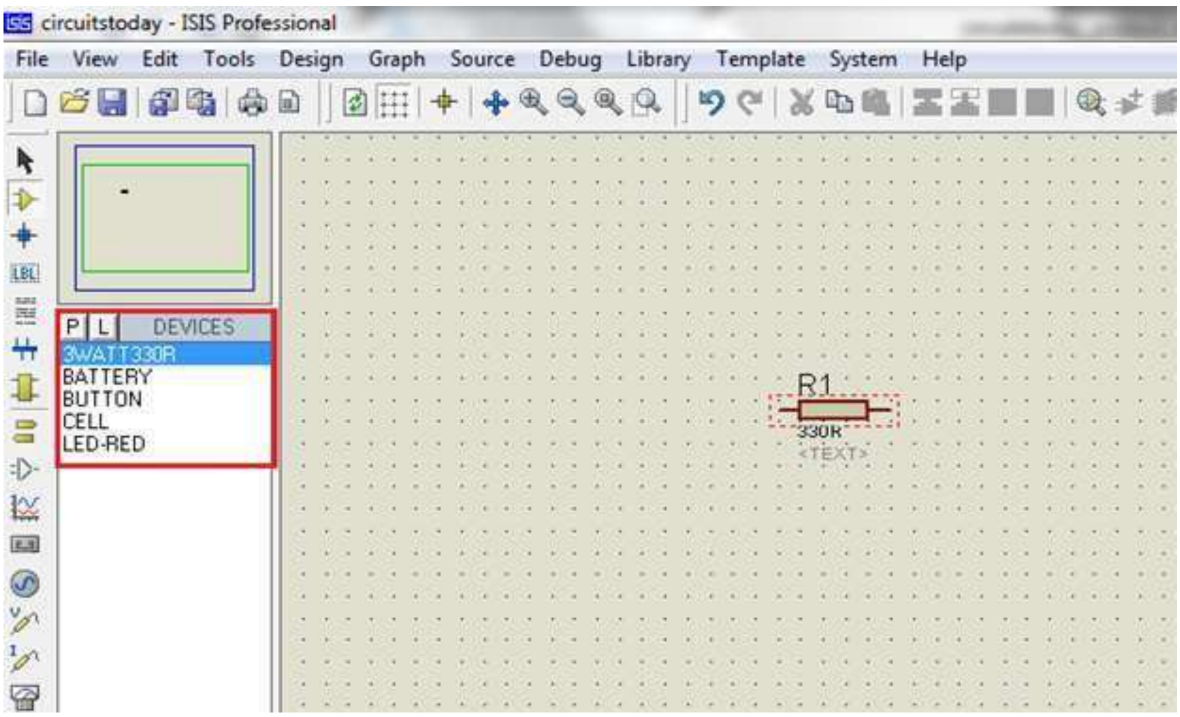


Fig.5.2.8 Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

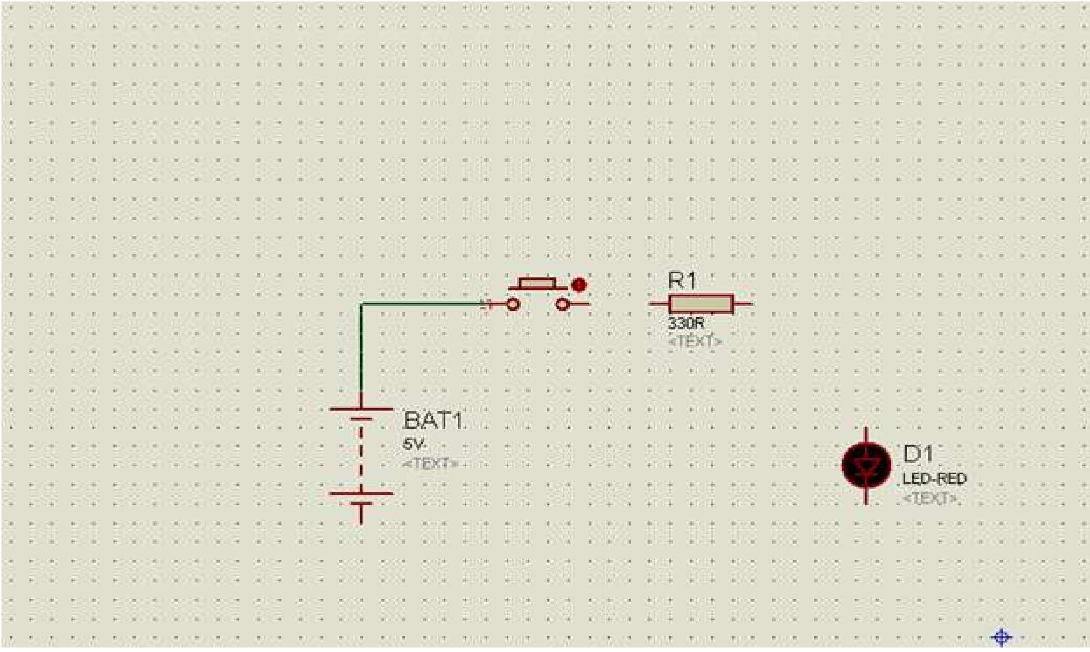


Fig.5.2.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

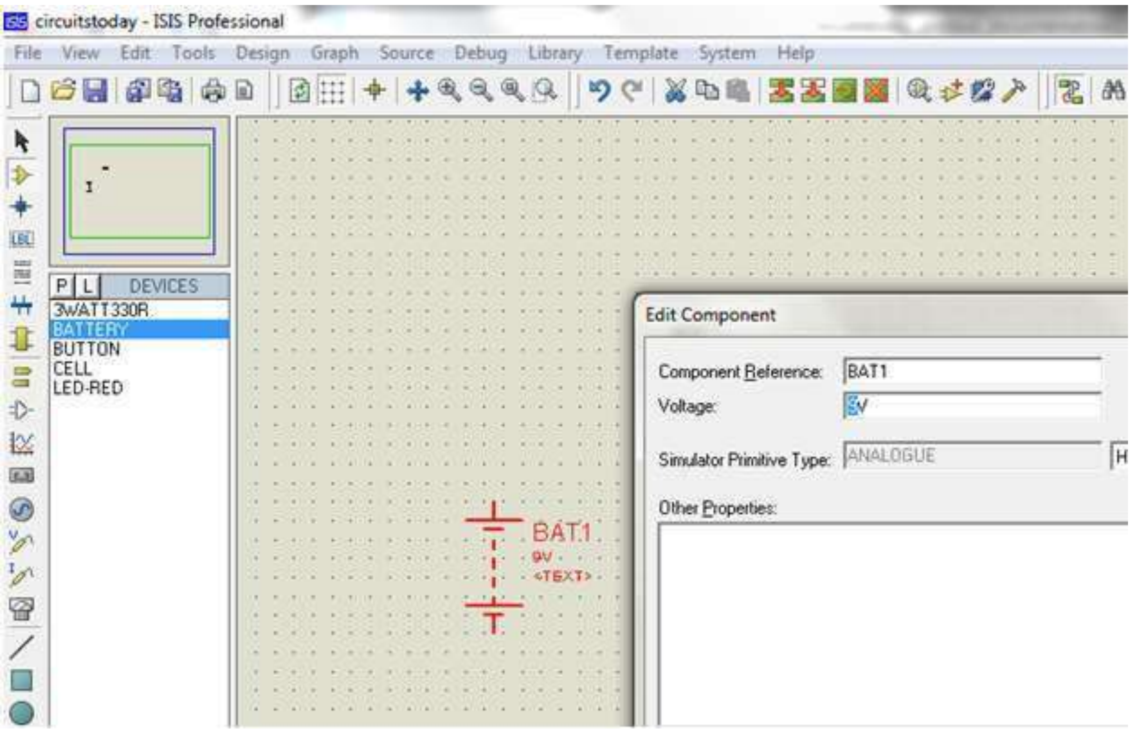


Fig.5.2.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

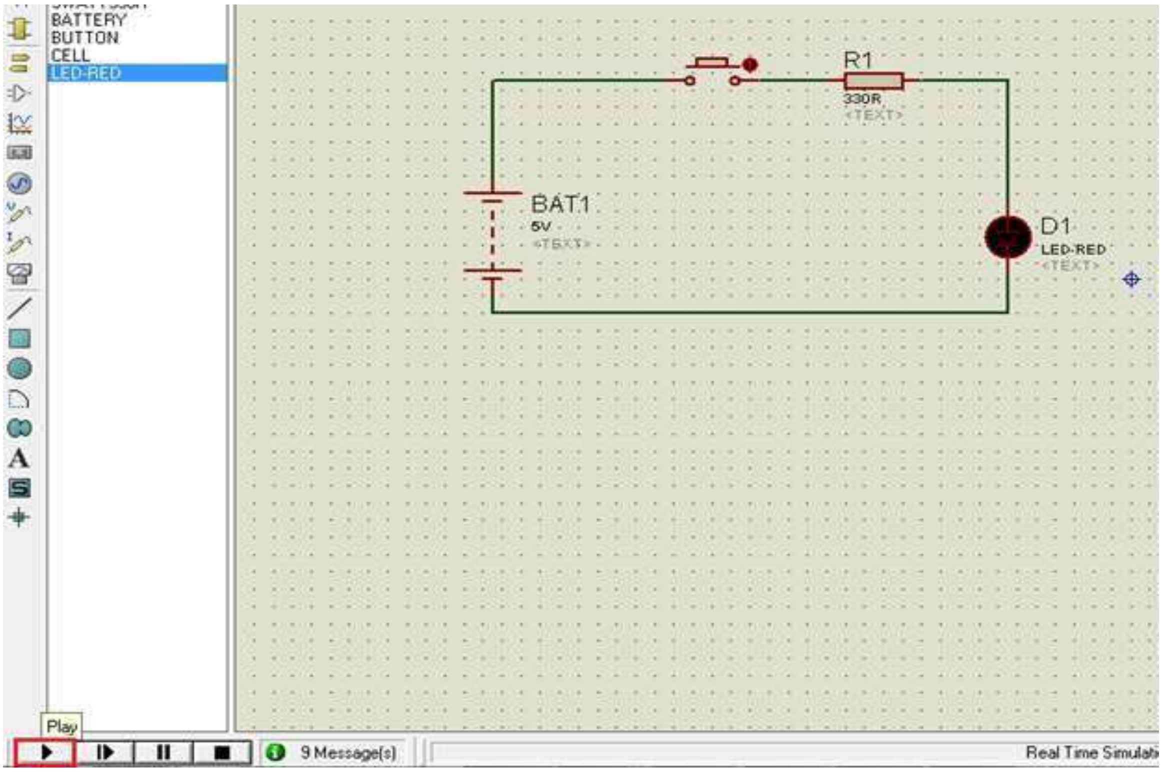


Fig.5.2.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

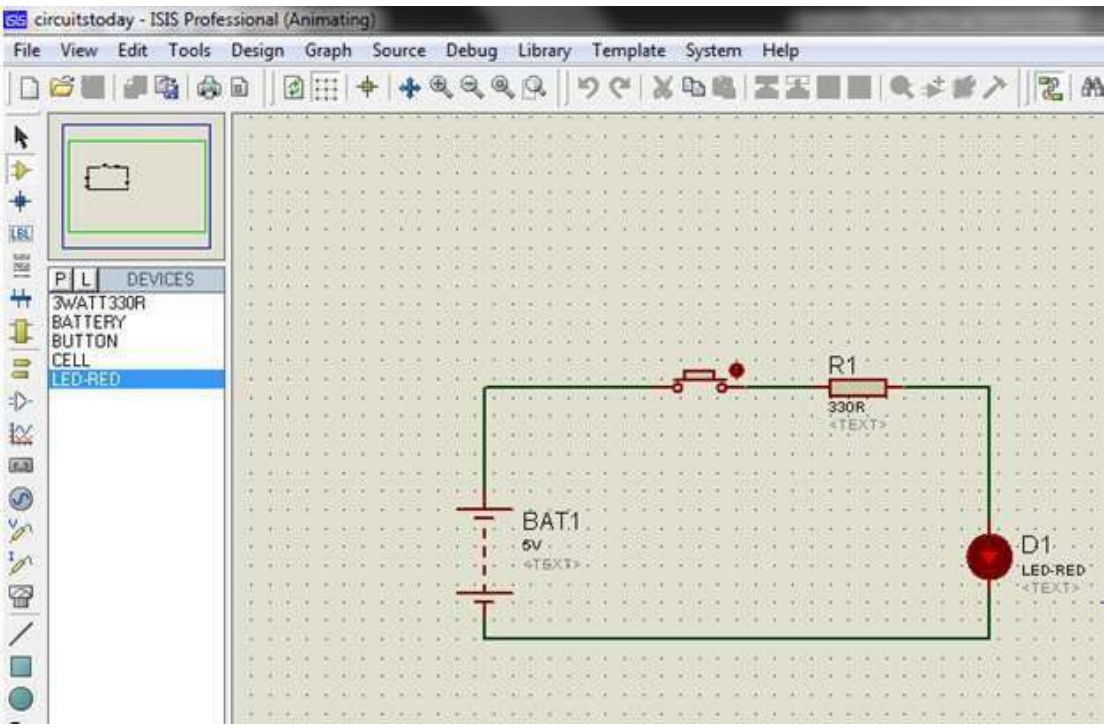


Fig.5.2.13 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

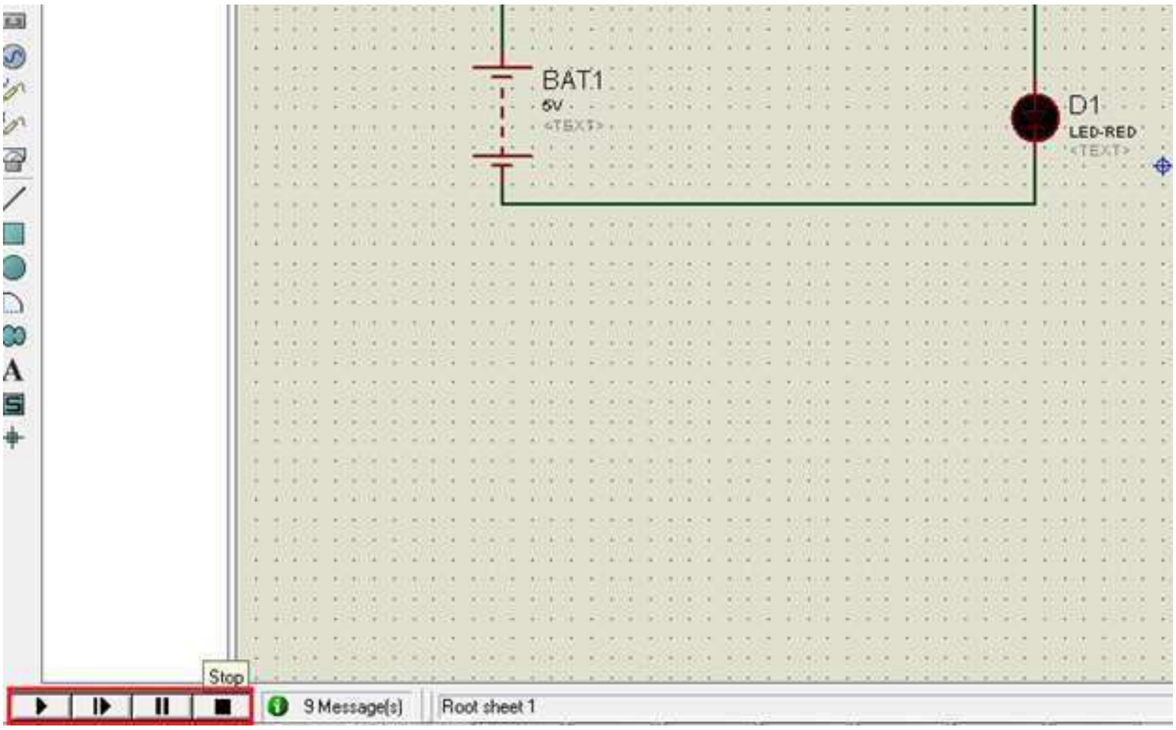


Fig.5.2.14 Simulation Step-Pause-Stop Buttons/

CHAPTER-6

PROJECT TESTING AND CODE IMPLEMENTATION

6.1 SOFTWARE TESTING:

Software Testing is a process of executing the application with an intent to find any software bugs. It is used to check whether the application met its expectations and all the functionalities of the application is working. The final goal of testing is to check whether the application is behaving in the way it is supposed to under specified conditions. All aspects of the code are examined to check the quality of application. The primary purpose of testing is to detect software failures so that defects may be uncovered and corrected. The test cases are designed in such way that scope of finding the bugs is maximum.

6.2 TESTING LEVELS:

There are various testing levels based on the specificity of test.

- **Unit testing:** Unit testing refers to tests conducted on a section of code in order to verify the functionality of that piece of code. This is done at the function level.
- **Integration Testing:** Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Its primary purpose is to expose the defects associated with the interfacing of modules.
- **System Testing:** System testing tests a completely integrated system to verify that the system meets its requirements.
- **Acceptance testing:** Acceptance testing tests the readiness of application, satisfying all requirements.
- **Performance testing:** Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or device such as response time or millions of instructions per second etc.

6.3 SYSTEM TEST CASES:

A test case is a set of test data, preconditions, expected results and post conditions, developed for a test scenario to verify compliance against a specific requirement. We have designed and executed a few test cases to check if the project meets the functional requirements.

6.4 CODE IMPLEMENTATION:

6.4.1 PROJECT CODE:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2,3);

unsigned int hbpin = 4;
int hcount = 0, hbval = 0;
 int temp = A4,sound = A3,water = A5;
char res[130];

void serialFlush(){
  while(Serial.available() > 0) {
  char t = Serial.read();
  }
}

void myserialFlush(){
  while(mySerial.available() > 0) {
  char t = mySerial.read();
  }
}

char check(char* ex,int timeout)
{
  int i=0;
  int j = 0,k=0;
  while (1)
  {
    sl:
    if(mySerial.available() > 0)
    {
      res[i] = mySerial.read();
```

```
      if(res[i] == 0x0a || res[i]=='>' || i == 100)
      {
        i++;
        res[i] = 0;break;
      }
      i++;
    }
    j++;
    if(j == 30000)
    {
      k++;
      //Serial.println("kk");
      j = 0;
    }
    if(k > timeout)
    {
     // Serial.println("timeout");
      return 1;
     }
  }//while 1

  if(!strncmp(ex,res,strlen(ex)))
  {
    //Serial.println("ok..");
    return 0;
   }
  else
  {
   // Serial.print("Wrong  ");
   // Serial.println(res);
    i=0;
    goto sl;
   }
}
```

```
char buff[200],k=0;
int phvalue=0;
void upload1();

const char* ssid = "project";
const char* password = "project1235";
int T;
int tt;
void setup() {
    int i=0;
  char ret;
    pinMode(temp,INPUT);
    pinMode(sound,INPUT);
    pinMode(water,INPUT);

  Serial.begin(9600);
  mySerial.begin(115200);
  lcd.begin(16,2);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("WELCOME");
  delay(3000);
//  serialFlush();
st:
  mySerial.println("ATE0");
  Serial.println("ATE0");
  ret  = check((char*)"OK",50);
  mySerial.println("AT");
  Serial.println("AT");
  ret  = check((char*)"OK",50);

  if(ret != 0)
  {
    delay(1000);
   goto st;
  }
```

```
  lcd.clear();lcd.setCursor(0, 0);
lcd.print("CONNECTING");
mySerial.println("AT+CWMODE=1");
Serial.println("AT+CWMODE=1");
   ret  = check((char*)"OK",50);
 cagain:

  myserialFlush();
  Serial.print("AT+CWJAP=\"");
  mySerial.print("AT+CWJAP=\"");
  mySerial.print(ssid);
  Serial.print(ssid);
  mySerial.print("\",\"");
  Serial.print("\",\"");
  mySerial.print(password);
  Serial.print(password);
  mySerial.println("\"");
  Serial.println("\"");
  if(check((char*)"OK",300))goto cagain;
  mySerial.println("AT+CIPMUX=1");
  Serial.println("AT+CIPMUX=1");
  delay(1000);

 lcd.clear();lcd.setCursor(0, 0);lcd.print("WAITING");
}
unsigned long int duration = 0;
int hbeat = 0;
void loop() {
  duration = pulseIn(hbpin, LOW, 5000000) / 1000;
  if (duration == 0)
    hbeat = 0;
  else
    hbeat = 64 + duration % 18;
  delay(1000);
```

```
 lcd.clear();
 lcd.setCursor(0, 1);
 hbval = hbeat;
 Serial.println(hbval);
 int td = analogRead(temp)/4;
 int sd = digitalRead(sound);
 int wd = digitalRead(water);

 Serial.print(td);
 Serial.print("\r\n");
 Serial.print(wd);
 Serial.print("\r\n");
 lcd.setCursor(0, 0);lcd.print("T:");lcd.print(td);lcd.print("    ");
 lcd.setCursor(5, 0);lcd.print("S:");lcd.print(sd);lcd.print("    ");
 lcd.setCursor(10, 0);lcd.print("W:");lcd.print(wd);lcd.print("    ");
lcd.setCursor(0, 1); lcd.print("HB:"); lcd.print(hbval); lcd.print(" ");
 delay(1000);

if(td > 35 )
{
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("HIGH TEMPRATURE");
 delay(3000);
 upload1(td,sd,wd,hbval);
 delay(500);
}

if(wd == 0 )
{
 lcd.clear();
 lcd.setCursor(0, 0);lcd.print("WATER detected");
 delay(3000);
upload1(td,sd,wd,hbval);
```

```
  delay(500);

}
/*if(sd == 0 )
{
  lcd.clear();
  lcd.setCursor(0, 0);lcd.print("WET DETECTED");
  delay(3000);
 upload1(td,sd,wd,phvalue);
  delay(500);

}*/
if(sd == 0 )
{
  lcd.clear();
  lcd.setCursor(0, 0);lcd.print("CRY DETECTED");
  delay(3000);

upload1(td,sd,wd,hbval);
  delay(500);

}
else
{

//upload1(td,sd,wd,phvalue);

}

if (hbval > 78 )
    {
      lcd.setCursor(0, 1); lcd.print("HIGH HEARTBEAT  ");

     //sendmsg(number,"HIGH HEART RATE:");
```

```
     delay(200);
//      sendmsg1(number1,"HIGH HEART RATE:");
   // delay(3000);
upload1(td,sd,wd,hbval);
delay(500);
    }



if (hbval>10 &&hbval < 50 )
    {
      lcd.setCursor(0, 1); lcd.print("LOW HEARTBEAT  ");



      upload1(td,sd,wd,hbval);


       delay(3000);


    }

}//loop



char bf2[100];
void upload1(unsigned char *chr ,unsigned char *chr1,unsigned char *chr2,unsigned char *chr3)
{
  delay(2000);
  lcd.clear();lcd.setCursor(0, 1);lcd.print("UPLOADING");
  myserialFlush();
  mySerial.println("AT+CIPSTART=4,\"TCP\",\"api.thingspeak.com\",80");
 // Serial.println("AT+CIPSTART=4,\"TCP\",\"embeddedspot.top\",80");
      delay(8000);
      sprintf(buff,"GET
https://api.thingspeak.com/update?api_key=ZHR6DIPXM6G1SBYN&field1=0=%u&field2=
%u\r\n\r\n",chr,chr1,chr2,chr3);
```

```
    //sprintf(buff,"GET
http://embeddedspot.top/iot/storedata.php?name=sensors010&s1=%u\r\n\r\n",chr);
    myserialFlush();
    sprintf(bf2,"AT+CIPSEND=4,%u",strlen(buff));
    mySerial.println(bf2);

      delay(5000);



       myserialFlush();
       mySerial.print(buff);
       Serial.print(buff);
          delay(2000);

          mySerial.println("AT+CIPCLOSE");
          Serial.println("AT+CIPCLOSE");
          lcd.setCursor(0, 1);lcd.print("UPLOADDED");  lcd.clear();

}
```

CHAPTER-7

PROJECT IMPLEMENTATION

7.1 BLOCK DIAGRAM OF PROJECT:

The system consists of an Arduino UNO microcontroller that is inbuilt with a Wi-Fi module. The UNO is a microcontroller based on ESP8266.It has 17 GPIO PINS(0-16) out of which 6 pins (GPIO 6-11) are connected to the flash memory chip .It has both input/output digital pins ,a single analog inputs, inbuilt Wi-Fi module and a USB connection . It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The micro-controller is attached with a temperature and humidity sensor (DHT11), a sound sensor, APR module, BJT and a servo motor. The sound sensor is used in order to detect the sound of the baby that is mainly when the baby cries it detects the sound and notifies the MCU. The MCU in turn notifies the Servo motor that makes the cradle swing. When a baby cries in the cradle, the microphone detects it and converts the sound signal into an electrical signal.

The electrical signal is then fed into an amplifier. The microcontroller then notifies the blynk server which in turn notifies the parent that the baby is crying. At the same time the microcontroller also notifies the parent through Wi-Fi module. If the baby relaxes and stops crying the parent is again notified that the baby is fine and stopped crying. The cradle stops swinging after a certain time. The temperature and humidity sensor consists of a material called thermistor and an IC on the back side of the sensor. For measuring humidity they use the humidity sensing component which has two electrodes with moisture holding substrate between them. So as long as the humidity change, the conductivity of the substrate changes or the resistance between these electrodes changes. On the other hand, for measuring temperature this sensors use a thermistor. A thermistor is actually a variable resistor that changes its resistance with the change of the temperature.

These sensors are made by sintering of semi conductive materials such as ceramics or polymers in order to provide larger changes in the resistance with just small changes in temperature. The resistance decreases with increase in the temperature. These changes in resistance is measured and processed by the IC which makes it ready to be read by a microcontroller. The microcontroller then notifies the blynk server which in turn notifies the parent about the temperature and wetness change. The microcontroller also notifies the BJT that in turn rotates the fan when the parent allows it to do so. The fan would also rotate incase the room temperature is high automatically.

The blynk app notifies every situation to the parent. Customization in blynk app can be easily done. The feeding time of the baby is accordingly preset and notified.

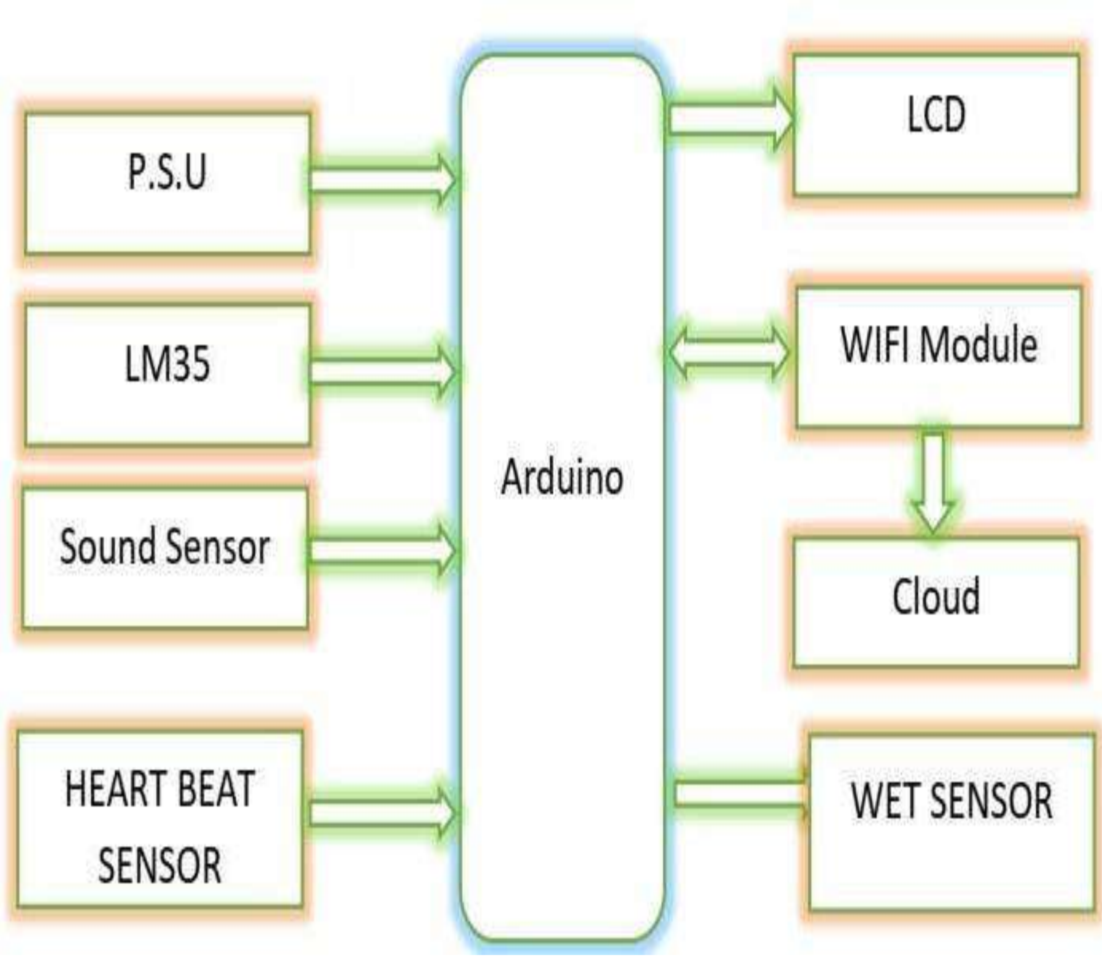


Fig 7.1 Block Diagram Of Baby Cradle And Health Monitoring System Using IOT

7.2 FLOW CHART:

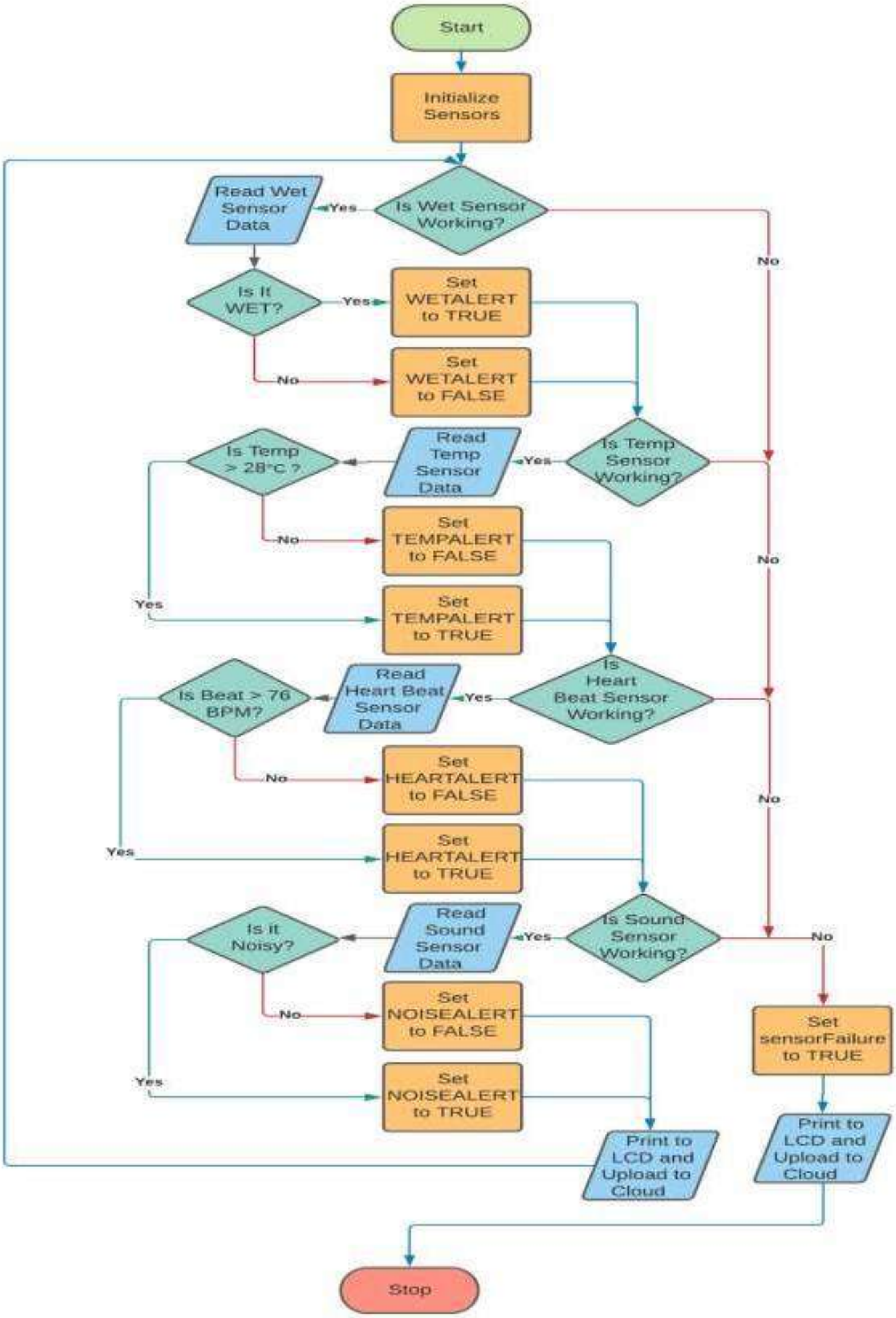


Fig.7.2 Flow Chart

7.3 RESULTS:

In this section, the results, discussion, and validation of IoT-BBMS are presented in detail. Figure 7.3.1 shows the final prototype of the developed smart cradle. Several manufacturing steps were carried out prior to the implementation of the control system for the smart cradle. The data measured by the utilized sensors, namely, sound sensor, heart beat sensor temperature, and wet sensor are updated to the Internet and can be accessed via the Adafruit MQTT server and MQTT Dash mobile applications. A mobile MQTT application, which is compatible with the Android operating system mobile phone, can be downloaded from the Play Store. The application's name is ''MQTT Dash,'' which can be synchronized with the Adafruit MQTT server. After some optimization and testing, we select this application for developing our dashboard due to its simplicity and user friendliness in terms of the view and usage.

During the synchronization process, the targeted MQTT server address is required, followed by the host port, the username, and the specified authentication token provided by the server. Figure 10 shows the requirements of the synchronization process for the mobile apps to share the same data received from the sensors. The password represents the authentication token (AIO key) generated by the MQTT server. As shown in Figures 7.3.2, the sensors data is uploading to server and MQTT Dash apps are synchronized and displayed the same readings uploaded by the NodeMCU microcontroller of the baby monitoring system. The smart cradle can swing automatically whenever sound is detected by the sound sensor. A notification, shown in Figure 7.3.2 is sent to the user through IFTTT mobile application to notify the user that crying is detected on the baby monitoring system. The user can also remotely control the cradle to swing manually by toggling the switch in the MQTT server or mobile apps. The mini fan is designed to turn on automatically whenever the room's temperature, measured by the temperature sensor, is higher than 28 ◦C. The user can also remotely switch the mini fan ON and OFF through the MQTT server and mobile application. A musical toy, which can be controlled by the user, is installed to entertain the baby. For visual monitoring of the baby's situation.

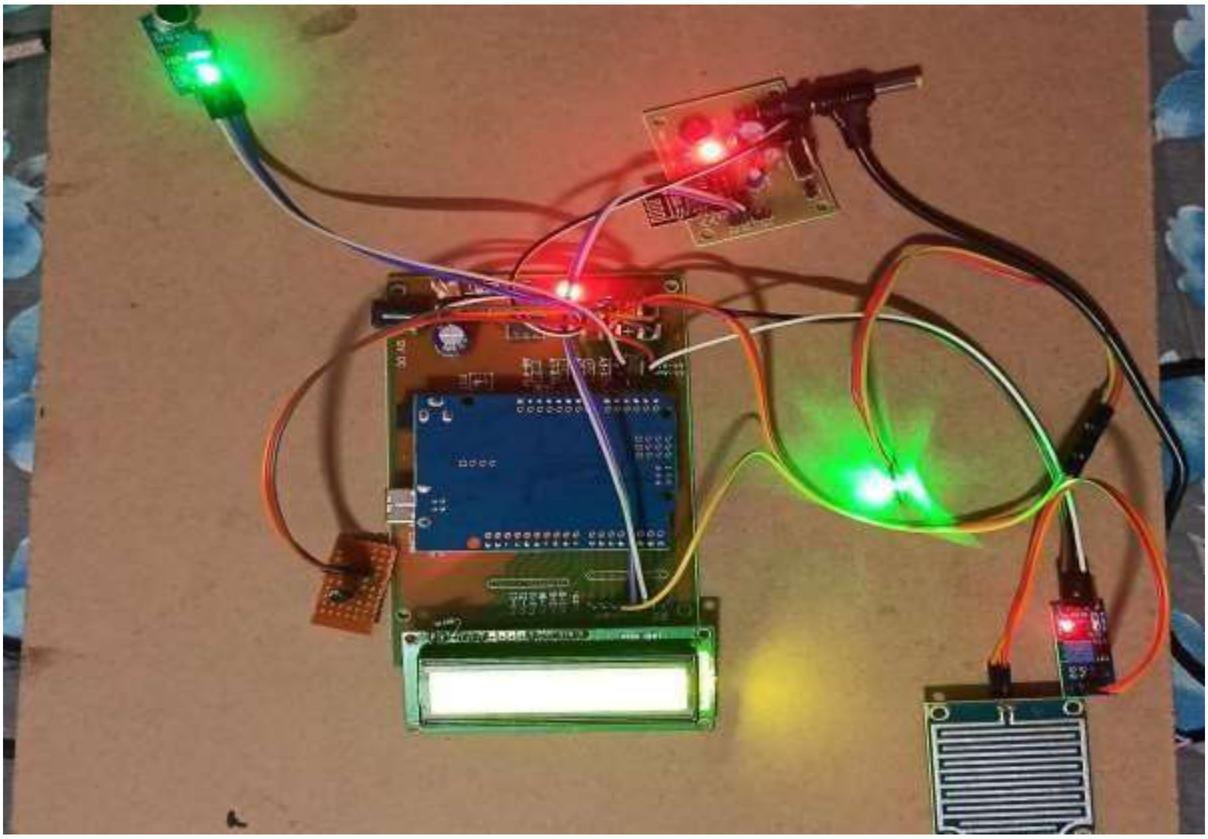


Fig 7.3.1 Result Of Working Model

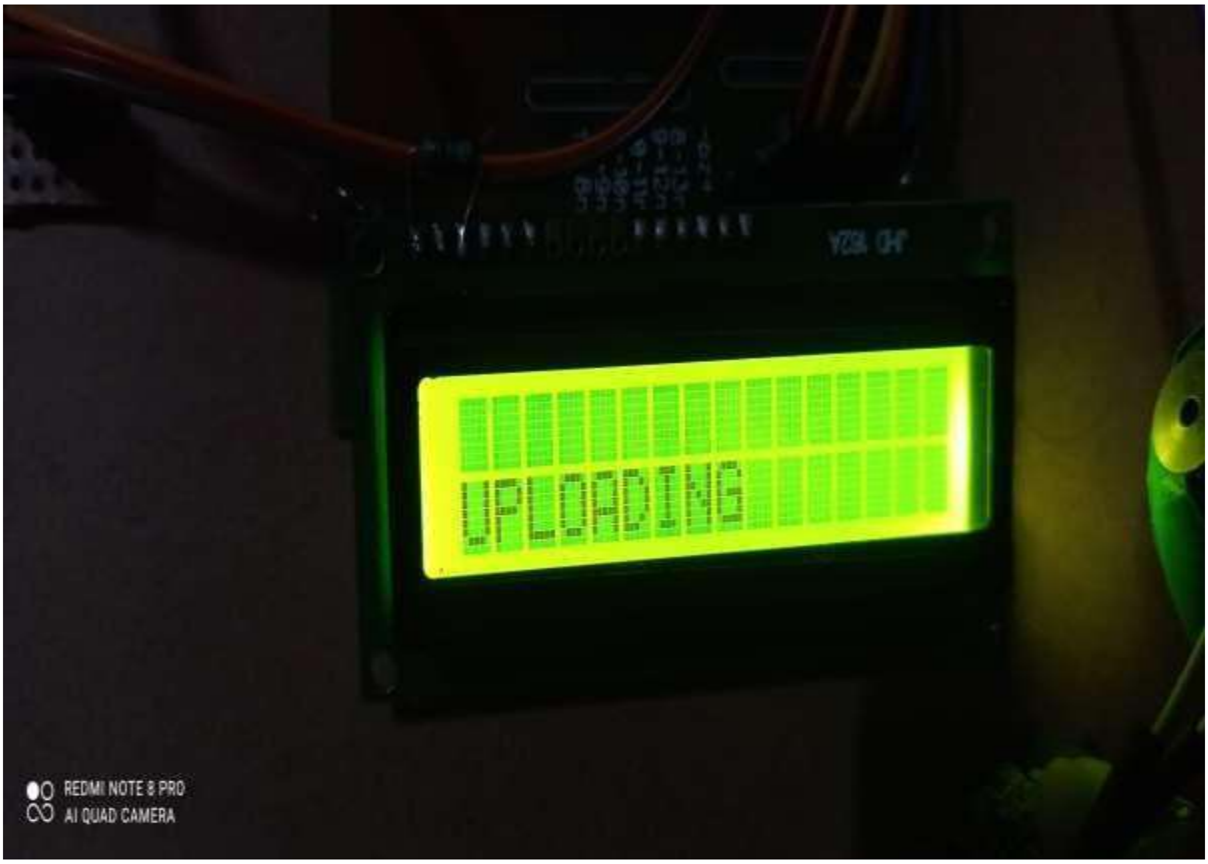


Fig.7.3.2 Uploading To Server

- The following figure 7.3.3, it shows the output result of temperature sensor when temperature is detected.

Fig.7.3.3 Temperature Sensor Output

- The following figure 7.3.4, it shows the output result of the wet sensor when wet is detected.

Fig.7.3.4 Wet Sensor Output

- The following figure 7.3.6, it shows the output result of sound sensor when sound is detected.

Fig.7.3.5 Sound Sensor Output

- The following figure 7.3.6, it shows the output of heart beat sensor when the heart rate is detected.

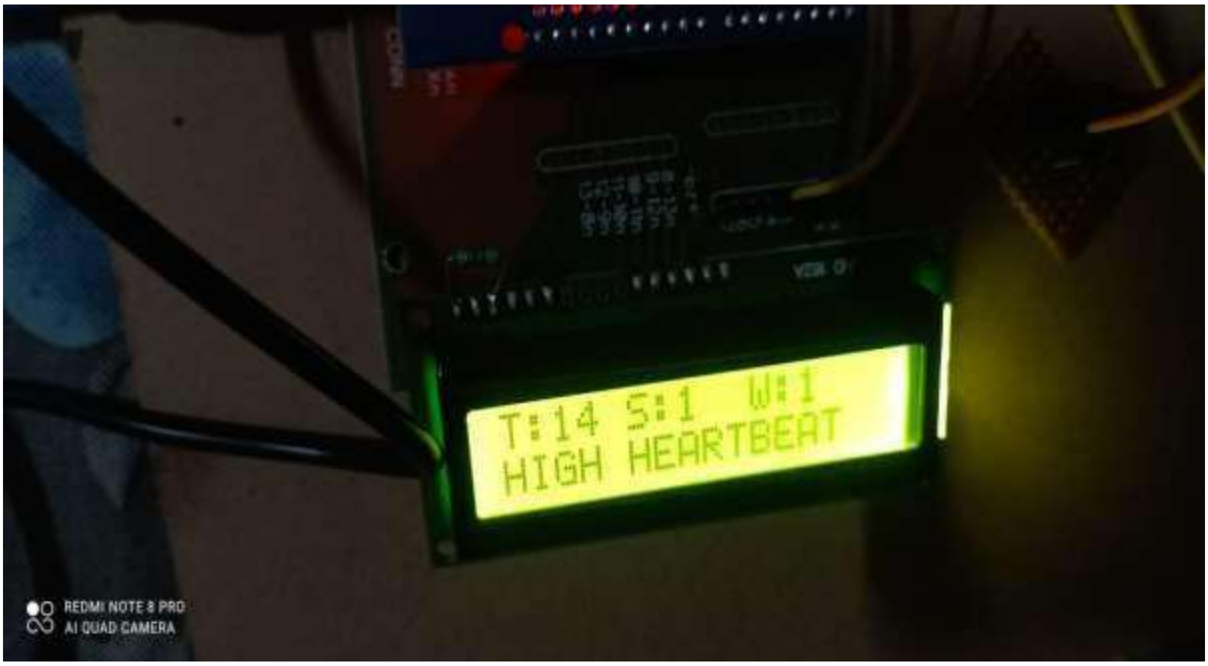


Fig.7.3.6 Heart Beat Sensor Output

- The following figure 7.3.7 shows, it is the channel statistics of the sensors with respective time interval.

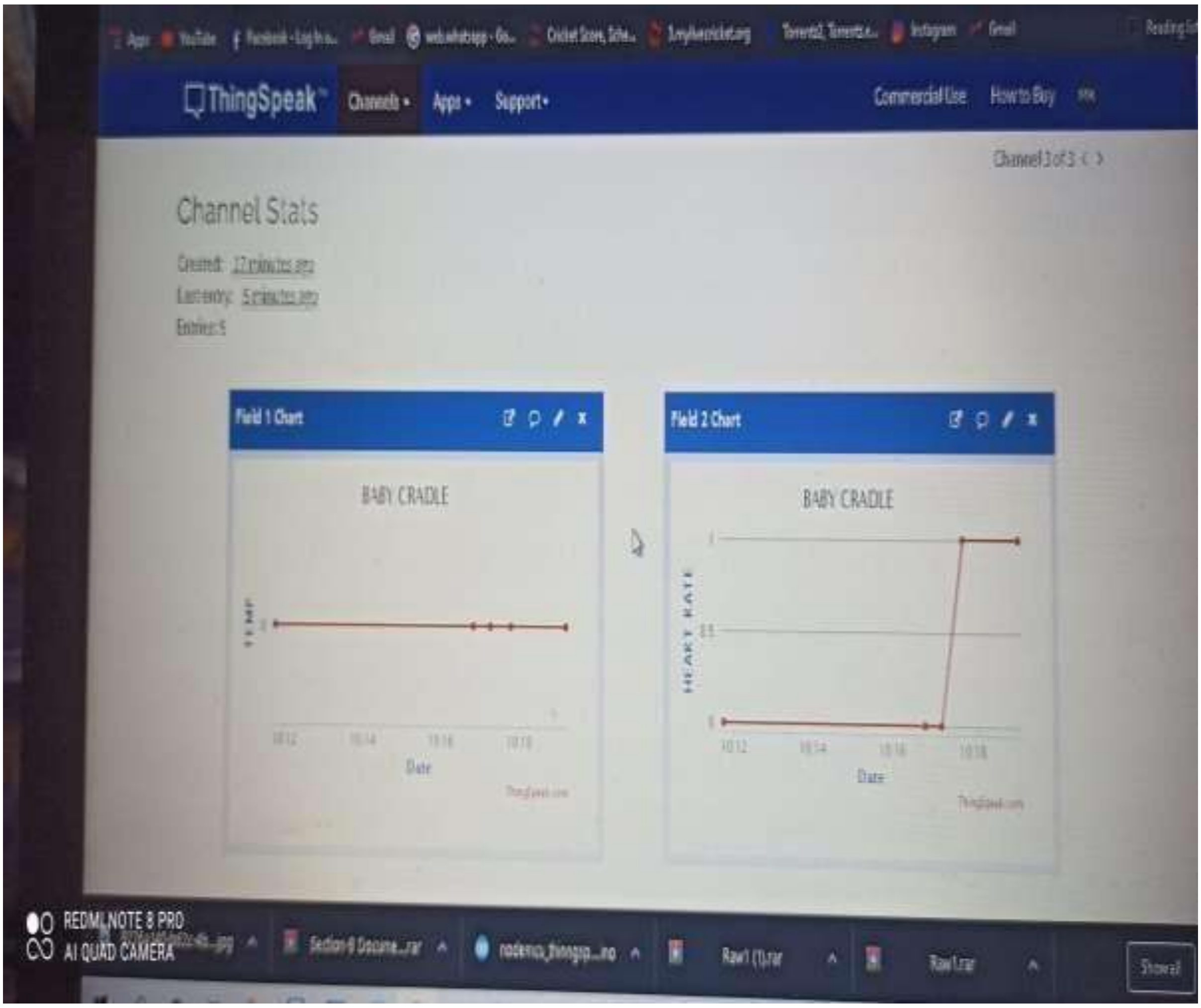


Fig. 7.3.7 Channel statistics

CHAPTER-8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 ADVANTAGES:

- Cost efficient
- User friendly
- Ensure safety
- Minimum manual work
- Baby sooth comfortably
- Baby stays healthy

8.2 APPLICATIONS:

- Provide a smart solution for child care industry.
- Help working women balance their work and domestic chores.
- Can be used in day care centers for the aid of human beings.
- Can be utilized in nurses in maternity units of hospital.

8.3 CONCLUSION:

The integration of various sensors with Arduino will provide a better way for the monitoring of the baby. Recent advances in sensor technologies and wireless communication technologies enable the creation of a new generation of healthcare monitoring systems with wearable electronics and photonics. Parents in the present world are busy in their professional life, so they do not get sufficient time to take care of their babies. It may be expensive for the household to afford a nanny. The constant recording of multiple biological parameters of a baby and analysis of overall health helps a mother to understand the overall health condition of the baby, and can take actions for improvement, if necessary. Cradle temperature and wetness is also measured. This is an efficient system for monitoring the baby's health condition from any distance.

8.4 FUTURE ENHANCEMENT:

To enhance the security of the baby apart from the basic requirement more modules can be added like PIR sensor to detect the motion, camera to see the surroundings or the person who has been around the baby. The mobile app we used for the prototype model is the built in app for Bluetooth module. Depending upon the requirement, android or iOS app can be made which increases the scope of project in app development framework.

PUBLICATION

Submitted Paper in the Conference ICSMEC-21 with paper ID (ICSMEC21-0034) and got Acceptance for the Paper.

REFERENCES

[1] S. Brangui, M. El Kihal, and Y. Salih-Alj, ‘‘An enhanced noise cancelling system for a comprehensive monitoring and control of baby environments,’’ in *Proc. Int. Conf. Elect. Inf. Technol. (ICEIT)*, 2015, pp. 404–409.

[2] M. Goyal and D. Kumar, ‘‘Automatic E-baby cradle swing based on baby cry,’’ *Int. J. Comput. Appl.*, vol. 975, p. 8887, Jan. 2013.

[3] R. Palaskar, S. Pandey, A. Telang, A. Wagh, and R. M. Kagalkar, ‘‘An automatic monitoring and swing the baby cradle for infant care,’’ *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 12, pp. 187–189, 2015.

[4] C.-T. Chao, C.-W. Wang, J.-S. Chiou, and C.-J. Wang, ‘‘An Arduino-based resonant cradle design with infant cries recognition,’’ *Sensors*, vol. 15,no. 8, pp. 18934–18949, 2015.

[5] A. F. Symon, N. Hassan, H. Rashid, I. U. Ahmed, and S. M. T. Reza, ‘‘Design and development of a smart baby monitoring system based on Raspberry Pi and Pi camera,’’ in *Proc. 4th Int. Conf. Adv. Elect. Eng.(ICAEE)*, 2017, pp. 117–122.

[6] A. Kaur and A. Jasuja, ‘‘Health monitoring based on IoT using Raspberry PI,’’ in *Proc. Int. Conf. Comput., Commun. Autom. (ICCCA)*, May 2017, pp. 1335–1340.

[7] S. P. Patil and M. R. Mhetre, ‘‘Intelligent baby monitoring system,’’ *ITSI Trans. Elect. Electron. Eng.*, vol. 2, no. 1, pp. 11–16, 2014.

[8] E. Saadatian, S. P. Iyer, C. Lihui, O. N. N. Fernando, N. Hideaki,A. D. Cheok, A. P. Madurapperuma, G. Ponnampalam, and Z. Amin, ‘‘Low cost infant monitoring and communication system,’’ in *Proc. IEEE Colloq. Humanities, Sci. Eng.*, Dec. 2011, pp. 503–508.

[9] D. N. F. M. Ishak, M. M. A. Jamil, and R. Ambar, ‘‘Arduino based infant monitoring system,’’ in *Proc. IOP Conf. Ser., Mater. Sci. Eng.*, 2017,vol. 226, no. 1, Art. no. 012095.

[10] J.-R. C. Chien, ‘‘Design of a home care instrument based on embedded system,’’ in *Proc. IEEE Int. Conf. Ind. Technol.*, Apr. 2008, pp. 1–6.

BIBILOGRAPHY

[1]. Mohit Kumar and Mrs. Suryakala,“Temperature control and monitoring of incubators using IoT”, International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume 22 Issue 1 – MAY 2016.

[2]. Mrudula Borkar, Neha Kenkre and Harshada Patke, “An Innovative approach for Infant Monitoring System using Pulse Rate and Oxygen level”, International Journal of Computer Applications (0975 – 8887) Volume 160 – No 5, February 2017.

[3]. Suruthi and S. Suma, “Microcontroller based baby incubator using sensors”, International Journal of Innovative Research in Science, Engineering and Technology (An ISO 3297: 2007 Certified Organization), Vol. 4, Issue 12, December 2015.

[4]. Shijo Joseph Mathew, S.Mathankumar and S.Vaishnodevi, “Portable neonatal Intensive care unit”, International Journal of Innovative Research in Science, Engineering and Technology”, (An ISO 3297: 2007 Certified Organization), Vol. 4, Issue 5, May 2015.

[5]. N.A.A Hadi, M.H.C. Hasan, N.M.Z. Hashim, N.R.Mohamad, A.S. Rahimi and K.A.M. Annuar, “Temperature Monitoring System for Infant Incubator Using Arduino”, International journal for advance research, Volume 3, Issue VI, June 2015 ISSN 2320-6802.

[6]. Prof. Kranti Dive and Prof. Gitanjali Kulkarani, “Design of Embedded Device for Incubator for the Monitoring of Infants”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 11, November 2013.

[7]. Faruk AKTAS, Emre KAVUS and Yunus KAVUS, “A Real-Time Infant Health Monitoring System for Hard of Hearing Parents by using Android-based Mobil Devices”, Faruk AKTAS et al. / IU-JEEE Vol. 17(1), (2017), 3107-3112.

[8]. Sujal Rane1, Kajal Sutar2, Vaidehi Temghare3, Prof. Rahul Patil4 and Prof. Sandip Chavan, “Baby Health Monitoring System using Wireless and Remote Access Technology”, International Journal of Recent Trends in Engineering & Research

APPENDICES

This project shows, that the functionalities of the baby monitoring system can be completely managed using the 'Internet of Things' concept based on the IOT technology. In reference to survey of baby care by working parents, we examine and report on the conditions of total time spent to take care of the baby and by a particular working parents and further evaluate total time wastage as well as human errors involvement, while doing so. Finally, to provide an optimal solution, we consent with the idea of making baby cradle and health monitoring system completely automated, with assistance of IOT technology.

A

MAJOR PROJECT REPORT

On

GSM BASED INDUSTRY PROTECTION SYSTEM

Submitted by

1) Mr. C. Sai Manish(17K81A04J7) **2) Ms. M. Dona Thomas (17K81A04M4)**

3) Mr. M. Harish Reddy (17K81A04M2)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Esteemed Guidance of

Ms. T. Sowmya

Assistant Professor

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA & NAAC A+ Accredited

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA & NAAC A+ Accredited

Department of Electronics & Communication Engineering

JUNE 2021

BONAFIDE CERTIFICATE

This is to certify that the project entitled **"GSM BASED INDUSTRY PROTECTION SYSTEM",** is being submitted by **1. C. Sai Manish(17K81A04J7) 2. M. Dona Thomas(17K81A04M4) 3. M. Harish Reddy(17K81A04M2)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS & COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

PROJECT GUIDE

Ms. T. SOWMYA
Assistant Professor
Department of ECE

HEAD OF THE DEPARTMENT

Dr. B. HARI KRISHNA
Professor
Department of ECE

Internal Examiner

External Examiner

Place:

Date:

DECLARATION

We, the students of **'Bachelor of Technology in Department of Electronics & Communication Engineering'**, session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled **'GSM BASED INDUSTRY PROTECTION SYSTEM'** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

C. Sai Manish(17K81A04J7)
M. Dona Thomas(17K81A04M4)
M. Harish Reddy(17K81A04M2)

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **CHILKURI SAI MANISH** WITH ROLL NO.**17K81A04J7, M HARISH REDDY** WITH ROLL NO.**17K81A04M2**, **MARIA DONA THOMAS** WITH ROLL NO.**17K81A04M4,** OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**GSM BASED INDUSTRY PROTECTION SYSTEM**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS LASYA IT
SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email: contact@lasyainfotech.com, ov@lasyainfotech.com
Website: www.lasyainfotech.com | contact: 7330666881/82/83/84/86

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, our beloved Principal, St. Martin's Engineering College for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr. B. Hari Krishna**, **Head & Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our Dept. project coordinator Dr. A. Anand Kumar** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide, **Ms. T. Sowmya, Assistant Professor**, Department of ECE and our class project coordinator **Ms. P. Pushpa**, **Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Malayadri**, **Professor in IT** and **Dr. R. Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. C. Sai Manish
2. M. Dona Thomas
3. M. Harish Reddy

ABSTRACT

With heavy automation taking place these days, the number of industries as well as number of accidents in those industries have increased substantially. Human errors and manual safety systems lead to increase in industrial accidents. So here we propose a GSM based industrial protection system that detects gas, temperature as well as light to keep track of accidents. The system needs to detect if the temperature of a machine or environment goes above a certain level, if it goes above a level, it is an indication of a malfunction. So here we use temperature sensor to detect a possible accident. Next indication is the malfunction or overloading of machines leading to smoke. We use gas detectors to detect hazardous gas and similarly a light sensor to detect light due to spark or any flash caused due to malfunctions. All these indicators/sensors constantly provide their input to the microcontroller connected to them. The microcontroller constantly processes this data. On encountering a warning signal from any of these sensors the microcontroller now communicates this data to a GSM modem interfaced to it and sends out an SMS message to intended user as a warning.

.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSORY OF TERMS

GSM- (Global System for Mobile Communications)

A digital mobile phone standard used extensively in Europe, the Middle East, Africa, Asia and in parts of America and Canada. First introduced in 1991, the GSM standard has been deployed at three different frequency bands: 900 MHz, 1800 MHz and 1900 MHz GSM 1900 is primarily deployed in North America. Named after its frequency band around 900 MHz, GSM-900 has provided the basis for several other networks using GSM technology. GSM uses narrowband TDMA which allows eight simultaneous calls on the same radio frequency. Along with CDMA and TDMA it represents the second generation of wireless networks.

2G: Second-generation mobile network or service. Generic name for second generation networks, for example GSM

Bandwidth: The range of frequencies available to be occupied by signals. In analogue systems, it is measured in terms of Hertz (Hz) and in digital systems in bit/s per second (bit/s). The higher the bandwidth, the greater the amount of information that can be transmitted in a given time. High bandwidth channels are referred to as broadband which typically means 1.5/2.0 Mbit/s or higher. Bit (binary digit): A bit is the primary unit of electronic, digital data. Written in base-2, binary language as a “1” or a “0”. Bit/s: Bits per second. Measurement of the transmission speed of units of data (bits) over a network. Also, kbit/s: kilobits (1’000) per second; Mbit/s: megabits (1’000’000) per second, and Gbit/s:

Gigabits (1’000’000’000) per second.

Sensor: A device, such as a photoelectric cell, that receives and responds to a signal or stimulus.

Server: (1) A host computer on a network that sends stored information in response to requests or queries.

(2) The term server is also used to refer to the soft-ware that makes the process of serving information possible.

SIM: Subscriber identity module (card). A small printed circuit board inserted into a GSM-based mobile phone. It includes subscriber details, security information and a memory for a personal directory of numbers. This information can be retained by subscribers when changing handsets.

SMS (Short Message Service): A service available on digital networks, typically enabling messages with up to 160 characters to be sent or received via the message Centre of a network operator to a subscriber's mobile phone.

Spectrum: The radio frequency spectrum of hertzian waves used as a transmission medium for cellular radio, radiopaging, satellite communication, over-the-air broadcasting and other services.

TD-SCDMA: Time Division Synchronous Code Division Multiple Access. A third-generation mobile standard under the IMT-2000 project. It uses spread spectrum CDMA technology in the TDD technique.

Wireless: Generic term for mobile communication services which do not use fixed-line networks for direct access to the subscriber.

Mobile: As used in this report, the term refers to mobile cellular systems.

Analogue: Transmission of voice and images using electrical signals. Analogue mobile cellular systems include AMPS, NMT and TACS.

Digital: Representation of voice or other information using digits 0 and 1. The digits are transmitted as a series of pulses. Digital networks allow for higher capacity, greater functionality and improved quality

Coverage: Refers to the range of a mobile cellular network, measured in terms of geographic coverage (the percentage of the territorial area covered by mobile cellular) or population coverage (the percentage of the population within range of a mobile cellular network).

CHAPTER-1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT:

Protection is the main concern for every industry. Every industry wants to work in safe and secured that are beneficial for the Employees and specially for their production process say for raw materials in the industry. Every industry wants their workers to keep safe and secured from various incidents like accidents caused due to Fire detector or accidents due to fire in their go down or their machinery department.

Industry security is the most significant one for every industry owner. To get the absolute peace of mind whether you are at industry or out of industry you must ensure that your industry is installed with the perfect industry security monitoring system. This GSM Bases industrial security system can be used to provide security system for home, industrial, office, School, Collages using GSM technique. Security systems are certain electronic devices which are used to detect intrusions in industry.

The basic components of an industry automation security system are motion detectors, fire detectors and motion detector. It is cheaper and maintained easily than any other security device. All the sensors are activated on the Security system. Whenever systems experience abnormal condition in the industry like any fire/smoke occurs in the industry and any intrusion into the industry the Security system alerts the security personnel as well as the owner of the industry by sending SMS alerts to the users of the industry. The system operates with the help of sensors installed in this system.

1.2 OBJECTIVES OF THE STUDY:

The main objectives of this project Industrial Protection Systems using GSM are:

i. To understand and gain knowledge about the function of Arduino, wireless sensors system and the basic concept of GSM Technology.

ii. To design an Industrial Protection Systems using GSM that requires low cost, low power consumption and user friendly.

iii. Industrial safety is needed to check all the possible chances of accidents for preventing loss of life and permanent disability of any industrial employee, any damage to machine and material.

iv. To eliminate accidents causing work stoppage and production loss.

v. To reduce workman's compensation, insurance rate, and all the cost of accidents.

vi. To achieve better morale among industrial employees.

vii. To prevent accidents in the industry by reducing any hazards.

Specific Objectives

i. To interface LM35 temperature sensor, GSM SIM800l Modem, 16X2 LCD, and process control devices (through relays) with the Arduino UNO.

ii. To send the measured temperature to distant user wirelessly with the help of GSM module (SIM 800) Via SMS.

iii. To send commands through user's mobile phone to control the process control devices based on sensed sensor readings.

iv. To test and implement the GSM Based Industrial Process Monitoring and Control System.

1.3 SCOPE OF THE STUDY:

The scope for this project focuses primarily on the protection system in the industries. In this study, GSM technology is used to establish a protection system for industries and by applying the wireless sensory system to detect and collects the parameter information like smoke, gas, temperature, and fire. The other purpose is that the designated system could be able to transmit and receive the information through SMS (Short Message Service) or Call using GSM (Global System for Mobile communication) technology in case of crisis or emergency event occurred. The Arduino Uno act as the micro-controller for the protection system. The idea is to see how well this project can benefit the industries in term of secureness and protection. The scope includes connecting the different industrial process control systems to relays for controlling the industrial environment complex tasks like nuclear plants and reactors in the industry. A GSM server is implemented with Arduino, sensors and relays. The GSM Modem can provide the necessary data related to the industry to a maintenance officer located anywhere at any time. According to data received officer will take some action by sending some AT commands to Arduino through mobile unit to GSM modem. Arduino decodes the commands and controls the industrial process devices through relays.

Significance of the project

i. Industrial and Factory process monitoring and control delivers increased productivity and improves product quality.
ii. It also presents real-time data that helps in making informed decisions.

1.4 MATERIAL REQUIREMENT:

1.4.1 HARDWARE REQUIREMENT:

- Arduino UNO
- LCD
- Power Supply
- Buzzer
- GSM Module(800l)
- Temperature Sensor (LM35)
- Light Sensor (LDR Sensor)
- Gas Sensor (MQ2)
- 12V DC Motor

1.4.2 SOFTWARE REQUIREMENT:

- Proteus software
- Arduino software

1.5 PROCUMENT OF EQUIPMENT:

We collected all the required components and to interface it ,checked the working condition of all the sensors.
Also, verified the threshold values of Sensors, which are helpful in calculating the exact values.

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH AREA:

Majority of the companies in INDIA have not implemented Automatic protection systems in industry. Except few large industries majority of the companies cannot afford to invest huge amount of money in the existing costly setups to meet the requirements of Industrial Automation.

Existing methods widely use the following technologies to communicate the information from one end to the other end of the company.

Using Bluetooth — But it is limited to short range.

Using Zigbee/ IEEE802.15.4 — Range is up to only few Kms maximum.

Using Wi-Fi — Requires costly equipment setup and high-power consumption.

All the methods discussed above are quite expensive and complex to implement and not very reliable. The availability of information at various nodes simultaneously is not achieved.

Bluetooth Technology

Bluetooth Technology is a radio frequency (RF)-based, short-range connectivity technology that promises to change the face of computing and wireless communication. It is designed to be an inexpensive, wireless networking system for all classes of portable devices. The projected cost of the Radio chip was around $5.

A complete Bluetooth system will require these elements:

- An RF portion for receiving and transmitting data includes short-range radio transceiver, an external antenna, and a clock reference (required for synchronization)
- A module with a baseband microprocessor
- Memory
- An interface to the host device (such as a mobile phone)

Its normal range of operation is 10m (at 1mW transmit power) and can be increased up to 100m by increasing the transmit power to 100mW. The system operates in unlicensed 2.4 GHz frequency-band; hence it can be used worldwide without any licensing issues. It provides an aggregate bit rate of approximately 1Mbps.

Zigbee Technology

The Zigbee radio specification designed for low cost and power consumption than Bluetooth. The specification is based on IEEE 802.15.4 standard. The radio operates in the same ISM band as Bluetooth and is capable of connecting 255 devices per network. The specification supports data rates of up to 250Kbps at a range of up to 30m. These data rates are slower than Bluetooth, but in exchange the radio consumes significantly with low power with a large transmission range. The goal of Zigbee is to provide radio operation for months or years without recharging, thereby targeting applications such as sensor networks and inventory tags. The beauty of Zigbee is that devices from different manufacturers will be able to work together, as long as all are compliant to the standard. It has been suggested that the name evokes the haphazard paths that bees follow as they harvest pollen, similar to the way packets would move through a mesh network.

Zigbee is standardized at two levels – the radio chips must follow certain design rules, and the protocol layers that actually make the network function are defined and controlled by the Zigbee Alliance. Advantages are: Reliable and self-healing, supports large number of nodes, Easy to deploy, very long battery life, Secure and Low cost.

Wi-Fi Technology

Wi-Fi is the name given by the Wi-Fi Alliance to the IEEE 802.11 suite of standards. 802.11 defined the initial standard for wireless local area networks (WLANs).

But because of its costly equipment setup and high-power consumption this technology is not preferred.

2.2 REVIEW ON RELATED LITERATURE:

GSM (Global System for Mobile communication) is a digital mobile telephone system that is widely used in Europe and other parts of the world. GSM uses a variation of Time Division Multiple Access (TDMA) and is the most widely used of the three digital wireless telephone technologies (TDMA, GSM, and CDMA). GSM digitizes and compresses data, then sends it down a channel with two other streams of user data, each in its own time slot. It operates at either the 900 MHz or 1,800 MHz frequency band. GSM is an international digital cellular telecommunication. The GSM standard was released by ETSI (European Standard Telecommunication Standard) back in 1989. In less than ten years since the first GSM network was commercially launched, it became, the world's leading and fastest growing mobile standard, spanning over 190 countries.

GSM characteristics

Mobile Frequency Range RX: 925-960; TX: 880-915

Multiple Access Method TDMA/FDMA

Duplex Method FDD

Number of Channels 124 (8 users per channel)

Channel Spacing 200kHz

Modulation GMSK (0.3 Gaussian Filter)

Channel Bit Rate 270.833Kb

Global System for Mobile (GSM) is a second-3

generation cellular standard developed to

deliver high quality and secure mobile voice and data services (such as SMS/ Text Messaging) with full roaming capabilities across the world using digital modulation.

Rozita Teymourzadeh, Salah Addin Ahmed, Kok Wai Chan, and Mok Vee Hoong suggested Smart GSM Based Home Automation System. This research work investigates the potential of 'Full Home Control', which is the aim of the house Automation Systems in near future. The implementation of the house automation technology using Global System for Mobile Communication (GSM) modem to manage home appliances like security system via SMS, light conditional system. The proposed research work is targeting functionality of the GSM protocol, which allows the user to manage the target system faraway from residential using frequency bandwidths. The concept of AT commands Nd serial communication has been applied towards development of the smart GSM-based home automation system. Home owners are going to be ready to receive feedback status of any home appliances in check whether switched on or off remotely from their mobile phones. PIC16F887 microcontroller with the mixing of GSM provides the smart automated house system with the specified baud of 9600 bps. The proposed prototype of GSM based home automation system was implemented and tested with maximum of 4 loads and shows the accuracy of ≥98%.

2.3 CONCLUSION ON REVIEWS:

In this chapter it will review and discuss about the "Industrial Protection System using GSM". The idea of this project originates from the problem faced by any user to inspect the condition/surrounding area in industries. Sensor detection of a remote place could be difficult or sometimes impossible for example in places where humans are prohibited to enter. The problem can be solved by using wireless sensor network to ensure the safeties of the area can be in check continuously by the user at any time and from anywhere. This chapter will explain about all the components that will be used and some related researches regarding this project. This chapter begins with a general outline of GSM technology, wireless sensor network, microcontroller (Arduino), and others related components. The second part of this chapter is dedicated to a related research that has correlation with this project "Industrial Protection System Using GSM". The third part of this chapter is the overview of the GSM standard, wireless sensor network, microcontroller (Arduino), and others hardware or software that are used in this project.

CHAPTER-3

PROJECT DESIGN

3.1 OVERVIEW OF THE DESIGN:

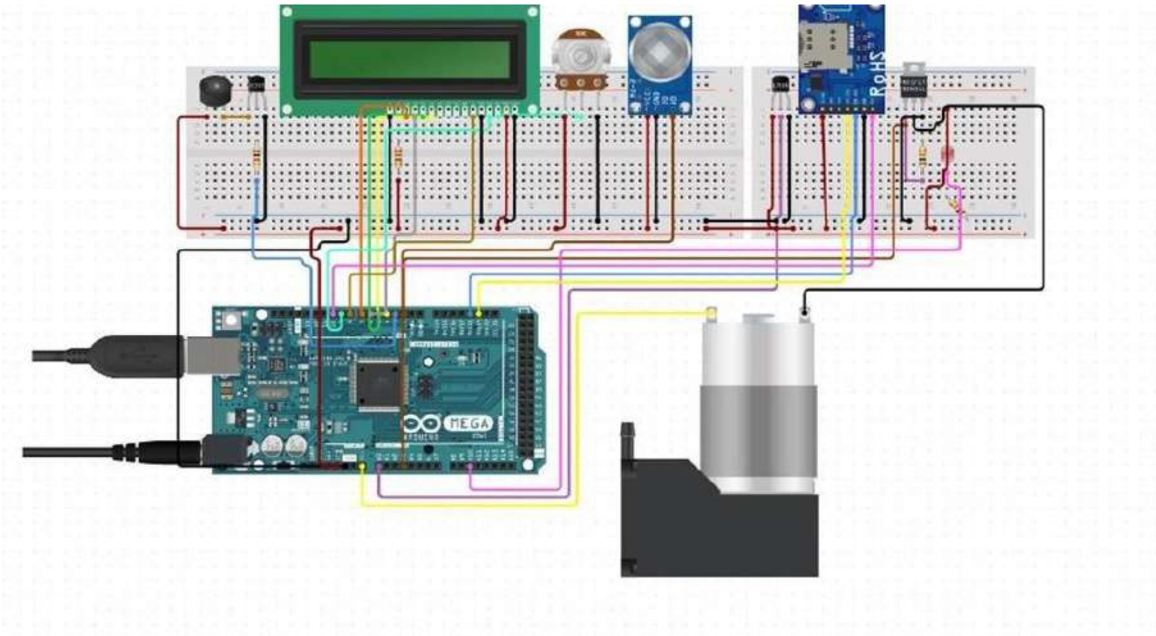


Fig.3.1 Overview of the design

3.2 EQUIPMENT ANALYSIS:

3.2.1 ARDUINO:

3.2.1.1 INTRODUCTION:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

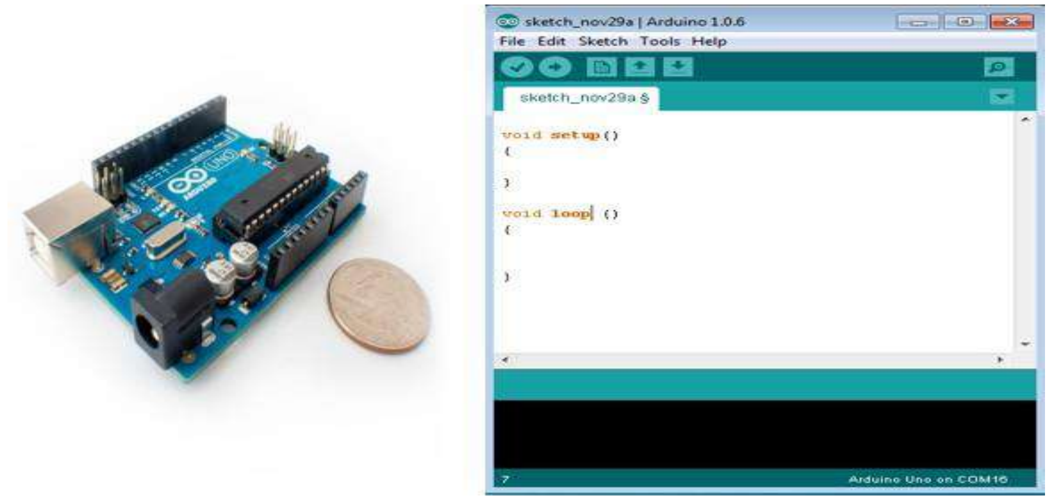


Fig 3.2.1.1 Arduino Uno

3.2.1.2 BOARD TYPES

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header

Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.2.1.2.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB

LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 3.2.1.2.2 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 3.2.1.2.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560	3.3V	84MHz	54	12	12	4	USB native

R3							

Table 3.2.1.2.4 Arduino boards based on AT91SAM3X8E microcontroller

3.2.1.3 BOARD DESCRIPTION:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

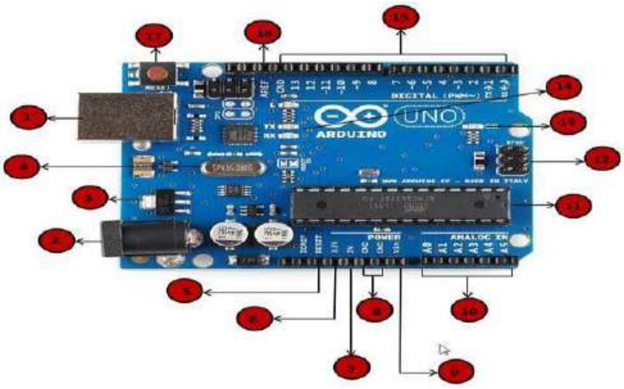


Fig.3.2.1.2 Board Description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of

	MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

3.2.1.4 ARDUINO FAMILY

Arduino makes several different boards, each with different capabilities. In addition, part of being open-source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

Fig.3.2.1.3: Arduino Family

3.2.1.5 SHIELDS:

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

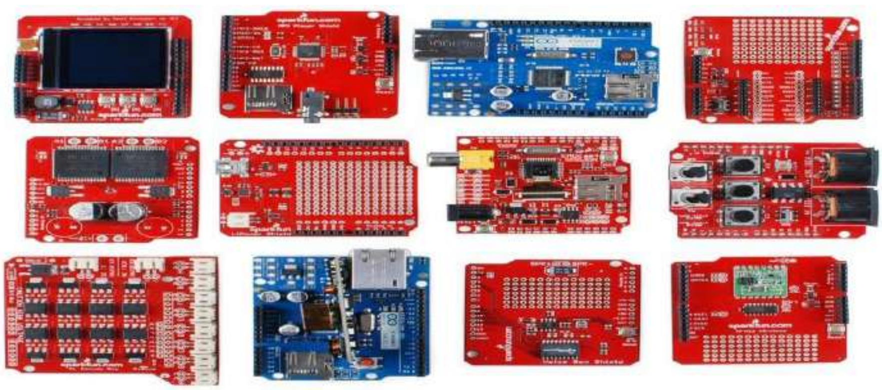


Fig.3.2.1.4: Arduino Shields

3.2.1.6 PIN DESCRIPTION OF ATMEGA328:

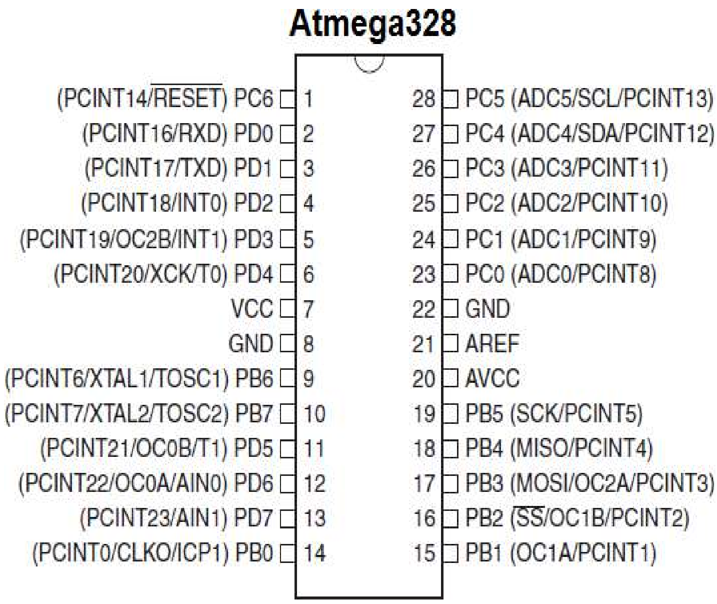


Fig.3.2.1.5 Pin description of ATMEGA328

3.2.1.7 APPLICATIONS:

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

3.2.2 16*2 LCD SCREEN DISPLAY

3.2.2.1 INTRODUCTION

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

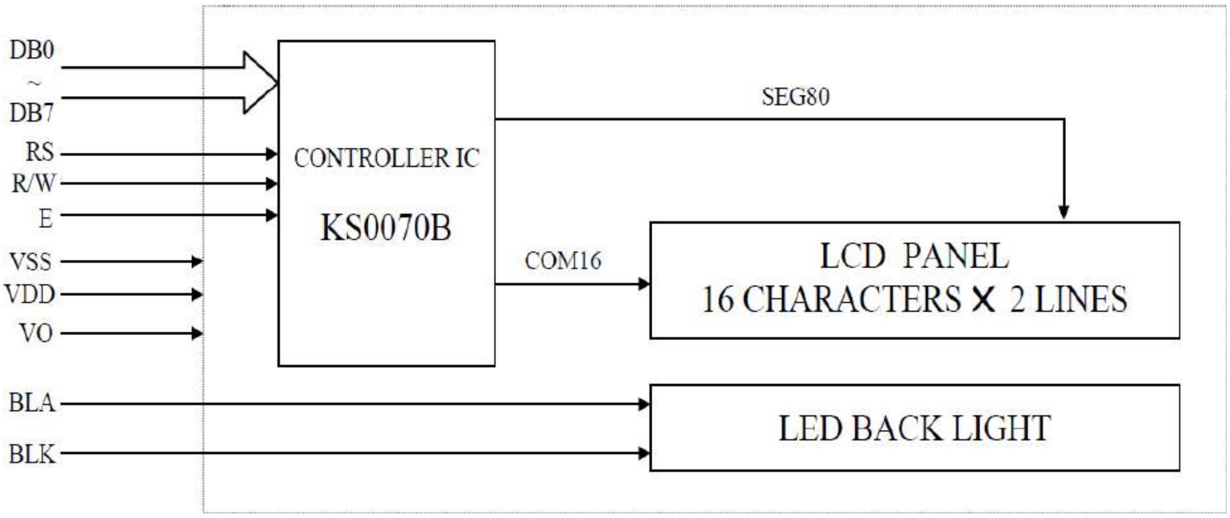


Fig.3.2.2.1 Lcd block diagram

3.2.2.2 DISPLAY DATA RAM:

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So, whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.

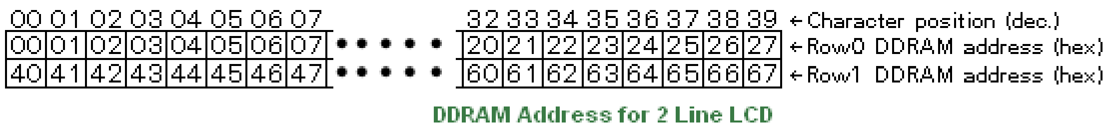


Fig.3.2.2.2 DDRAM addresses of 2 Line LCD

3.2.2.3 CHARACTER GENERATOR ROM:

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM.

3.2.2.4 BUS FLAG:

Busy Flag is a status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e., BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

3.2.2.5 INSTRUCTION REGISTER (IR) AND DATA REGISTER (DR):

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g., LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD. Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

3.2.2.6 16 X 2 ALPHANUMERIC LCD MODULE FEATURES:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

3.2.2.7 CIRCUIT DESCRIPTION:

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore, by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore, we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result, we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly. The 2-line x 16-character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

3.2.3 BUZZER

3.2.3.1 MAGNETIC TRANSDUCER

Magnetic transducers contain a magnetic circuit consisting of an iron core with a wound coil and a yoke plate, a permanent magnet and a vibrating diaphragm with a movable iron piece. The diaphragm is slightly pulled towards the top of the core by the magnet's magnetic field. When a positive AC signal is applied, the current flowing through the excitation coil produces a fluctuating magnetic field, which causes the diaphragm to vibrate up and down, thus vibrating air. Resonance amplifies vibration through resonator consisting of sound hole(s) and cavity and produces a loud sound.

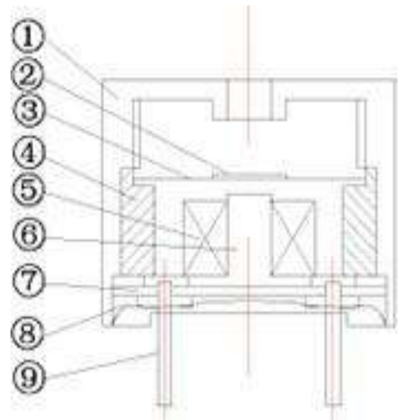


Fig.3.2.3.1 Magnetic Transducer

3.2.3.2 MAGNETIC BUZZER(SOUNDER)

Buzzers like the TMB-series are magnetic audible signal devices with built-in oscillating circuits. The construction combines an oscillation circuit unit with a detection coil, a drive coil and a magnetic transducer. Transistors, resistors, diodes and other small devices act as circuit devices for driving sound generators. With the application of voltage, current flows to the drive coil on primary side and to the detection coil on the secondary side. The amplification circuit, including the transistor and the feedback circuit, causes vibration. The oscillation current excites the coil and the unit generates an AC magnetic field corresponding to an oscillation frequency. This AC magnetic field magnetizes the yoke comprising the magnetic circuit. The oscillation from the intermittent magnetization prompts the vibration diaphragm to vibrate up and down, generating buzzer sounds through the resonator.

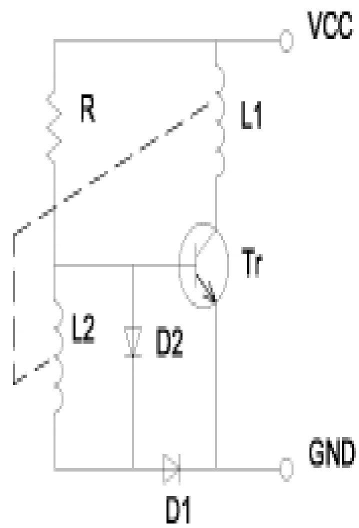


Fig.3.2.3.2 Magnetic Buzzer

3.2.3.3 SPECIFICATIONS:

Rated Voltage: A magnetic buzzer is driven by 1/2 square waves (V o-p).
Operating Voltage: For normal operating. But it is not guaranteed to make the minimum Sound Pressure Level (SPL) under the rated voltage.
Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.
Direct Current Resistance: The direct current resistance is measured by ammeter directly.
Sound Output: The sound output is measured by decibel meter. Applying rated voltage and 1/2 square waves, and the distance of 10 cm.
Rated Frequency: A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.
Operating Temp: Keep working well between -30℃ and +70℃.

How to choose:

Driving methods: AX series with built drive circuit will be the best choice when we cannot provide frequency signal to a buzzer, it only needs direct current.

Dimension: Dimension affects frequency, small size result in high frequency.

Voltage: Depend on V o-p (1/2 square waves)

Fixed methods: From the highest cost to the lowest- DIP, wires/ connector, SMD.

Soldering methods: AS series is soldered by hand, the frequency is lower because of the holes on the bottom. On the other hand, we suggest AC series for the reflow soldering, the reliability is better.

3.2.3.4 HOW TO CHOOSE A BUZZER:

There are many different kinds of buzzer to choose, first we need to know a few parameters, such as voltage, current, drive method, dimension, mounting type, and the most important thing is how much SPL and frequency we want.

Operating voltage: Normally, the operating voltage for a magnetic buzzer is from 1.5V to 24V, for a piezo buzzer is from 3V to 220V. However, in order to get enough SPL, we suggest giving at least 9V to drive a piezo buzzer.

Consumption current: According to the different voltage, the consumption current of a magnetic buzzer is from dozens to hundreds of mill amperes; oppositely, the piezo type saves much more electricity, only needs a few mill amperes, and consumes three times current when the buzzer start to work.

Driving method: Both magnetic and piezo buzzer have self-drive type to choose. Because of the internal set drive circuit, the self-drive buzzer can emit sound as long as connecting with the direct current. Due to the different work principle, the magnetic buzzer need to be driven by 1/2 square waves, and the piezo buzzer need square waves to get better sound output.

Dimension: The dimension of the buzzer affects its SPL and the frequency, the dimension of the magnetic buzzer is from 7 mm to 25 mm; the piezo buzzer is from 12 mm to 50 mm, or even bigger.

Connecting way: Dip type, Wire type, SMD type, and screwed type for big piezo buzzer are usuallyseen.

Sound Pressure Level (SPL): Buzzer is usually tested the SPL at the distance of 10 cm, if distance double, the SPL will decay about 6 dB; oppositely, the SPL will increase 6 dB when the distance is shortened by one time. The SPL of the magnetic buzzer can reach to around 85 dB/ 10 cm; the piezo buzzer can be designed to emit very loud sound, for example, the common siren, are mostly made of piezo buzzer.

3.2.3.5 INTRODUCTION OF PIEZO BUZZER:

Piezo buzzers are simple devices that can generate basic beeps and tones. They work by using a piezo crystal, a special material that changes shape when voltage is applied to it. If the crystal pushes against a diaphragm, like a tiny speaker cone, it can generate a pressure wave which the human ear picks up as sound. Simple change the frequency of the voltage sent to the piezo and it will start generating sounds by changing shape very quickly!

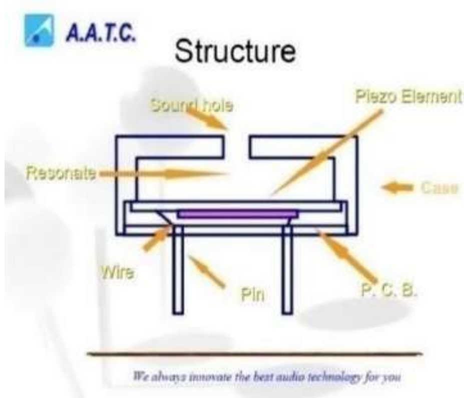


Fig.3.2.3.5 piezo buzzer

3.2.3.6 SPECIFICATIONS:

Rated Voltage: A piezo buzzer is driven by square waves (Vp-p).

Operating Voltage: For normal operating. But it is not guaranteed to make the minimum SPL under the rated voltage.

Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.

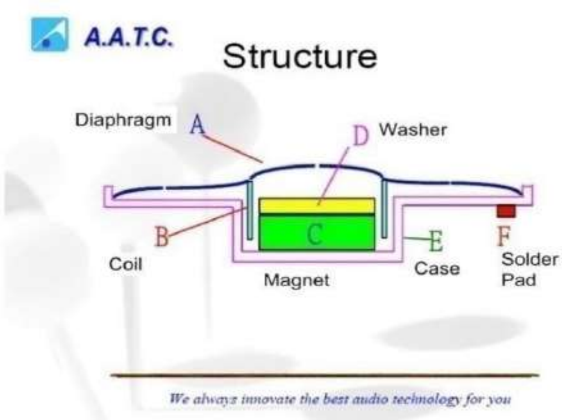


Fig. 3.2.3.6 structure of piezo buzzer

Capacitance: A piezo buzzer can make higher SPL with higher capacitance, but it consumes more electricity.

Sound Output: The sound output is measured by decibel meter.

Applying rated voltage and square waves, and the distance of 10 cm. **Rated Frequency:** A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency. **Operating Temp.:** Keep working well between -30℃ and +70℃.

3.2.4 TEMPERATURE SENSOR(LM35):

Temperature Sensor which converts temperature value into electrical signals. We used IC called LM 35 as a temperature sensor. LM35 series sensors are precision integrated-circuit temperature sensors whose output voltage is linearly proportional to the Celsius temperature. The LM35 requires no external calibration since it is internally calibrated. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°C over a full −55 to +150°C temperature range.

The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60µA from its supply, it has very low self-heating, less than 0.1°C in still air.

3.2.4.1 Features:

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteed (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±1⁄4°C typical

- Low impedance output, 0.1 W for 1 mA load

Fig 3.2.4.1: Temperature Sensor

The characteristic of this LM35 sensor is:

For each degree of centigrade temperature, it outputs 10milli volts. ADC accepts the output from LM35 and converts that data into digital form which is sent to microcontroller for further processing.

3.2.4.2 TEMPERATURE SENSING CIRCUIT:

The methods of temperature measurement may be divided into two main classes according as the exchange of heat between the testing body and the hot system takes place by contact, or by radiation across a space. In the contact methods, thermometers or thermocouples are used and they are immersed in solids or liquids. The thermodynamic equilibrium between the hot body and the testing body is established by material contact. In the non-contact methods, the thermodynamic equilibrium is established by the radiation emitted as excited atom and molecules in the hot body return to the ground state.

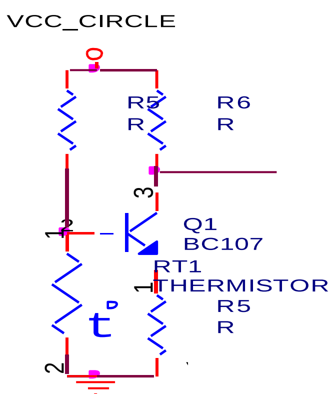


Fig.3.2.4.2: Temperature Sensing Circuit

3.2.5 GAS SENSOR:

- ➢ Gas sensors main aim is to sense hazardous gases that evolve its surroundings
- ➢ Gas sensor detects the concentrations of combustible gas in the air and outputs its reading as an analog voltage. The sensor can measure concentrations of flammable gas of 300 to 10,000 ppm. The sensor can operate at temperatures from -20 to 50°C and consumes less than 150 mA at 5 V.
- ➢ High sensitivity to LPG, Propane and Hydrogen.

3.2.6 SOFTWARE EXPLANATION:

- Arduino software
- Proteus simulation

3.2.6.1 ARDUINO SOFTWARE:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are
completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

3.2.6.2 WHY ARDUINO?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low-cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step-by-step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handy board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and
wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows. Wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.

- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for

beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

- Open source and extensible software - The Arduino software is published as open-source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under

a Creative Commons license, so experienced circuit designers can make their own version Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money

- Getting Started with Arduino and Genuino products: -

Install the Arduino Software (IDE) on Windows PCs-
This document explains how to install the Arduino Software (IDE) on Windows machines.

✓ Download the Arduino Software (IDE)

✓ Proceed with board specific instructions.

3.2.6.3 HOW TO DOWNLOAD THE ARDUINO SOFTWARE (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.
When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

3.2.6.4 INSTALLATION:

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega

2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.3.2.6.4.1 Standard USB Cable (A plug to B plug)

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig. 3.2.6.4.2 Standard USB Cable (an A to Mini-B plug)

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

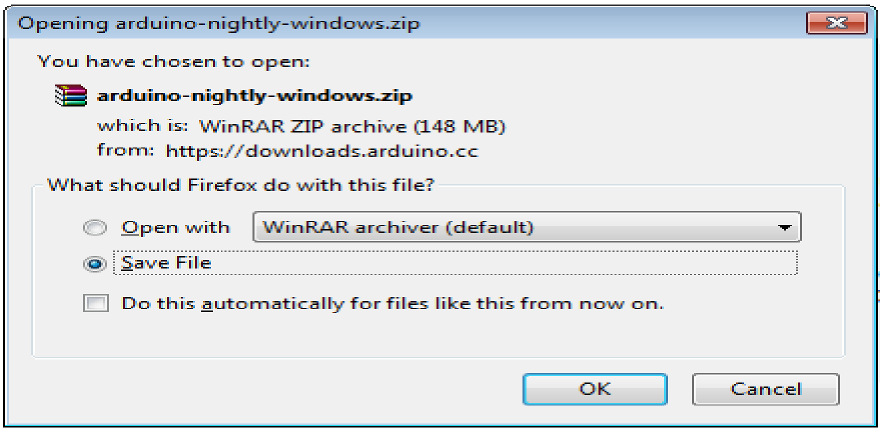


Fig. 3.2.6.4.3 Download Arduino IDE Software

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

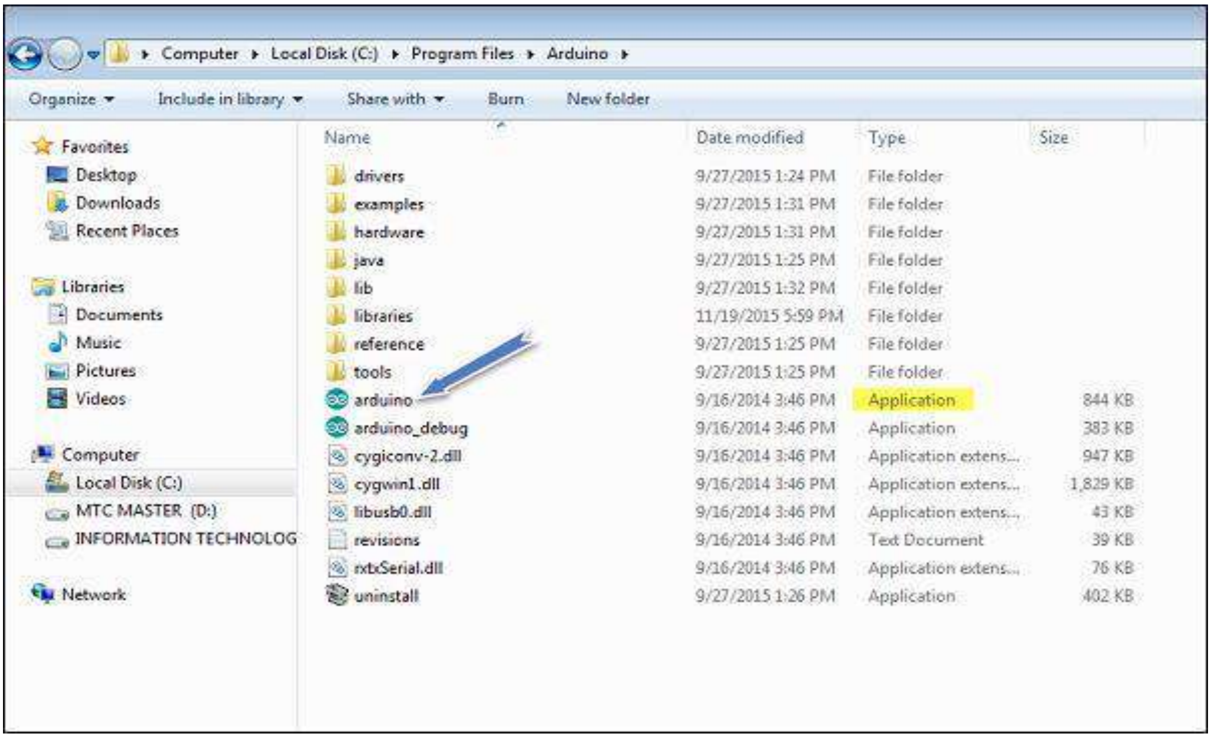


Fig. 3.2.6.4.4 Launch Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**

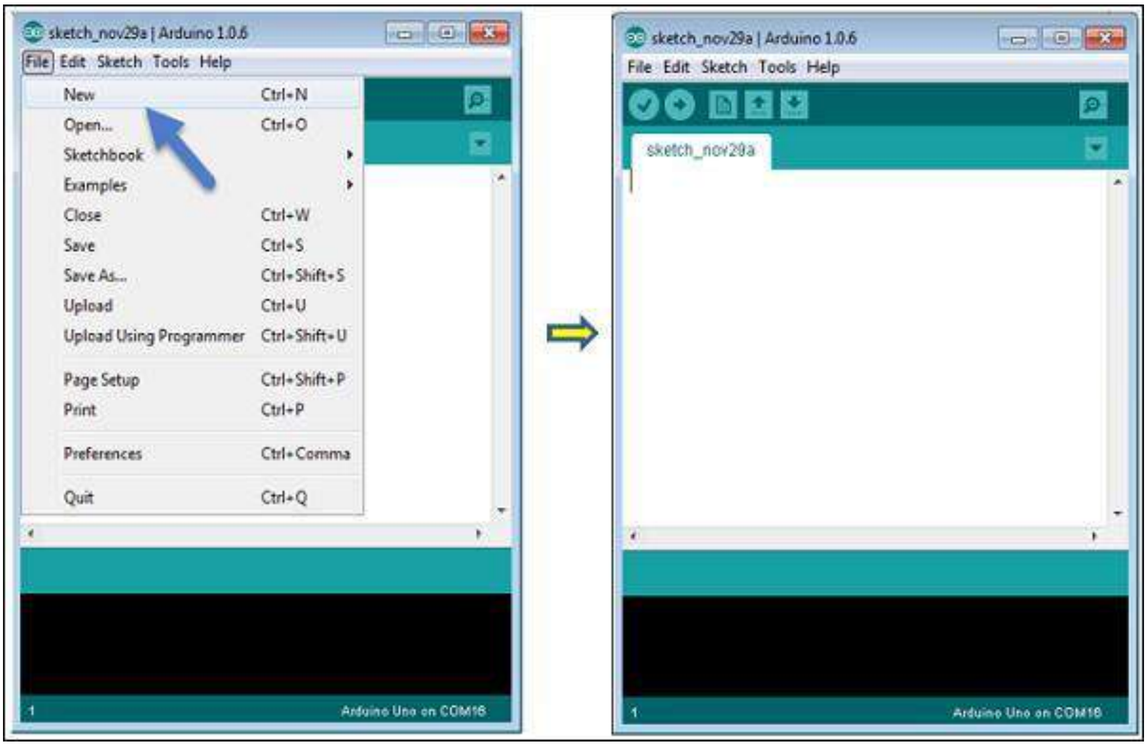
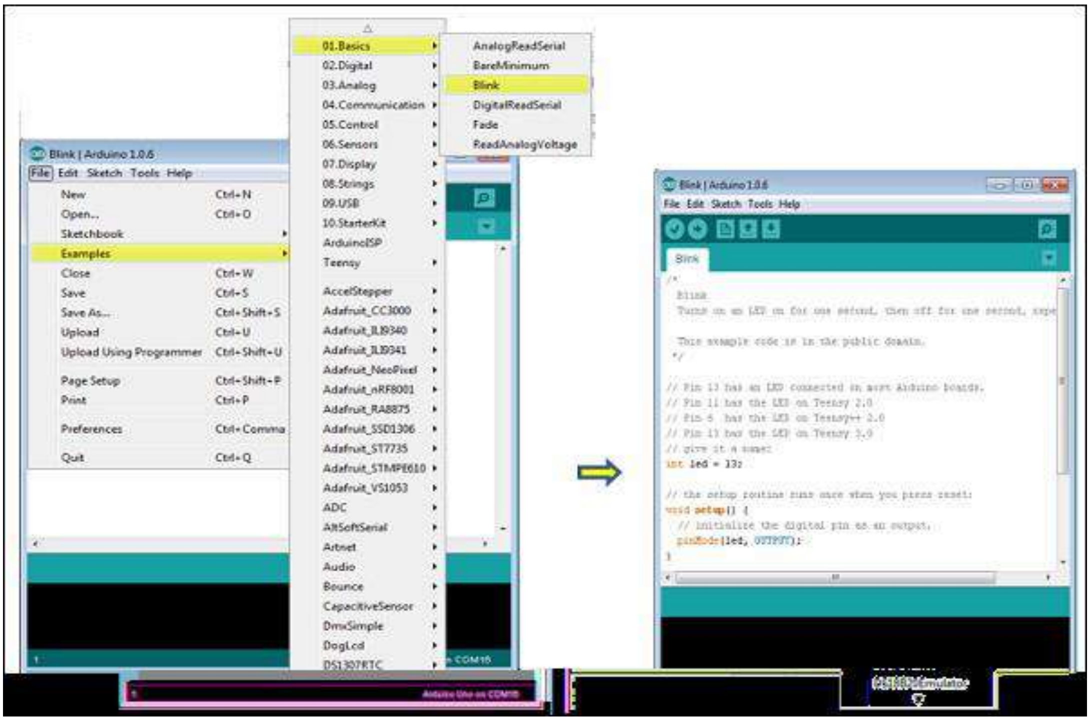


Fig. 3.2.6.4.5 Open your first project.

To open an existing project example, select File → Example → Basics → Blink.



.

Fig. 3.2.6.4.6 Open your first program.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

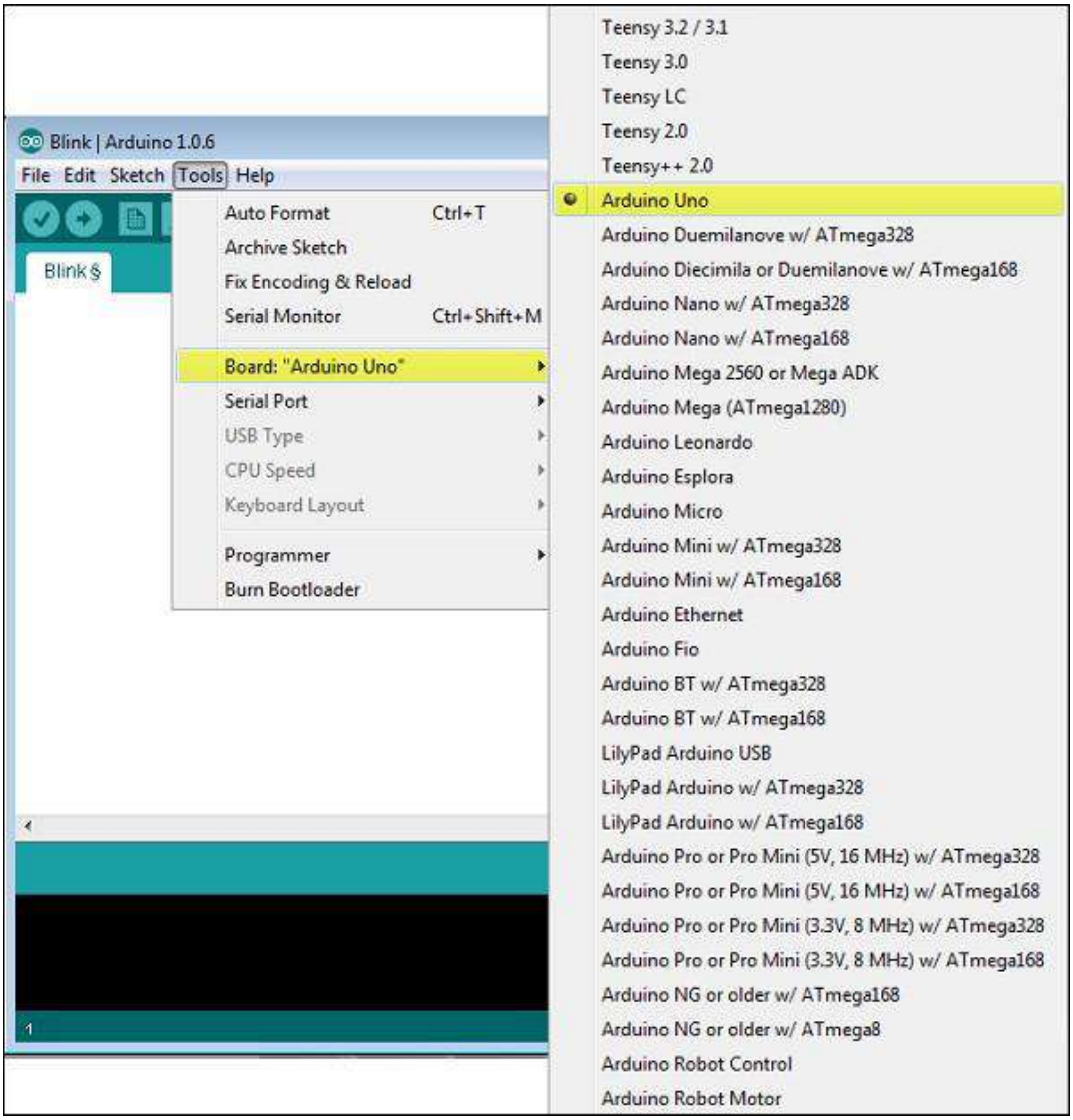


Fig. 3.2.6.4.7 Select your Arduino board.

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

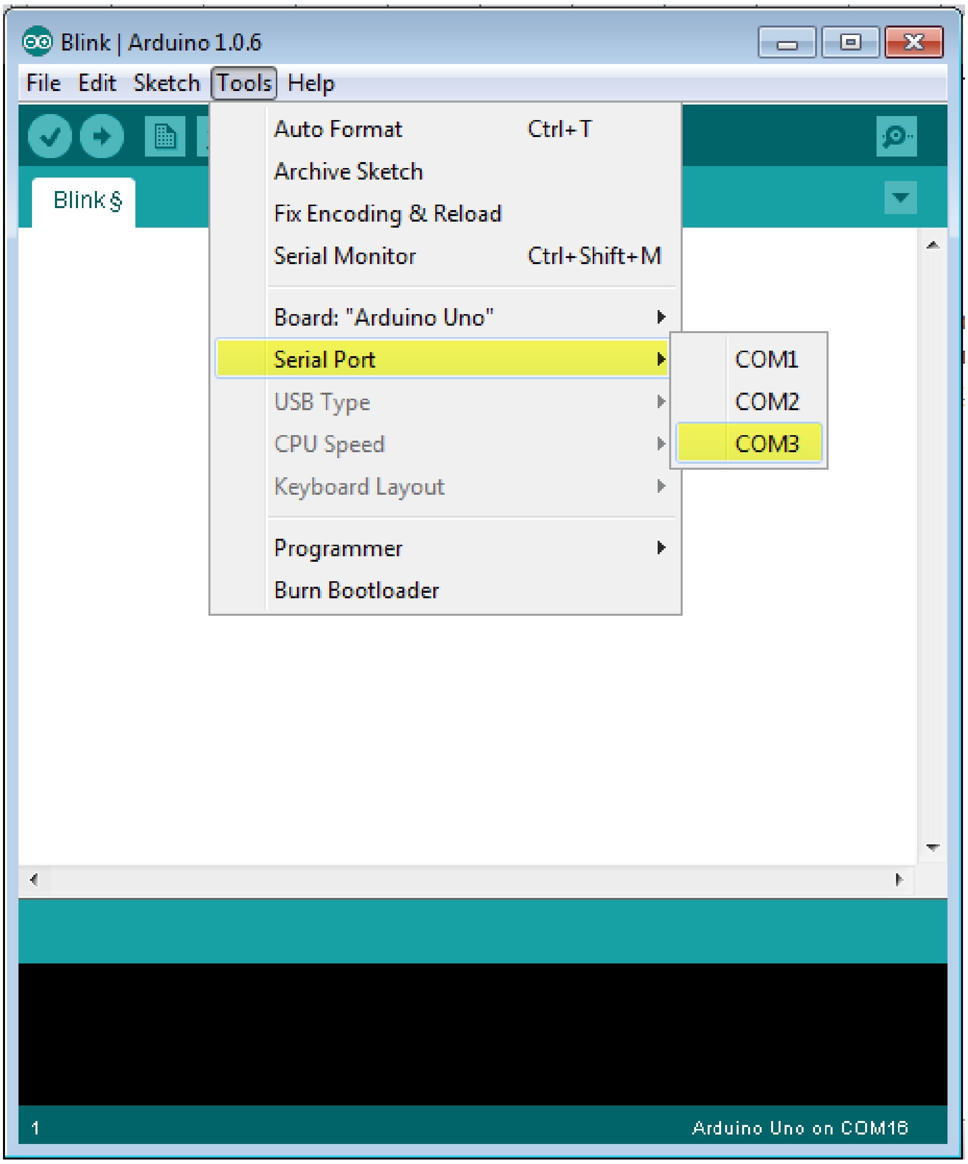


Fig. 3.2.6.4.8 Select your serial port.

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

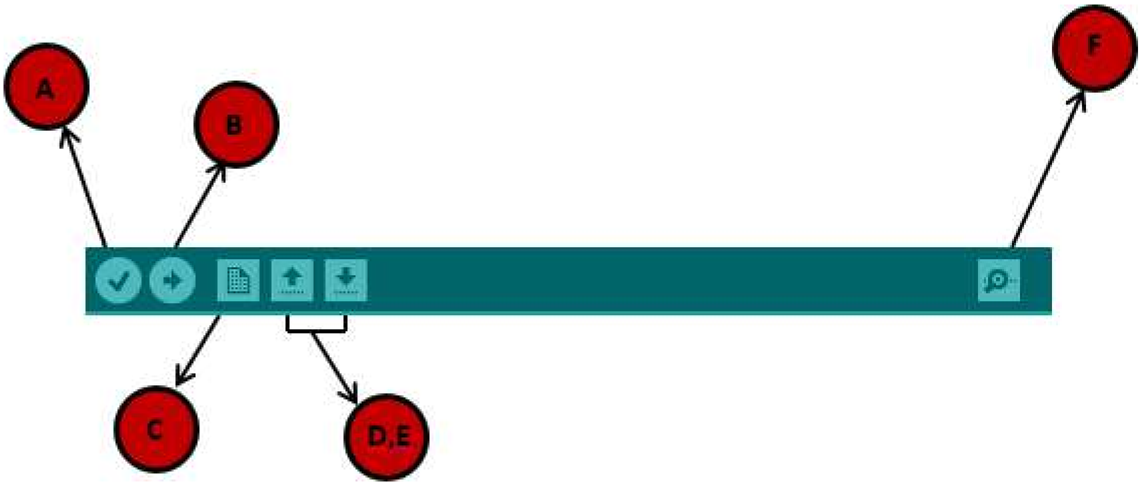


Fig. 3.2.6.4.9 Upload the program to your board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from board & send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

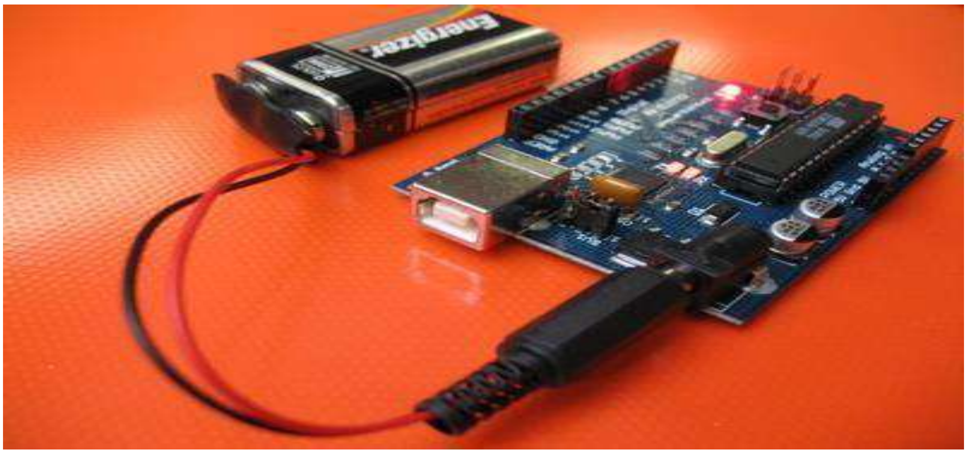


Fig. 3.2.6.4.10 Connecting a Battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called "sketches", but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

Fig. 3.2.6.4.11 Example Program

3.2.7 PROTEUS:

PROTEUS:

Proteus is a simulation and design software tool developed by Lab Centre Electronics for Electrical and Electronic circuit design. It also possesses 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole

packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design:

Step 1: Open ISIS software and select new design in File menu

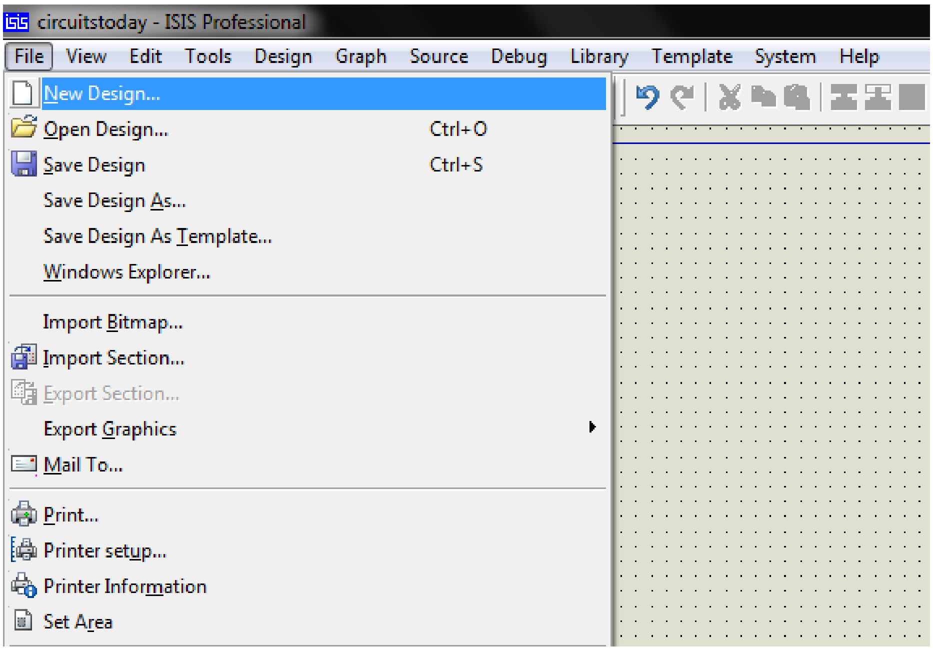


Fig. 3.2.7.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

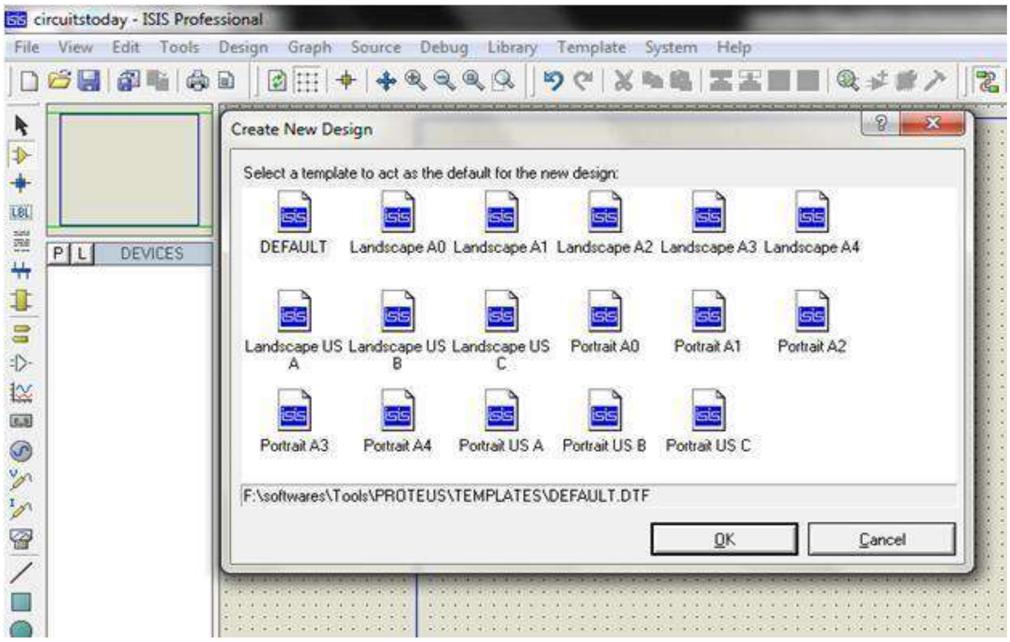


Fig. 3.2.7.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

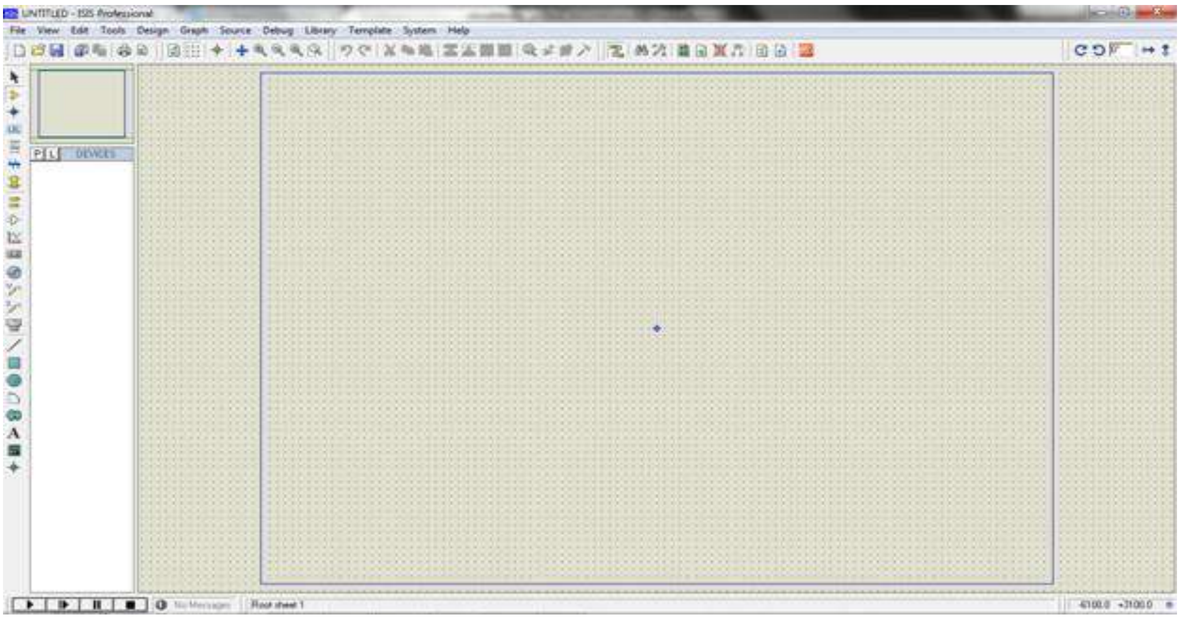


Fig. 3.2.7.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

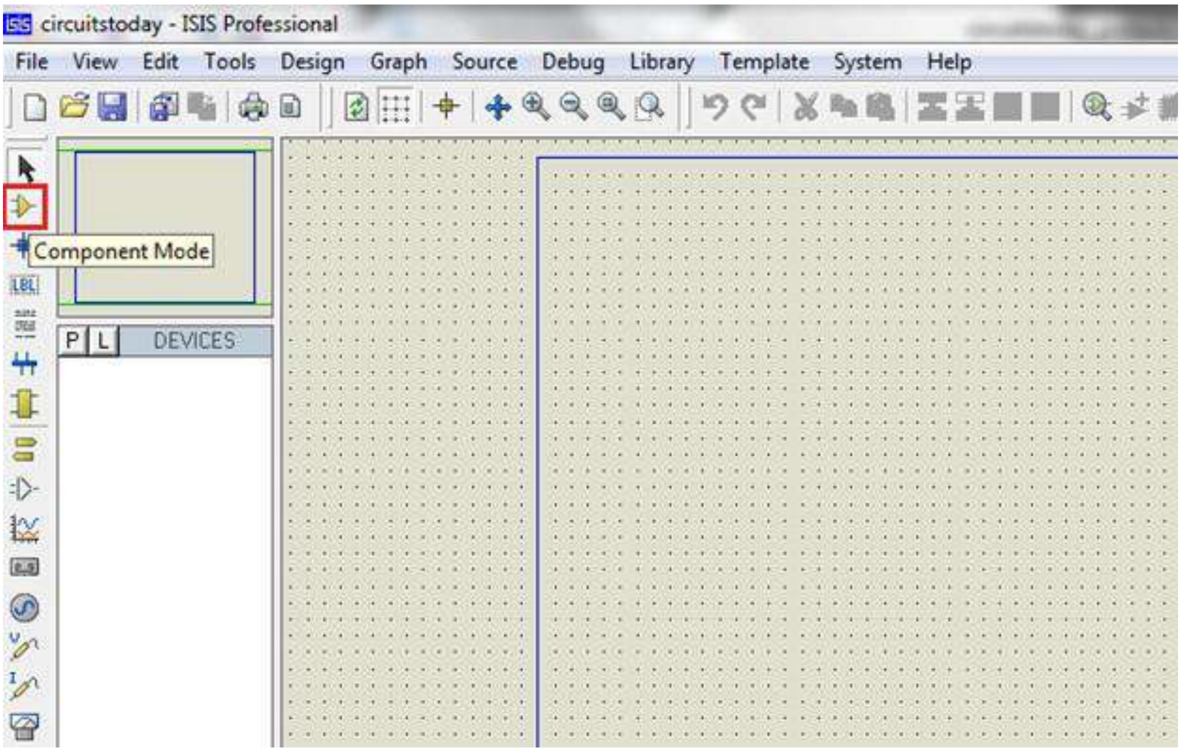


Fig. 3.2.7.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

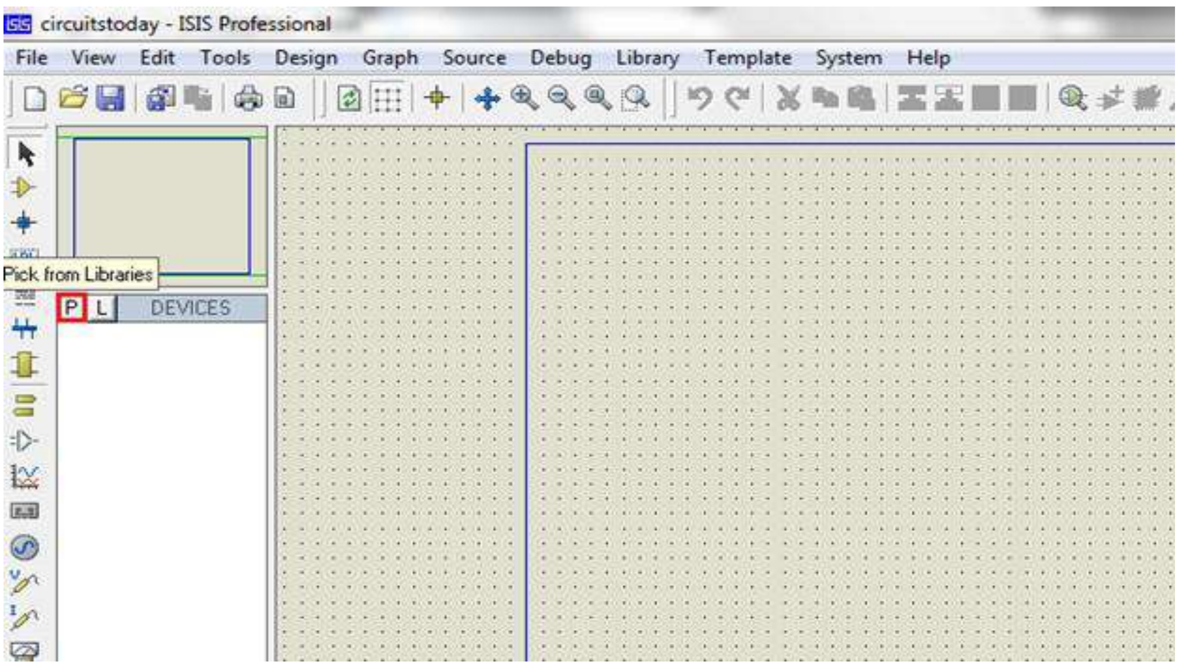


Fig. 3.2.7.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

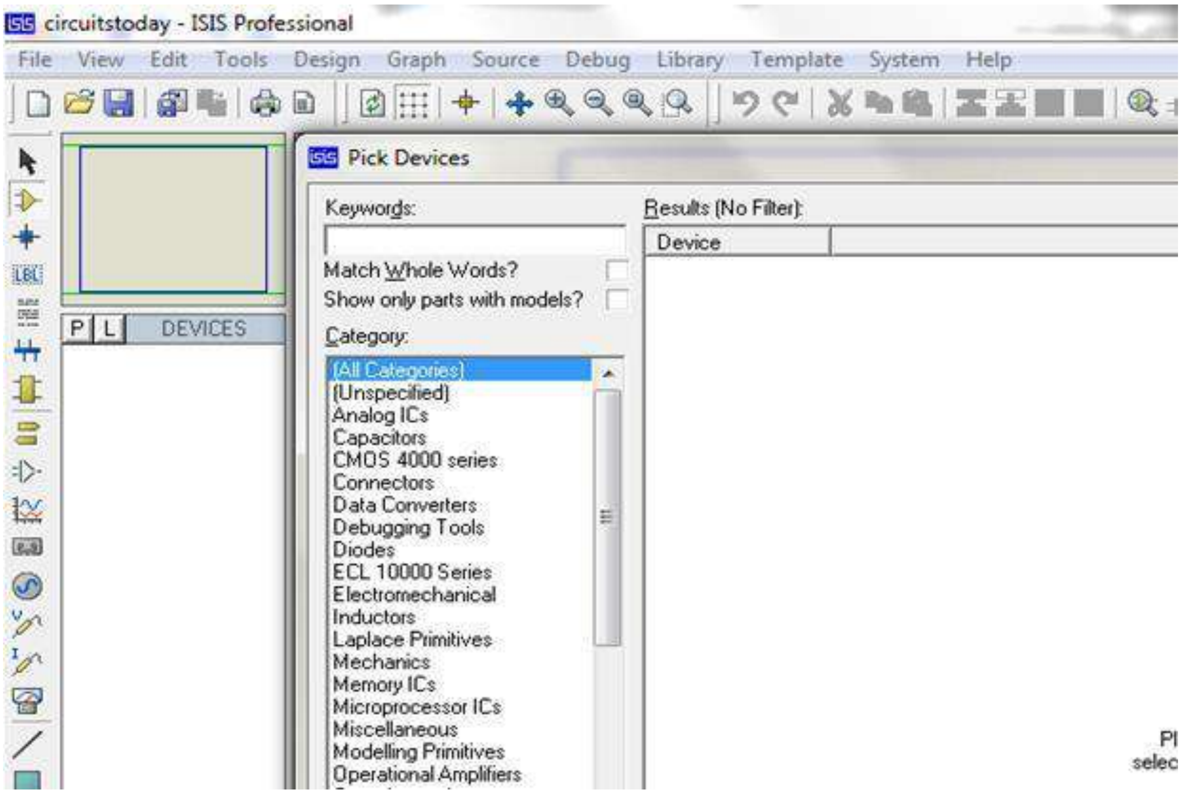


Fig. 3.2.7.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

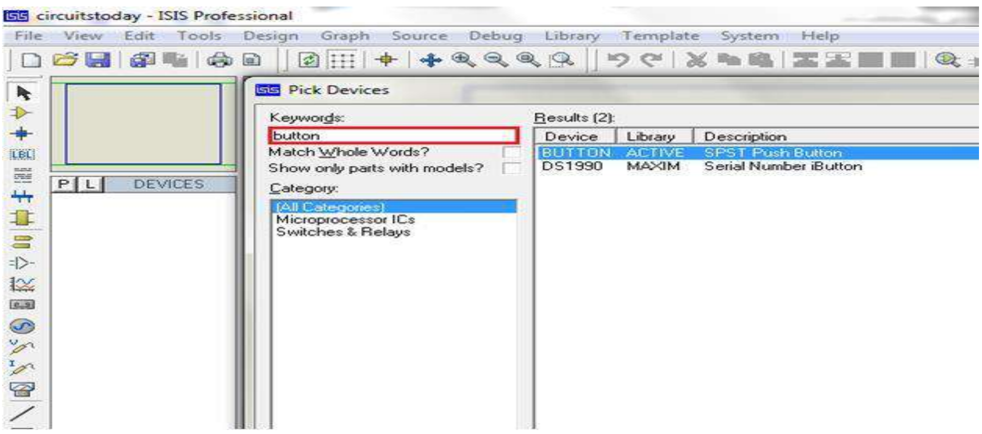


Fig. 3.2.7.7 :Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

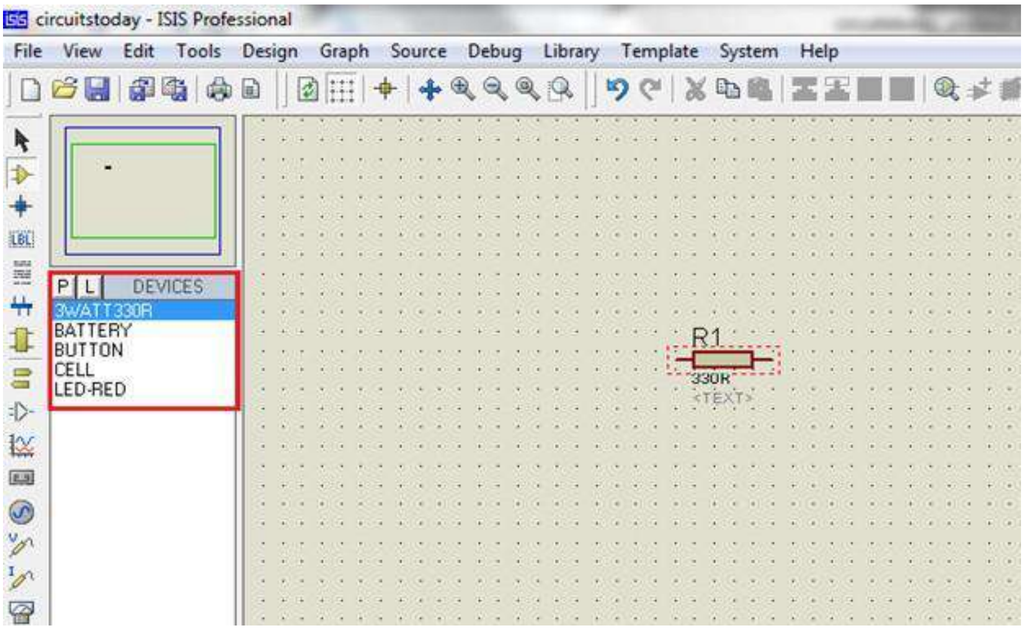


Fig. 3.2.7.8 Component Selection

Place all the required components and route the wires i.e., make connections. Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

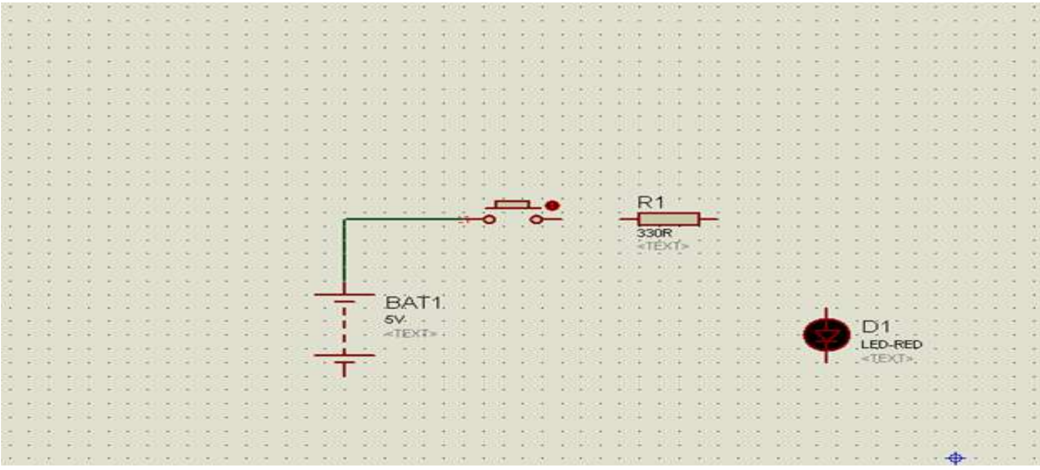


Fig. 3.2.7.9 :Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

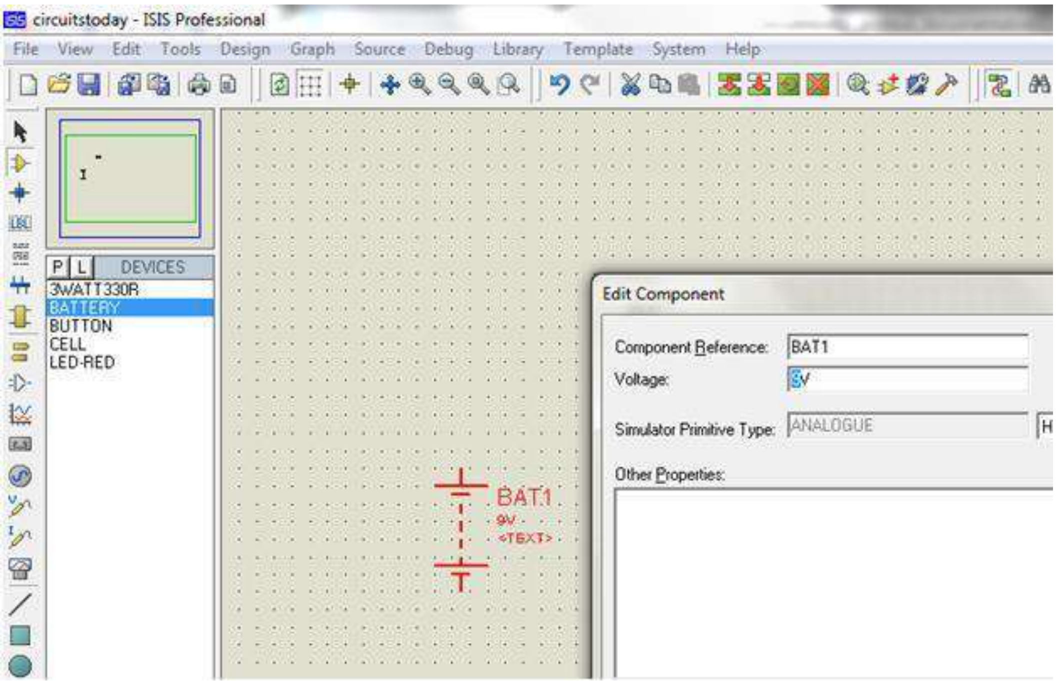


Fig. 3.2.7.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

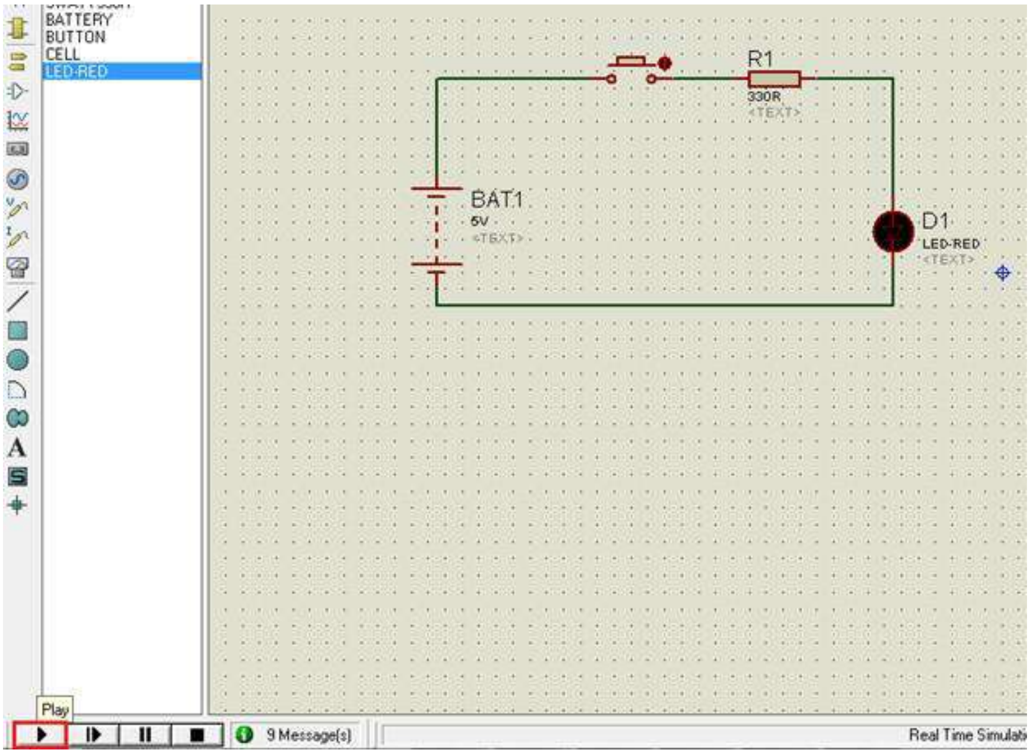


Fig. 3.2.7.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

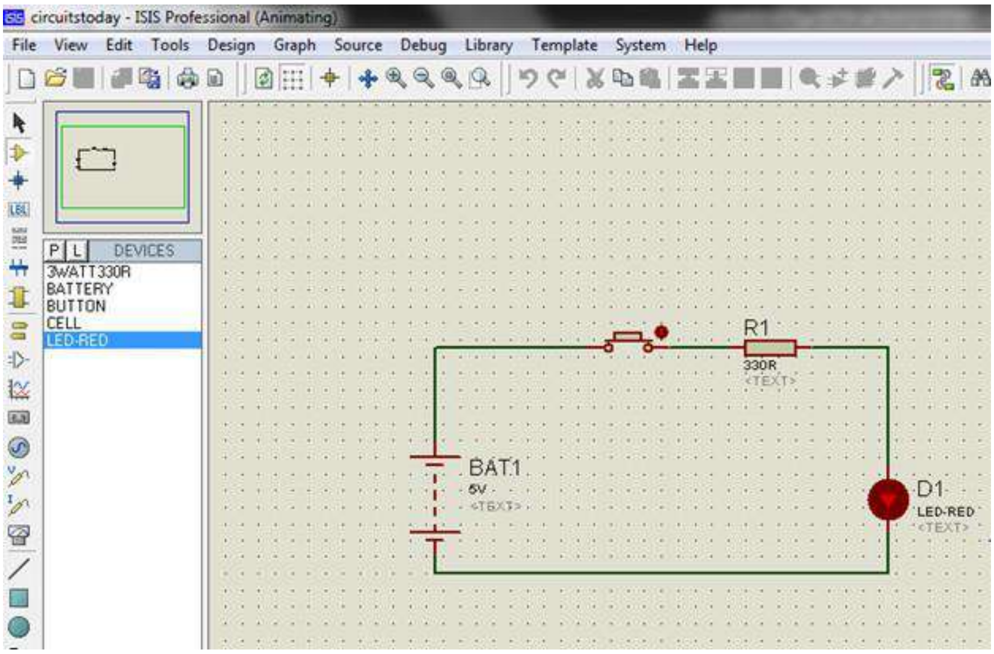


Fig. 3.2.7.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

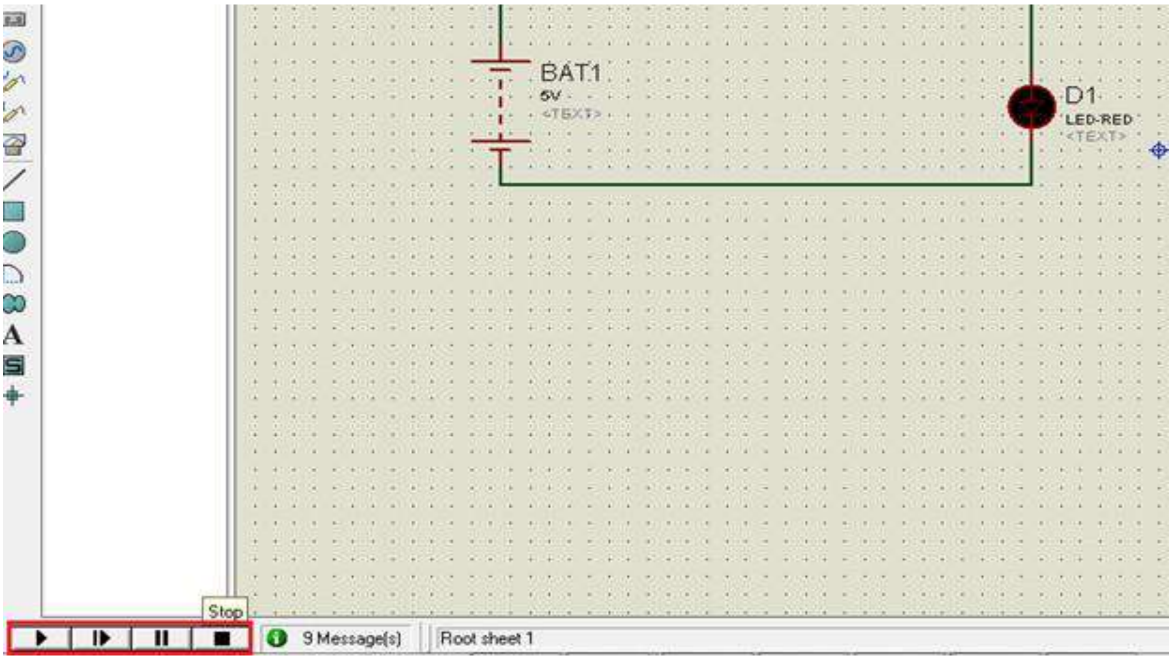


Fig. 3.2.7.13 Simulation Step-Pause-Stop Buttons

According to its developers, Things Speak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. Things Speak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates.

Things Speak was originally launched by iot Bridge in 2010 as a service in support of IoT applications.

Things Speak has integrated support from the numerical computing software MATLAB from MathWorks, allowing Things Speak users to analyse and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license from Math works.[citation needed].

Things Speak has a close relationship with Math works, Inc. In fact, all of the Things Speak documentation is incorporated into the Math works MATLAB documentation site and even enabling registered Math works user accounts as valid login credentials on the Things Speak website. The terms of service and privacy policy of ThingSpeak.com are between the agreeing user and Math works, Inc.

Setup Things Speak

Create a free MathWorks account or sign into Things Speak using an existing account.

Select the Things Speak channel you want your data to stream into. See Collect Data in a New Channel for help creating a new channel.

Record the following for the selected channel:

Channel ID, which is listed at the top of the channel view.

Write API key, which can be found on the API Keys tab of your channel view.

3.3 DEFINE THE MODULES:

3.3.1 GSM MODULE

GSM is a mobile communication modem; it is stands for global system for mobile communication (GSM). The Theory of GSM was developed at Bell Laboratories in 1970.Mobile communication system is used widely throughout the world. GSM is an open and digital cellular technology used for transmitting data and mobile voice operates at the 850MHz, 900MHz frequency bands. For communication purposes GSM system was developed as a digital system using time division multiple access technique. A GSM digitizes and reduces the data, then sends the data through a channel with two different streams of client data, each in its own stream of data the digital system can carry 64 kbps to 120 Mbps of data rates.

There are various cell sizes in a GSM system such as macro, micro and Pico cells. Each cell varies as per the implementation domain. There are various cell sizes in a GSM network namely macro, micro, Pico and umbrella cells. The coverage area of each cell varies according to the implementation of the environment.

SIM 800L:

SIM800L is a miniature cellular module which allows for GPRS transmission, sending and receiving SMS and making and receiving voice calls. Low cost and small footprint and quad band frequency support make this module perfect solution for any project that require long range connectivity. After connecting power module boots up, searches for cellular network and login automatically. On board LED displays connection state (no network coverage - fast blinking, logged in - slow blinking).

This module has two antennas included. First is made of wire (which solders directly to NET pin on PCB) - very useful in narrow places. Second - PCB antenna - with double sided tape and attached pigtail cable with IPX connector. This one has better performance and allows to put your module inside a metal case - as long the antenna is outside.

3.3.1.1 Specifications:

- Supply voltage: 3.8V - 4.2V
- Recommended supply voltage: 4V
- Power consumption:
  - Sleep mode < 2.0mA
  - Idle mode < 7.0mA
  - GSM transmission (avg): 350 mA
  - GSM transmission (peek): 2000mA
- Module size: 25 x 23 mm
- Interface: UART (max. 2.8V) and AT commands
- SIM card socket: micro-SIM (bottom side)
- Supported frequencies: Quad Band (850 / 950 / 1800 /1900 MHz)
- Antenna connector: IPX
- Status signalling: LED
- Working temperature range: -40 do + 85 ° C

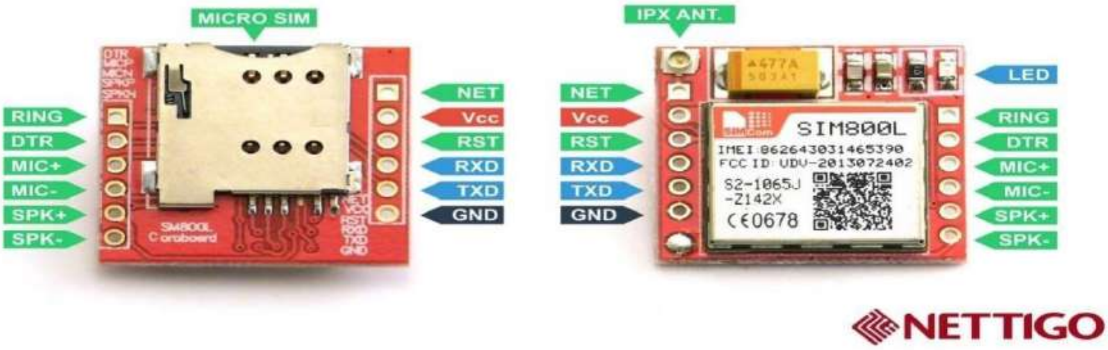


Fig.3.3.1: GSM Module

Pinout (bottom side - left):

- RING (not marked on PBC, first from top, square) - LOW state while receiving call
- DTR - sleep mode. Default in HIGH state (module in sleep mode, serial communication disabled). After setting it in LOW the module will wake up.
- MICP, MICN - microphone (P + / N -)
- SPKP, SPKN - speaker (P + / N -)

Pinout (bottom side - right):

- NET - antenna
- VCC - supply voltage
- RESET - reset
- RXD - serial communication
- TXD - serial communication
- GND – ground

3.3.1.2 GSM Commands:

Command	Description
AT+CMGD	DELETE SMS MESSAGE
AT+CMGF	SELECT SMS MESSAGE FORMAT
AT+CMGL	LIST SMS MESSAGES FROM PREFERRED STORE
AT+CMGR	READ SMS MESSAGE
AT+CMGS	SEND SMS MESSAGE
AT+CMGW	WRITE SMS MESSAGE TO MEMORY
AT+CMSS	SEND SMS MESSAGE FROM STORAGE
AT+CMGC	SEND SMS COMMAND
AT+CNMI	NEW SMS MESSAGE INDICATIONS
AT+CPMS	PREFERRED SMS MESSAGE STORAGE
AT+CRES	RESTORE SMS SETTINGS
AT+CSAS	SAVE SMS SETTINGS
AT+CSCA	SMS SERVICE CENTER ADDRESS
AT+CSCB	SELECT CELL BROADCAST SMS MESSAGES
AT+CSDH	SHOW SMS TEXT MODE PARAMETERS
AT+CSMP	SET SMS TEXT MODE PARAMETERS
AT+CSMS	SELECT MESSAGE SERVICE

Table 3.3.1: GSM Commands

CHAPTER-4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM OF PROJECT:

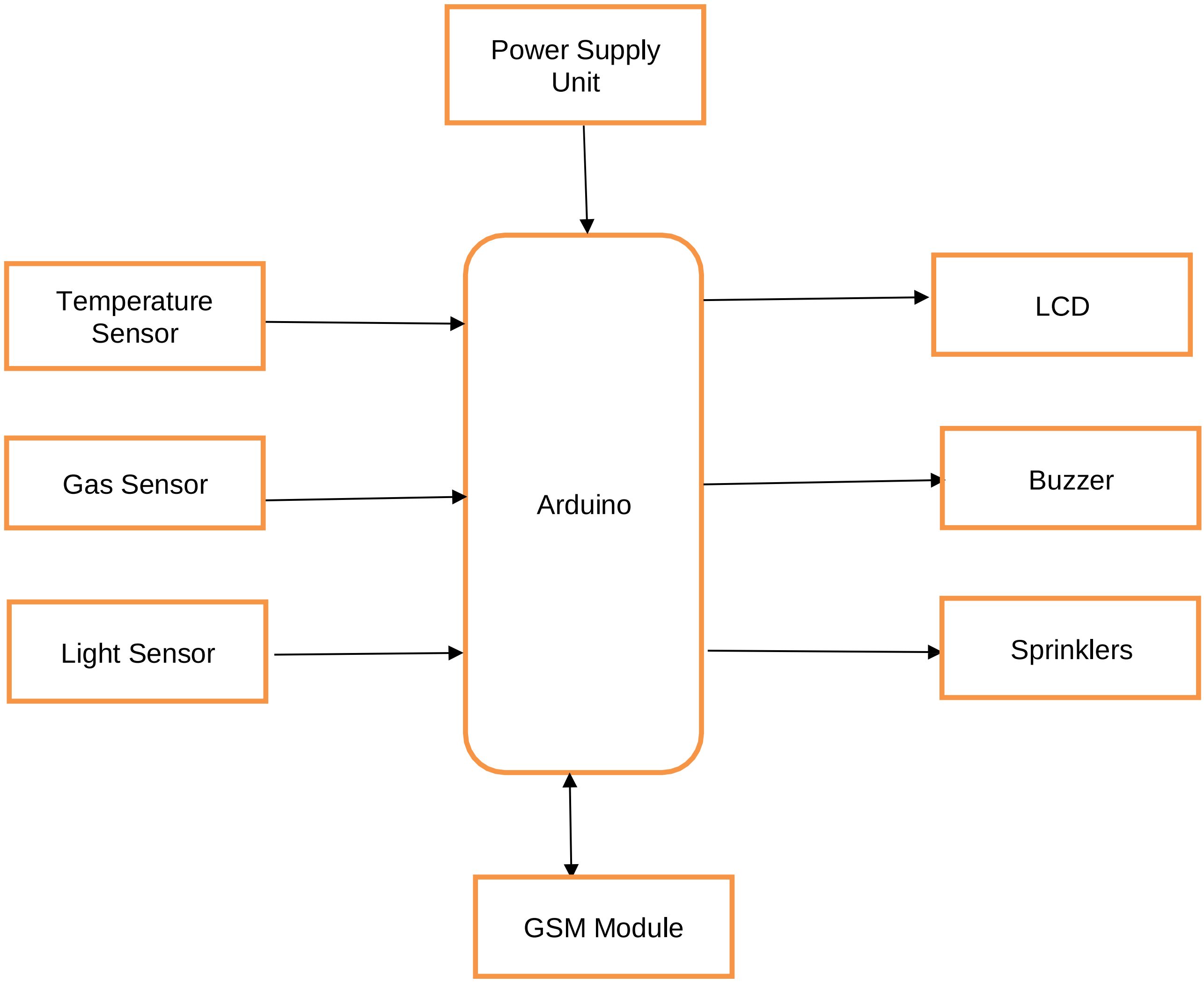


Fig.4.1: Block diagram of the project

4.2 FLOW CHART

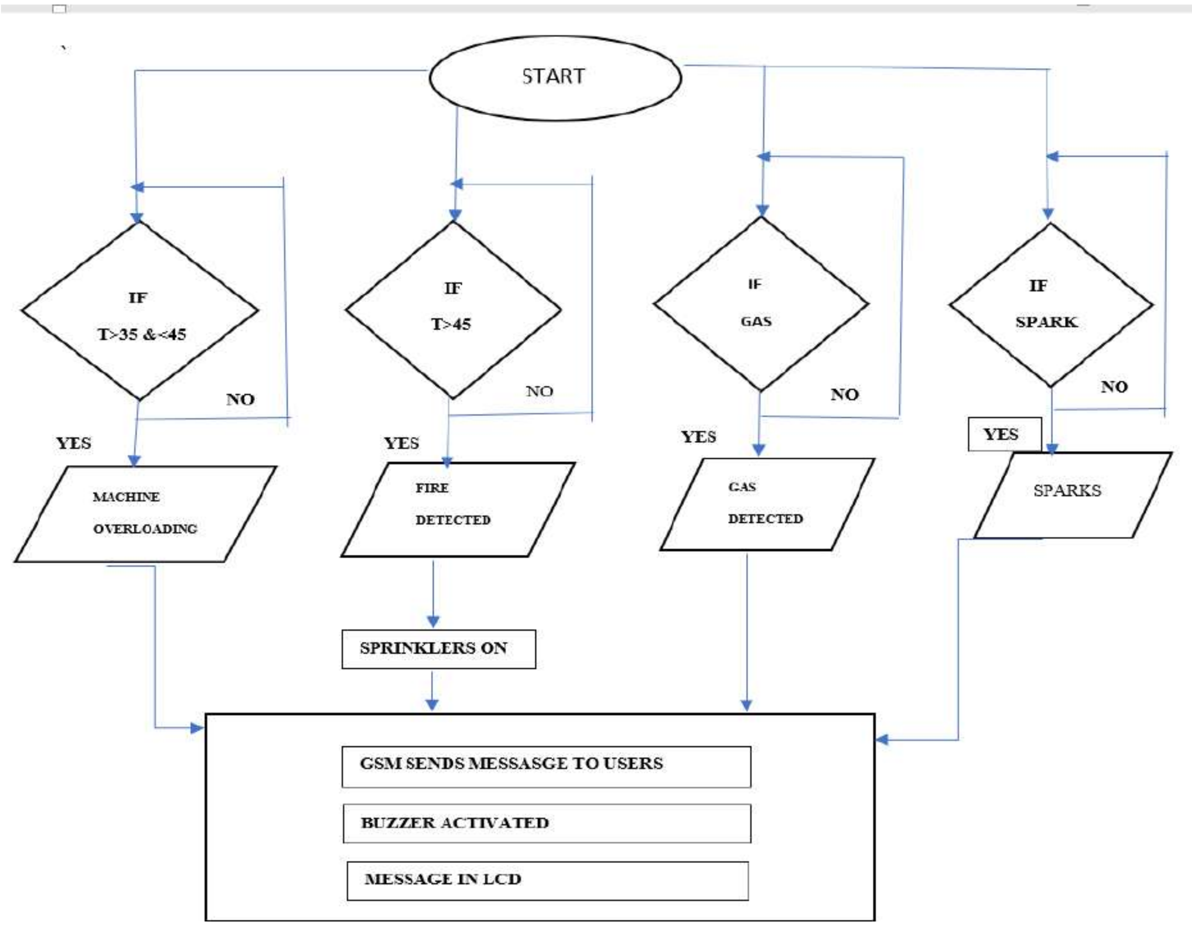


Fig.4.2 Flow Chart

4.3 IMPLEMENTATION STAGES

The following figures shows the output of proposed system in different test cases

Fig.4.3.1: Test case-1

The Fig.4.3.1 represents values of sensors. Since we have taken the outputs as active low here L:1 means the light sensor has not detected any sparks or light and it is in off state, similarly the G:1 means the gas sensor has not detected any gas and it is in off state. T:15 means the temperature detected by temperature sensor is 15.

Fig.4.3.2: Test case-2

The Fig.4.3.2 shows data displaying on LCD screen when machine is overloaded.

Fig.4.3.3: Test case-3

The Fig.4.3.3 shows data displaying on LCD screen when Sparks are detected. This tells us that once the sparks have been detected the light sensor detects the intensity of light and if it has crossed the threshold limit, this information will be passed to GSM and the message saying ‘SPARKS DETECTED’ will be sent to the authority and it will be displayed on the lcd to alert people in the respective area.

Fig.4.3.4: Test case-4

The Fig.4.3.4 shows data displaying on LCD screen when hazardous gas is detected. Once gas has been leaked in the particular area, if it has crossed the threshold limit it will be detected by the gas sensor and this information will be initiated by the GSM and a message

'GAS DETECTED' will be sent to authority and a message saying 'GSM INIT' and 'GAS DETECTED' will be displayed on lcd to alert people.

Fig.4.3.5: Test Case-5

The Fig.4.3.5 shows the conformation of the message sent. This figure shows us that the message has been sent regarding the accident taken place in the respective area under the surveillance of industry protection system.

4.4 PROJECT CODE:

```
#include <LiquidCrystal.h> //LCD
#include <SoftwareSerial.h> //COMMINCATION GSM MODULE

LiquidCrystal lcd(13,12,11,10,9,8);  //RS 13,EN 12, D4 11 ,D5 10,D6 9,D7 8
SoftwareSerial mySerial(2,3);//rx , tx  //GSM

int ldr = A3;
int temp=A4;
int gas = A5;
int  buz=7;
int relay=4;
int relay2=5;
void serialFlush ()
{
  while (Serial.available() > 0) {
    char tt = Serial.read();
  }
}
```

```
void sendmsg (char *num, char * msg)
{
  Serial.print("AT+CMGS=\"");
  Serial.print(num);
  Serial.println("\"");
  delay(800);
  Serial.println(msg);
  delay(800);
  Serial.write(0x1a);
  delay(2000);
}
const char* number = "9676107709\0"; /*Give the required mobile num to send the msg*/


void setup(void)
{

  Serial.begin(9600);
  mySerial.begin(9600);
  lcd.begin(16,2);
  lcd.clear();lcd.setCursor(0, 0);
  lcd.print("WELCOME");
  delay(1000);// 1st romw 0th colom
  pinMode(gas, INPUT);
  pinMode(temp, INPUT);
  pinMode(ldr, INPUT);
  pinMode(buz, OUTPUT);
  pinMode(relay, OUTPUT);
  pinMode(relay2, OUTPUT);
  digitalWrite (buz, HIGH); //OFF STATE
  Serial.println("AT"); delay(1000);
  Serial.println("AT+CMGF=1"); delay(1000);
  Serial.println("AT+CNMI=1,2,0,0"); delay(1000);
  Serial.println("AT+CSMP=17,167,0,16"); delay(1000);
```

```
  delay(1000);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("READY")
}
void loop()
{
    int gas_data = digitalRead(gas);
    int ld = digitalRead(ldr);
    int td = analogRead(temp)/4;

    Serial.println(gas_data);
    lcd.clear();
    lcd.setCursor(0, 0); lcd.print("GASDETECTION GSM ");
    delay(2000);
    lcd.clear();
    lcd.setCursor(0, 1); lcd.print("G:");lcd.print(gas_data);lcd.print("    ");
    lcd.setCursor(6, 1);lcd.print(" T:");lcd.print(td);
    delay(2000);

     if(gas_data == LOW)
     {
      lcd.clear();lcd.setCursor(0, 1);lcd.print("GAS DETECTED");
      mySerial.println("GAS DETECTED"); delay(500);
      lcd.clear();lcd.setCursor(0, 0);lcd.print("GSM INIT");
      Serial.println("GSM CONN... ");
      Serial.println("AT");delay(1000);
      Serial.println("AT+CMGF=1");delay(1000);
      Serial.println("AT+CNMI=1,2,0,0");delay(1000);
      Serial.println("AT+CSMP=17,167,0,0");delay(1000);
      lcd.clear();lcd.setCursor(0, 0);lcd.print("READY");
      Serial.println("GSM DONE");
      sendmsg(number,"GAS DETECTED");
      delay(3000);
      lcd.clear();lcd.setCursor(0, 0);lcd.print("MESSAGE SENT");
      digitalWrite(buz,LOW);
```

```
        delay(1000);
     }
  else
{
  digitalWrite(buz, HIGH);
  delay(500);
}
if(td > 35 &&td < 45)
{
  lcd.clear();
  lcd.setCursor(0, 0);lcd.print("MACHINE OVERLOAD");
  mySerial.println("MACHINE OVERLOAD"); delay(500);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("GSM INIT");
  Serial.println("GSM CONN. ..");
  Serial.println("AT");delay(1000);
  Serial.println("AT+CMGF=1");delay(1000);
  Serial.println("AT+CNMI=1,2,0,0");delay(1000);
  Serial.println("AT+CSMP=17,167,0,0");delay(1000);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("READY");
  Serial.println("GSM DONE");
       sendmsg(number,"MACHINE OVERLOAD");
       delay(3000);
       lcd.clear();lcd.setCursor(0, 0);lcd.print("MESSAGE SENT");
       digitalWrite(buz,LOW);
       //digitalWrite(relay2,HIGH);
        delay(1000);
     }
     else
{
  digitalWrite(buz,HIGH);
  digitalWrite(relay2,LOW);
  delay(500);
}
if(td > 45 )
```

```
{
 lcd.clear();
 lcd.setCursor(0, 0);lcd.print("FIRE OCCURED");
 mySerial.println("FIRE OCCURED"); delay(500);
 lcd.clear();lcd.setCursor(0, 0);lcd.print("GSM INIT");
 Serial.println("GSM CONN. ..");
 Serial.println("AT");delay(1000);
 Serial.println("AT+CMGF=1");delay(1000);
 Serial.println("AT+CNMI=1,2,0,0");delay(1000);
 Serial.println("AT+CSMP=17,167,0,0");delay(1000);
 lcd.clear();lcd.setCursor(0, 0);lcd.print("READY");
 Serial.println("GSM DONE");
      sendmsg(number,"FIRE OCCURED");
      delay(3000);
      lcd.clear();lcd.setCursor(0, 0);lcd.print("MESSAGE SENT");
      digitalWrite(buz,LOW);
      digitalWrite(relay2,HIGH);
       delay(1000);
     }
     if(ld== LOW)
     {
      lcd.clear();lcd.setCursor(0, 1);lcd.print("SPARKS DETECTED");
      mySerial.println("SPARKS DETECTED"); delay(500);
      lcd.clear();lcd.setCursor(0, 0);lcd.print("GSM INIT");
 Serial.println("GSM CONN. ..");
 Serial.println("AT");delay(1000);
 Serial.println("AT+CMGF=1");delay(1000);
 Serial.println("AT+CNMI=1,2,0,0");delay(1000);
 Serial.println("AT+CSMP=17,167,0,0");delay(1000);
 lcd.clear();lcd.setCursor(0, 0);lcd.print("READY");
 Serial.println("GSM DONE");
 sendmsg(number,"SPARKS DETECTED");
 delay(3000);
 lcd.clear();lcd.setCursor(0, 0);lcd.print("MESSAGE SENT");
```

```
 digitalWrite(buz,LOW);
   delay(1000);
     }
}
```

4.5 RESULTS

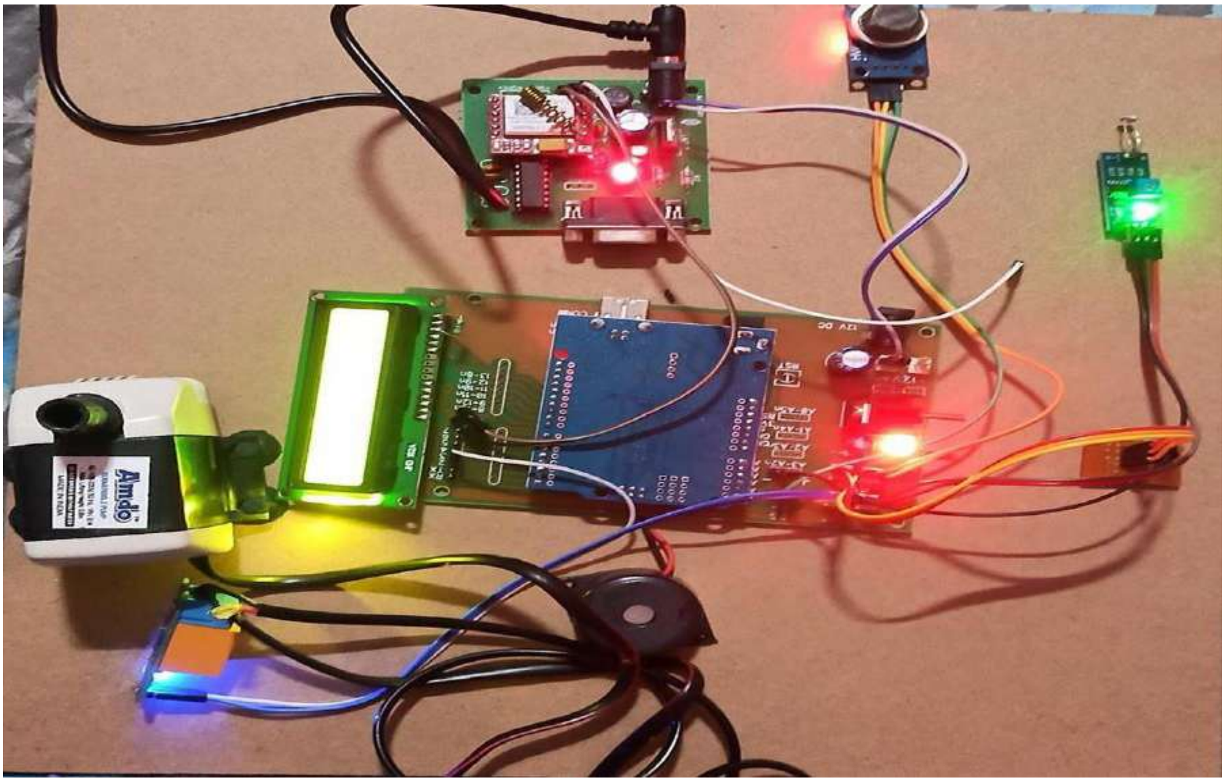


Fig.4.5 Final Output

The above figure is the circuit obtained after making the necessity connections on PCB board as per project requirements.

CHAPTER-5

PROJECT TESTING

5.1 OVERVIEW OF TESTING METHODS

- Security system in industry using ZigBee technology uses a ZigBee transmitter at the site location and another ZigBee receiver is placed at the control panel. Information received at the receiver will be sent to the PC. The security people will take appropriate action according to the problem failure like fire generated, gas leakage etc.
- But a drawback of this system is that the information can be passed only to few meters.
- The proposed system uses GSM module which enables us to know the security status of industry, by sending commands via SMS and this information can be sent to multiple members simultaneously.
- This system also uses sprinklers to put out fire in case of a fire accident.
- The project industrial protection system works on SMS sent by the user with the help of a GSM modem. The device works by comparing the readings of the gas sensor, temperature and light sensor with a threshold limit initially set by the programmer.
- As soon as the reading of the sensors reaches the threshold limit SMS would be generated to inform the concerned authorities regarding the failure or the signal of the industrial safety under threat.

CHAPTER-6

CONCLUSION AND FUTURE ENHANCEMENT

The main agenda of our project was to provide a safe environment for the workers several industrial accidents like fire and vibration and the incidents of unauthorized access can be resolved using our system A simple system to improve the standards is developed. It is a real time monitor able system developed with simple hardware which simplifies the possibility of error free security system. This system can be easily implemented with maximum reliability and high security with low cost,

- One can implement a few more sensors and connect it to the satellite as a global feature of this system.
- Adding more sensor to monitor other environmental parameters such as Co2, Pressure and Oxygen Sensor.
- In aircraft, navigation and military there is a great scope of this real -time system.
- It can also be implemented in hospitals or medical institutes for the research & study in "Effect of Weather on health and Diseases", hence to provide better precaution alerts.
- Voice announcement system can be added to indicate device conditions. We can add voice announcement system along with the buzzer so if there are hazardous parameters detected then respective voice message will be announced.

PUBLICATION

Submitted Paper in the Conference ICSMEC-21 with Paper ID (ICSMEC21-0018) and got Acceptance for the Paper.

REFERENCES

[1] ArbabWaheed Ahmad, Neem Jan, Saeed Iqbal and Chankil Lee. “Implementation of ZigBee-GSM based home security monitoring and remote-control system”. Research assistant, Hanyang university, Ansan, Korea. 978-61284-857 IEEE 2011

[2] A Alheraish. “Design and Implementation of home automation system”. King saud university, Riyadh, Saudi Arabia 00983063 IEEE 2004.

[3] Rozita Teymourzadeh, Salah Addin Ahmed, Kokwaichan and Mok vee hong. “Smart GSM based home automation system”. Faculty of Engineering Technology and Built environment UCSI university.

[4] R. Anandan, B. Karthik, Dr.T.V.U. kiran Kumar. “Wireless home and industrial automation security system using GSM”. Journal of global research in computer science. Volume 4, No.4, April 2013.

[5] M.Sravan Kumar, M. Mounika, L. Ramya Pavani, E. Ranadeep, B.Siddhartha, K.B.V.S.R. Subramanyam. “GSM based industrial security system”. ISSN(PRINT):2393-8374, (ONLINE):2394- 0697, VOLUME-2, ISSUE-5, 2015

[6] Onengiye M. Georgewill, Chukwunazo J. Ezeofor. “Design and implementation of SMS-based industrial/homes gas leakage monitoring and detection alarm System”. International Journal of Engineering Trends and Technology (IJETT)-Volume 35 Number 9-May 2016.

[7] KashmiraThul, PritiDhote, Ashwini Chokole, Samir Raipurkar. “GSM based industrial security system”. International Journal of Innovations in Engineering and Science, e-ISSN:2456-3463 Vol.2 No.4, 2017.

[8] E. Isa and N Sklavos. “Smart home Automation:GSM security system design and implementation”. Journal of Engineering Science and Technology Review 10(3)(2017) 170-174.

[9] LDR Sensor- P. Satya, Ravi Teja, A. Sai Srikar, V. Kushal and K. Srinivasan. "Photosensitive Security System for theft detection and control using GSM technology". National institute of Technology, Trichy, India.SPACES-2015.

[10] Temperature Sensor- T. Murugan, AzhaPeriasamy, S. Murugananad. “Embedded based industrial temperature monitoring system using GSM”. International Journal of Computer Applications (0975-8887) volume 58-No. 19, November 2012.

BIBILOGRAPHY

1. GSM: Switching, Services and Protocols. June 2001.

2. Micro mechatronics: Modelling, Analysis and Design with MATLAB, second edition.

3. Arduino and Scilab based products.

APPENDICES

A practical goal of industrial safety is to lighten the environmental impact on the manufacturing unit and each person and the role of the industrial safety professionals is to find leverage or opportunities for considerable improvement using practical effort.

The main aim of this proposed project is to design and implement a flexible, cost-effective and powerful GSM Based Industrial protection system. A GSM based Industrial Protection system is needed for the occupant's convenience and safety. This system helps you to detect burglary, leaking of harmful gas, smoke caused due to fire and after detecting suspicious activity, it sends an alarm message to the owner number as well as security personnel. The concerned person will take some action as soon as possible, by sending some commands to the Microcontroller unit through registered mobile and controls the industrial devices through Relays.

A

PROJECT REPORT

on

DETECTION OF DIGITAL IMAGE FORGERY USING FFT AND LOCAL FEATURES

Submitted by

Ms. D. Pavani	**(17K81A04K1)**
Ms. D. Sai Bhargavi	**(17K81A04K2)**
Mr. B. Harish	**(17K81A04J5)**

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under the Guidance of

Ms. R. ALEKYA, M.E

Assistant Professor

ST. MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)
Dhulapally, Secunderabad-500 100

2020-2021

ST. MARTIN'S ENGINEERING COLLEGE

UGC AUTONOMOUS

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

2020-2021

BONAFIDE CERTIFICATE

This is to certify that the project entitled "Detection of Digital Image Forgery using FFT and Local Features" is being submitted by **Ms. D. Pavani 17K81A04K1, Ms. D. SaiBhargavi 17K81A04K2, Mr. B. Harish 17K81A04J5** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

INTERNAL GUIDE **HEAD OF DEPARTMENT**

Ms. R. ALEKYA
M E, Assistant Professor

Dr. B. HARI KRISHNA
PH. D(ECE)

EXTERNAL EXAMINER

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT D. PAVANI WITH ROLL NO.17K81A04K1, D.SAI BHARGAVI WITH ROLL NO.17K81A04K2, B. HARISH WITH ROLL NO.17K81A04J5, OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "DETECTION OF DIGITAL IMAGE FORGERY USING FFT AND LOCAL FEATURES " AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. ★ Kompally-500014 ★

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of '**Bachelor of Technology in Department of Electronics and Communication Engineering**', session: 2017-2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project work entitled '**Detection of Digital Image Forgery using FFT and Local Features'** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

D. Pavani	**D. Saibhargavi**	**B. Harish**
17K81A04K1	**17K81A04K2**	**17K81A04J5**

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success**.**

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr. P. Santosh Kumar Patra**, our beloved **Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B. Hari Krishna**, **Head & Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our department **Project Coordinator, Dr. A. Anand Kumar, Professor** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come. We sincerely thank our **Class Project Coordinator, Ms. P. Pushpa,** Department of ECE and our project **Internal guide, Ms. R. Alekya, Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level **Project Coordinators Dr. Maalyadri**, **Professor in IT** and **Dr.R. Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. D. Pavani
2. D. Saibhargavi
3. B. Harish

ABSTRACT

Multimedia security is one of the key challenges in today's world, as dependency on multimedia information is increasing day by day. Easily available image editing software have enabled every common user of a smart phone and computer, to hack into the information of the images and video and alter it to some extent. To authenticate the genuineness of images, detection of image tempering is need of the time. Various techniques have been proposed to use image features for detection of image forgery. The techniques of forgery detection work in two domains of image forgery; copy-move forgery detection (CMFD) and image splicing detection (ISD). This paper presents a comprehensive comparative analysis for the use of local texture descriptors i.e. local binary pattern (LBP) and local ternary pattern (LTP) for forgery detection in an image. The project also presents a technique to integrate fast Fourier transform (FFT) with local texture descriptors for image forgery detection using existing block-based methodology. Performance of the technique(s) and descriptor(s) is tested for benchmarked dataset CASIA v1.0. Results are evaluated by using standard detection metrics detection accuracy and recall.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY OF COMMONLY USED TERMS

IMAGE: An image is represented as a two dimensional function f(x, y) where x and y are spatial co-ordinates and the amplitude of 'f' at any pair of coordinates (x, y) is called the intensity of the image at that point.

MATLAB: Matlab is a programming and numeric computing platform used millions of engineers and scientists to analyze the data.

DWT (Discrete wavelet transform): The **discrete wavelet transform** (**DWT**) is a computerized technique to compute fast wavelet transform of a signal. The **discrete wavelet transform** (**DWT**) is an optimum solution for computational time overhead.

BLOCK PROCESSING: Some image processing operations involve processing an image in sections called blocks, rather than processing an entire image at once. The input image is divided into blocks, each block calls the specified function and results are consolidated to form an output image.

FEATURE EXTRACTING: Feature extraction is a part of the dimensionality reduction process, in which, an initial set of the raw data is divided and reduced to more manageable groups. These features are easy to process, but still able to describe the actual data set with the accuracy and originality.

SVM (SUPPORT VECTOR MACHINE): Support vector machines (**SVMs**) are particular linear **classifiers** which are based on the margin maximization principle. They perform structural risk minimization, which improves the complexity of the **classifier** with the aim of achieving excellent generalization performance.

IMAGE SPLICNG FORGERY: Image splicing is an image editing method to copy a part of an image and paste it onto another image, and it is commonly followed by postprocessing such as local/global blurring, compression, and resizing.

COPY MOVE FORGERY: Copy-move forgery is a specific type of image tampering where a part of the image is copied and pasted somewhere else in the image with the intent to cover an important image feature.

CHAPTER 1

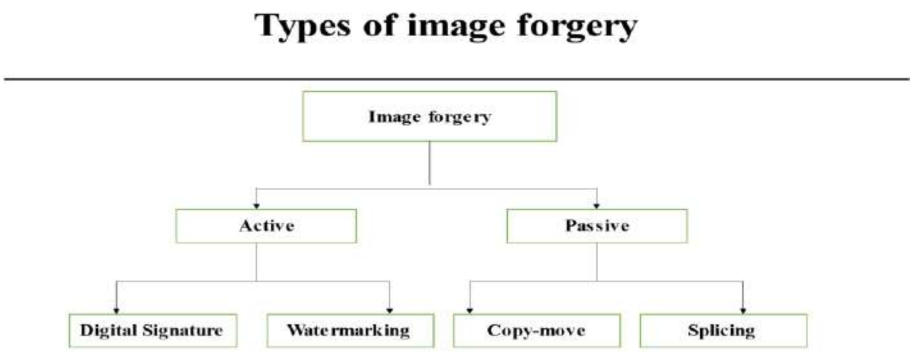
INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

Multimedia security is one of the key challenges in today's world, as dependency on multimedia information is increasing day by day. Easily available image editing software have enabled every common user of a smart phone and computer, to hack into the information of the images and video and alter it to some extent. To authenticate the genuineness of images, detection of image tempering is need of the time. Various techniques have been proposed to use image features for detection of image forgery. The techniques of forgery detection work in two domains of image forgery,copy-move forgery detection (CMFD) and image splicing detection (ISD). This project presents a comprehensive comparative analysis for the use of local texture descriptors i.e. local binary pattern (LBP) and local ternary pattern (LTP) for forgery detection in an image. The project also presents a technique to integrate fast fourier transform (FFT) with local texture descriptors for image forgery detection using existing block-based methodology. So, the main objective of the project is to detect the forged/tempered part of the image.

1.2 OBJECTIVES OF THE STUDY

Digital images are used in many fields as the basic source of information and for proofs. The process of creating a fake image has become a simple process by using existing technology and powerful editing software. Image splicing is a technique to combine number of multiple images into a single image. Copy move image is a technique to copy and paste the certain parts of image at different parts of image.



multi-image forgery in the images may be detected through analysis of local texture and LBP. It has been used by various researchers for detecting image forgery. Image splicing detection has been achieved in the Forgery Detection using LBP and LTP In first approach chrominance component is divided in overlapping blocks of size 3x3 and then two type of features are extracted from these blocks i.e. LBP and LTP.

1) Local Binary Pattern (LBP): LBP was initially proposed in for extracting local features of an image. LBP possesses a two valued code matrix .

$$LBP_{x,y} = \sum_{x=0}^{X-1} s(g_x - g_t)2^x$$
$$s(x) = \begin{cases} 1, if\, x \geq 0; \\ 0, otherwise \end{cases} \quad (1)$$

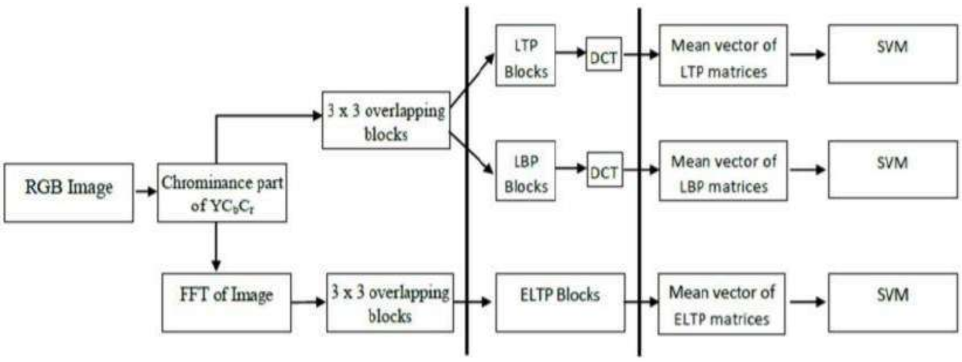


Fig. 1.1 Image splicing detection methodology

where ‘gx’ and ‘gt’ are the gray value of current neighborhood pixel and central pixel respectively. ‘X’ is the number of pixels in the block. A binary matrix is returned by the above equation which is then converted to decimal for convenience as shown in Figure 4.

2) Local Ternary Pattern (LTP): LBP is prone to random and quantization noise in near-uniform regions because its value depends center pixel value. Local ternary pattern (LTP) was proposed as an enhancement to LBP and generates 3- valued code by following equation 2. present work by following the methodology shown in Figure 1. RGB image is firstly converted into YCBCR format. Chrominance part of the converted image is used for further evaluation, as chrominance part include suitable information for forgery detection. Two different approaches have been suggested in the methodology for forgery detection.

$$s'(g_x, i_t, th) = \begin{cases} 1, & g_x \geq g_t + th \\ 0, & g_x - g_t < th \\ -1, & g_x \leq g_x - th \end{cases} \qquad (2)$$

where ‘th’ is the constant threshold value. DCT is applied on these feature matrices to generate feature coefficients (LBP/LTP). Mean for every DCT block is calculated for dimensionality reduction and generating the final feature vector. This vector is then fed to SVM classifier for classifying the image as authentic or forged.

A. Forgery Detection using FFT and ELTP (FFT-ELTP)

In the second approach, chrominance component is transformed using fast Fourier transform and then segmented into overlapping bocks. Instead of using LBP or LTP for feature extraction, the present paper evaluates extension of LTP as enhanced local ternary pattern (ELTP) [27] code for these blocks to generate a feature vector:

1) Fast Fourier Transform: Fast fourier transform (FFT) is an efficient way to compute discrete fourier transform (DFT). An image matrix is transformed to discrete fourier coefficients by formulation in equation 3.

$$Y_{p+1,q+1} = \sum_{j=0}^{m-1} \sum_{k=0}^{n-1} w_m^{jp} w_n^{kq} X_{j+1,k+1} \qquad (3)$$

where ‘wm’ and ‘wn’ are complex root of unity, defined as and respectively and ‘i' is the imaginary unit.

2) **Enhanced Local Ternary Pattern:** Yuan et. al have extended the LTP to attain better features for the image by introducing ELTP. The constant value in the LTP makes it less robust for gray level transformations. ELTP uses a dynamic threshold value based on the mean absolute deviation(mad) of the respective block. ELTP achieves significantly better results in texture classification in comparison to LBP and its variants. The generated feature vector in this approach will contain ELTP code for every FFT block of the image. This is fed to SVM classifier for classification of the image as authentic or forged.

1.3 SCOPE OF THE STUDY

The scope of this project focuses on proposing a forgery detection technique using both FFT and local features method, with both descriptor and feature matching methods, and its performance on accurately detecting tampered images. Apart from the proposed CMFD techniques an existing CMFD technique namely SIFT, which is used for feature extraction method.

The performance of the proposed FFT and local features method is evaluated using images which have underwent different geometrical attacks, namely translation, different degrees of rotation, symmetrical and asymmetrical scaling. The proposed technique is also tested on images with multiple copy-move regions. The proposed work is implemented using Matlab and tested using various datasets of images.

1.4 ORGANIZATION OF CHAPTERS

For proper presentation of the work, the project will be divided into five (5) chapters. Chapter one will cover the introduction which talks about the overview and scope of the study, objectives of the study. Chapter two will deal with review of related Literature on the topic and this is broken down into appropriate sub-topics. Chapter three will deal with the methodology the researcher employed which comprises of the research design for the study and the introduction to matlab software. Chapter four deals with the implementation stages, results of the project. Chapter five deals with the conclusion and future enhancement of the project.

1.4.1 INTRODUCTION

The project proposes a method of detection technology to detect the digital image forgery by using both FFT and Local Features. This proposed method has higher accuracy of detection of tampered part of image than the existing methods.

1.4.2 LITERATURE SURVEY

The survey is done on existing technology used for detection of image forgery and also the methods used for detection of image forgery.

1.4.3 PROJECT DESIGN

The project was designed by using existing block-based technology and DCT and DWT as main algorithm.

1.4.4 PROJECT IMPLEMENTATION

The project is implemented step by step process for obtaining final output

1.4.5 CONCLUSION AND FUTURE ENHANCEMENT

The proposed technique increase accuracy of detection of tampered part of image when compared to existing technique

CHAPTER 2

LITERATURE REVIEW

2.1 LITERATURE REVIEW ON RESEARCH AREA

In today's world, visual media is the primary source of communication. Visual media is paragon at explaining the situation at its best. Malicious modification of digital images with the intent to deceive for the sake of altering the public perception is termed as Digital Image Forgery. Forgery has a sole purpose of changing public perception. Forging an image takes just one thing into consideration that people usually believe what they see. Decisions can be easily manipulated by altering the images with user-friendly and easily available image editing tools like Sumopaint, Paintshop Pro, Photoshop CC and HitFilm Express. These easily available tools make forgery no longer restricted to specialists. Manipulation is done so precisely that it hardly leaves any visible traces. An observer cannot sense manipulation with naked eye and needs some scientific methods to detect forgery in the image. Image is commonly manipulated with two basic operations of copy-paste or image splicing. Image splicing is very basic and harmful type of forgery. Image Splicing creates a composite image by blending cropped regions to same or different image. Some post processing operations like blurring are performed after pasting the region to completely merge the pasted portion with the background. Figure 1 highlights the process of image splicing and Figure 2 elaborated an example of the same. Image forgery detection can be active or passive. An active approach makes use of prior information, embedded into the image to verify its authenticity. Such information is embedded using digital watermark or digital signatures. Passive approaches are prioritized over the active approach as they do not need any prior information but the image itself. These approaches have no knowledge about the origin of the image. As tampering is not visually detectable, passive approach analyses the underlying statistical characteristic of the image which are supposed to be definitely disturbed by the manipulation process. Image forensics predominantly uses passive approach to verify image authenticity. This paper discusses techniques to detect image splicing forgery.

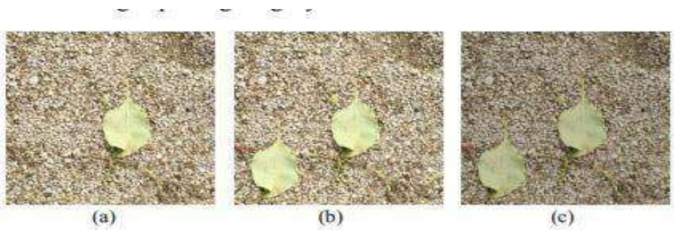


Fig. 2.0. (a) Original Image (b) Forged Image (c) Image Retouching

Fig. 2.1 Image Splicing by using two different images

Taking into account the fact that pasting of a cropped portion onto a different image, replaces the existing micro edge patterns of the host image with its own patterns, which make the pasted portion different from the rest of the image and such disturbance is peculiar along its boundary, visual descriptor LBP and its variants are used to encode the micro edge patterns and DCT is used to encode the frequency content with respect to each blocks LBP value. This approach utilizes the chrominance component of the RGB image which is believed to capture the disturbance created by forgery better than any other colour channel. A technique of forgery detection is presented that uses fast fourier transform of the chrominance component for extracting local features of the image. SVM Classifier is used for classifying the images as forged or authentic.

2.2 REVIEW ON RELATED LITERATURE

A detection technique based on the combination of LBP, discrete wavelet transforms (DWT) and principal component analysis (PCA) was proposed with support vector machine (SVM). The combination of LBP and DWT asserted for the improvement in accuracy rate in CASIAv1.0 is 97.21% and in case of Columbia dataset is 95.13%. Another method which was also based on LBP and DCT. This method was suggested to detect copy move and image splicing forgery. It used DCT to convert the LBP image into its frequency domain, in-order to discover any chances of tampering more precisely. After that, for each block statistical measures of computed DCT coefficients are calculated. Such methods train the SVM classifier to differentiate between the authentic and forged images. This method yielded the best precision on CASIAv1.0 as 97%, CASIAv2.0 as 97.50% and CISDE dataset as 97.77%. had proposed a technique to detect image splicing forgery. This technique was imitation of the magnitude component of 2D arrays which were acquired by applying multi-size block discrete cosine transform (MBDCT) on testing images, using LBP. The dimensionality of feature set was deducted by using the kernel principal component analysis (K-PCA), to ensure it to be more efficient mathematically. Using this reduced feature set, SVM was able to classify the authentic images and tampered images. This method attained a precision of 90.46%. A tampering detection technique is proposed in using multi- scale LBP and DCT. This technique divides the image into non-overlapping blocks of various sizes 32*32, 64*64, 16*16. After dividing in the blocks, they were passed to DCT to extract the coefficients. In-order to build a feature set of the image, standard deviation was calculated with respect to computed coefficients. In the proposed method, the classifier was trained by using the SVM along with radial basis function (RBF) kernel. It attained the precision rate on CASIAv1.0 96% and on CASIAv2.0 97.3%. Other research paper put forward another method to determine the image authenticity. It this method, Markov features were extracted discretely in all the three domains. It was the first method which combined three domains, they were – spatial domain, DCT and DWT domain. Efficient classifier used in the proposed method, reduced the computational complexity as well as yield an efficient TPR, TNR and precision rate without making use of PCA. It attained a precision rate of 99.80% on Columbia Image Splicing Detection Evaluation Dataset (CISDE).

Another method to detect image splicing forgery, by using Markov features in QDCT domain was developed. It took RGB image without converting into grey scale to avoid any color distortion and it also resulted in improved precision. Markov feature were extracted from quaternion discrete cosine transform (QDCT) frequency domain of the blocked color image. SVM classifier was trained by these extracted features. This method enhanced the precision level to 92.38%. proposed a method to discover whether an image was authentic or tampered, based on spatial and DCT based Markov features. First, Markov features were extracted from spatial and DCT domains and then they were combined. In order to reduce the dimensionality, the most relevant features were extracted from combined feature set, by using a PCA.

SVM along with RBF kernel method was used, to optimize the classification process. This method was determined to be more than 98% precise, when assessed on Columbia Image Splicing Detection Evaluation Dataset (CISDE). One more method to detect image splicing forgery proposed in [10] deploy quantization- based Markov feature extraction. The performance was enhanced by reducing the loss of information, so quantization is used in this method. Two Markov feature selection method made use of the summation and maximization of the color feature. The proposed method had been tested on CASIAv1.0, CASIAv2.0 and Columbia color datasets and the results were precise to be 98.95%, 97.25% and 95.24% respectively used a noise level evaluation method which was intensity dependent, to determine image splicing forgery in the digital image. Here, the base used for splicing forgery detection was a variable noise level which vary depending on image taken from different sources. This method was outstanding, as it was the only method with varying noise level of blocks taken from different image sources and captured by the same camera. For localization of image splicing, [28] had proposed another method to highlight forged region in the image. This method utilized previous noise level estimation algorithm and PCA based algorithms, in order to determine block wise noise level of a testing image. K-means clustering was used to differentiate between the spliced region and the original region. When difference between spliced region and original region was small noise, then this method attained high performance. Contrary, when the spliced region and the original region had same noise level, then this method failed to localize the tampering.

2.3 PROPOSED METHOD

The project has presented two approaches for detecting image splicing. Both the approaches use overlapping blocks to extract image features. First approach extracts LBP or LTP features based on gray values of the image chrominance whereas second approach extract the ELTP features from fast fourier transform of the chrominance channel. Results of these techniques have been presented in comparative manner. It can be observed that LBP and ELTP perform as better features to classify the image as forged or authentic. ELTP in particular comes out to be a significantly better feature in comparison to existing image splicing detection techniques. The FFT-ELTP technique perform fairly by achieving an accuracy of 88.62% on compressed images of CASIAv1.0 dataset. However, all of the presented approaches involve complex transformations like DCT and FFT which increases the complexity of the methodology.

CHAPTER-3

PROJECT DESIGN

3.1 OVERVIEW OF THE DESIGN

The image that has to be tested for genuiness is given as the input image. The input image can be either RGB or grey image. The given input image undergoes preprocessing part which involves conversion of RGB image into chrominance part of image and then block processing is applied to divide the image into 3*3 or 4* 4 overlapping blocks. Then feature extraction of image is done by applying DWT and DCT algorithm and then the output is given to the image classifier SVM(support vector machine).The mentioned process is shown in the below figure.

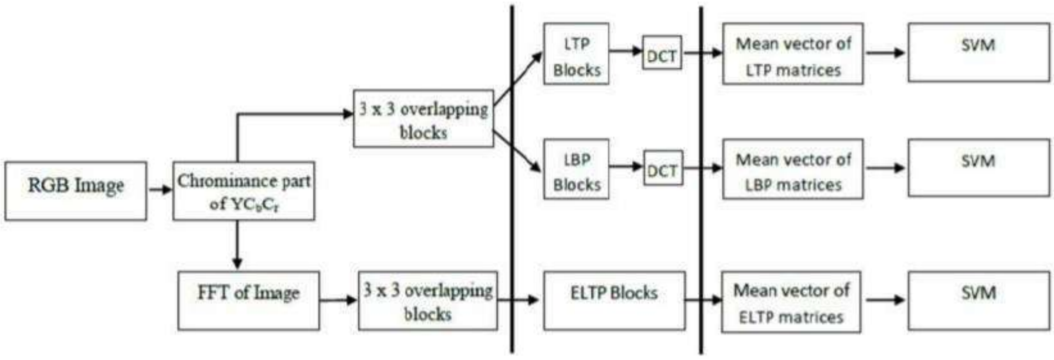


Fig 3.1 Overview of the Design

3.2 INTRODUCTION TO IMAGE PROCESSING

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces. The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by combination of methods, especially in a pseudo-photograph.

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

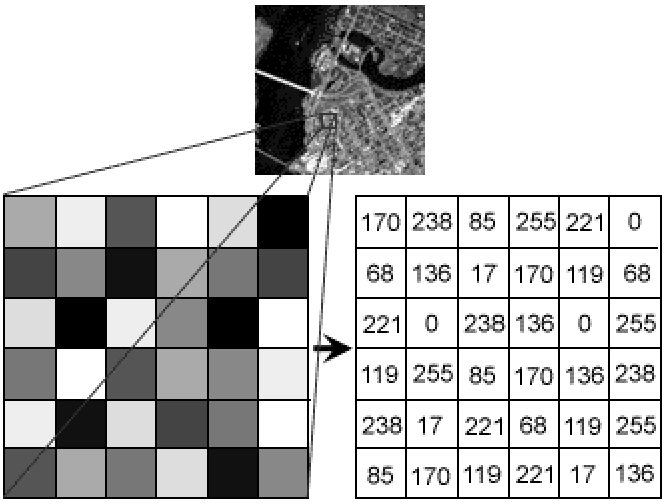


Fig 3.1 Image pixel

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

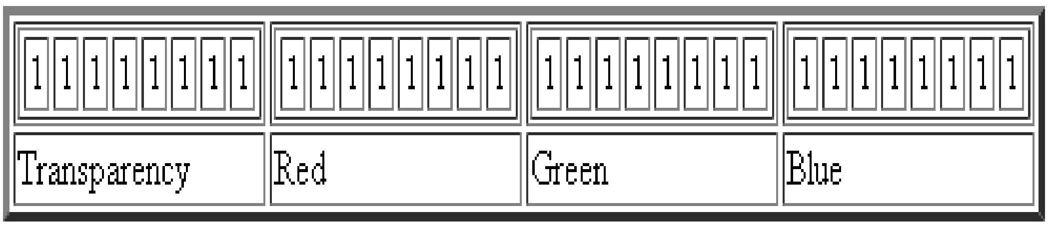


Fig 3.2 Transparency image

3.3 IMAGE FILE SIZES

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

3.4 IMAGE FILE FORMATS

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

3.4.1 RASTER FORMATS

These formats store images as bitmaps (also known as pixmaps)

- **JPEG/JFIF**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

- **EXIF**

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

- **TIFF**

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

- **PNG**

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

- **GIF**

GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

- **BMP**

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

3.4.2 VECTOR FORMATS

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

- **SVG**

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

3.5 TYPES OF IMAGES

Binary Images

Binary images have a very specific meaning in MATLAB.A binary image is a logical array 0s and1s.Thus, an array of 0s and 1s whose values are of data class, say unit8, is not considered as a binary image in MATLAB .A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, we create an array B using the statement.

B=logical (A)

If A contains elements other than 0s and 1s.Use of the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.

Using relational and logical operators also creates logical arrays.

To test if an array is logical we use the I logical function: islogical(c).

If c is a logical array, this function returns a 1. Otherwise returns a 0. Logical array can be converted to numeric arrays using the data class conversion functions.

Indexed Images

An indexed image has two components:

A data matrix integer, x

A color map matrix, map

Matrix map is an m*3 arrays of class double containing floating point values in the range [0, 1].The length m of the map are equal to the number of colors it defines. Each row of map specifies the red, green and blue components of a single color. An indexed images uses “direct mapping” of pixel intensity values color map values. The color of each pixel is determined by using the corresponding value the integer matrix x as a pointer in to map. If x is of class double ,then all of its components with values less than or equal to 1 point to the first row in map, all components with value 2 point to the second row and so on. If x is of class units or unit 16, then all components value 0 point to the first row in map, all components with value 1 point to the second and so on.

RGB Image

An RGB color image is an M*N*3 array of color pixels where each color pixel is triplet corresponding to the red, green and blue components of an RGB image, at a specific spatial location. An RGB image may be viewed as “stack” of three gray scale images that when fed in to the red, green and blue inputs of a color monitor

Produce a color image on the screen. Convention the three images forming an RGB color image are referred to as the red, green and blue components images. The data class of the components images determines their range of values. If an RGB image is of class double the range of values is [0, 1].

Similarly the range of values is [0,255] or [0, 65535].For RGB images of class units or unit 16 respectively. The number of bits use to represents the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8bit image, the corresponding RGB image is said to be 24 bits deep.

Generally, the number of bits in all component images is the same. In this case the number of possible color in an RGB image is (2^b) ^3, where b is a number of bits in each component image. For the 8bit case the number is 16,777,216 colors.

3.6 IMAGE PROCESSING

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

3.6 FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING

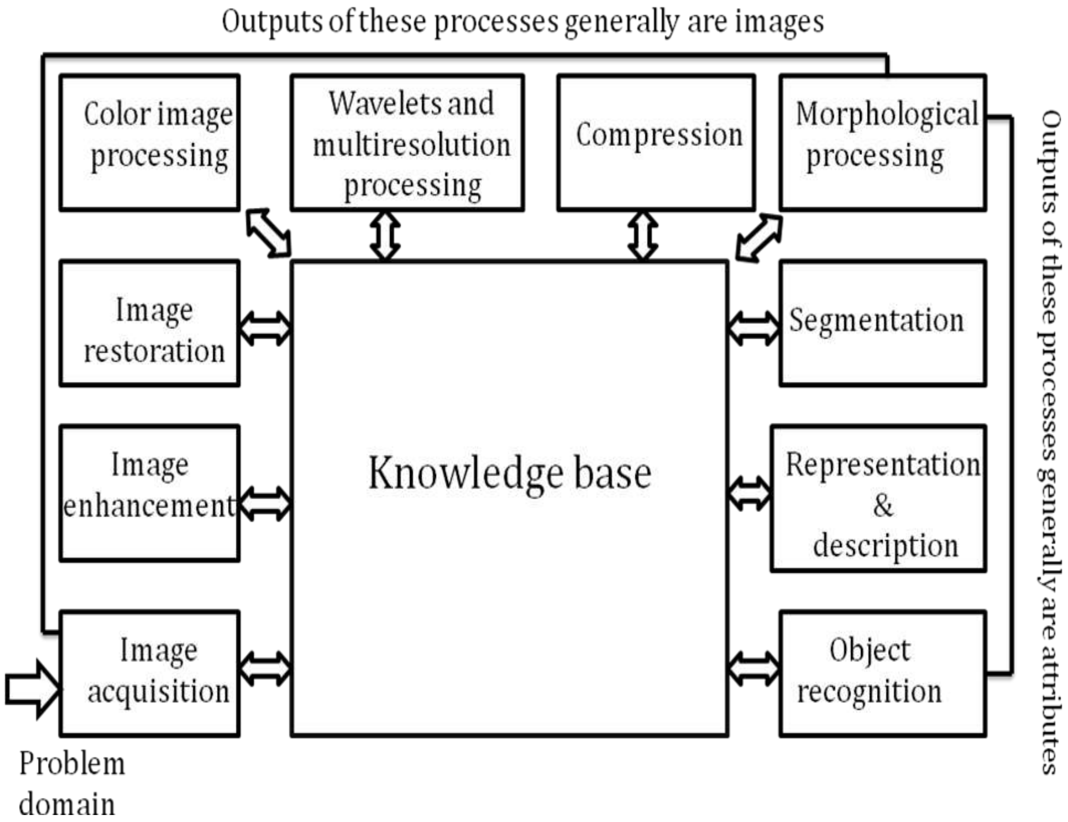


Fig 3.3 Image fundamental

3.6.1 IMAGE ACQUISITION

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application

3.6.2 IMAGE ENHANCEMENT

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because "it looks better." It is important to keep in mind that enhancement is a very subjective area of image processing.

3.6.3 IMAGE RESTORATION

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a "good" enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, where as removal of image blur by applying a deblurring function is considered a restoration technique.

3.6.4 COLOR IMAGE PROCESSING

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

3.6.5 WAVELETS AND MULTIRESOLUTION PROCESSING

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform-based image processing since the late1950's, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called **Multiresolution** theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

3.6.6 COMPRESSION

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image cmpression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

3.6.7 MORPHOLOGICAL PROCESSING

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

3.6.8 SEGMENTATION

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

3.6.9 REPRESENTATION AND DESCRIPTION

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

3.6.10 OBJECT RECOGNITION

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

3.6.11 KNOWLEDGEBASE

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an inter related to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented pre-processing techniques using MATLAB.

3.7 INTRODUCTION TO SOFTWARE: MATLAB (2013rb version)

3.7.1. INTRODUCTION TO MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

3.7.2 THE MATLAB SYSTEM

The MATLAB system consists of five main parts

- **DEVELOPMENT ENVIRONMENT**

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **THE MATLAB MATHEMATICAL FUNCTION LIBRARY**

  This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **THE MATLAB LANGUAGE**

  This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

- **GRAPHICS**

  MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **THE MATLAB APPLICATION PROGRAM INTERFACE (API)**

  This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

3.7.3 GRAPHICAL USER INTERFACE (GUI)

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in

M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.
- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type

guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted,

GUIDE opens a new (i.e., blank) window.

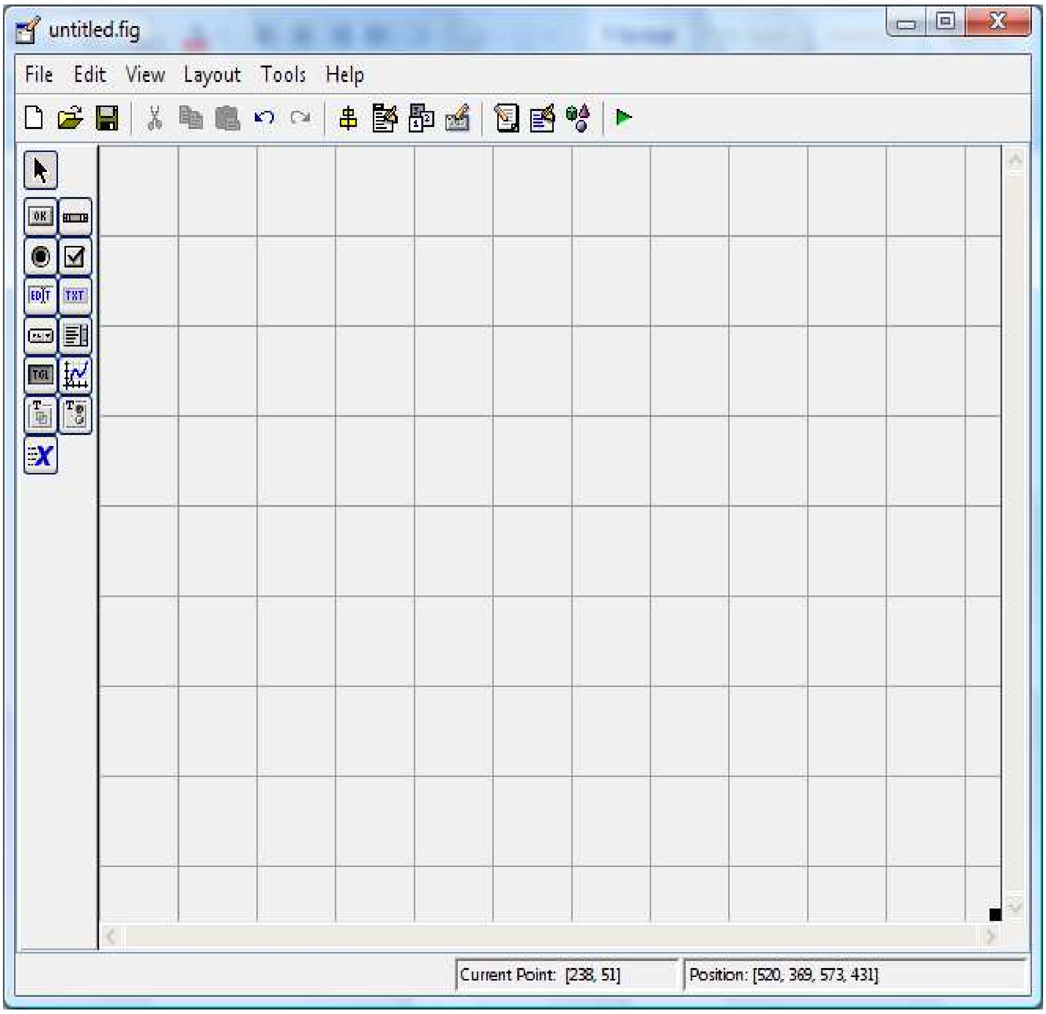


Fig 3.4 GUI window

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

3.8 GETTING STARTED

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

- **Introduction** - describes the components of the MATLAB system.
- **Development Environment** - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.
- **Manipulating Matrices** - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.
- **Graphics** - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.
- **Programming with MATLAB** - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

3.9 DEVELOPMENT ENVIRONMENT

3.9.1 INTRODUCTION

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

STARTING AND QUITTING MATLAB

3.9.2 STARTING MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

3.9.3 QUITTING MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

3.9.4 MATLAB DESKTOP

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

3.9.5 DESKTOP TOOLS

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

COMMAND WINDOW

Use the Command Window to enter variables and run functions and M-files.

COMMAND HISTORY

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

RUNNING EXTERNAL PROGRAMS

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

LAUNCH PAD

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

HELP BROWSER

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type helpbrowser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

HELP NAVIGATOR

Use to Help Navigator to find information. It includes:

PRODUCT FILTER - Set the filter to show documentation only for the products you specify.

CONTENTS TAB - View the titles and tables of contents of documentation for your products.

INDEX TAB - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

SEARCH TAB - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

FAVORITES TAB - View a list of documents you previously designated as favorites.

DISPLAY PANE

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

BROWSE TO OTHER PAGES - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

BOOKMARK PAGES - Click the Add to Favorites button in the toolbar.

PRINT PAGES - Click the print button in the toolbar.

FIND A TERM IN THE PAGE - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

CURRENT DIRECTORY BROWSER

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

SEARCH PATH

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

WORKSPACE BROWSER

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

ARRAY EDITOR

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

EDITOR/DEBUGGER

Use the Editor/Debugger to create and debug M-files, which are programs you write to run MATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

3.10 MANIPULATING MATRICES

3.10.1 ENTERING MATRICES

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.

- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Dürer's matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

MATLAB displays the matrix you just entered.

```
A =

    16     3     2    13

     5    10    11     8

     9     6     7    12

     4    15    14     1
```

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

3.10.2 EXPRESSIONS

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables

- Numbers
- Operators
- Functions

VARIABLES

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

NUMBERS

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

3.10.3 OPERATORS

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

Table 3.1 Expression of Arithmetic Operations in MATLAB

3.10.4 FUNCTIONS

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.14159265...
I	Imaginary unit, √-1
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2- \varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

Table 3.2 Expressions of Functions in MATLAB

3.11 GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include.

3.11.1 CREATING GUIs WITH GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

3.11.2 GUI DEVELOPMENT ENVIRONMENT

The process of implementing a GUI involves two basic tasks.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

THE IMPLEMENTATION OF A GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of itschildren (uicontrols and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and thecallbacks, which are defined as subfunctions. This M-file is referred to as the application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

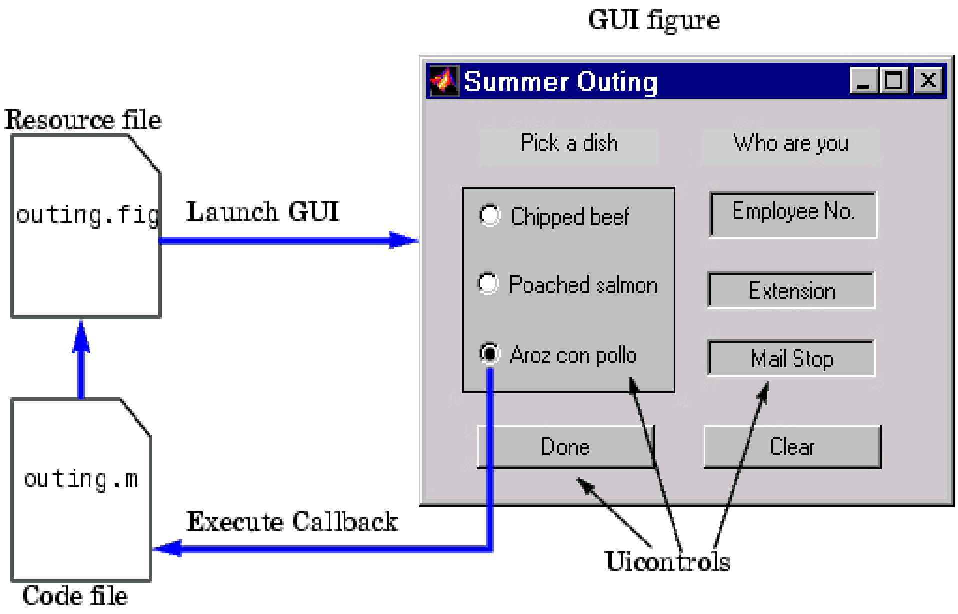


Fig 3.5 Graphical User Blocks

3.11.3 FEATURES OF THE GUIDE-GENERATED APPLICATION M-FILE

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

3.11.4 BEGINNING THE IMPLEMENTATION PROCESS

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques used in the application M-file.

Application Examples - a collection of examples that illustrate techniques which are useful for implementing GUIs.

COMMAND-LINE ACCESSIBILITY

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

3.11.5 USER INTERFACE CONTROL

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- Listboxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

PUSH BUTTONS

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

PROPERTIES TO SET

String - set this property to the character string you want displayed on the push button.

TAG - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

PROGRAMMING THE CALLBACK

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

TOGGLE BUTTONS

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

PROGRAMMING THE CALLBACK

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

FROM THE GUIDE APPLICATION M-FILE

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)

button_state = get(h,'Value');

if button_state == get(h,'Max')

   % toggle button is pressed

elseif button_state == get(h,'Min')

   % toggle button is not pressed
```

end

ADDING AN IMAGE TO A PUSH BUTTON OR TOGGLE BUTTON

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);

a(:,:,2) = rand(16,128);

a(:,:,3) = rand(16,128);

set(h,'CData',a)
```

RADIO BUTTONS

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

IMPLEMENTING MUTUALLY EXCLUSIVE BEHAVIOR

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)

set(off,'Value',0)
```

OBTAINING THE RADIO BUTTON HANDLES

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)

off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];

mutual_exclude(off)

% Continue with callback

   .

   .

   .
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

CHECKBOXES

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)
if (get(h,'Value') == get(h,'Max'))
   % then checkbox is checked-take approriate action
else
   % checkbox is not checked-take approriate action
end
```

EDIT TEXT

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)
user_string = get(h,'string');
% proceed with callback...
```

OBTAINING NUMERIC DATA FROM AN EDIT TEST COMPONENT

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)

user_entry = str2double(get(h,'string'));

if isnan(user_entry)

   errordlg('You must enter a numeric value','Bad Input','modal')

end

% proceed with callback...
```

TRIGGERING CALLBACK EXECUTION

On UNIX systems, clicking on the menubar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menubar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

STATIC TEXT

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

FRAMES

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uncontrol's can appear within frames (axes cannot).

PLACING COMPONENTS ON TOP OF FRAMES

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

LIST BOXES

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer

SINGLE OR MULTIPLE SELECTION

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

SELECTION TYPE

Listboxes differentiate between single and double clicks on an item and set the figure SelectionType property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

TRIGGERING CALLBACK EXECUTION

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure SelectionType property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

POPUP MENUS

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

PROGRAMMING THE POPUP MENU

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

switch val

case 1

% The user selected the first item

case 2

% The user selected the second item

% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

string_list = get(h,'String');
```

selected_string = string_list{val}; % convert from cell array to string

% etc.

ENABLING OR DISABLING CONTROLS

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is

grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

AXES

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

AXES CALLBACKS

Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes ButtonDownFcn property to define the callback.

3.11.6 PLOTTING TO AXES IN GUIS

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIS WITH MULTIPLE AXES

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

```
axes(handles.axes1)
```

makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axes. See GUI with Multiple Axes for and example that uses two axes.

FIGURE

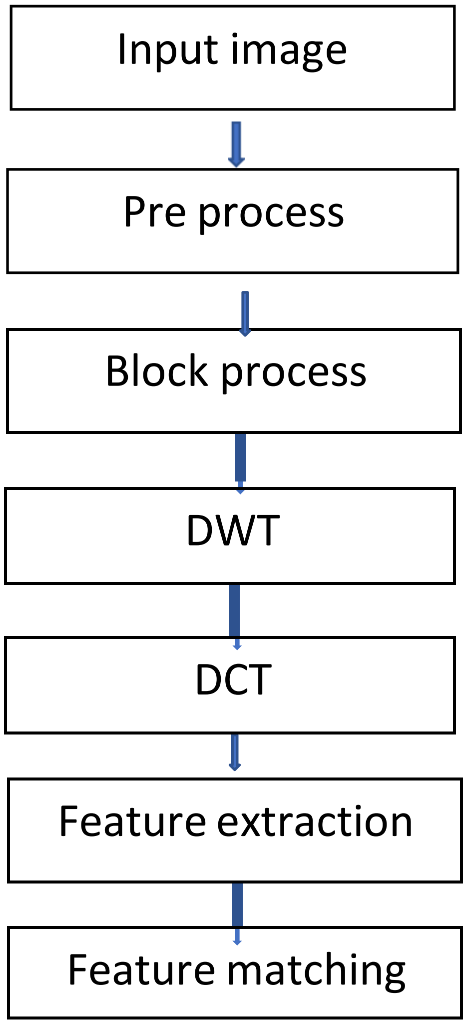
Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control

CHAPTER 4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM OF PROJECT

The basic procedure that every image should go through during image forgery detection is shown the following block diagram. The first step includes preprocessing which is used for image enhancement and followed by block process to divide the image into 3*3 or 4*4 overlapping blocks and then DWT and DCT algorithm is applied followed by feature extraction and feature matching is done and post processing is done then output image is given to image classifier to detect the tampered part and the displayed as final output. This basic process is shown in following block diagram



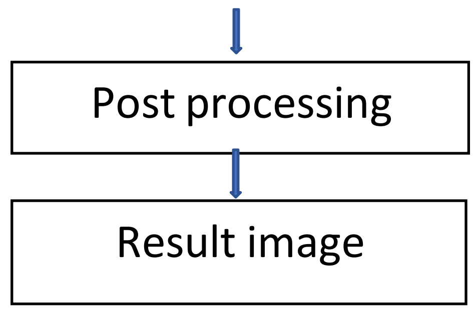
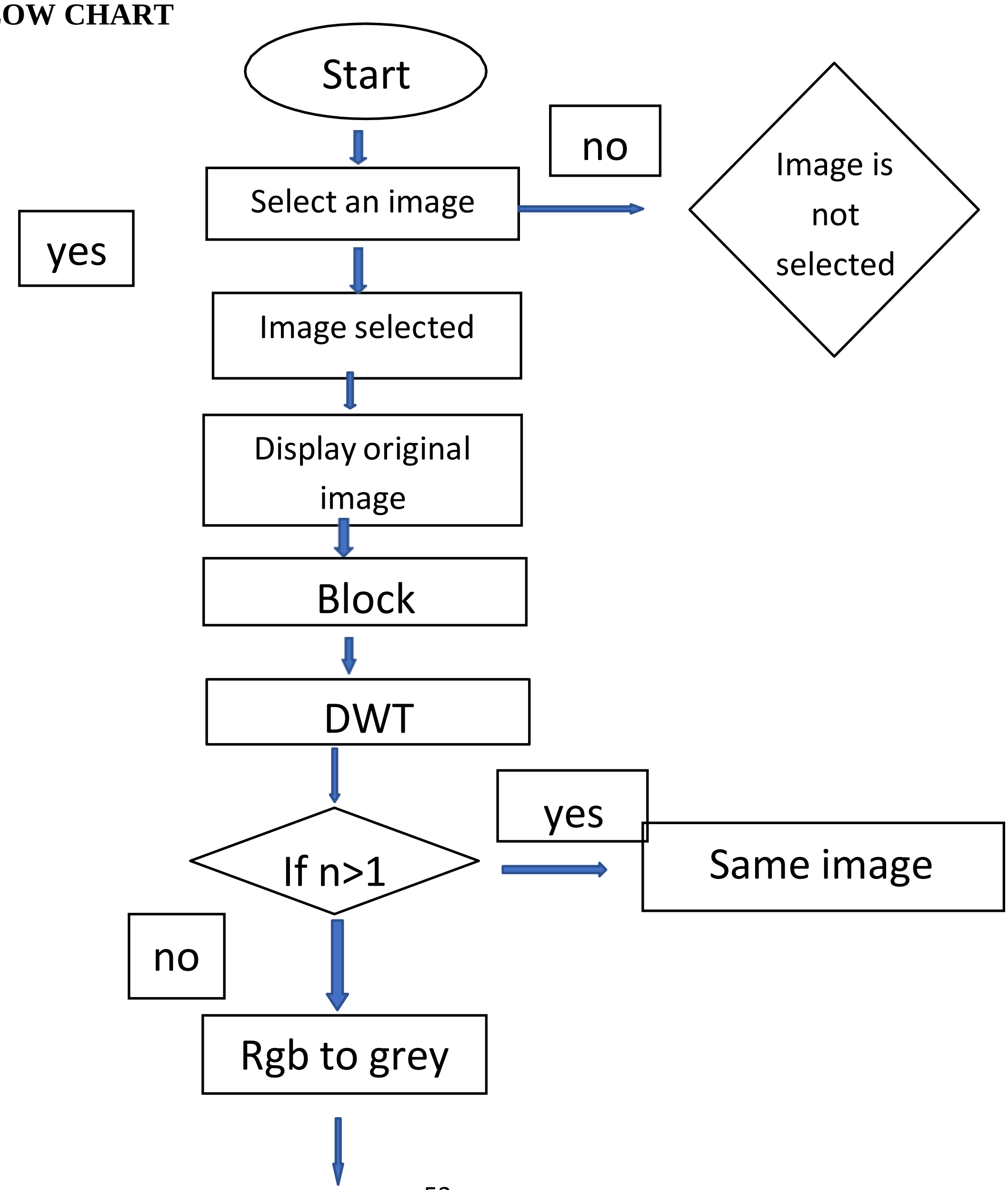


Fig.4.1: Block diagram of the project

4.2 FLOW CHART



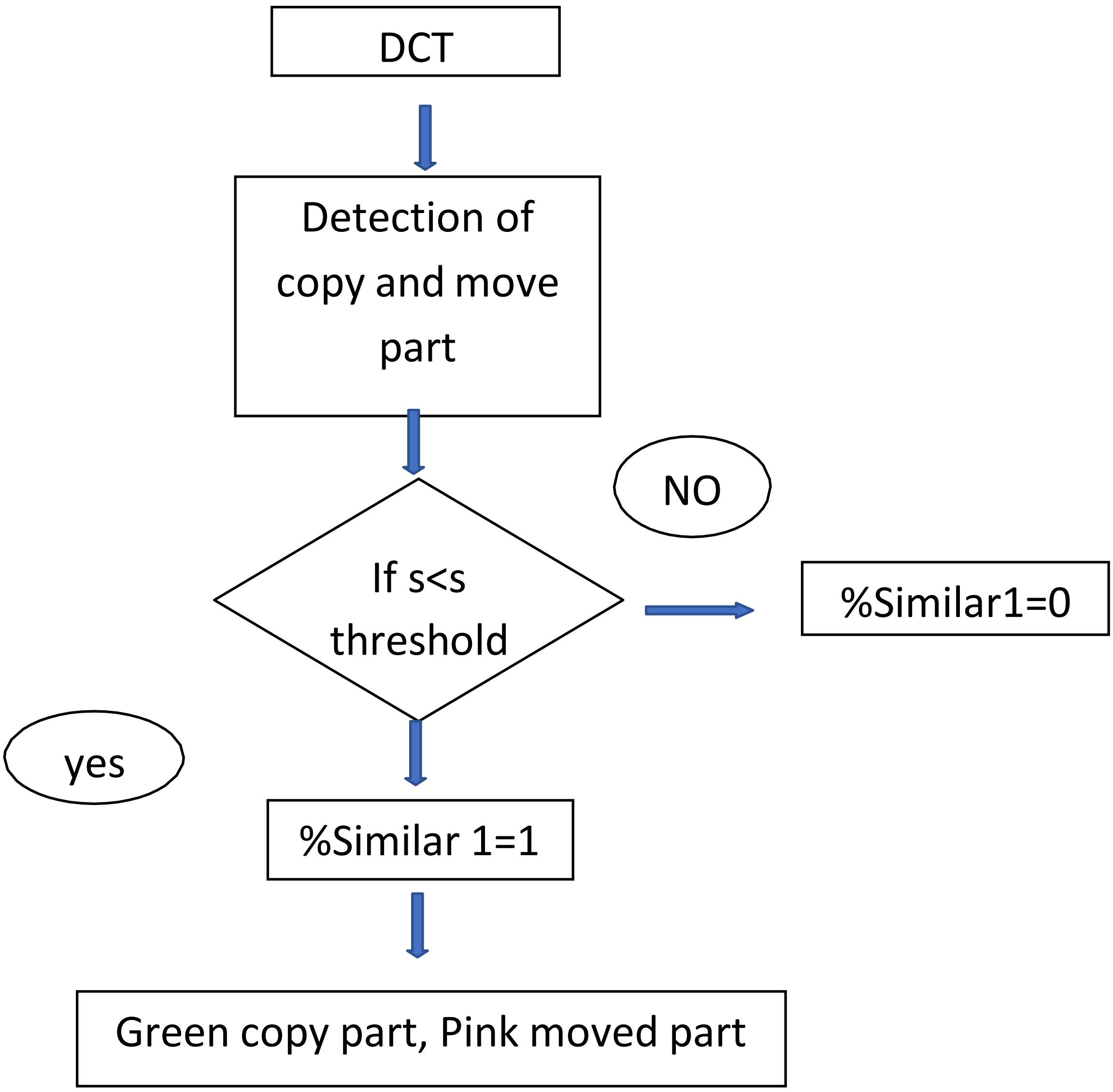


Fig.4.2 Flow Chart

4.3 IMPLEMENTATION STAGES

Stage-1: First an image that to be tested will be given as input.

Fig.4.3 Input image

Stage-2: As a step of pre-processing of an image block process is done.

Fig.4.4 Block Processing image

Stage-3: As a step 3 DWT is applied to the block processing image.It gives four types of image LL,LH,HL,HH based contrast and with respective coefficients

Fig.4.5 LL Image

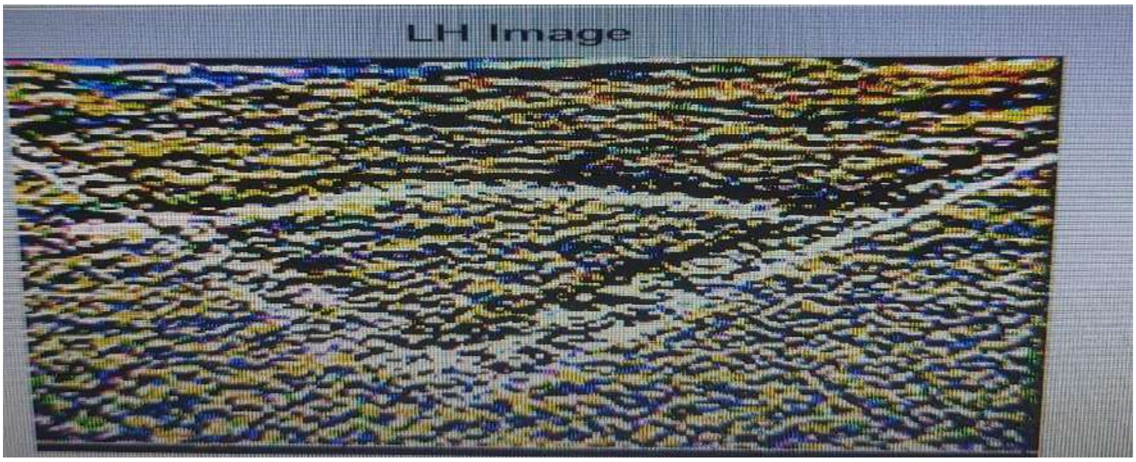


Fig.4.6 LH Image

Fig.4.7 HL Image

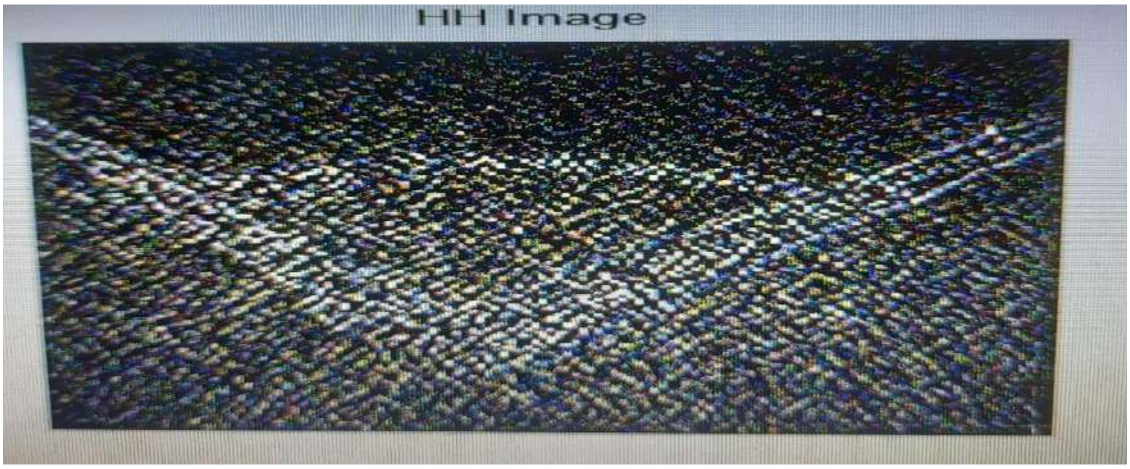


Fig.4.8 HH image

Stage-4: As a step 4 post processing is done by extracting required features.

Fig.4.9 Processing Image

4.5 RESULTS

By Performing feature matching and sorting copy and move part is detected.

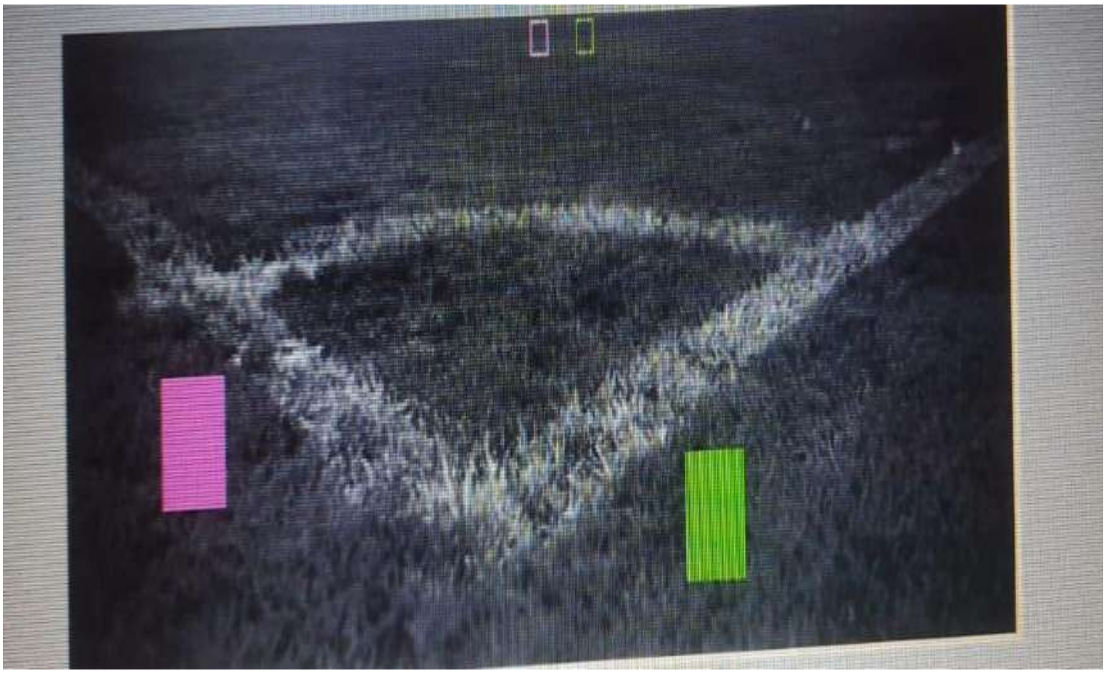


Fig 4.10: Final Output

CHAPTER-5

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION

The proposed approach can be used to detect the digital image forgery by using FFT and Local features successfully. It can be observed that LBP and ELTP perform as better features to classify the image as forged or authentic. The presented approaches involve complex transformations like DCT and FFT which increases the complexity of the methodology. The accuracy of detecting the copied parts of the image is increased when compared to existing techniques. The time taken for detection of forgery is reduced. The proposed system had successfully overcome the difficulties in the existing techniques.The proposed forgery detection had used block-based division techniques and feature extraction and SVM for image classifiers for accuracy of detection.

FUTURE ENHANCEMENT

The proposed method consists of FFT and DCT as main algorithm which increases complexity of the system. In future development complexity can be reduced. In future based on key point division the forgery detection technique can be developed. Further accuracy and reduction of time can be achieved in future practices.

PUBLICATION

Submitted Paper in the Conference ICSMEC- 21 with Paper ID (ICSMEC21-0045) and got Acceptance for the Paper. The paper was submitted under the name of Detection of Digital Image Forgery using FFT and Local features. The authors and coauthors of the project D. Saibhargavi, D. Pavani, B. Harish. This project has been completed under the guidance of Ms. R. Alekya Assistant professor ECE dept.

REFERENCES

[1] Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, G. Bebis, and H. Mathkour, “Passive detection of image forgery using dct and local binary pattern,” Signal, Image and Video Processing, vol. 11, no. 1, pp. 81–88, 2017.

[2] G. K. Birajdar and V. H. Mankar, “Digital image forgery detection using passive techniques: A survey,” Digital investigation, vol. 10, no. 3, pp. 226–245, 2013.

[3] Cavalin and L. S. Oliveira, “A review of texture classification methods and databases,” in Graphics, Patterns and Images Tutorials (SIBGRAPIT), 2017 30th SIBGRAPI Conference on. IEEE, 2017, pp. 1– 8.

[4] A. Doegar, M. Dutta, and G. Kumar, “A review of passive image cloning detection approaches,” in Proceedings of 2nd International Conference on Communication, Computing and Networking. Springer, 2019, pp. 469– 478.

[5] J. Dong, W. Wang, and T. Tan, “Casia image tampering detection evaluation database,” in Signal and Information Processing (ChinaSIP), 2013 IEEE China Summit & International Conference on. IEEE, 2013, pp. 422–426.

[6] E.-S. M. El-Alfy and M. A. Qureshi, “Combining spatial and dct based markov features for enhanced blind detection of image splicing,” Pattern Analysis and Applications, vol. 18, no. 3, pp. 713–723, 2015.

[7] H. Farid, “Digital doctoring: how to tell the real from the fake,” pp. 162– 166, 2006.

[8] F. Hakimi, M. Hariri, and I. Azad, “Image-Splicing Forgery Detection Based on Improved LBP and K-Nearest Neighbors Algorithm,” no. September 2015, 2015.

[9] F. Hakimi, M. Hariri, and F. GharehBaghi, “Image splicing forgery detection using local binary pattern and discrete wavelet transform,” in Knowledge-Based Engineering and Innovation (KBEI), 2015 2nd International Conference on. IEEE, 2015, pp. 1074–1077.

[10] J. G. Han, T. H. Park, Y. H. Moon, and I. K. Eom, “Quantizationbased markov feature extraction method for image splicing detection,” Machine Vision and Applications, vol. 29, no. 3, pp. 543–552, 2018.

[11] Z. He, W. Sun, W. Lu, and H. Lu, “Digital image splicing detection based on approximate run length,” Pattern Recognition Letters, vol. 32, no. 12, pp. 1591–1597, 2011. [Online].

[12] P. Heckbert, “Fourier transforms and the fast fourier transform (fft) algorithm,” Computer Graphics, vol. 2, pp. 15–463, 1995.

BIBILOGRAPHY

1.https://www.researchgate.net/publication/311957994_Digital_image_forgery_detection_techniques_a_survey.

2.https://ieeexplore.ieee.org/abstract/document/7377269?section=abstractCopy move image forgery detection using Hessian and center symmetric local binary pattern

3.https://www.researchgate.net/publication/342113801_An_Image_Forgery_Detection_using_SIFT-PCA.

4.https://link.springer.com/chapter/10.1007/978-981-15-5421-6_49Copy-Move Forgery Detection Methods

5.https://www.researchgate.net/publication/328824216_Visual_Feature_Based_Image_Forgery_Detection

6.http://bth.diva-portal.org/smash/get/diva2:1085560/FULLTEXT02Splicing Forgery Detection and the Impact of Image Resolution.

APPENDIX

The project proposes a technology to detect the genuiness of the image, this project is used to detect whether the image is genuine or forged. This project had got motivated from the existing methods published in IEEE papers under the name of Image-Splicing Forgery Detection Based on Improved LBP and K-Nearest Neighbors Algorithm," no. September 2015.This method uses both FFT and LOCAL FEATURES to detect the tampered part of the image and final result is displayed by performing certain processing techniques. This proposed techniques gives greater accuracy compared to existing methods .

PROJECT CODE

```
% Detection of Digital Image Forgery using Fast Fourier
% Transform and Local Features

clc
clear all;
close all;

[filename, pathname] = uigetfile({'*.bmp';'*.jpg'},'pick an image');

if isequal(filename,0) || isequal(pathname,0)
   warndlg('File is not selected');
else
input_image1=imread(filename);
imwrite(input_image1,'forgery_img.jpg');
input_image=imresize(input_image1,[512 512]);

figure,imshow(input_image);title('Original image');
end
%% block processing

fun = @(block_struct) imresize(block_struct.data,0.15);
block_img = blockproc(input_image,[4 4],fun);
block_img=imresize(block_img,[512 512]);

figure;
imshow(block_img);title('block procee image');
%%%%%%%%%%%%%%%%%% Apply Discrete wavelet transform
%%%%%%%%%%%%%%%%

[cA_img,cH,cV,cD]=dwt2(block_img ,'haar');
figure,imshow(mat2gray(cA_img));title('LL Image');
figure,imshow(cH);title('LH Image');
figure,imshow(cV);title('HL Image');
figure,imshow(cD);title('HH Image');
```

```
%%
mtx_cal=8;
overlapp=1;
Nd=16;
Th=0.9999;
s_threshold=1;

img=imread(filename);
[r,c,n] = size(img);
if n > 1
im=rgb2gray(img);
else
   im=img;
end

tic,
a=1;
figure,imshow(img), title('processing Image')
for j=1:overlapp:(c-mtx_cal)+1
   for i=1:overlapp:(r-mtx_cal)+1

      sondos(a).block=im(i:i+mtx_cal-1,j:j+mtx_cal-1);
      sondos(a).position=[i j];
      sondos(a).index=a;
      a=a+1;
   end
end

%% dct trnasformation for gray level checking
sz=size(sondos,2);
DC=[];
QZ=4;
for a=1:sz
   FDCT=dct2(sondos(a).block);
   FDCT=round(FDCT./QZ);
   DC(a,1)=FDCT(1,1);
end


G=[];
numclass=4;
[centers,mincenter,mindist,q2,quality] = FastKmean(DC,numclass,1);
for n=1:numclass
   ind= find(mincenter==n);
   for i=1:length(ind)

      G(n,i)= ind(i);

   end
end
```

```
%% detect copy and move part
figure,imshow(im),title('Result Image')
Th=0.99;
TotalMatch=0;
%parfor nG=1:size(G,1)
   for nG=1:size(G,1)
   emp=find( G(nG,:)==0);
   if isempty(emp)==0
      if emp(1)==1
         time=toc
         disp('time')
         disp(time)
      end
      A=[];
      for a=1:emp(1)-1
         [f,vec]=featureExtraction(sondos(G(nG,a)).block);
         A(a,1:9)=f;
         A(a,10)=sondos(G(nG,a)).position(1);
         A(a,11)=sondos(G(nG,a)).position(2);
      end
   else
      A=[];
      for a=1:size(G,2)
         [f,vec]=featureExtraction(sondos(G(nG,a)).block);
         A(a,1:9)=f;
         A(a,10)=sondos(G(nG,a)).position(1);
         A(a,11)=sondos(G(nG,a)).position(2);
      end
   end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
   Asorted=RadixSort(A,9);

   for i=1:size(Asorted,1)-1

      similar=[];
      for l=1:9

         s=abs(Asorted(i+1,l)-Asorted(i,l));

         if s<s_threshold
            %similar%
            similar(l)=1;

         else
            %not similar%
            similar(l)=0;
         end
```

```
        end
        if isempty(find(similar==0))

            x1=Asorted(i,10);
            x2=Asorted(i+1,10);
            y1=Asorted(i,11);
            y2=Asorted(i+1,11);
            D= sqrt((x1-x2)^2+(y1-y2)^2);
            if D>Nd

                TotalMatch=TotalMatch+1;

                rectangle('Position',[y1,x1,mtx_cal,mtx_cal],'Edgecolor','m');
                rectangle('Position',[y2,x2,mtx_cal,mtx_cal],'Edgecolor','g');
                %end


            end
        end


    end


   toc;
    end

    disp('m is copy part');
    disp('g is move part');


% end
```

A

MAJOR PROJECT REPORT

On

IOT BASED UNDERGROUND CABLE FAULT DETECTOR

Submitted by

Ms. Josyula Rohini (17K81A04L1)
Ms. K. Prardhana Merlyn (17K81A04L6)
Ms. Mubeena Begum (17K81A04N1)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Esteemed Guidance of

Mrs. Aleti Soumya

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad – 500 100

NBA & NAAC A+ Accredited

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA & NAAC A+ Accredited

JUNE 2021

Department of Electronics & Communication Engineering

BONAFIDE CERTIFICATE

This is to certify that the project entitled "IOT BASED UNDERGROUND CABLE FAULT DETECTOR", is being submitted by 1.**Ms.Josyula Rohini(17K81A04L1), 2.Ms.K.Prardhana Merlyn(17K81A04L6), 3.Ms.Mubeena Begum(17K81A04N1)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

PROJECT GUIDE	HEAD OF THE DEPARTMENT
Mrs.ALETI SOUMYA	Dr. B. HARI KRISHNA
Assistant Professor	Professor
Department of ECE	Department of ECE

Internal Examiner	External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **JOSYULA ROHINI** WITH ROLL NO.**17K81A04L1**, **K PRARDHANA MERLYN** WITH ROLL NO.**17K81A04L6**, **MUBEENA BEGUM** WITH ROLL NO.**17K81A04N1**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**IOT BASED UNDERGROUND CABLE FAULT DETECTOR**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. ★Kompally-500014.★

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of **Bachelor of Technology in Department of ELECTRONICS AND COMMUNICATION ENGINEERING**, session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled "**IOT BASED UNDERGROUND CABLE FAULT DETECTOR**" is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

JOSYULA ROHINI	**PRARDHANA MERLYN**	**MUBEENA BEGUM**
(17K81A04L1)	**(17K81A04L6)**	**(17K81A04N1)**

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna**, **Head &Professor**, Department of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in our memory for years to come.

We would like to express our gratitude to our dept. project coordinator **Dr.A.Anand, Professor** for his involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time to come.

We sincerely thank our Class Project Coordinator **Ms. P. Pushpa, Assistant Professor**, Department of Electronics and Communication Engineering and our project guide **Mrs.A.Soumya, Assistant Professor**, Department of Electronics and Communication Engineering for guidance and encouragement in carrying out this project work.

1. Josyula Rohini (17K81A04L1)
2. K.Prardhana Merlyn (17K81A04L6)
3. Mubeena Begum (17K81A04N1)

ABSTRACT

Underground cables are prone to a wide variety of faults due to underground conditions, wear and tear, rodents etc. Also detecting fault source is difficult and entire line is to be dug in order to check entire line and fix faults. So here we propose an cable fault detection over IOT that detects the exact fault position over IoT that makes repairing work very easy. The repairmen know exactly which part has fault and only that area is to be dug to detect the fault source. This saves a lot of time, money and efforts and also allows to service underground cables faster. We use IOT technology that allows the authorities to monitor and check faults over internet. The system detects fault with the help of potential divider network laid across the cable. Whenever a fault gets created at a point shorting two lines together, a specific voltage gets generated as per the resistors network combination. This voltage is sensed by the microcontroller and is updated to the user. The information conveyed to the user is the distance to which that voltage corresponds to. The microcontroller retrieves the fault line data and displays over LCD display, also it transfers this data over internet to display online. We use IoT Gecko to develop the online system that links with the system to display the cable faults online.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER-1
INTRODUCTION

1.1 OVERVIEW OF THE PROJECT:

Power supply networks are growing continuously and their reliability getting more important than ever. The complexity of the whole network comprises numerous components that can fail and interrupt the power supply for end user. For most of the worldwide operated low voltage and medium voltage distribution lines, underground cables have been used for many decades. Underground high voltage cables are used more and more because they are not influenced by weather conditions, heavy rain, storm, snow and pollution. Even though the Cable manufacturing technology is improving steadily; there are still influences which may cause cable to fail during test and operation. A cable in good condition and installed correctly can last a lifetime of about 30 years. However cables can be easily damaged by incorrect installation or poorly executed jointing, while subsequent third party damage by civil works such as trenching or curb edging.

1.1.1 Types of Faults in Cables

1.1.1.1 Open Circuit Fault

When there is a break in the conductor of the cable, it is called open circuit fault of the cable. The open circuit fault can be checked by megger. For this purpose, the three conductors of the 3-core cable at the far end are shorted and earthed. Then resistance between each conductor and earth is measured by a megger. The megger will indicate zero resistance in the circuit of the conductor that is not broken. However, if the conductor is broken, the megger will indicate infinite resistance in its circuit.

1.1.1.2. Short Circuit Fault

When two conductors of a multi-core cable come in electrical contact with each other due to insulation failure, it is called short-circuit fault. The two terminals of the megger are connected to any two conductors. If the megger gives zero reading, it indicates short-circuit fault between these two conductors. The same step can be repeated for other conductors taking two at a time.

1.1.1.3. Earth Fault

When the conductor of the cable comes in contact with earth, it is called earth fault or ground fault. To identify this fault, one terminal of the megger is connected to the conductor and the other terminal connected to earth. If megger indicates zero reading, it means the conductor is earthed. The same procedure is repeated for other conductors of the cable. This project is used to detect the location of fault in digital way. Locating the faulty point in an underground cable helps to facilitate quicker repair, improve the system reliability and reduced outage period.

1.2 OBJECTIVES OF THE STUDY:

The evaluation of IoT in the electrical Power Industry transformed the way things performed in usual manner. IoT increased the use of wireless technology to connect power industry assets and infrastructure in order to lower the power consumption and cost. The applications of IoT are not limited to particular fields, but span a wide range of applications such as energy systems, homes, industries, cities, logistics, heath, agriculture and so on. Since 1881, the overall power grid system has been built up over more than 13 decades, meeting the ever increasing demand for energy. Power grids are now been considered to be one of the vital components of infrastructure on which the modern society depends. It is essential to provide uninterrupted power without outages or losses. It is quiet hard to digest the fact that power generated is not equal to the power consumed at the end point due to various losses. It is even harder to imagine the after effects without power for a minute. Power outages occur as result of short circuits. This is a costly event as it influences the industrial production, commercial activities and consumer lifestyle. Government & independent power providers are continuously exploring solutions to ensure good power quality, maximize grid uptime, reduce power consumption, increase the efficiency of grid operations and eradicate outages, power loss & theft. Most importantly, the solution should provide a real-time visibility to customers on every penny paid for their energy. There is an increasing need of a centralized management solution for more reliable, scalable, and manageable operations while also being cost effective, secure, and interoperable. In addition, the solution should enable power providers and utilities to perform effective demand forecasting and energy planning to address the growing need for uninterrupted quality power [5]. The goal of IoT is not just only connecting things such as machines, devices and appliances, but also allowing the things to communicate, exchanging control data and other necessary information while executing applications. It consists of IoT devices that have unique identities and are capable of performing remote sensing, monitoring and actuating tasks. These devices are capable of interacting withone another directly or indirectly. Data collection is performed locally or remotely via centralized servers or cloud based applications. These devices may be data collection devices to which various sensors are attached such as temperature, humidity, light, etc., or they may be data actuating devices to which actuators are connected, such as relays. IoT system is composed of three layers: the perception layer, the network layer, and the application layer as shown in Figure 1. The perception layer includes a group of Internet-enabled devices that can percept, detect objects, collect systems information, and exchange information with other devices through the Internet communication networks. Sensors, Global Positioning Systems (GPS), cameras, and Radio Frequency Identification Devices (RFID) are examples of devices that exist at perception layer. The network layer is responsible of forwarding data from perception layer to the application layer under the constraints of devices' capabilities, network limitation and the applications' constraints. IoT systems use a combination of Internet and short-range networks based on the communicated parties. Short-range communication technologies such as Bluetooth and ZigBee are used to carry the information from perception devices to a nearby gateway. Other technologies such as Wi-Fi, 2G, 3G, 4G, and Power line Communication (PLC) carry the information for long distances based on the application. The upper layer is the application layer, where incoming information is processed to induce insights for better power's distribution design and management strategies.

1.3 SCOPE OF THE STUDY:

- The work can be extended for open circuit fault, short circuit Line to Line Fault (LL) and double Line to Ground Fault (LLG).
- The open circuit fault can be detected using a capacitor in ac circuit which measures the change in impedance and calculate the distance of fault.
- The proposed system in this paper detect only the location of Short Circuit fault in underground cable line, and also detect the location of open circuit fault, to detect the

open circuit fault capacitor is used in circuit which measure the change in resistance & calculate the distance of fault.

- For future research, the system would proceed with similar neural networks structure for different types fault section and fault location estimation.
- We can further develop a better user interface by which detection of open circuit fault is possible in near future.
- To find the fault in ac circuits, fluctuation in impedance is measured with the help of capacitor. In this way we can find the fault distance.
- We use an Ethernet shield to establish connection to the web page. If we use a Wi-Fi module instead of an Ethernet shied then we can make it wireless.
- With the help of an SD card, we can develop a better user interface. In the future it can be used to find the error in open circuit faults
- In this project we detect the exact location of short circuit fault in the underground cable from feeder end in km by using Arduino.
- In future, this project can be implemented to calculate the impedance by using a capacitor in an AC circuit and thus measure the open circuit fault.

1.4 MATERIAL REQUIREMENT:

1.4.1 HARDWARE REQUIREMENT:

- Arduino uno
- LCD
- Power Supply
- WI-FI Module ESP8266
- Switches
- Relay
- Load

1.4.2 SOFTWARE REQUIREMENT:

- Proteus software
- Arduino software

1.5 PROCUMENT OF EQUIPMENT:

We brought all the required hardware components in an electronic manufacturing shop and before assembling and interfacing hardware components to Arduino UNO micro-controller, we connected individual components to Arduino in Proteus Software and understood working of every individual component in Proteus Software.

Now, we assembled all components and interfaced to Arduino UNO micro-controller using connecting wires through bread board and we wrote program to work all the hardware components in Arduino IDE we understood the working of all components in a practical way.

We were able to monitor the underground cable fault with better efficiency, with less time delay and the results were also satisfactory and we successfully got the output in mobile telnet app and LCD screen. We would like to improve the project in future for further developments.

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH AREA:

The idea of Underground transmission cable was first introduced by W. M. Callender before the American Institute of Electrical Engineers, at Philadeithia, September, 1884.he told in his paper that — For underground work, the first thing to be decided upon is the conduit A practical algorithm with wavelet Multi Resolution Analysis (MRA) coefficients are designed by Tag Eldin (2010) for fault detection and classification. The line currents at the two ends are evaluated through an online WT algorithm with MRA for fault identification. The magnitudes of spikes in wavelet coefficients are used in the fault detection and classification process. To segregate the ground faults from phase-to-phase faults, the smooth coefficient examination of neutral current is employed. The designed technique by Reddy et al. (2009) with a mixture of impedance calculation and CWT method identifies the disturbance with fault occurrence. There are many schemes with the relay for identification of the proper faulty phase in order to protect the power system from transmission line faults. The scheme employs the NN, WT or both, to address the issues. WT with exact tools are used by Mohammad Ali et al. (2014) for transient signal in transmission lines. An ANN has a large variation between the measured signal and associated signal. Transmission lines are used to distribute the electric power at the destination. The fault localization issues are addressed by Mohammad Abdul Baseer (2013) by traveling wave voltage and current signals on a transmission line network. Fourier Transform (FT) is used to get the frequency components in fault signal. The wavelet MRA is essential in signal analysis and compatible with traveling wave signals. A fault detection method has been designed by Durga Prasad & Srinivasu (2015) for protection of transmission lines by root mean square values of 3-phase power. Input signal for detection algorithm is changed during faults and non-fault cases. The designed method is authenticated for faults of many types at various moments and various locations. The transmission line is an important part, susceptible to fault of electrical power system, because of its large physical dimension. A model has been designed by Andre de Souza Gomes (2013) to functionally characterize the phases of a transmission line. The detection and classification plans are designed by a set of fault simulation and real record. Dhivya Dharani.A, Sowmya.T the paper titles as―Development of a Prototype Underground Cable Fault Detector ―Cable faults are damage to cables which affects the resistance in the cable. If allowed to persist, this can lead to a voltage breakdown. To locate a fault in the cable, the cable must first be tested for faults. This prototype is assembled with a set of resistors representing cable length in Kilo meters and fault creation is made by a set of switches at every known Kilo meters (km's) to cross check the accuracy of the same. The fault occurring at what distance and which phase is displayed on a 16X2 LCD interfaced with the microcontroller. Nikhil Kumar Sain, Rajesh Kajla paper titled as ―Underground Cable Fault Distance Conveyed. This paper proposes fault location model for underground power cable using microcontroller. The aim of this project is to determine the distance of underground cable fault from base station in kilometres. This project uses the simple concept of ohm's law. When any fault like short circuit occurs, voltage drop will varydepending on the length of fault in cable, since the current varies. R.K.Raghul Mansingh, R.Rajesh, S.Ramasubramani, G.Ramkumar titled as ―Underground Cable Fault Detection using Raspberry Pi and Arduino‖-The aim of this project is to determine the underground cable fault. This project uses the simple concept of CT Theory. When any fault like short circuit occurs, voltage drop will vary depending on the length of fault in cable, since the current varies CT is used to calculate the varying. The signal conditioner manipulates the change in voltage and a microcontroller is used to make the necessary calculations so that the fault distance is displayed by IOT devices.

2.2 REVIEW ON RELATED LITERATURE:

Sectionalizing:

Sectionalizing procedure reduces cable reliability, because it depends on physically cutting and splicing the cable. Dividing the cable into successively smaller sections and measuring both ways with an ohmmeter or high-voltage insulation resistance (IR) tester enable to narrow down search for a fault. This laborious procedure normally involves repeated cable excavation.

Thumping:

When high voltage is supplied to faulty cable, the resulted high current arc makes a noise loud enough to hear above ground. While this method eliminates the sectionalizing method's cutting and splicing, it has its own drawback. Thumping requires a current on the order of tens of thousands of amps at voltages as high as 25 kV to make an underground noise loud enough to hear above ground. The heating from this high current often causes some degradation of the cable insulation. The limit of damage can be reduced by passing minimum required power to conduct the test.

Time-Domain Reflectometry:

The Time domain reflectometer (TDR) is an electronic instrument that uses time domain reflectometry to characterize and locate faults in metallic cables. The TDR sends a lowenergy signal through the cable, causing no insulation degradation. A theoretically perfect cable returns that signal in a known time and in a known profile. Impedance variations in a "real-world" cable alter both the time and profile, which the TDR screen or printout graphically represents. One weakness of TDR is that it does not pinpoint faults.

Arc Reflection Method:

This method is often referred to as a high voltage radar technique that overcomes the 200 Ω limitation of low-voltage radar. In addition to the TDR, an arc reflection filter and surge generator is required. The surge generator is used to create an arc across the shunt fault which creates a momentary short circuit that the TDR can display as a downward-going reflection. The filter protects the TDR from the high voltage pulse generated by the surge generator and routes the low-voltage pulses down the cable. Arc reflection is the most accurate and easiest pre location method. The fault is displayed in relation to other cable landmarks such as splices, taps and transformers and no interpretation is required. Arc reflection makes it possible for the TDR to display "before" and "after" traces or cable signatures. The "before" trace is the low-voltage radar signature that shows all cable landmarks but does not show the downward reflection of a high resistance shunt fault. The "after" trace is the high-voltage signature that includes the fault location even though its resistance may be higher than 200 Ω. This trace is digitized, stored and displayed on the screen and the cursors are positioned in order to read the distance to the high resistance fault.

2.3 CONCLUSION ON REVIEWS:

Today, the methods to find fault became increasingly complex and these methods include usage of Fourier analysis, Wavelet Transformation etc. But still, the time is taken to report the fault and to rectify it is reduced slightly. Thus came the use of Microprocessor followed in by IoT. The fault is identified and is stored in the cloud. The stored data is updated in the database for an app and HTML page. Thus the repairmen are given as much information about the fault is provided.

Programs uploaded in Arduino UNO kit to detect faults from the underground cables. When a fault occurs in the underground cables, we can find out faults through Arduino controller kit. LCD display which displays the faults in Kilometer. In this project we created faults manually. Cable has many types. Every cable has different resistance which depends upon the material used. The value of the resistance is depends upon the length of the cable. In here resistance is the leading role of the project. If any deviation occurs in the resistance, the value of the voltage will be changed that particular point is called FAULT. We are finding out those faults.

CHAPTER-3

PROJECT DESIGN

3.1 OVERVIEW OF THE DESIGN:

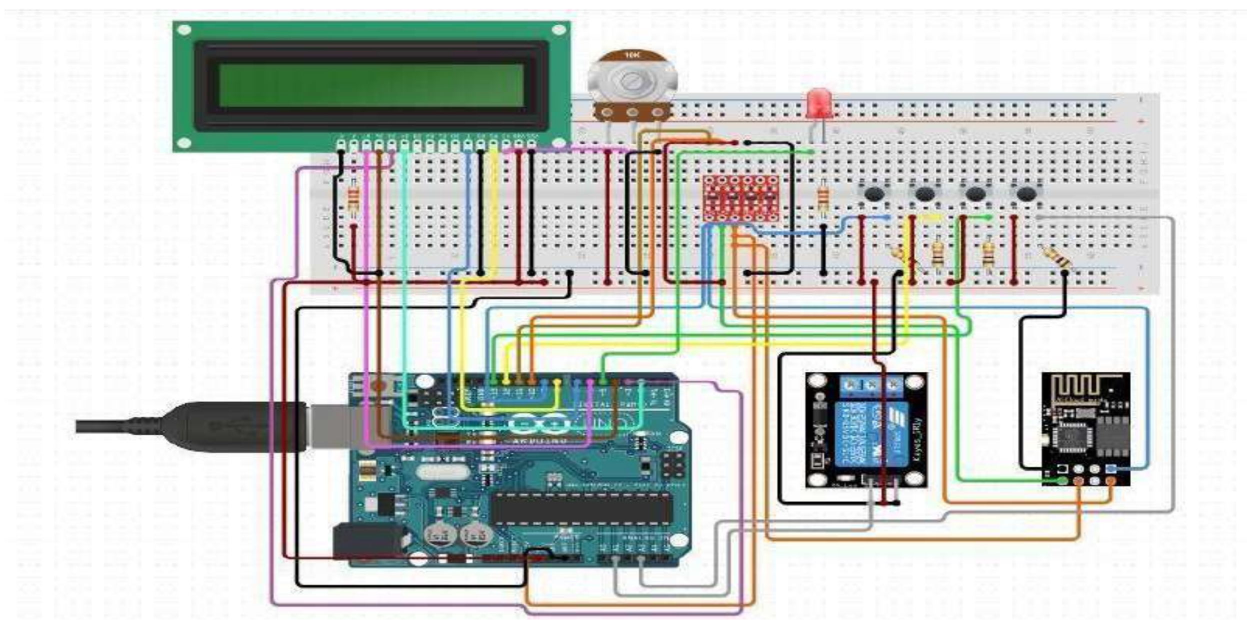


Fig.3.1 Overview of the design

3.2 EQUIPMENT ANALYSIS:

3.2.1 ARDUINO:

3.2.1.1 INTRODUCTION:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

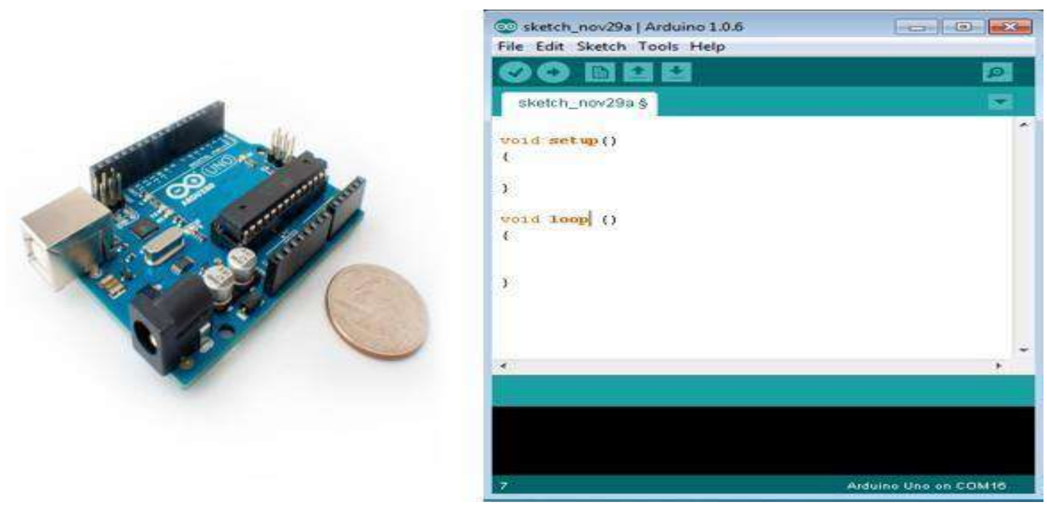


Fig 3.2.1.1 Arduino Uno

3.2.1.2 BOARD TYPES

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V. Here is a list of different Arduino boards available.

Board Name	**Operating Volt**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header

Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.2.1.2.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 3.2.1.2.2 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible

							Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 3.2.1.2.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table 3.2.1.2.4 Arduino boards based on AT91SAM3X8E microcontroller

3.2.1.3 BOARD DESCRIPTION:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

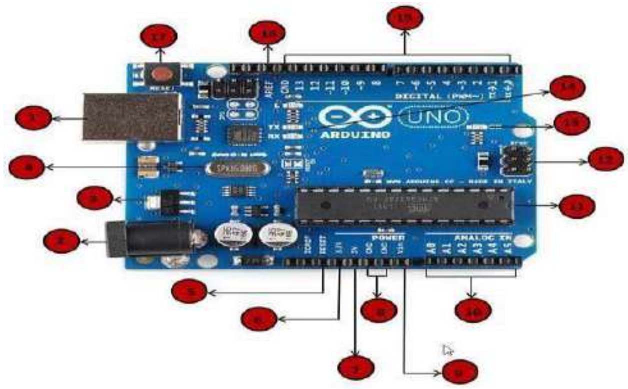


Fig.3.2.1.3 Board Description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt

	• 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs**

	On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Table 3.2.1.3.1 Board description of Arduino UNO

3.2.1.4 ARDUINO FAMILY

Arduino makes several different boards, each with different capabilities. In addition, part of being open-source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

Fig.3.2.1.4: Arduino Family

3.2.1.5 SHIELDS:

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

Fig.3.2.1.5: Arduino Shields

3.2.1.6 PIN DESCRIPTION OF ATMEGA328:

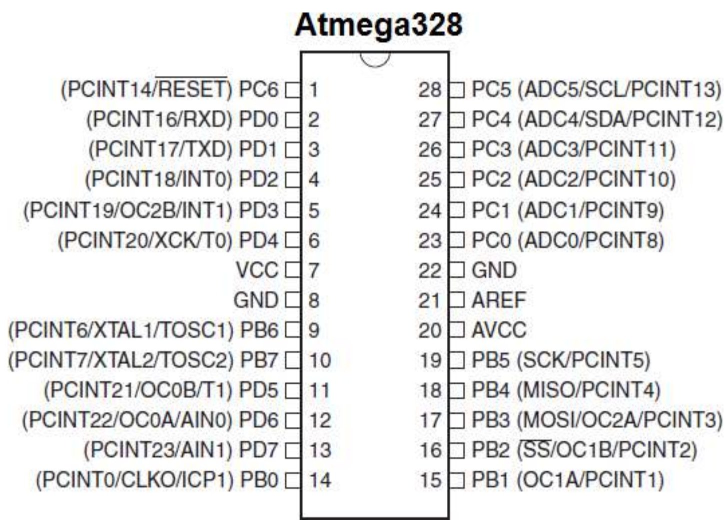


Fig.3.2.1.6 Pin description of ATMEGA328

3.2.1.7 APPLICATIONS:

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

3.2.2 16*2 LCD SCREEN DISPLAY

3.2.2.1 INTRODUCTION

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

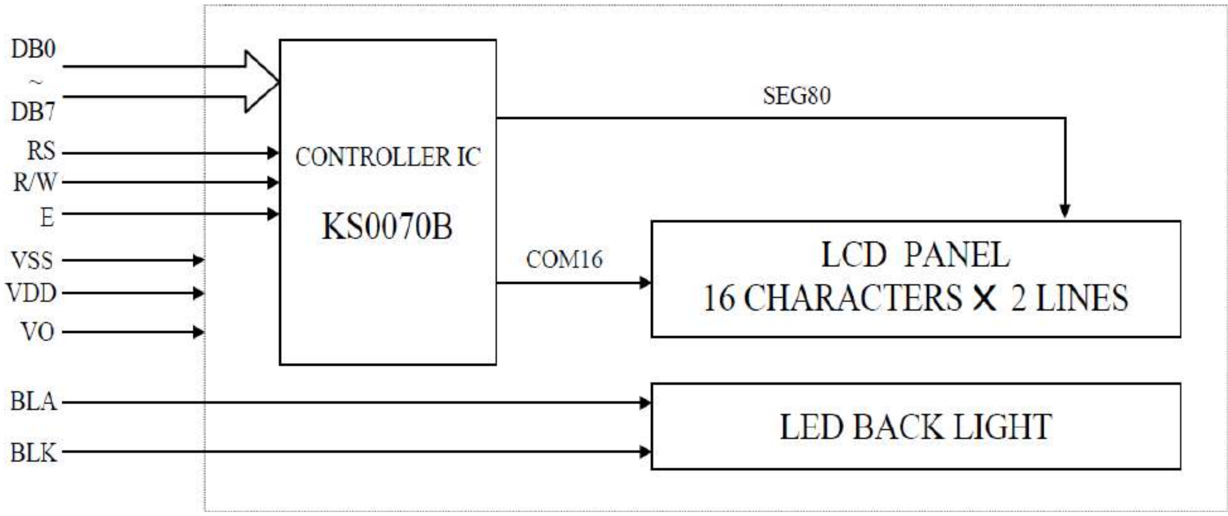


fig.3.2.2.1 Lcd block diagram

3.2.2.2 DISPLAY DATA RAM:

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.
Figure below will show you the DDRAM addresses of 2 Line LCD.

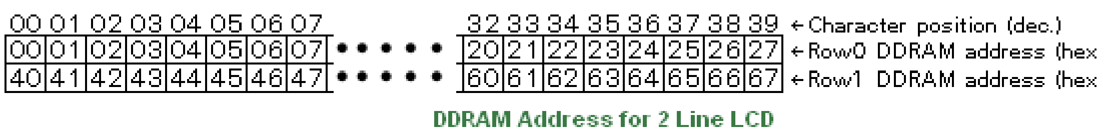


Fig.3.2.2.2 DDRAM addresses of 2 Line LCD

3.2.2.3 CHARACTER GENERATOR ROM:

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM.

3.2.2.4 BUS FLAG:

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e. BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

3.2.2.5 INSTRUCTION REGISTER (IR) AND DATA REGISTER (DR):

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g. LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD.

Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

3.2.2.6 16 X 2 ALPHANUMERIC LCD MODULE FEATURES:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible

- Connector for standard 0.1-pitch pin headers

3.2.2.7 CIRCUIT DESCRIPTION:

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

3.2.3 POWER SUPPLY UNIT

3.2.3.1 INTRODUCTION:

The power supplies are designed to convert high voltage AC mains electricity to a suitable l voltage supply for electronic circuits and other devices. A power supply can by broken down into a ser of blocks, each of which performs a particular function. A d.c power supply which maintains the out voltage constant irrespective of a.c mains fluctuations or load variations is known as “Regulated D.C Po Supply”.

3.2.3.2 BLOCK DIAGRAM OF POWER SUPPLY:

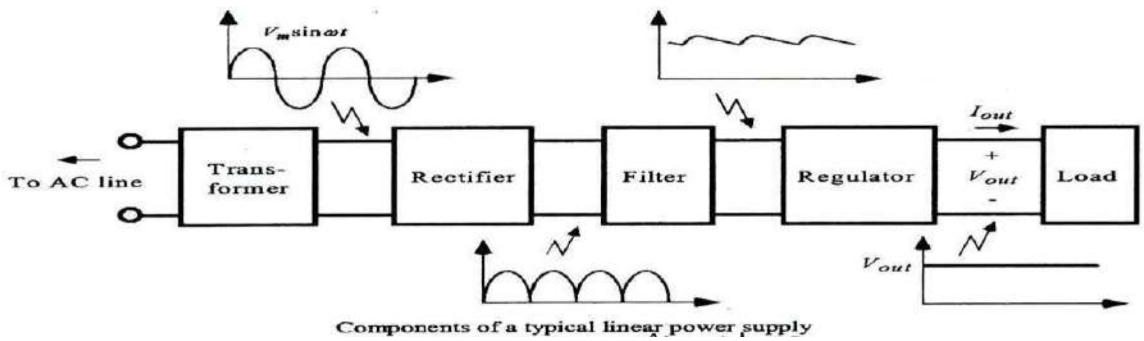


Figure 3.2.3.2.1: Block Diagram of Power Supply

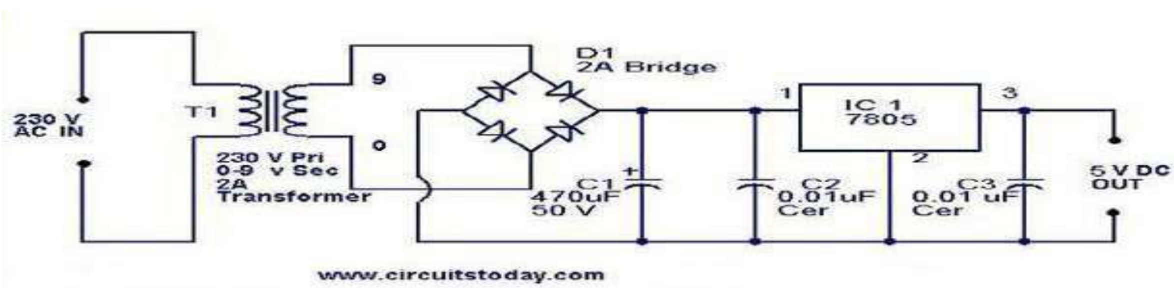


Figure 3.2.3.2.2: Schematic Diagram of Power Supply

3.2.3.3 DESCRIPTION OF POWER SUPPLY:

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert aa different form of energy – such as solar, mechanical, or chemical - into electrical energy.

A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered. For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

3.2.3.4 TRANSFORMER:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

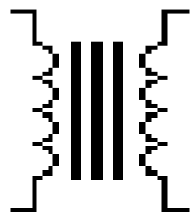


Figure 3.2.3.4.1 : Transformer Symbol

Transformer is a device that converts the one form energy to another form of energy like a transducer.

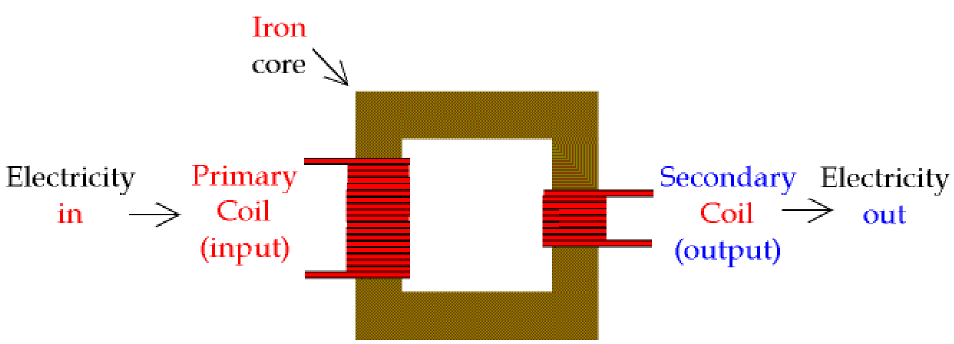


Figure 3.2.3.4.2 : Transformer

3.2.3.5 BASIC PRINCIPLE OF TRANSFORMER:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

From Faraday's Law

$$\frac{V_S}{V_P} = \frac{N_S}{N_P}$$

For ideal transformer

The voltage ratio is equal to the turns ratio, and power in equals power out.

From conservation of energy

$$P_P = V_P I_P = V_S I_S = P_S$$

3.2.3.6 WORKING OF TRANSFORMER:

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

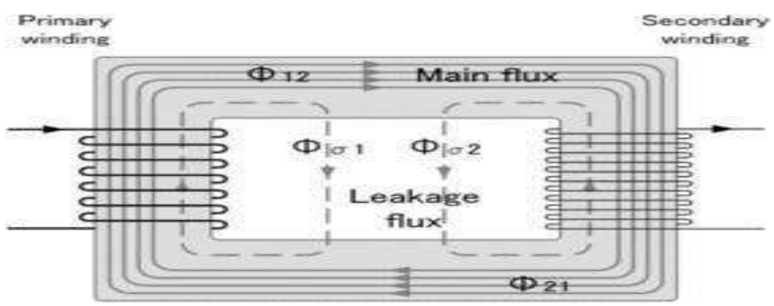


Figure 3.2.3.6 : Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f. The iron core is laminated to prevent 'eddy currents' flowing in the core.

These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

3.2.3.7 . Classification of Transformer:

- Step-Up Transformer
- Step-Down Transformer

3.2.3.7.1 Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn's ratio of the two sets of windings determines the amount of voltage transformation.

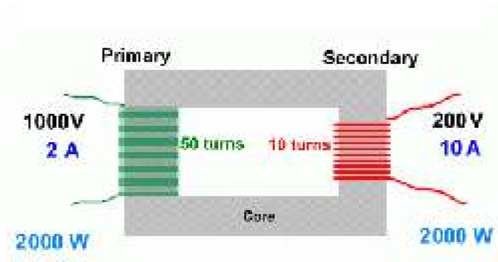


Figure 3.2.3.7.1 : Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single phase steps down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase steps up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

3.2.3.7.2 Step-Up Transformer:

A step up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply. A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

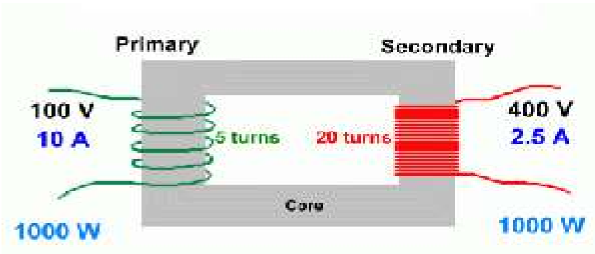


Figure 3.2.3.7.2 : Step-Up Transformer

3.2.3.8 DIODES:

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

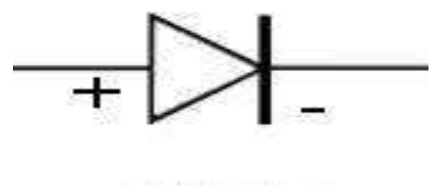


Figure 3.2.3.8 : Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

3.2.3.9 Rectifier:

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

3.2.3.10 The Half-Wave Rectifier:

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure.

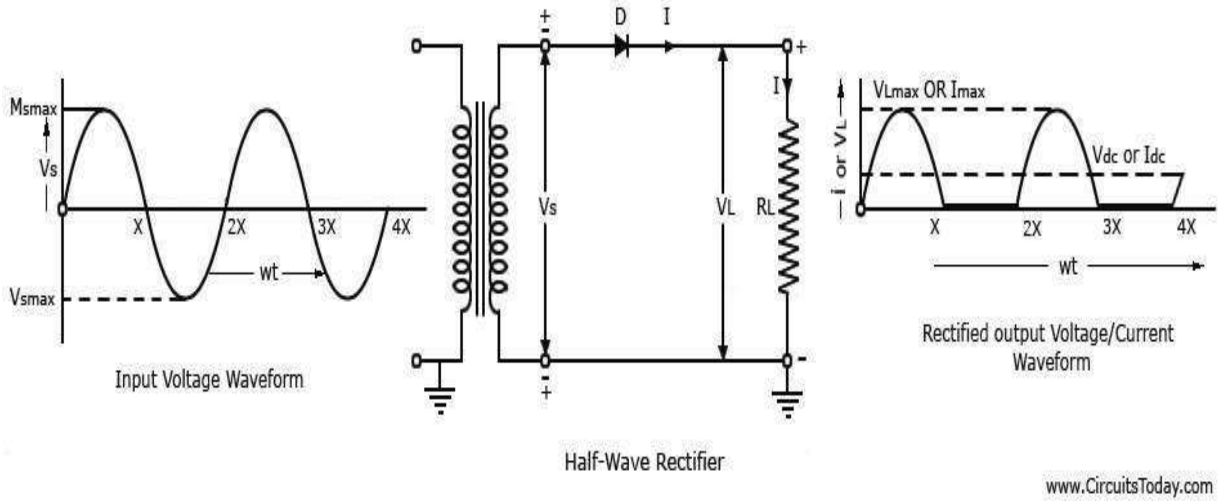


Figure 3.2.3.10 : Half Wave Rectifier

Figure 2 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

3.2.3.11 The Full-wave Rectifier:

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC.

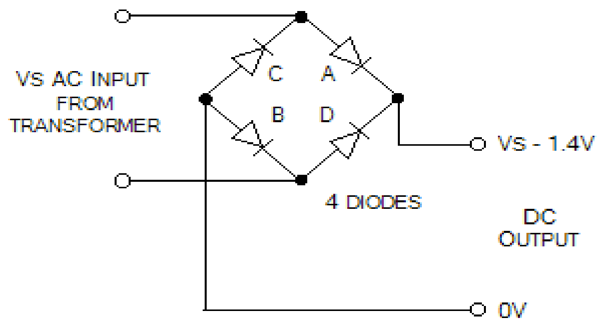


Figure 3.2.3.11.1: Full-Wave Rectifier

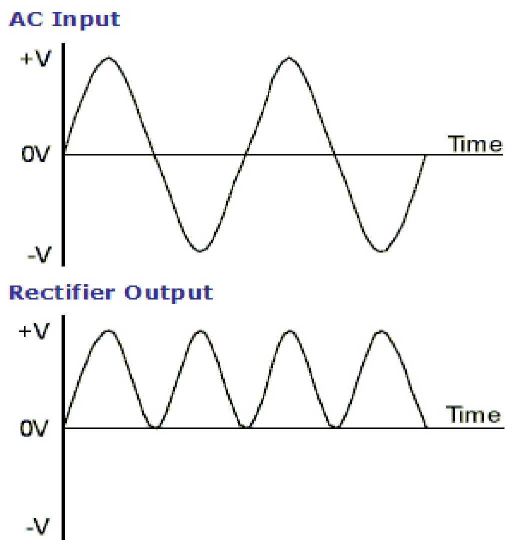


Figure 3.2.3.11.2: Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased. While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

3.2.3.12 CAPACITOR FILTER:

The capacitor-input filter, also called "Pi" filter due to its shape that looks like the Greek letterpi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

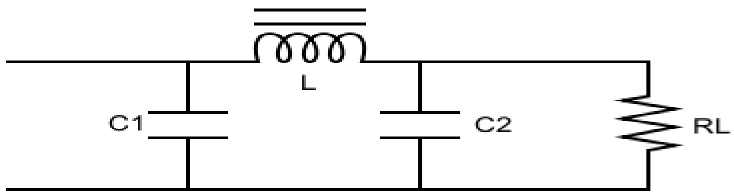


Figure 3.2.3.12.1 : Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result, the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result, only the DC component appears across the load RL.

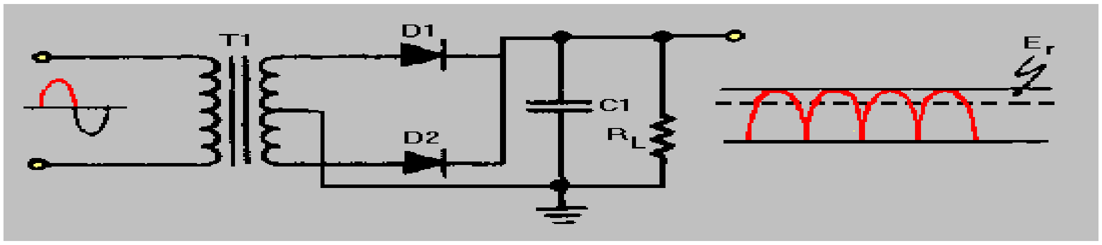


Figure 3.2.3.12 : Centered Tapped Full-Wave Rectifier with a Capacitor Filter

3.2.3.13 . VOLTAGE REGULATOR:

A voltage regulator is an electricalregulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx'indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx'indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consist the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

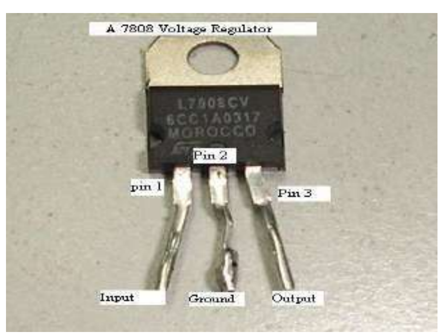


Fig.3.2.3.13 Voltage Regulator

3.2.4 Relay

3.2.4.1 Introduction

A relay is used to isolate one electrical circuit from another. It allows a low current control circuit to make or break an electrically isolated high current circuit path. The basic relay consists of a coil and a set of contacts. The most common relay coil is a length of magnet wire wrapped around a metal core. When voltage is applied to the coil, current passes through the wire and creates a magnetic field. This magnetic field pulls the contacts together and holds them there until the current flow in the coil has stopped. The diagram below shows the parts of a simple relay.

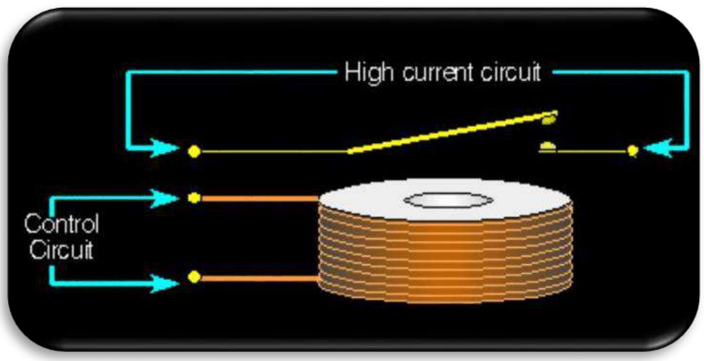


Figure 3.2.4.1 : Relay

3.2.4.2 Operation:

When a current flows through the coil, the resulting magnetic field attracts an armature that is mechanically linked to a moving contact. The movement either makes or breaks a connection with a fixed contact. When the current is switched off, the armature is usually returned by a spring to its resting position shown in figure 6.6(b). Latching relays exist that require operation of a second coil to reset the contact position.

By analogy with the functions of the original electromagnetic device, a solid-state relay operates a thyristor or other solid-state switching device with a transformer or light-emitting diode to trigger it.

3.2.4.3 Pole and throw

3.2.4.3.1 SPST

SPST relay stands for Single Pole Single Throw relay. Current will only flow through the contacts when the relay coil is energized.

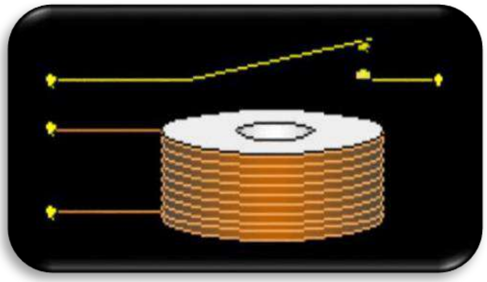


Figure 3.2.4.3.1 : SPST Relay

3.2.4.3.2 SPDT Relay

SPDT Relay stands for Single Pole Double Throw relay. Current will flow between the movable contact and one fixed contact when the coil is De-energized and between the movable contact and the alternate fixed contact when the relay coil is energized. The most commonly used relay in car audio, the Bosch relay, is a SPDT relay.

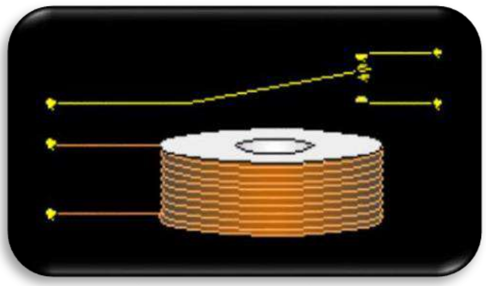


Figure 3.2.4.3.2 : SPDT Relay

3.2.4.3.3 DPST Relay

DPST relay stands for Double Pole Single Throw relay. When the relay coil is energized, two separate and electrically isolated sets of contacts are pulled down to make contact with their stationary counterparts. There is no complete circuit path when the relay is De-energized.

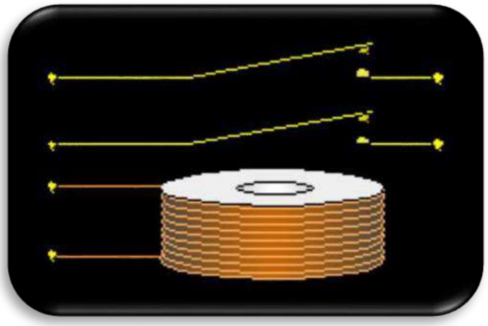


Figure 3.2.4.3.3 : DPST Relay

3.2.4.3.4 DPDT Relay

DPDT relay stands for Double Pole Double Throw relay. It operates like the SPDT relay but has twice as many contacts. There are two completely isolated sets of contacts.

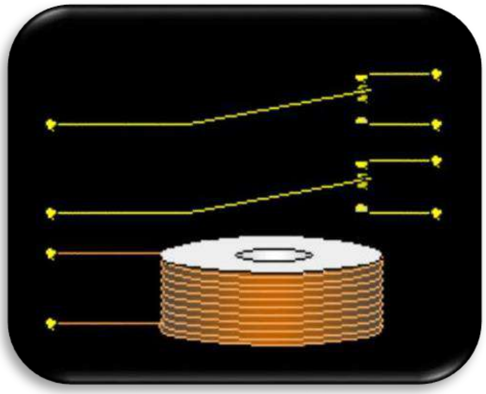


Figure 3.2.4.3.4.1 : DPDT Relay

This is a 4 Pole Double Throw relay. It operates like the SPDT relay but it has 4 sets of isolated contacts.

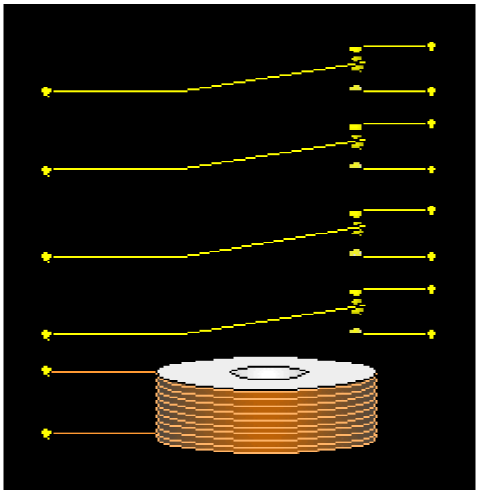


Figure 3.2.4.3.4.2 :Pole Double Throw relay

3.2.4.4 Types of relay:

1. Latching Relay
2. Reed Relay
3. Mercury Wetted Relay
4. Machine Tool Relay
5. Solid State Relay (SSR)

3.2.4.4.1 Latching relay

Latching relay, dust cover removed, showing pawl and ratchet mechanism. The ratchet operates a cam, which raises and lowers the moving contact arm, seen edge-on just below it. The moving and fixed contacts are visible at the left side of the image.

A **latching relay** has two relaxed states (bi-stable). These are also called "impulse", "keep", or "stay" relays. When the current is switched off, the relay remains in its last state. This is achieved with a solenoid operating a ratchet and cam mechanism, or by having two opposing coils with an over-center spring or permanent magnet to hold the armature and contacts in position while the coil is relaxed, or with a remanent core. In the ratchet and cam example, the first pulse to the coil turns the relay on and the second pulse turns it off. In the two coil example, a pulse to one coil turns the relay on and a pulse to the opposite coil turns the relay off. This type of relay has the advantage that it consumes power only for an instant, while it is being switched, and it retains its last setting across a power outage. A remanent core latching relay requires a current pulse of opposite polarity to make it change state.

Figure 3.2.4.4.1 : Latching relay

3.2.4.4.2 Reed Relay:

A **mercury-wetted reed relay** is a form of reed relay in which the contacts are wetted with mercury. Such relays are used to switch low-voltage signals (one volt or less) because of their low contact resistance, or for high-speed counting and timing applications where the mercury eliminates contact bounce. Mercury wetted relays are position-sensitive and must be mounted vertically to work properly. Because of the toxicity and expense of liquid mercury, these relays are rarely specified for new equipment. See also mercury switch.

3.2.4.4.3 Machine tool relay:

A **machine tool relay** is a type standardized for industrial control of machine tools, transfer machines, and other sequential control. They are characterized by a large number of contacts (sometimes extendable in the field) which are easily converted from normally-open to normally-closed status, easily replaceable coils, and a form factor that allows compactly installing many relays in a control panel. Although such relays once were the backbone of automation in such industries as automobile assembly, the programmable logic controller (PLC) mostly displaced the machine tool relay from sequential control applications.

3.2.4.4.4 Solid-state relay

A solid state relay (SSR) is a solid state electronic component that provides a similar function to an electromechanical relay but does not have any moving components, increasing long-term reliability. With early SSR's, the tradeoff came from the fact that every transistor has a small voltage drop across it. This voltage drop limited the amount of current a given SSR could handle. As transistors improved, higher current SSR's, able to handle 100 to 1,200 Amperes, have become commercially available. Compared to electromagnetic relays, they may be falsely triggered by transients.

Figure 3.2.4.4.4 : Solid relay, which has no moving parts

3.2.4.5 Specification

- Number and type of contacts – normally open, normally closed, (double-throw)
- Contact sequence – "Make before Break" or "Break before Make". For example, the old style telephone exchanges required Make-before-break so that the connection didn't get dropped while dialing the number.
- Rating of contacts – small relays switch a few amperes, large contactors are rated for up to 3000 amperes, alternating or direct current
- Voltage rating of contacts – typical control relays rated 300 VAC or 600 VAC, automotive types to 50 VDC, special high-voltage relays to about 15 000 V
- Coil voltage – machine-tool relays usually 24 VAC, 120 or 250 VAC, relays for switchgear may have 125 V or 250 VDC coils, "sensitive" relays operate on a few milli-amperes

3.2.4.5 Applications:

3.2.4.5.1 Relays are used:

- To control a high-voltage circuit with a low-voltage signal, as in some types of modems,
- To control a high-current circuit with a low-current signal, as in the starter solenoid of an automobile,

- To detect and isolate faults on transmission and distribution lines by opening and closing circuit breakers (protection relays),
- To isolate the controlling circuit from the controlled circuit when the two are at different potentials, for example when controlling a mains-powered device from a low-voltage switch. The latter is often applied to control office lighting as the low voltage wires are easily installed in partitions, which may be often moved as needs change. They may also be controlled by room occupancy detectors in an effort to conserve energy,
- To perform logic functions. For example, the boolean AND function is realized by connecting relay contacts in series, the OR function by connecting contacts in parallel. Due to the failure modes of a relay compared with a semiconductor, they are widely used in safety critical logic, such as the control panels of radioactive waste handling machinery.
- As oscillators, also called vibrators. The coil is wired in series with the normally closed contacts. When a current is passed through the relay coil, the relay operates and opens the contacts that carry the supply current. This stops the current and causes the contacts to close again. The cycle repeats continuously, causing the relay to open and close rapidly. Vibrators are used to generate pulsed current.
- To generate sound. A vibrator, described above, creates a buzzing sound because of the rapid oscillation of the armature. This is the basis of the electric bell, which consists of a vibrator with a hammer attached to the armature so it can repeatedly strike a bell.

3.2.5 SOFTWARE EXPLANATION:

Arduino software

Proteus simulation

3.2.5.1 ARDUINO SOFTWARE:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

3.2.5.2 WHY ARDUINO?

Thanks to its simple and accessible user experience, Arduino has been used in thousands ofdifferent projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics

principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key toolto learn new things. Anyone - children, hobbyists, artists, programmers

- can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX- 24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and

wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.Wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.

- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers c an make their own version Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money
- Getting Started with Arduino and Genuino products: -

Install the Arduino Software (IDE) on Windows PCs-

This document explains how to install the Arduino Software (IDE) on Windows machines.

- ✓ Download the Arduino Software (IDE)
- ✓ Proceed with board specific instructions.

3.2.5.3 HOW TO DOWNLOAD THE ARDUINO SOFTWARE (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

3.2.5.4 INSTALLATION:

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.3.2.5.4.1 Standard USB Cable (A plug to B plug)

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig. 3.2.5.4.2 Standard USB Cable (an A to Mini-B plug)

Step 2 – Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

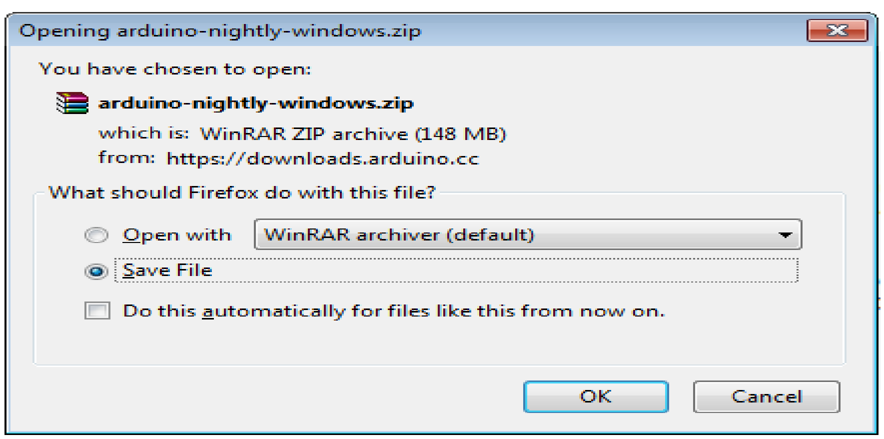


Fig. 3.2.5.4.3 Download Arduino IDE Software

Step 3 – Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 – Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

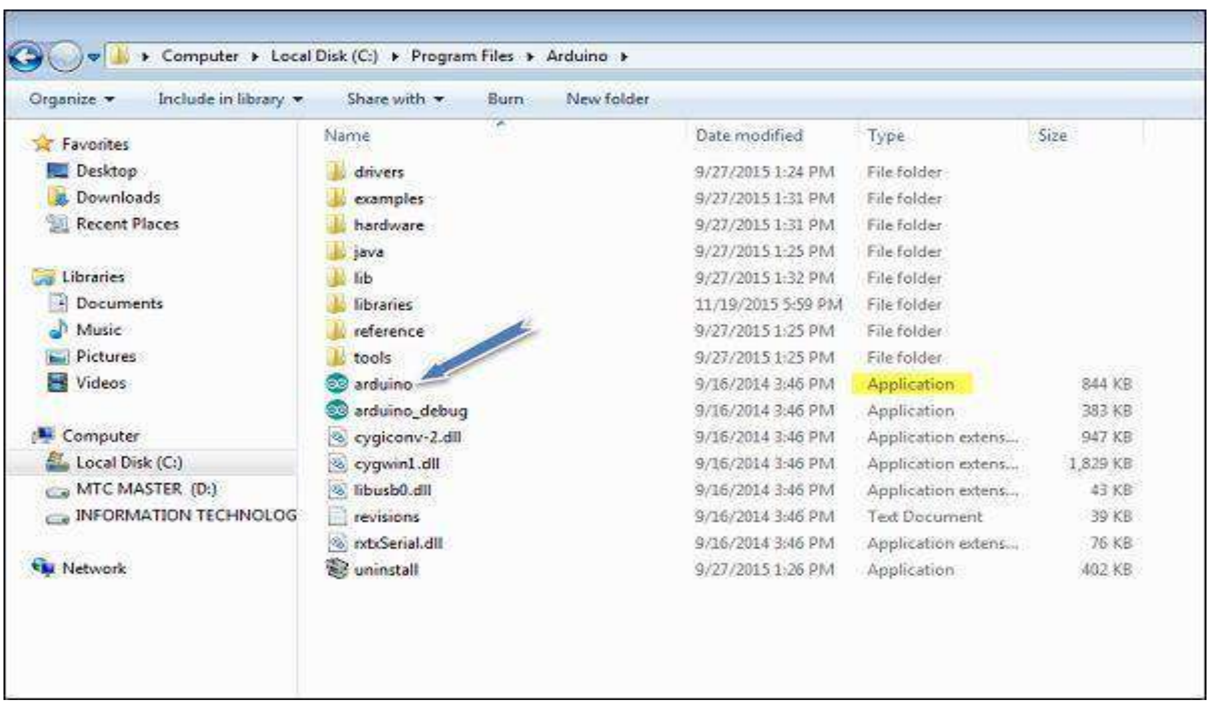


Fig. 3.2.5.4.4 Launch Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**

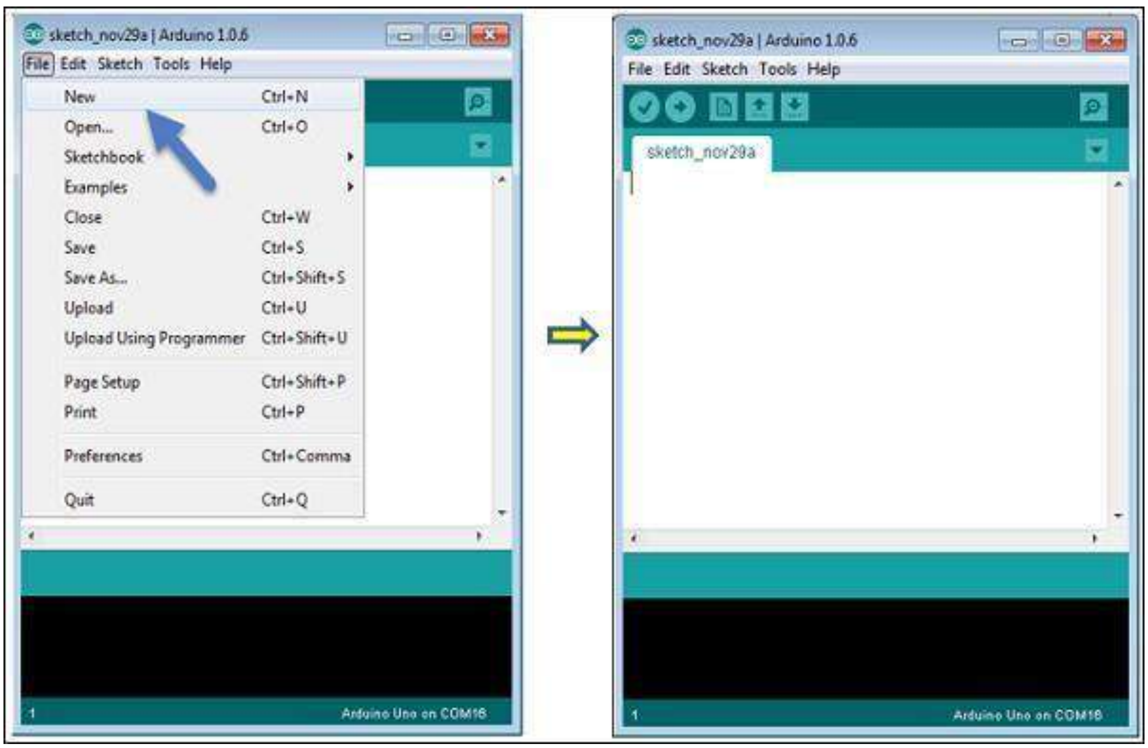


Fig. 3.2.5.4.5 Open your first project.

To open an existing project example, select File → Example → Basics → Blink.

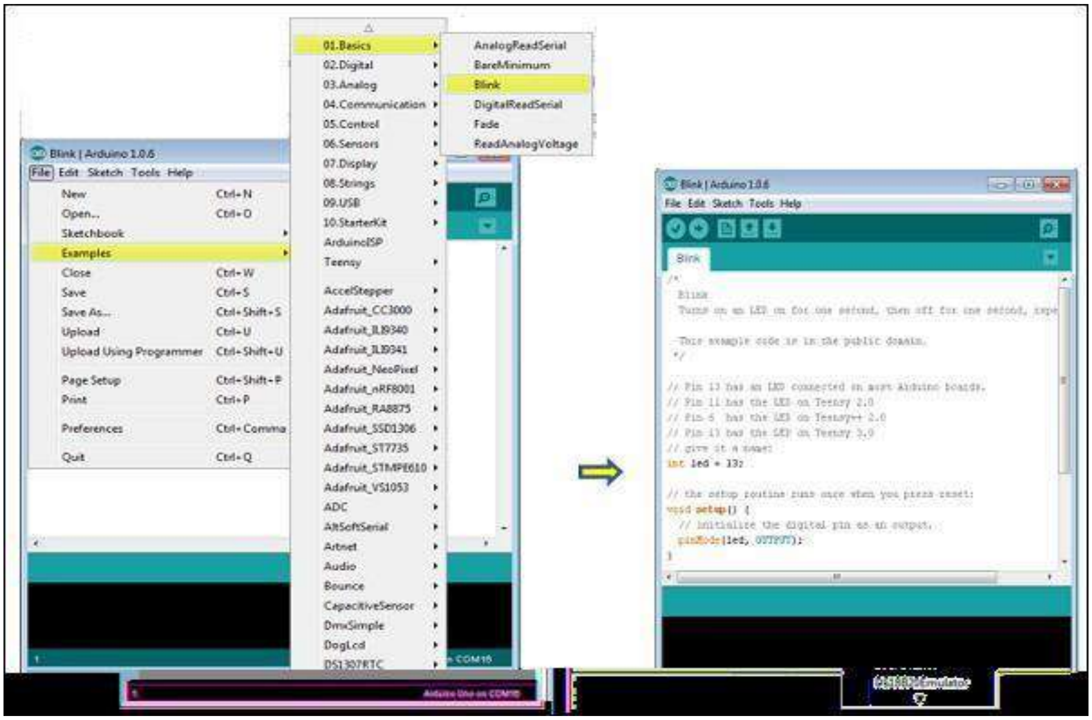


Fig. 3.2.5.4.6 Open your first program.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

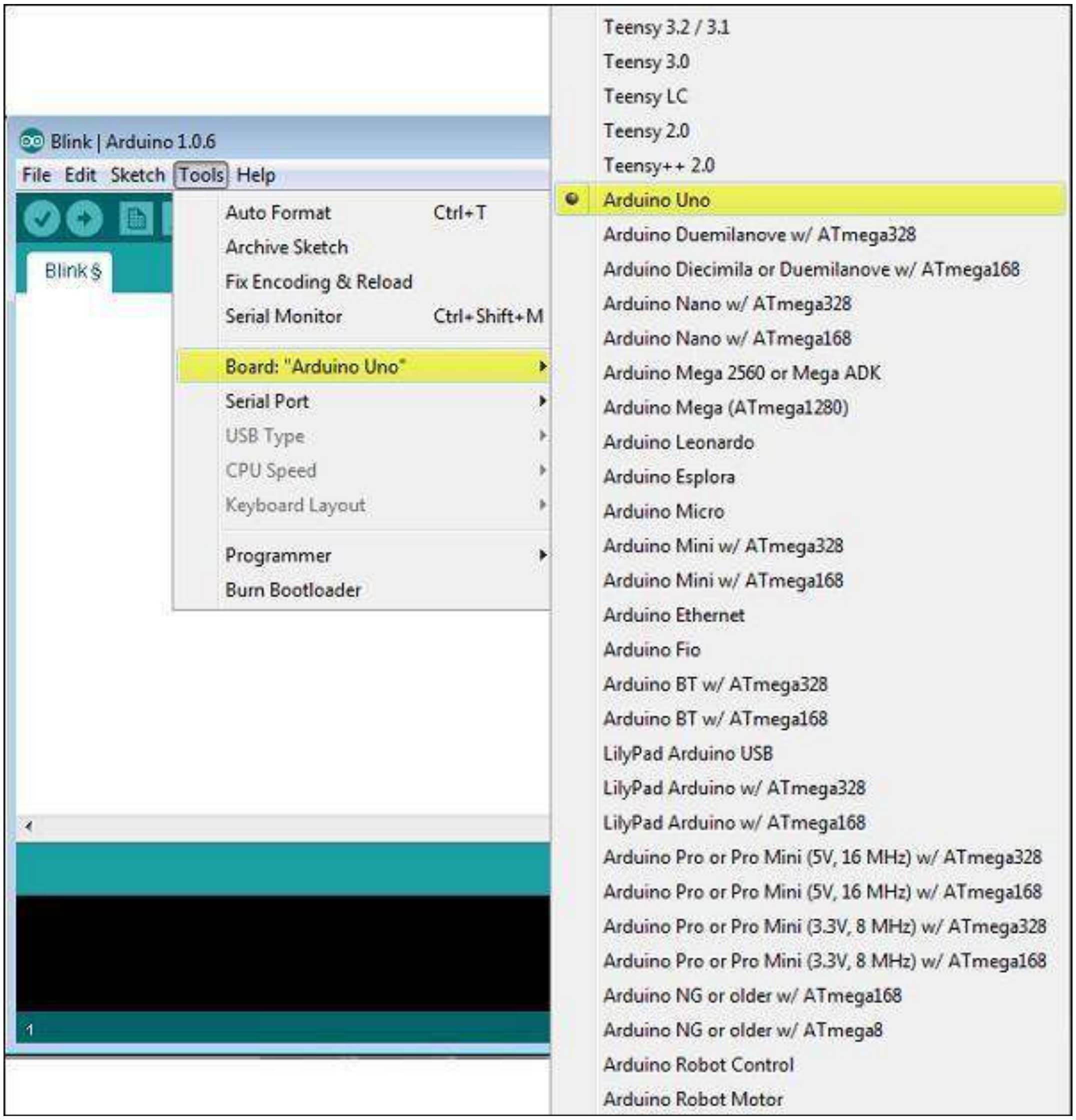


Fig. 3.2.5.4.7 Select your Arduino board.

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

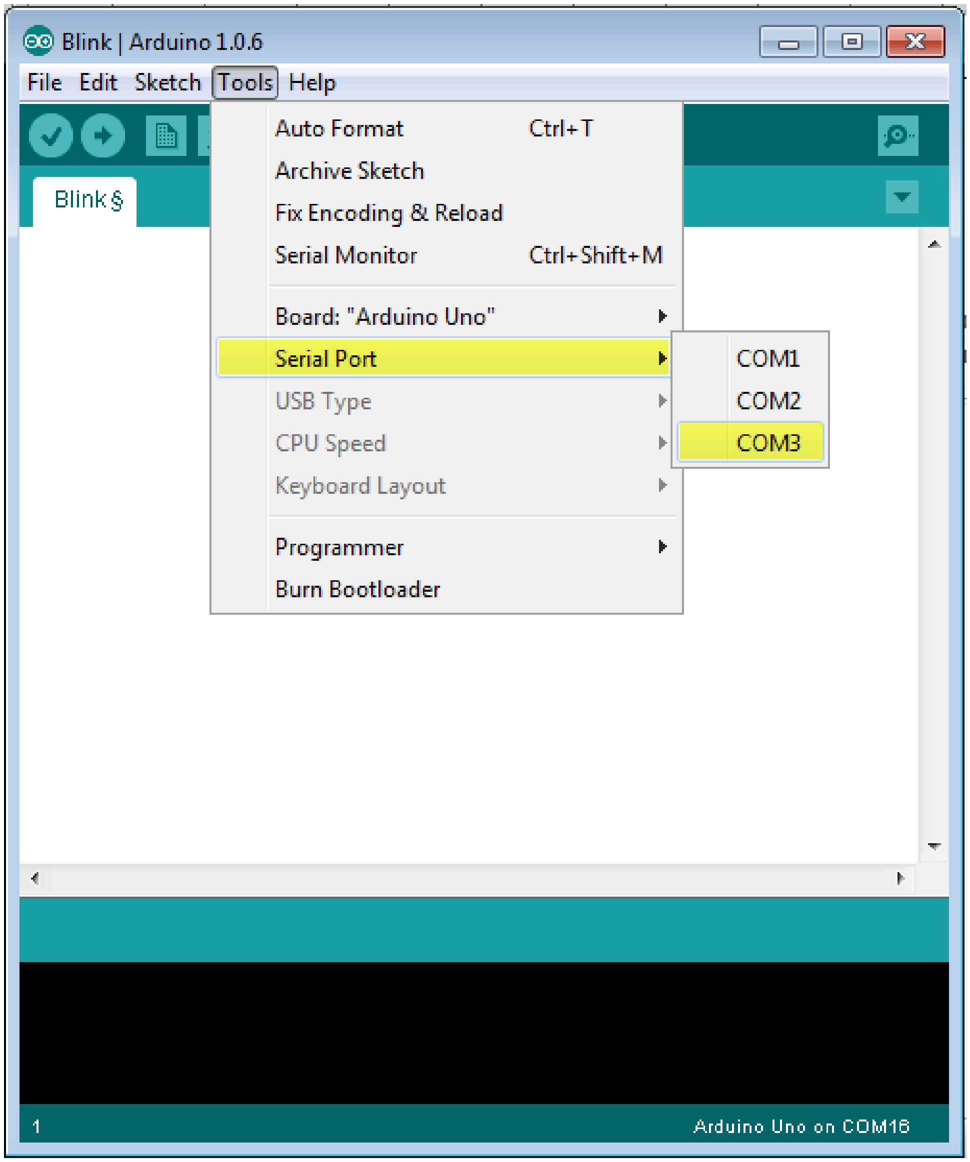


Fig. 3.2.5.4.8 Select your serial port.

Step 8 – Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

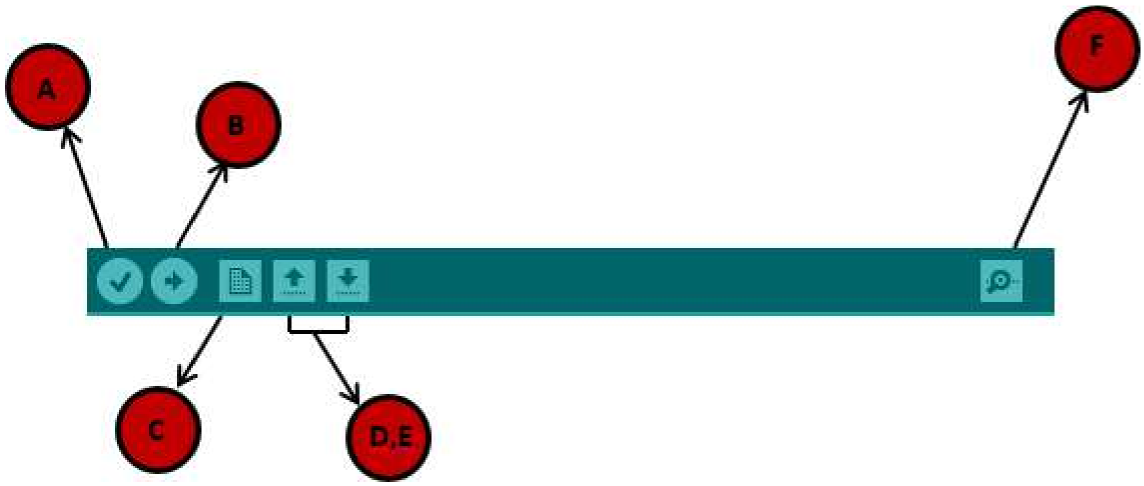


Fig. 3.2.5.4.9 Upload the program to your board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from board & send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USBconnection to the computer. While the external power can be anywhere in the range of 6 to 24V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

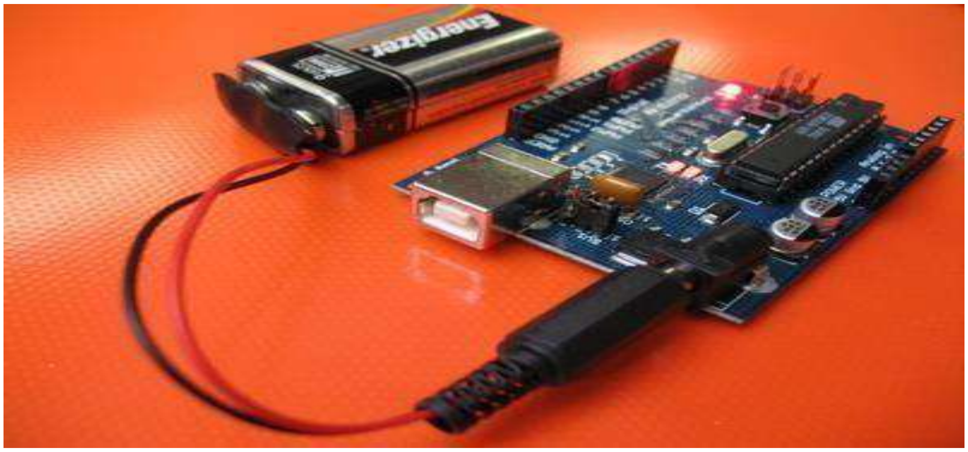


Fig. 3.2.5.4.10 Connecting a Battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called “sketches”, but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

Fig. 3.2.5.4.11 Example Program

3.2.6 PROTEUS:

PROTEUS:

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline "From concept to completion".

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design:

Step 1: Open ISIS software and select New design in File menu

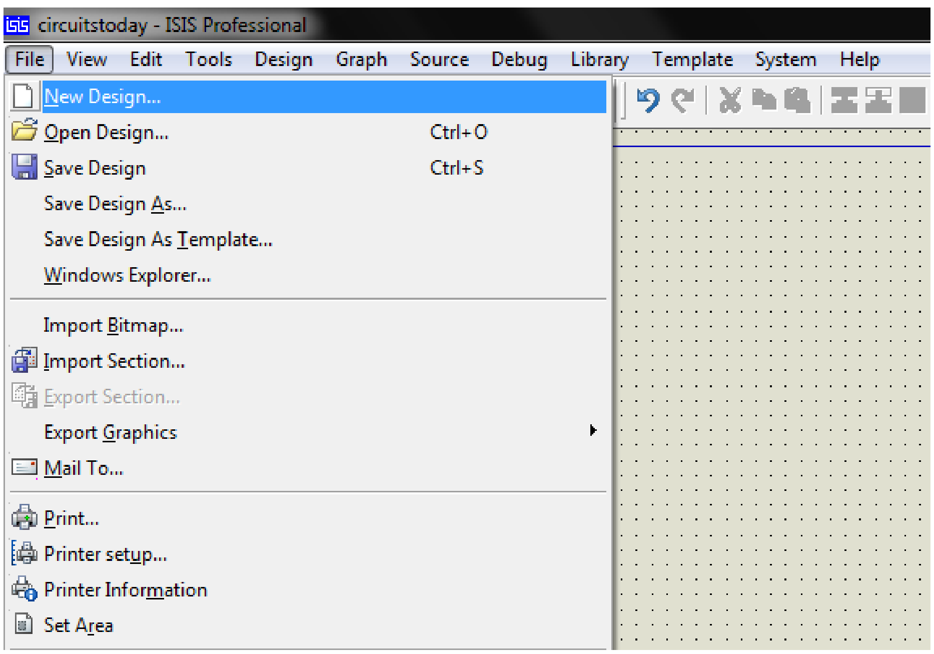


Fig. 3.2.6.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

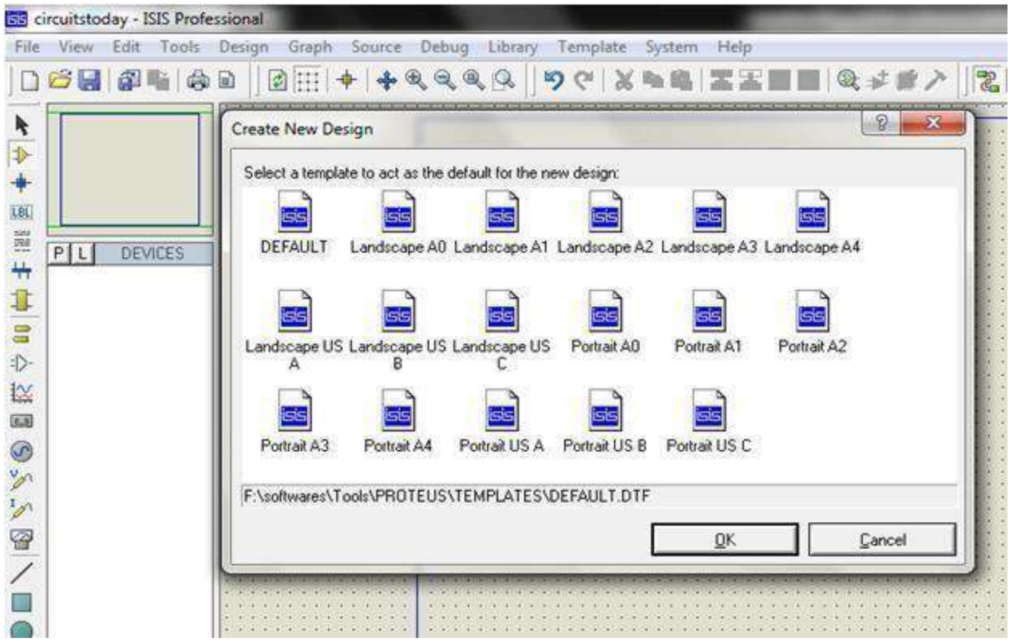


Fig. 3.2.6.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to createa new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

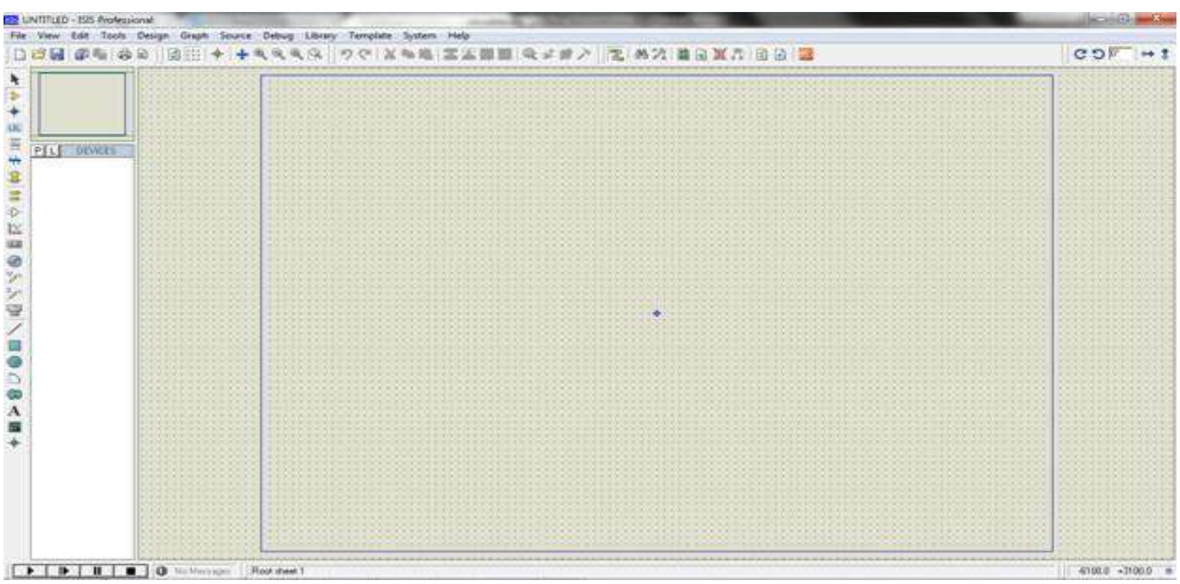


Fig. 3.2.6.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

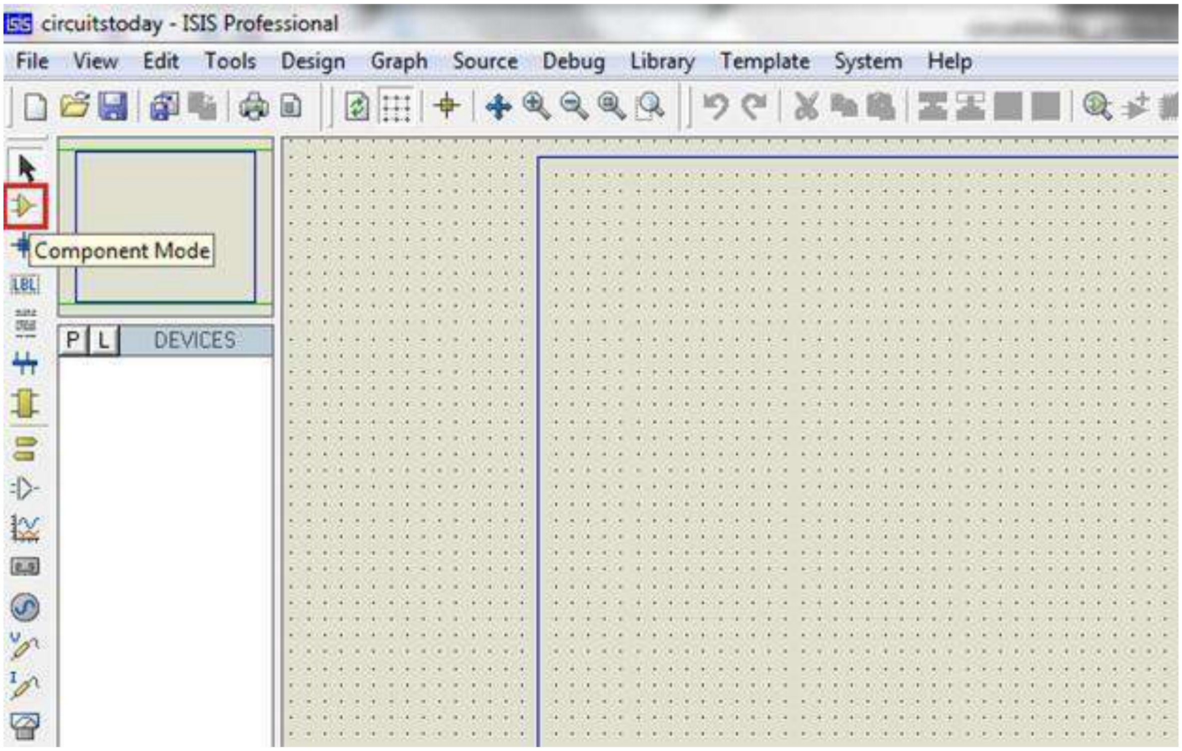


Fig. 3.2.6.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

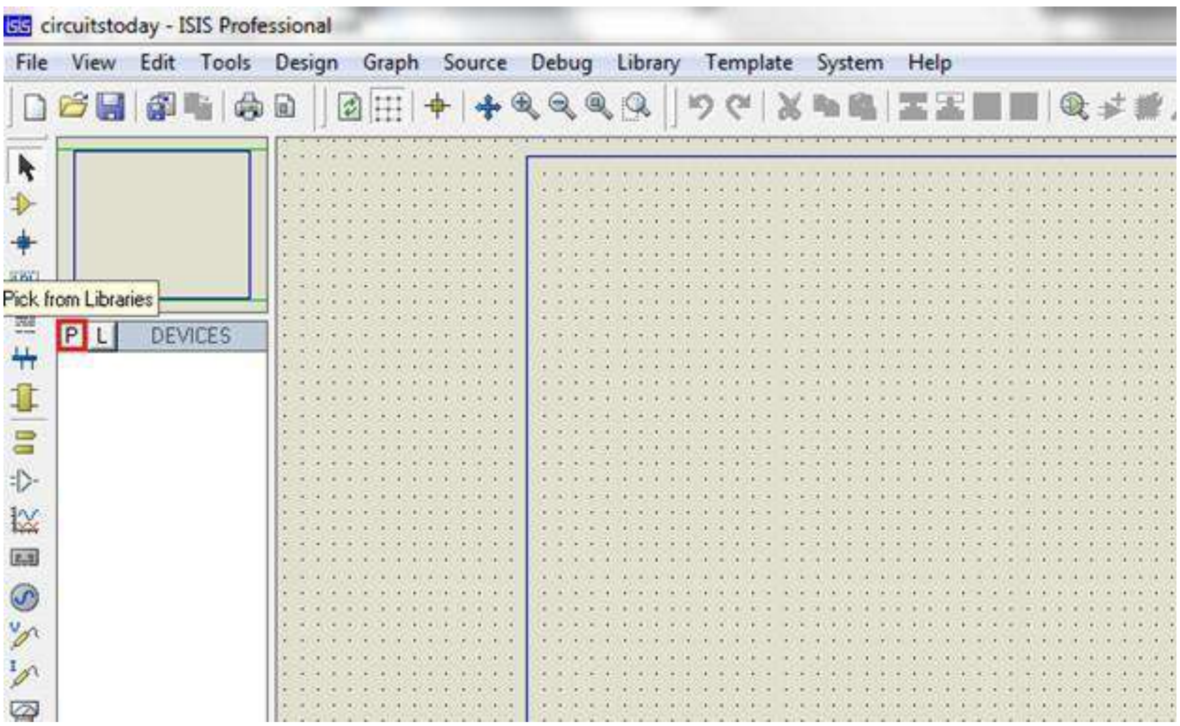


Fig. 3.2.6.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

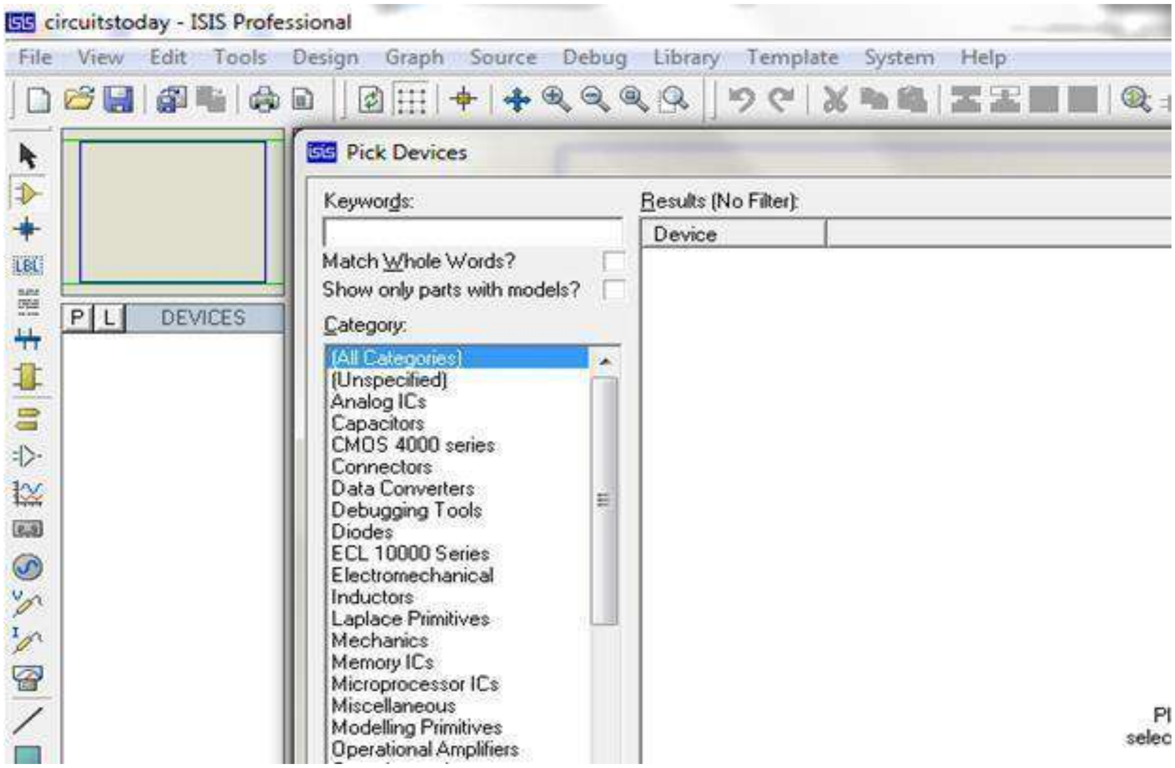


Fig. 3.2.6.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

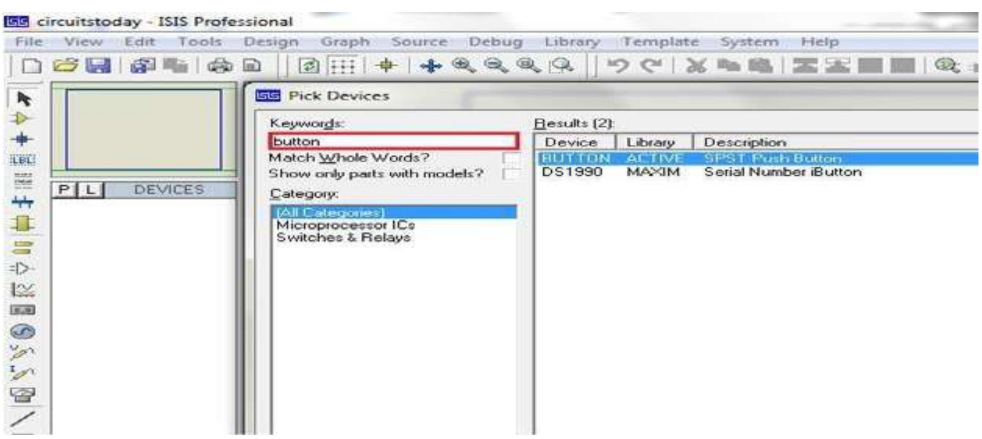


Fig. 3.2.6.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

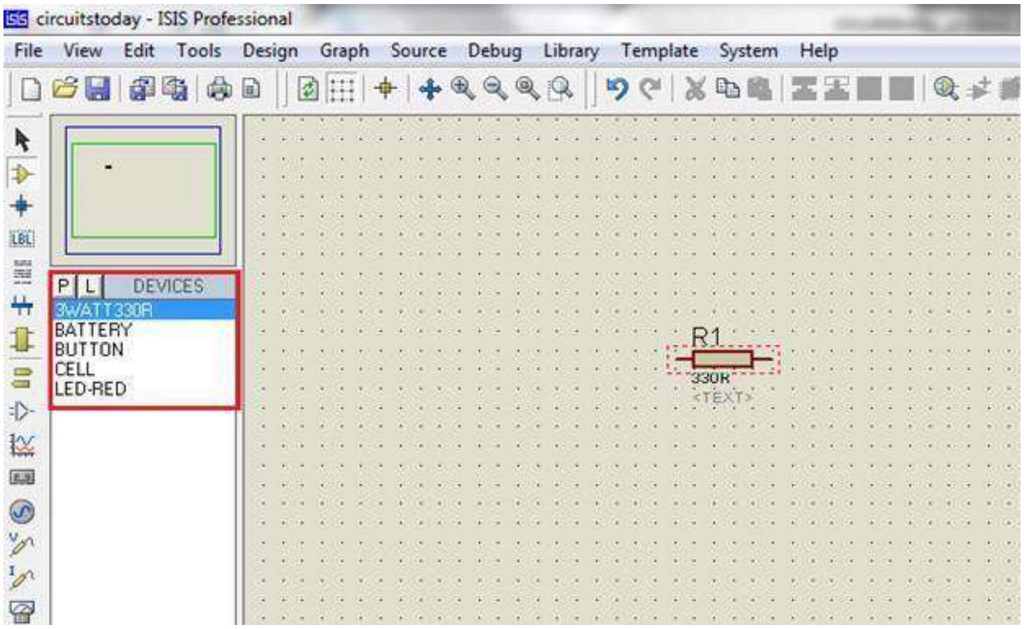


Fig. 3.2.6.8 Component Selection

Place all the required components and route the wires i.e., make connections. Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

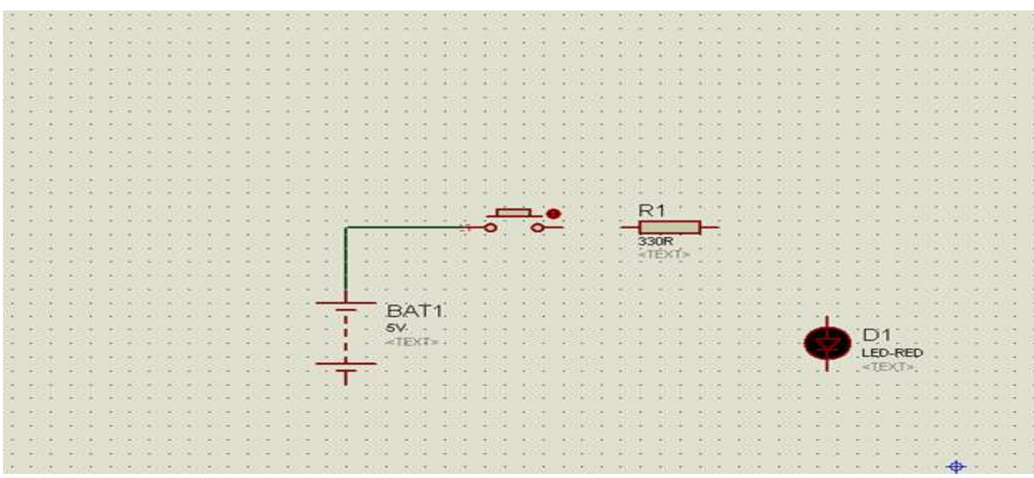


Fig. 3.2.6.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

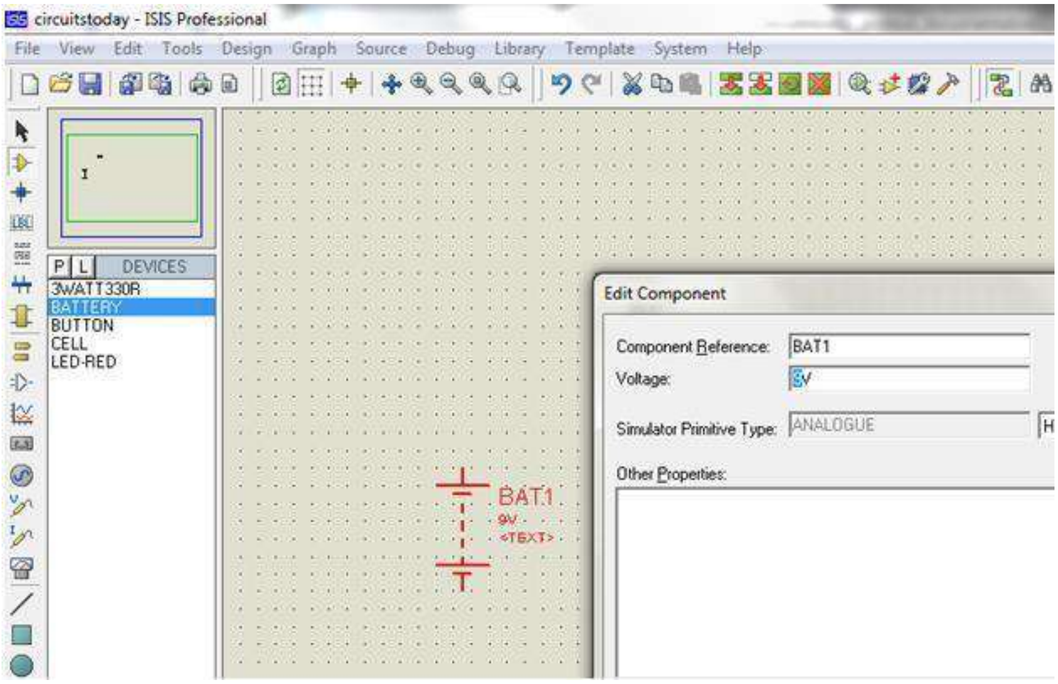


Fig. 3.2.6.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

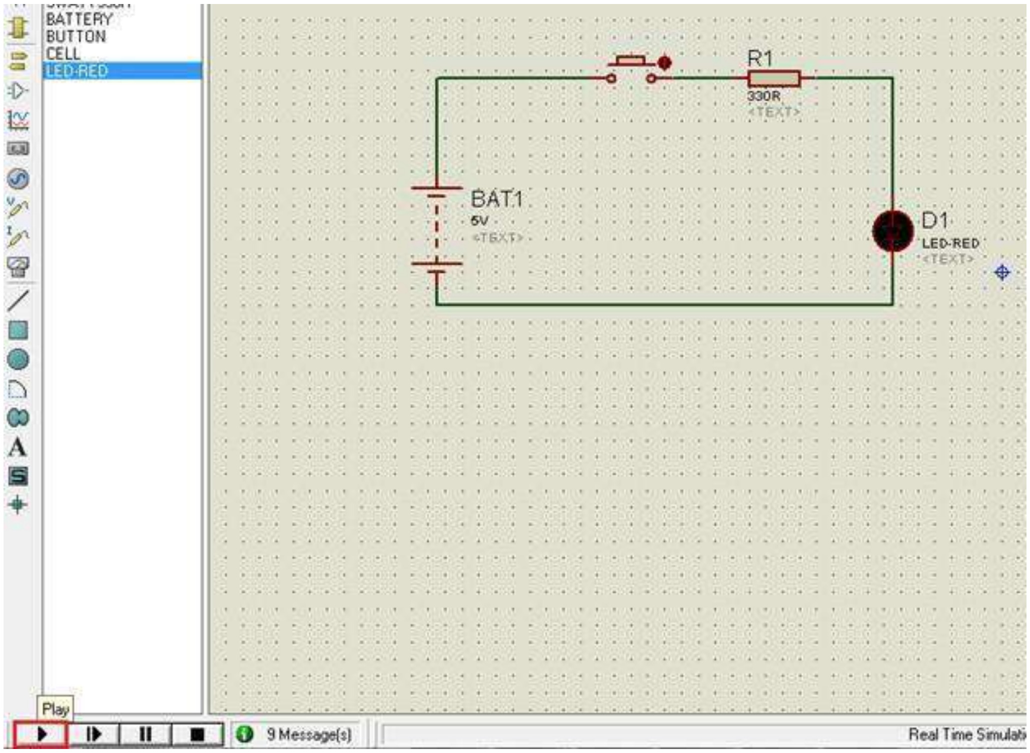


Fig. 3.2.6.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

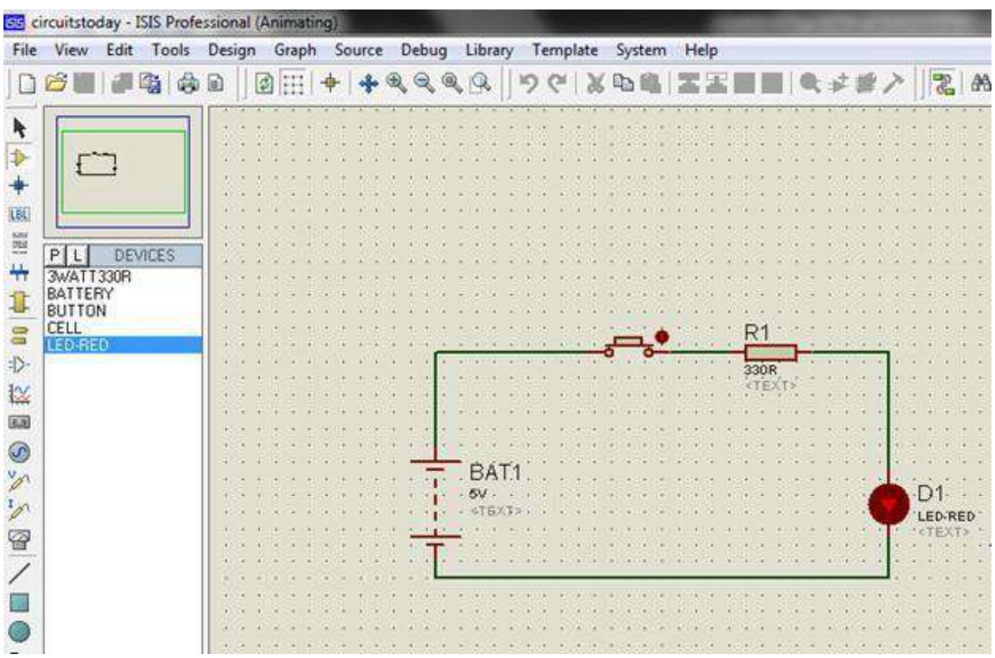


Fig . 3.2.6.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

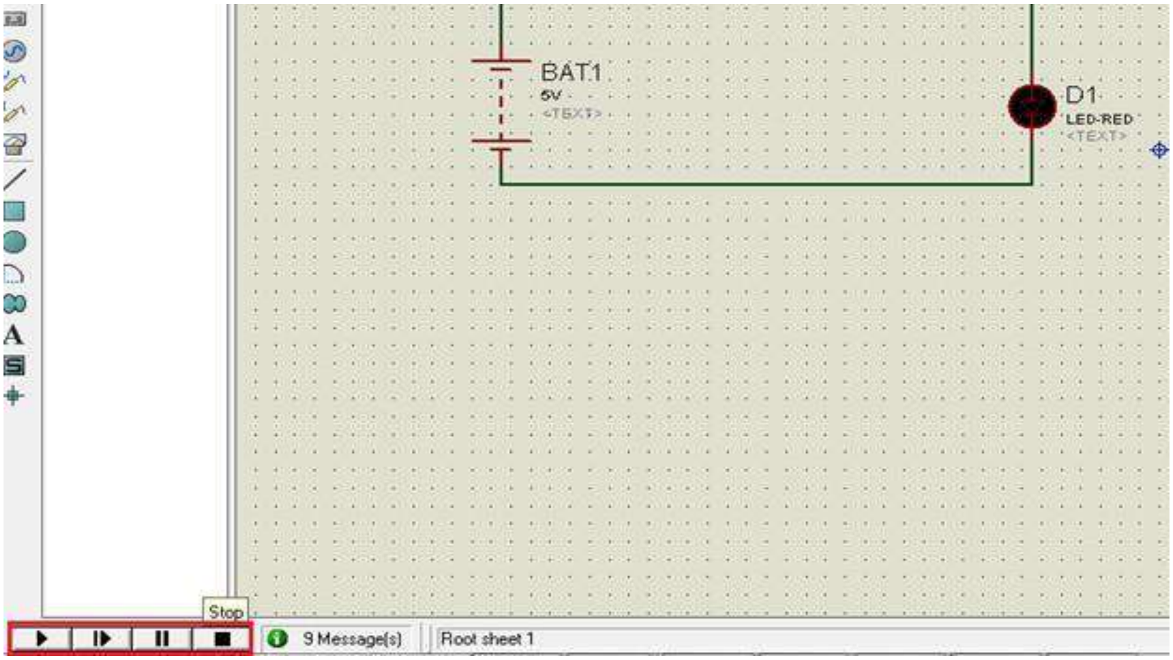


Fig. 3.2.6.13 Simulation Step-Pause-Stop Buttons

According to its developers, Things Speak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. Things Speak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates.

Things Speak was originally launched by iot Bridge in 2010 as a service in support of IoT applications.

Things Speak has integrated support from the numerical computing software MATLAB from MathWorks, allowing Things Speak users to analyse and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license from Math works.[citation needed].

Things Speak has a close relationship with Math works, Inc. In fact, all of the Things Speak documentation is incorporated into the Math works' Matlab documentation site and even enabling registered Math works user accounts as valid login credentials on the Things Speak website. The terms of service and privacy policy of ThingSpeak.com are between the agreeing user and Math works, Inc.

Setup Things Speak

Create a free MathWorks account or sign into Things Speak using an existing account.

Select the Things Speak channel you want your data to stream into. See Collect Data in a New Channel for help creating a new channel.

Record the following for the selected channel:

Channel ID, which is listed at the top of the channel view.

Write API key, which can be found on the API Keys tab of your channel view.

3.2.7 TELNET MOBILE APP:

Telnet is a client-server protocol, based on a reliable connection-oriented transport. Typically, this protocol is used to establish a connection to Transmission Control Protocol (TCP) port number 23, where a Telnet server application (telnetd) is listening. Telnet, however, predates TCP/IP and was originally run over Network Control Program (NCP) protocols.

Even though Telnet was an ad hoc protocol with no official definition until March 5, 1973,[3] the name actually referred to *Teletype Over Network Protocol* as the RFC 206 (NIC 7176) on Telnet makes the connection clear:[4]

The TELNET protocol is based upon the notion of a virtual teletype, employing a 7-bit ASCII character set. The primary function of a User TELNET, then, is to provide the means by which its users can 'hit' all the keys on that virtual teletype.[5]

Essentially, it used an 8-bit channel to exchange 7-bit ASCII data. Any byte with the high bit set was a special Telnet character. On March 5, 1973, a Telnet protocol standard was defined at UCLA[6] with the publication of two NIC documents: Telnet Protocol Specification, NIC 15372, and Telnet Option Specifications, NIC 15373.

Many extensions were made for Telnet because of its negotiable options protocol architecture. Some of these extensions have been adopted as Internet standards, IETF documents STD 27 through STD 32. Some extensions have been widely implemented and others are proposed standards on the IETF standards track (see below) Telnet is best understood in the context of a user with a simple terminal using the local Telnet program (known as the client program) to run a logon session on a remote computer where the user's communications needs are handled by a Telnet server program.

When Telnet was initially developed in 1969, most users of networked computers were in the computer departments of academic institutions, or at large private and government research facilities. In this environment, security was not nearly as much a concern as it became after the bandwidth explosion of the 1990s. The rise in the number of people with access to the Internet, and by extension the number of people attempting to hack other people's servers, made encrypted alternatives necessary.

3.3 DEFINE THE MODULES:

3.3.1 ESP8266 WIFI MODULE

3.3.1.1 INTRODUCTION

The **ESP8266 Wi-Fi Module** is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your **Wi-Fi** network. The **ESP8266** is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

3.3.1.2 CHARACTER CONFIGURATION:

- 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2),
- general-purpose input/output (16GPIO),
- Inter-Integrated Circuit (I^2C) serial communication protocol,
- analog-to-digital conversion (10-bitADC)
- Serial Peripheral Interface (SPI) serial communication protocol,
- I^2S (Inter-IC Sound) interfaces with DMA(Direct Memory Access) (sharing pins with GPIO),
- UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2),and
- Pulse-width modulation (PWM).

3.3.1.3 APPLICATIONS:

- Smart Home Appliances
- Improved Home safety and security

- Home air quality and water quality monitoring
- Natural Language-based voice assistants.

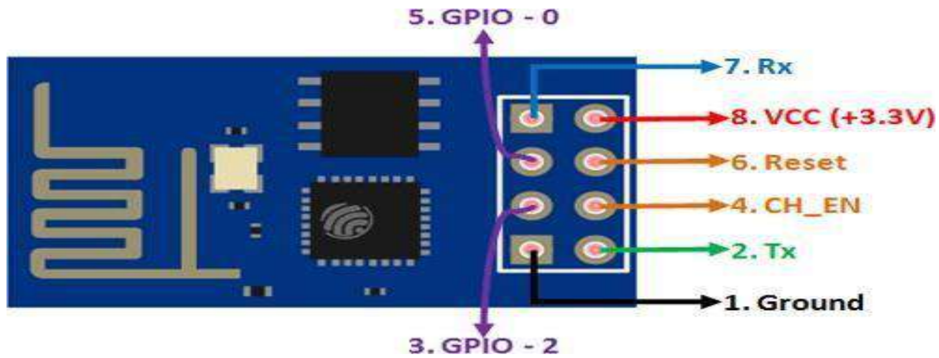


Fig.3.3.1.3 ESP8266 wifi module pin out

3.3.1.4 STRUCTURE AND CONFIGURATION:

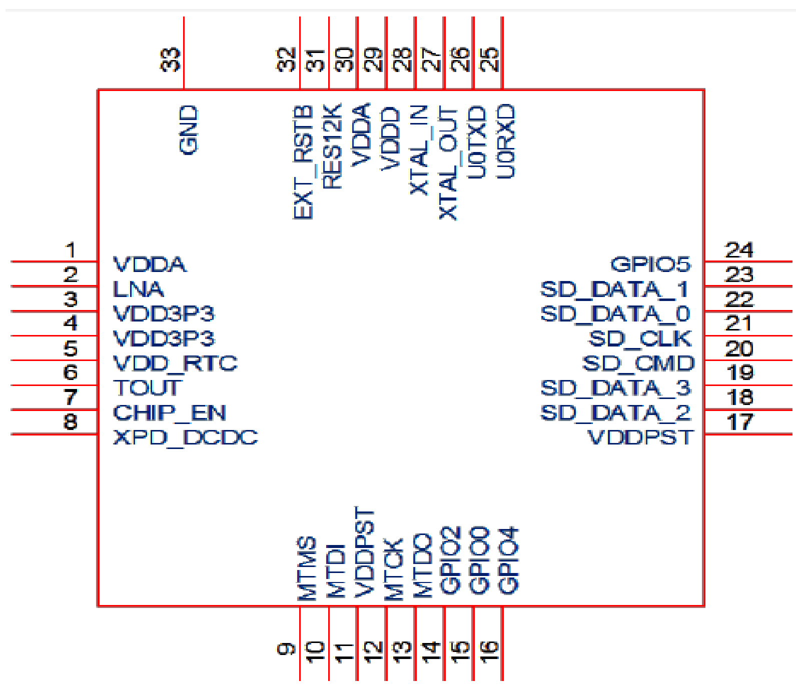


Fig.3.3.1.4 Structure and configuration

3.4 MODULE FUNCTIONALITIES:

3.4.1 ESP8266 WIFI MODULE

The ESP8266 WiFi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all WiFi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existance interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support. In the Documents section below you will find many resources to aid you in using the ESP8266, even instructions on how to transform this module into an IoT (Internet of Things) solution!

CHAPTER-4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM OF PROJECT:

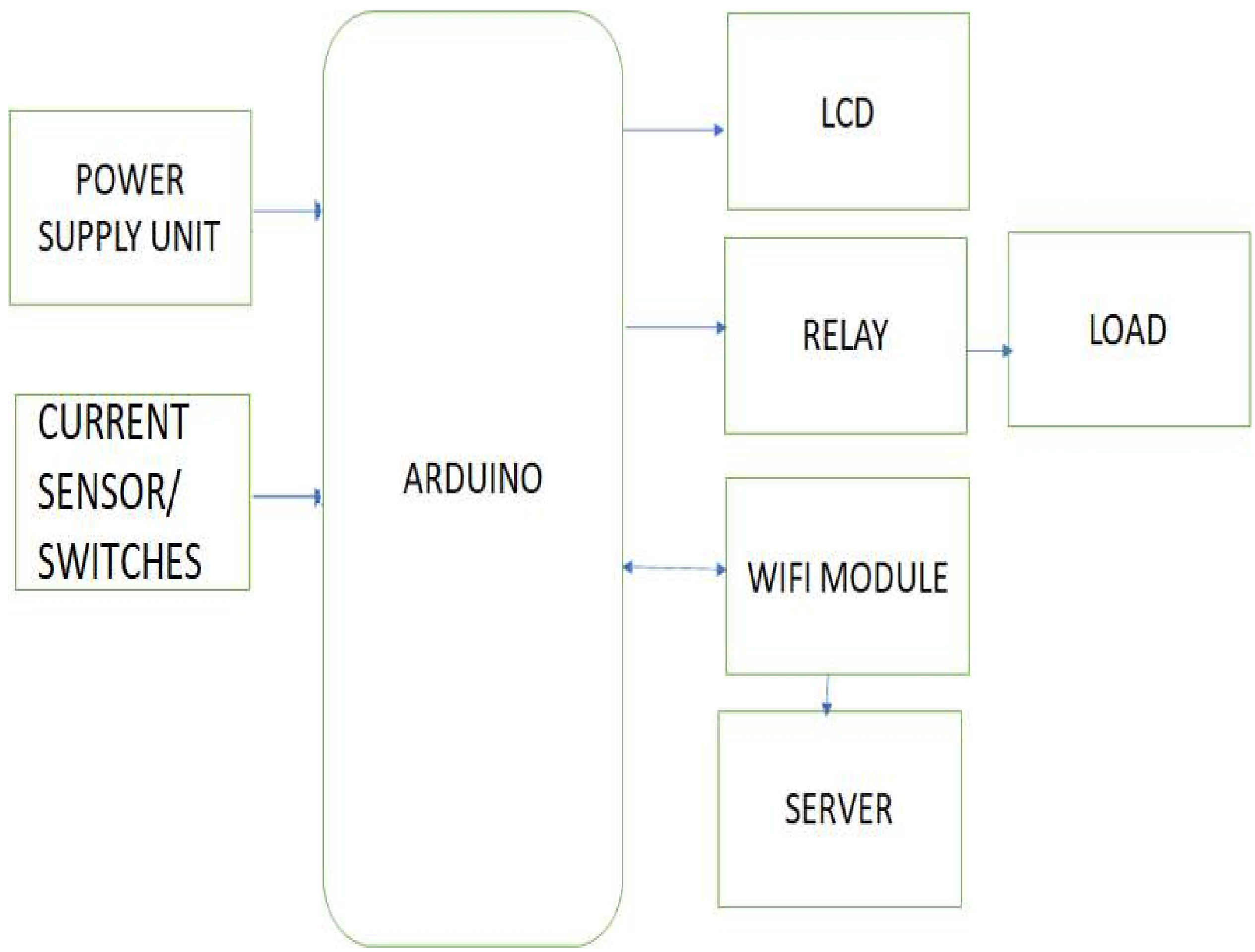


Fig.4.1: Block diagram of the project

4.2 FLOW CHART

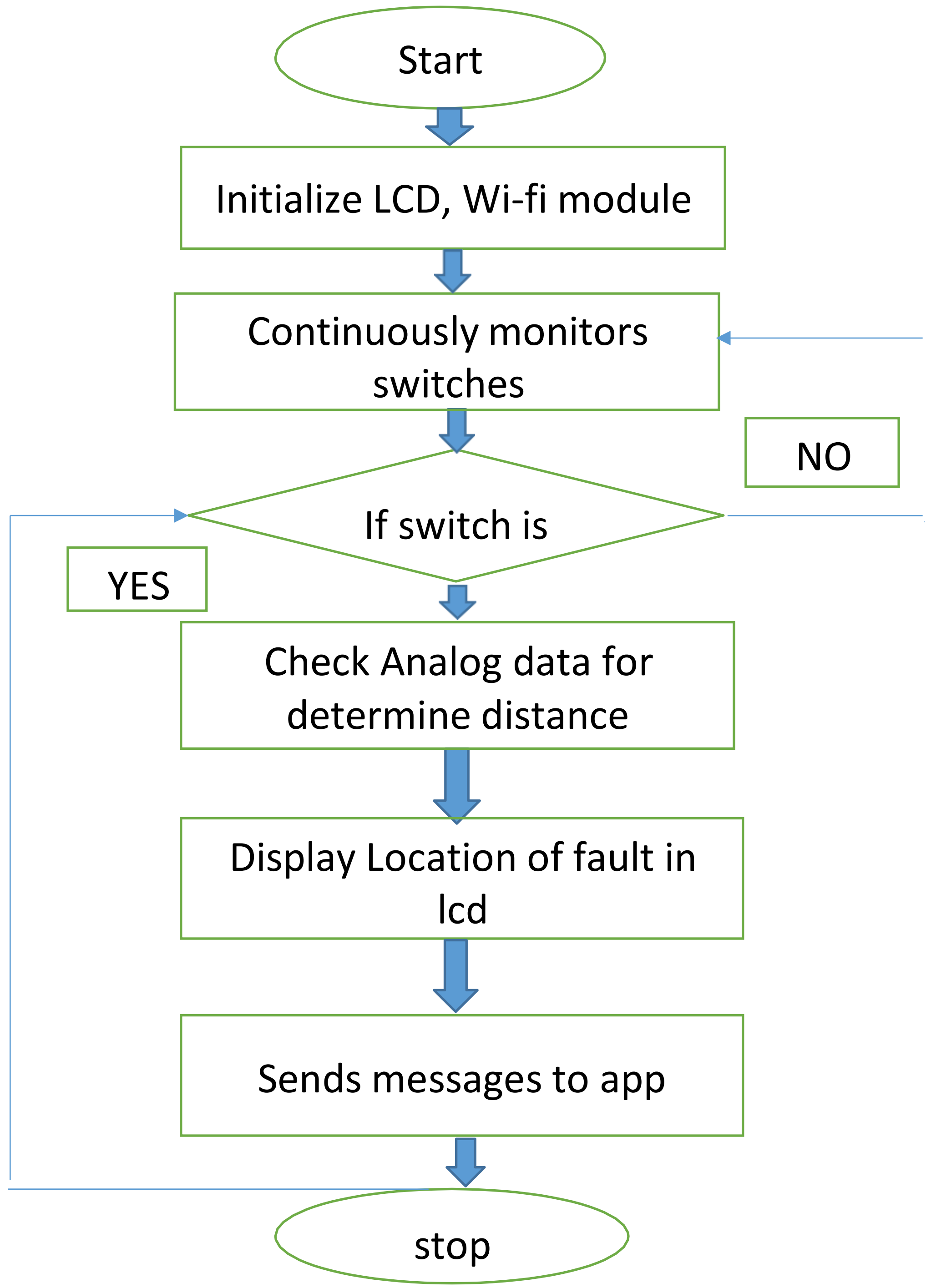


Fig.4.2 Flow Chart

4.3 IMPLEMENTATION STAGES

Stage-1: First on lcd we will get a message displaying waiting for link that indicates the wifi is not yet connected.

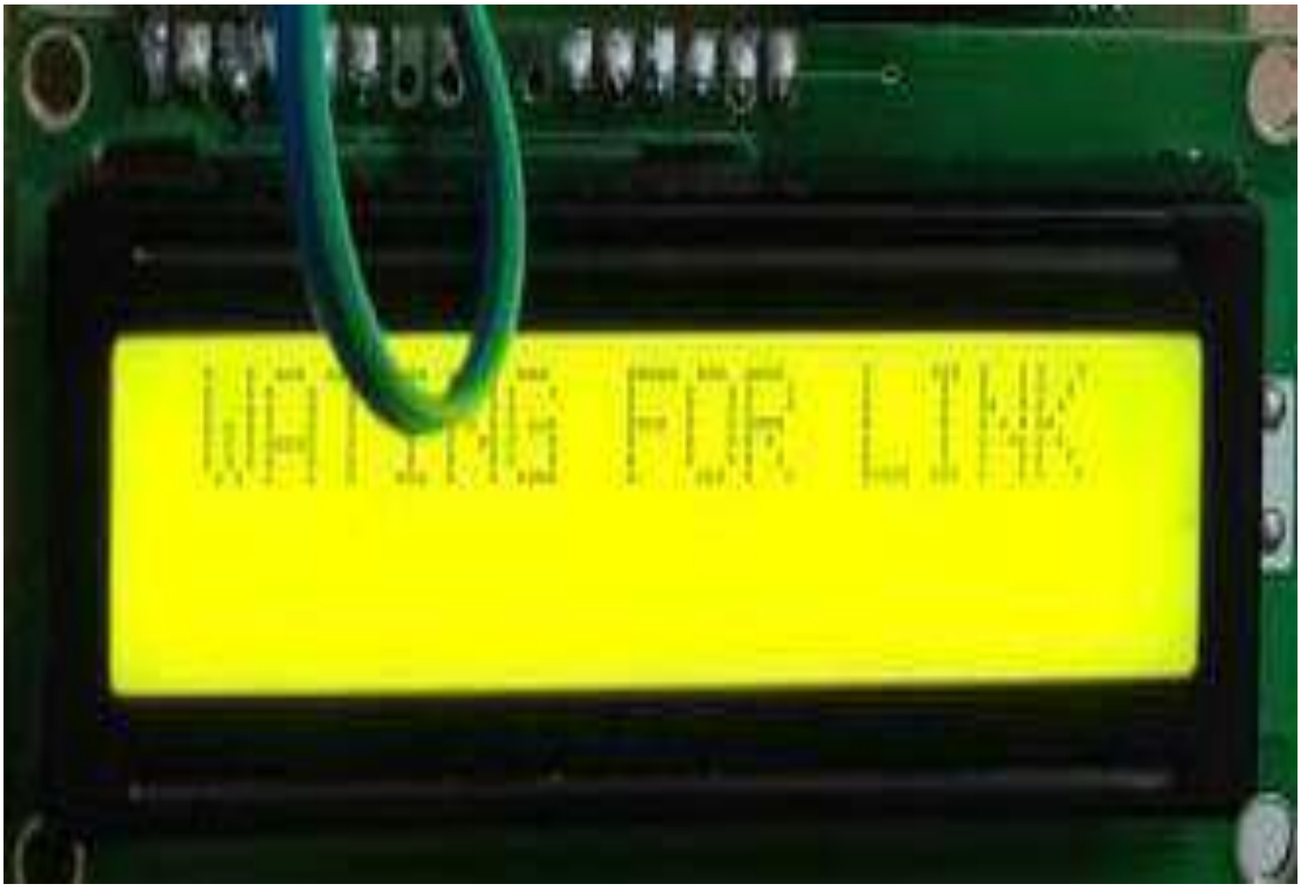


Fig.4.3.1 WAITING FOR LINK

Stage-2: In Mobile Telnet App, we will get a message like welcome when the wifi is connected to ESP8266 WIFI Module.

Fig.4.3.2 WELCOME

Stage 3: When Switch is pressed ,fault is created and the corresponding location is displayed on the LCD

Fig.4.3.3 WELCOME

Stage 4: When Switch is pressed ,fault is created and the corresponding location is displayed on the App

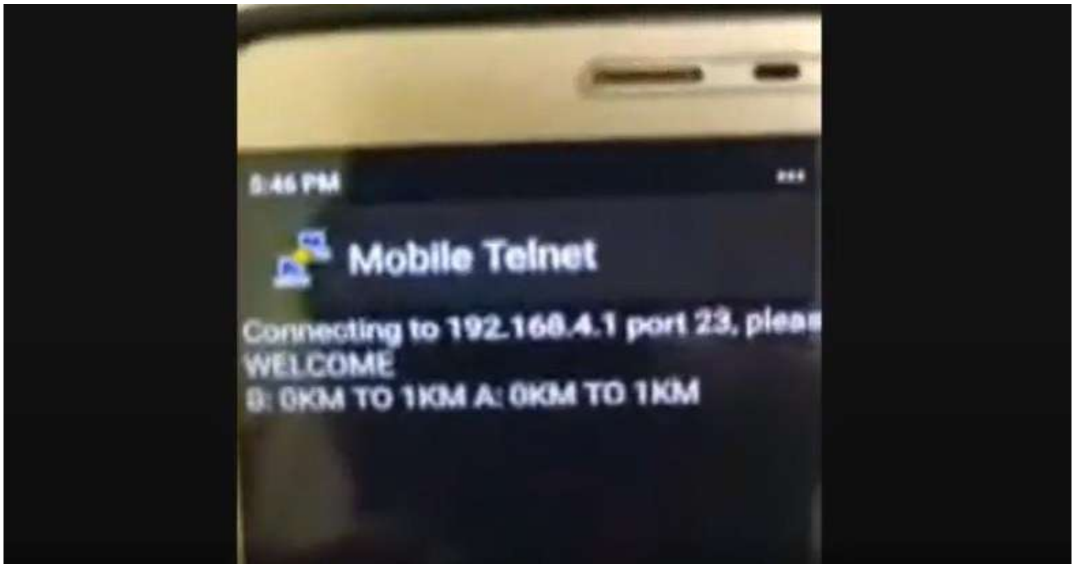


Fig.4.3.4 WELCOME

Stage 5: When there is no fault, “Everything Fine” is displayed on the LCD

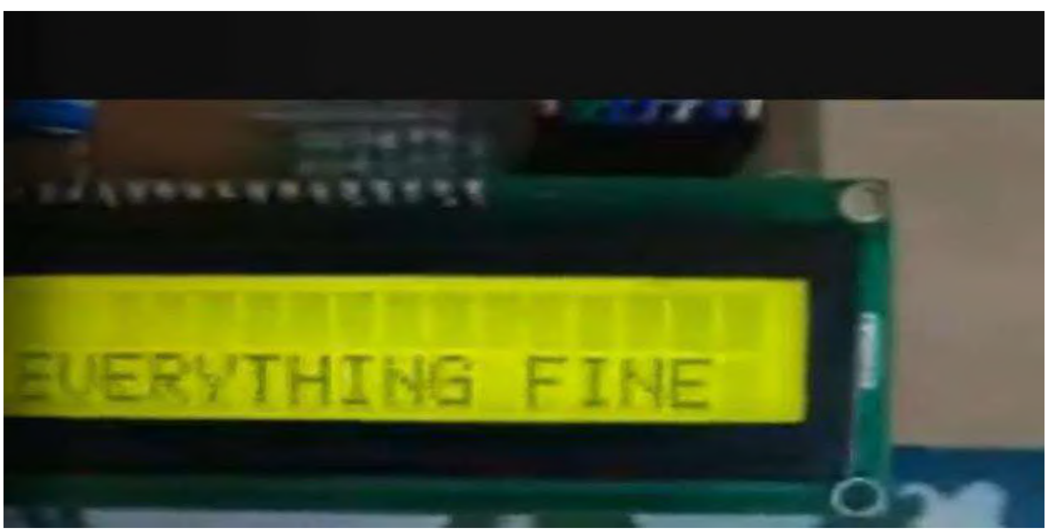


Fig.4.3.5 Output displayed on the LCD when there is no fault

4.4 PROJECT CODE

```
#include <SoftwareSerial.h>
//SoftwareSerial mySerial(2,3);

#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);
int sw1=2,sw2=3,sw3=4,sw4=5,load=6;
void serialFlush(){
while(Serial.available() > 0) {
    char t = Serial.read();
  }
}
void sendwifi(String chr,unsigned int len)
{
  Serial.print("AT+CIPSEND=0,")
Serial.println(len-1);
delay(2000);
Serial.print(chr);
delay(2000);
}

void setup()
{

   pinMode(load,OUTPUT);
   pinMode(sw1,INPUT_PULLUP);
   pinMode(sw2,INPUT_PULLUP);
   pinMode(sw3,INPUT_PULLUP);
   pinMode(sw4,INPUT_PULLUP);
   digitalWrite(load,HIGH);
```

```
Serial.begin(115200);
lcd.begin(16,2);
lcd.clear();lcd.setCursor(0, 0);lcd.print("WELCOME");
 Serial.print("AT\r\n");
 delay(1000);
 Serial.print("ATE0\r\n");
 delay(1000);
 Serial.print("AT+CWMODE=3\r\n");
 delay(1000);
 Serial.print("AT+CWSAP=\"Project\",\"project1235\",5,0\r\n\r\n");
 delay(1000);
 Serial.print("AT+CIPMUX=1\r\n");
 delay(1000);
 Serial.print("AT+CIPSERVER=1,23\r\n");
 delay(1000);

 lcd.clear();lcd.setCursor(0, 0);lcd.print("WATING FOR LINK");
 while(1)
 {
   if(Serial.available())
   {
     //if(Esp.find("0,LINK"))
    if(Serial.find("0,CONNECT"))
    {
       lcd.clear();lcd.setCursor(0, 0);lcd.print("LINK FOUND");
     break;
    }

   }
 }
delay(1000);
```

```
    sendwifi("WELCOME \r\n",11);
    delay(1000);
}


void loop()
{
    int sw1d = digitalRead(sw1);
    int sw2d = digitalRead(sw2);
    int sw3d = digitalRead(sw3);
    int sw4d = digitalRead(sw4);

   // lcd.clear();lcd.setCursor(0, 0);lcd.print("S2:");lcd.print(sw2d);lcd.print("      ");
    delay(1000);

    if(sw1d == LOW )
    {
        lcd.clear();lcd.setCursor(0, 1);lcd.print("A: 0KM TO 1KM ");
        sendwifi("A: 0KM TO 1KM \r\n",15);
        delay(1000);
        digitalWrite(load,LOW);
        delay(2000);
      }
      else if(sw2d == LOW )
      {
        lcd.clear();lcd.setCursor(0, 1);lcd.print("A: 1KM TO 2KM ");
        sendwifi("A: 1KM TO 2KM \r\n",15);
        delay(1000);
        digitalWrite(load,LOW);
        delay(2000);
      }
```

```
 else if(sw3d == LOW )
{
  lcd.clear();lcd.setCursor(0, 1);lcd.print("B: 0KM TO 1KM ");
  sendwifi("B: 0KM TO 1KM \r\n",15);
  delay(1000);
  digitalWrite(load,LOW);
  delay(2000);
}
else if(sw4d == LOW )
{
  lcd.clear();lcd.setCursor(0, 1);lcd.print("B: 1KM TO 2KM ");
  sendwifi("B: 1KM TO 2KM \r\n",15);
  delay(1000);
  digitalWrite(load,LOW);
  delay(2000);
}
else
{
  lcd.clear();lcd.setCursor(0, 1);lcd.print("EVERYTHING FINE ");
  delay(100);
  digitalWrite(load,HIGH);
  delay(1000);
}

}//loop
```

4.5 RESULTS

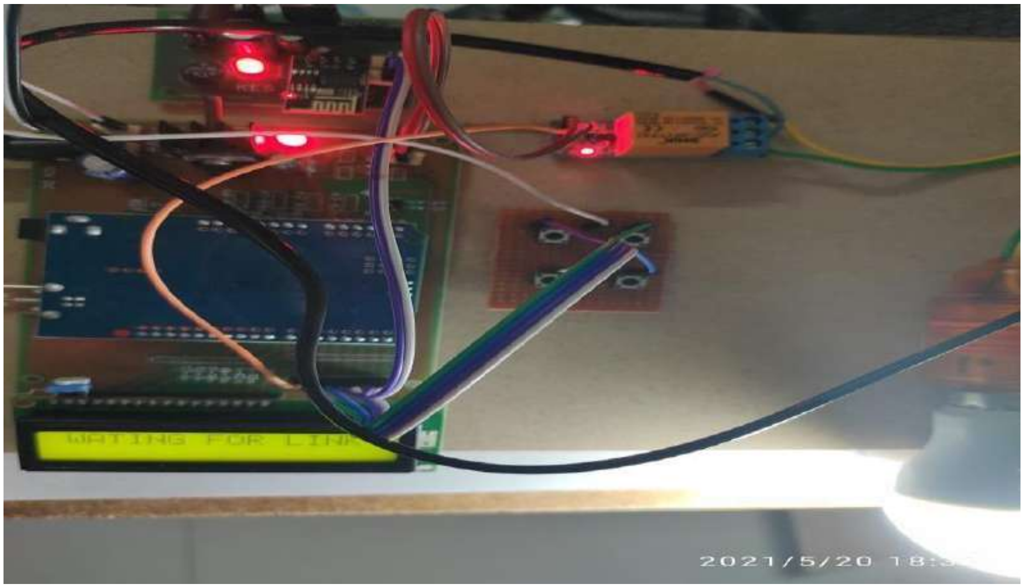


Figure 4.5.1: Output image of Load (bulb) glowing indicating there is no fault

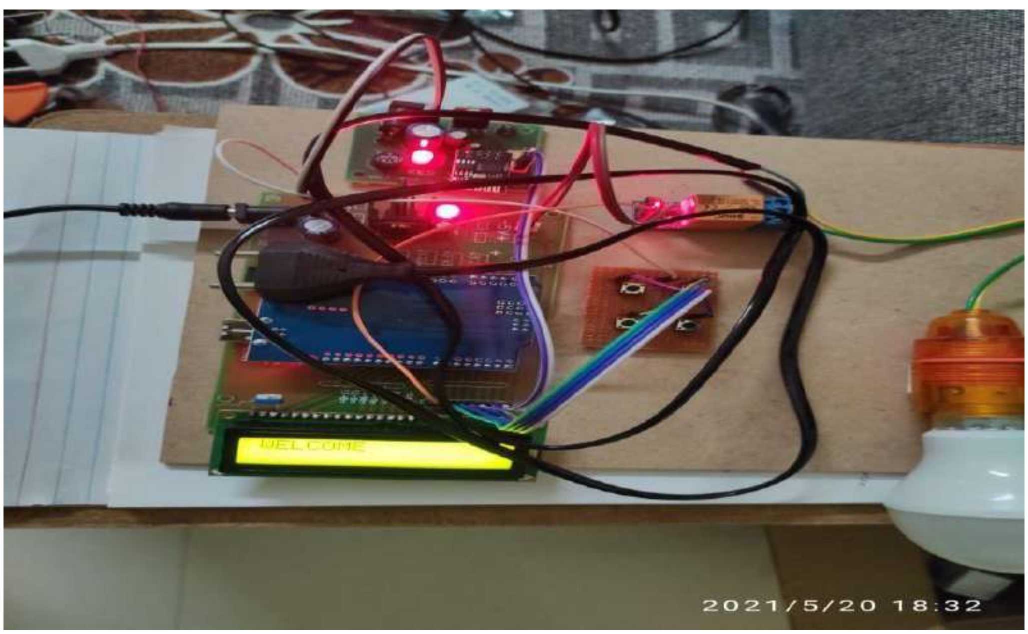


Figure 4.5.2: Output image of Load (bulb) off indicating there is a fault

CHAPTER-5

PROJECT TESTING

In general, fault location techniques for underground cable network can be categorized in two groups:

Tracer method

The tracer method is an exhaustive way to locate a faulted segment by ―walking‖ through the cable circuits. A faulted segment can be determined from audible or electromagnetic signals and requires dispatching crew member stop the outage area. There have been various techniques largely used in the industries, including the tracing approach

through acoustic, electromagnetic or current.

Terminal method

The terminal method is a technique used to determine a fault location of a distribution cable network from one or both ends without tracing exhaustively. A bridge technique is one of the most popular terminal methods that links with a resistor to determine a fault location .It is a technique used to detect fault location of cable from one or both ends without tracing.

Security: IoT is a combination of physical objects ("things"), sensors, embedded software and keeps a goal of effective connectivity and faithful data exchange. The current graphs imply that IoT market growing at remarkable rate. However, some of these devices suffer from limited memory, power consumption and processing power. These issues may cause IoT to become penetrable and hence security is at utmost priority to IoT domain.

As stated in a 2015 report by Hewlett-Packard on IoT research [3]:

1. Per 100, 70 devices still use un-encrypted network services
2. Per 100, 90 devices collected at least one piece of personally identifiable information (via device, cloud or mobile app)
3. Per 100, 70 devices (with cloud and mobile app components) enabled an attacker to identify valid user accounts through enumeration.
4. 80% of devices (with cloud and mobile app components) failed to require passwords of

sufficient complexity

5.Behind every 10 device every 6 devices user interfaces (UI) were vulnerable

Following are the vulnerabilities of the IoT as per The Open Web Application Security Project (OWASP)

1) Insufficient protection compositions.
2) Dubiously secured Software/Firmware
3) Doubtful Network Services
4) Doubtful Web Interface
5) Worst Physical Security
6) Doubtful Cloud Interface
7) Doubtful Mobile Interface
8) Privacy Concerns
9) Doubtful Authentication/Authorization
10) Inadequate Transport Encryption Ongoing approaches to secure IoT have attempted to put a grip on communication protocol-based mechanisms, such as encryption for data-at-rest or in-transit.

But this itself is doubtful if the respective endpoints themselves are capable of being modified either by local access or remote connections. Gartner claims that by year 2020 more than 25% of identified attacks in a particular company will be on IoT devices or systems, even though IoT will only contribute to less than 10% of IT security budgets.

While testing our project, we found the following risks

- Unexpected testing Result
- Failure of component
- Impact the quality, timeline, and cost
- Implementation may become difficult or impossible
- Include design, implementation, interface, verification and maintenance problem

While testing our project we found following constraints

- Additional Resources (Cost increase)
- Fast Completion (Risk Increase and Quality Decrease.

CHAPTER-6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The short circuit fault at a particular distance in the underground cable is located to rectify the fault efficiently using simple concepts of Ohms law. The work automatically displays the phase, distance and time of occurrence of fault with the help of ESP8266 Wi - Fi module in a webpage. The benefits of accurate location of fault are fast repair to revive back the power system, it improves the system performance, it reduce the operating expense and the time to locate the faults in the field.

6.2 FUTURE ENHANCEMENT

The work can be extended for open circuit fault, short circuit Line to Line Fault (LL) and double Line to Ground Fault (LLG). The open circuit fault can be detected using a capacitor in ac circuit which measures the change in impedance and calculate the distance of fault.

PUBLICATION

Submitted Paper in the Conference ICSMEC - 21 with Paper ID (ICSMEC21-0056) and got Acceptance for the Paper.

REFERENCES

[1] Xiaoning Kang; Xiuda Ma; Shuai Jiang; Xiaoyun Qu, Chao Zhang; Xiaoning Kang; Xiuda Ma; Shuai Jiang; Xiaoyun Qu 2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)

[2] Gilbert Cheung, Yuan Tian, Tobias Neier, Technics of Locating Underground Cable Faults inside conduits, International Conference on Condition Monitoring and Diagnosis IEEE (CMD 2016)

[3] Nikhil Kumar Sain, Rajesh Kajla, and Mr.Vikas Kumar, Underground Cable Fault Distance Conveyed Over GSM, International Organization of Scientific Research Journal of Electrical and Electronics Engineering, Volume 11, Issue 2, Mar-April 2016.

[4] C.Bharatiraja, S.Jeevananthan, J.L. Munda, A Timing Correction Algorihm based extended SVM for three level Neutral point clamped MLI in Over Modulation Zone IEEE Journal of Emerging and Selected topics in Power Electronics.

[5] ManarJaradat, Moath Jarrah, Abdel Kader Bousselham, YaserJararweh, Mahmoud AlAyyoubThe Internet of Energy: Smart Sensor Networks and Big Data Management for Smart Grid, Procedia Computer Science Elsevier, July 2015.

[6] Dhivya Dharani. A and Sowmya. T, Development of a Prototype of Underground Cable Fault Detector, International Journal Electrical, Electronics, and Computer Systems, Volume-2, 2014.

[7] Md. Fakhrul Islam, Amanullah M T O, Salahuddin. A. Azad, Locating Underground Cable Faults: A Review and Guideline for New Development, 2013 IEEE Conference

[8] M.Fonseca_Badillo, L. Negrete_Navarrete, A. Gonzalez_parada, A. Castaneda_Miranda, Simulation and analysis of underground power cables faults, 2012 Elsevier Procedia Engineering

[9] Abishek Pandey, Nicolas H. Younan Underground cable fault detection and identification using Fourier analysis, 2010 IEEE Conference

[10] Tobias Neier, Cable fault location practical experience, HV Technologies, version-1, June 2006.

[11] M.S. Choi, D.S. Lee, and X. Yang, A Line to Ground Fault Location Algorithm for Underground Cable System, Korean Institute of Electrical Engineers International Transactions on Power Engineering, pp. 267 – 273, Jun 2005.

[12] K.K. Kuan, Prof. K. Warwick, Real-time expert system for fault location on high voltage

underground distribution cables, IEEE Proceedings-C, Volume. 139, No. 3, MAY 1992.

[13] Ashlesha A. Patil and Dr. S. R. Suralkar. Review on-IOT Based Smart Healthcare System. International Journal of Advanced Research in Engineering and Technology, 8(3), 2017, pp 37–42

[14] Snehal R. Shinde, A. H. Karode and Dr. S. R. Suralkar, Review onIOT Based Environment Monitoring System, International Journal of Electronics and Communication Engineering and Technology, 8(2), 2017, pp. 103–108

[15] Viswanath Naik.S1 ,S.Pushpa Bai1 , Rajesh.P , MallikarjunaNaik.B, IOT Based Green House Monitoring System, International Journal of Electronics and Communication Engineering & Technology (IJECET), Volume 6, Issue 6, June (2015), pp. 45-47

BIBILOGRAPHY

- Jamali, S. (2010). Impedance based fault location method for single phase to earth faults in transmission systems. *10th IET International Conference on Developments in Power System Protection.* Manchester, UK.
- Liao, Y. (2009). Yuan Liao . *IEEE*.
- Ngaopitakkul, A. (2011). Study of characteristics for simultaneous faults in distribution underground cable using DWT. *IEEE*.
- P.S, P. (2015). Fault Detection Technique to pinpoint Incipient Fault for Underground Cables. *International Journal of Engineering Research and General Science*.
- Pandey, A. (2010). Underground cable fault detection and identification via fourier analysis. *Conference: High Voltage Engineering and Application.*

APPENDICES

The main aim of this project is to find the accurate and specific location of the fault occur in the underground cable with the help of IoT device and by using Arduino UNO microcontroller kit. The result is shown on a web page and also the message is being sent on the mobile phone with the help of ESP8266. The urban region uses an underground electrical cable wirerather than using the overhead lines. But it is hard to spot the fault location in underground cable accurately that leads to difficulty in repairing. This system finds the specific location of the fault in the cable. The underground cables are prone to get faults due to wear and tear, underground conditions, rodents etc. The whole cable is to be dugout for the invigilation and fault fixing as we not know the fault location accurately. Our purpose is to find the location of fault exactly that isto be fixed for the ease in the process of repairing. The potential divider network system is used to detect the fault that laid across the cable. When lines are short together then a fault is created at that point, a particular voltage is generated as per the network combination of the resistors. The microcontroller sensed the voltage change and send a signal to the user. The information about the location at which that voltage coincide is sent to the user.

A

Major Project Report

On

VEHICLE MOVEMENT STREET LIGHT WITH LIGHT SENSING USING ATMEGA

Submitted by

1) Ms. MAJJI ANUSHA – (17K81A04M3) **2) Ms. SHETLEM TULASI – (17K81A04P4)**

3) Mr. THATI SANTOSH KUMAR – (17K81A04P7)

in partial fulfilment of the requirement for the award of the degree

of

BACHELOR OF TECHNOLOGY

In

Electronics & Communication Engineering

Under The Esteemed Guidance Of

Ms. K. ANITHA

Assistant Professor

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

St. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad-500 100

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA & NAAC A+ Accredited

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the project entitled "VEHICLE MOVEMENT STREET LIGHT WITH LIGHT SENSING ATMEGA**" is being submitted by** 1. MAJJI ANUSHA (17K81A04M3), 2. SHETLEM TULASI (17K81A04P4), 3. THATI SANTOSH KUMAR(17K81A04P7) **in partial fulfilment of the requirements for the award of the degree of** BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING **is recorded of Bonafide work carried out by them. The results embodied in this report have been verified and found satisfactory.**

Head of the Department

Mrs. K. ANITHA — **Dr. B. HARIKRISHNA**

Department of ECE — Department of ECE

Internal Examiner — External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **MAJJI ANUSHA** WITH ROLL NO.**17K81A04M3, SHETLEM TULASI** WITH ROLL NO.**17K81A04P4**, **THATI SANTHOSH KUMAR** WITH ROLL NO.**17K81A04P7,** OF B. TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**VEHICLE MOVEMENT STREET LIGHT WITH LIGHT SENSING ATMEGA**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT

DIRECTOR

TRAININGS & PLACEMENTS

LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014

Email : contact@lasyainfotech.com, ov@lasyainfotech.com

Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of **Bachelor of Technology** in Department of Electronics and Communication Engineering, session: <2017 - 2021>, St. Martin’s Engineering College, Dhulapally, Kompally, Secundrabad, hereby declare that the project work entitled “VEHICLE MOVEMENT STREET LIGHT WITH LIGHT SENSING ATMEGA” is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied is this project report has not been submitted in any university for award of any degree.

MAJJI ANUSHA (17K81A04M3)

SHETLEM TULASI (17K81A04P4)

THATI SANTOSH KUMAR (17K81A04P7)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr. P. Santosh Kumar Patra**, **our beloved Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr. B. HARI KRISHNA**, **Head &Professor**, Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in our memory for years to come.

We would like to express our gratitude to our dept. Project coordinator project coordinator **Dr. A. Anand, Professor** for his involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time to come.

We sincerely thank our Project Internal Guide, **Mrs. K. Anitha, Assistant Professor,** Department of ECE and our class Project Coordinator **Ms**. **P. Pushpa, Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our Project Coordinators **Dr. Maalyadri, Professor in IT** and **Dr. R. Santosh Kumar, Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1.MAJJI ANUSHA (17K81A04M3)

2.SHETLEM TULASI (17K81A04P4)

3.THATI SANTOSH KUMAR (17K81A04P7)

ABSTRACT

The main purpose Vehicle Movement Based Street Lights with External Light Sensing Using ATMEGA project is that it saves energy by putting on the lights of the system only when the system detects movement of vehicle. The system switches on the street light ahead of the vehicle and switches off the trailing lights simultaneously.

The movement of vehicle gets detected by sensors. The system automatically puts on the lights that are ahead of the vehicle detected and as soon as the vehicle moves ahead, the trailing lights are switched off.

This system is more efficient for saving a lot of energy than existing system where the street lights are kept on always unlike this system where the street lights are put on only when movement of vehicle is detected. During day time these lights are dim as this system has the capability to sense external lights.

Thus, this system senses the external light and then accordingly switches ON or OFF the street lights. It uses PWM to control the intensity through microcontroller. The IR sensors sense the vehicle movements and send it to a ATMEGA microcontroller that initiates commands for switching the lights ON/OFF.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF OUPUT SCREENS

GLOSSARY

Manual System:

A manual system is like a bookkeeping system in which records maintenance is done by hand, without using a computer system or any automatic system. In this type of system transactions are written in journals, from which the information is manually retrieved into a set of financial statements. These systems suffer from higher rate of inaccuracy, and they are much slower than computerized systems. ”

Automated system:

Automated system is a combination of both software and hardware which is designed and programmed to work automatically without the need of any human operator to provide inputs and instructions for each operation. “

Automated systems allow you to monitor your processes in real time and identify problems as they arrive, enabling quick adjustments along the way. While manual systems can be difficult to coordinate, similar to the old cliche that "the right hand doesn't know what the left hand is doing," automated systems work in tandem on their own.

Pulse width modulation:

PWM is a way to control analog devices with a digital output. Another way to put it is that you can output a modulating signal from a digital device such as an MCU to drive an analog device. It’s one of the primary means by which MCUs drive analog devices like variable-speed motors, dimmable lights, actuators, and speakers. PWM is not true analog output, however. PWM “fakes” an analog-like result by applying power in pulses, or short bursts of regulated voltage.

Arduino:

The **ArduinoUno** is an open source board with the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.[1] The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts.

Arduino IDE:

The **Arduino Integrated Development Environment (IDE)** is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards.

Proteus:

The **Proteus Design Suite** is a proprietary software tool suite used primarily for automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

IR sensor:

An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment. While measuring the temperature of each color of light (separated by a prism), he noticed that the temperature just beyond the red light was highest. IR is invisible to the human eye, as its wavelength is longer than that of visible light (though it is still on the same electromagnetic spectrum)

LDR sensor:

A Light Dependent Resistor (LDR) is also called a photoresistor or a cadmium sulfide (CdS) cell. It is also called a photoconductor. It is basically a photocell that works on the principle of photoconductivity. The passive component is basically a resistor whose resistance value decreases when the intensity of light decreases. This **optoelectronic device** is mostly used in light varying sensor circuit, and light and dark activated switching circuits.

Relay:

A Light Dependent Resistor (LDR) is also called a photoresistor or a cadmium sulfide (CdS) cell. It is also called a photoconductor. It is basically a photocell that works on the principle of photoconductivity. The passive component is basically a resistor whose resistance value decreases when the intensity of light decreases. This **optoelectronic device** is mostly used in light varying sensor circuit, and light and dark activated switching circuits.

LED Bulbs:

An **LED lamp** or **LED light bulb** is an electric light that produces light using light-emitting diodes (LEDs). LED lamps are significantly more energy-efficient than equivalent incandescent lamps and can be significantly more efficient than most fluorescent lamps. The most efficient commercially available LED lamps have efficiencies of 200 lumens per watt (Lm/W). Commercial LED lamps have a lifespan many times longer than incandescent lamps.

CHAPTER-1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT:

Automation plays an increasingly very important role in the world economy and in daily life. Automatic systems are being preferred over any kind of manual system. We can also call it an "SMART STREET LIGHT SENSING". Intelligent light sensing refers to public street lighting that adapts to movement by pedestrians, cyclists and cars. Intelligent street lighting, also referred to as adaptive street lighting, dims when no activity is detected, but brightens when movement is detected. This type of lighting is different from traditional, stationery and illumination, or dimmable street lighting that dims at pre-determined times.

Street lights are doing more than ever in today's smart cities. With digital networks and embedded sensors, they collect and transmit information that help cities monitor and respond to any circumstance, from traffic and air quality to crowds and noise. They can detect traffic congestion and track available parking spaces. Those very same networks can remotely control LED lights to turn on and off, flash, dim and more, offering cities a chance to maximize low-energy lighting benefits while also improving pedestrian and bicyclist safety. With street lights creating a network canopy, those networks of data can be used by more than just lighting departments, empowering even schools and businesses via a lighting infrastructure that brightens the future of the digital city.

Smart lighting helps cities save energy, lower costs, reduce maintenance—all while better serving citizens and reducing energy use and CO2 emissions. Automation and networked control can further increase your energy savings and reduce maintenance spending. Networked street lighting built on a scalable platform can reduce crime up to 10% and make roadways safer through improved visibility. Leveraging intelligent control systems can rapidly increase lighting efficiencies and traffic management.

The research work shows automatic control of streetlights as a result of which power is saved to some extent. In the scope of industrialization, automation is a step beyond mechanization. Whereas mechanization provided human operators with machinery to assist the users with muscular requirements of work, automation greatly decreases the need for human sensory and mental requirements as well. Basically, street lighting is one of the important parts. Therefore, the street lamps are relatively simple but with the development of urbanization, the number of streets increases rapidly with high traffic density. There are several factors need to be considered in order to design a good street lighting system such as night-time safety for community members and road users, provide public lighting at cost effective, the reduction of crime and minimizing it is effect on the environment.

1.2 OBJECTIVE OF THE STUDY:

The idea of designing a new system for the streetlight that do not consume huge amount of electricity and illuminate large areas with the highest intensity of light is concerning each engineer working in this fields. Providing street lighting is one of the most important and expensive responsibilities of a city. Manual control is prone to errors and leads to energy wastages and manually dimming during mid night is impracticable. Also, dynamically tracking the light level is manually impracticable. The current trend is the introduction of automation and remote management solutions to control street lighting.

At the beginning, street lamps were controlled by manual control where a control switch is set in each of the street lamps which is called the first generation of the original street light. After that, another method that has been used was optical control method done using high pressure sodium lamp in their system. Nowadays, it is seen that the method is widely used in the country. The method operates by set up an optical control circuit, change the resistance by using of light sensitive device to control street lamps light up automatically at dusk and turn off automatically after dawn in the morning. Due to the technological development nowadays, road lighting can be categorized according to the installation area and performance, for an example, lighting for traffic routes, lighting for subsidiary roads and lighting for urban center and public amenity areas. The WSN helps in improving the network sensing for street lighting. Meanwhile, street light system can be classified according to the type of lamps used such as incandescent light, mercury vapor light, metal halide light, high pressure sodium light, low pressure sodium light, fluorescent light, compact fluorescent light, induction light and LED light. Different type of light technology used in lighting design with their luminous efficiency, lamp service life and their considerations. The LED is considered a promising solution to modern street lighting system due to its behavior and advantages. Apart from that, the advantages of LED are likely to replace the traditional street lamps such as the incandescent lamp, fluorescent lamp and high-pressure Sodium Lamp in future but LED technology is an extremely difficult process that requires a combination of advanced production lines, top quality materials and high- precision manufacturing process. Therefore, the research work highlights the energy efficient system of the street lights system using LED lamps with IR sensor interface for controlling and managing.

With the wide accessibility of adaptable lighting innovation like light transmitting diode lights and all over accessible remote web association, quick responding, dependable working, and power moderating street lighting frameworks get to be reality. The reason for this work is to portray the Smart Street Lighting framework, a first way to deal with perform the interest for adaptable smart lighting frameworks. The goal of this undertaking is to plan an automated lighting framework which focuses on the saving of power; to construct a vitally energy efficient smart lighting framework with integrated sensors and controllers.

1.3 SCOPE OF THE PROJECT:

- Automation systems are being preferred over the manual mode because it reduces the use of energy to saves energy. These automation systems play an essential role in making our daily life more comfortable and facilitate users from ceiling fans to washing machines and in other applications.
- Among all exciting applications, street lights play a vital role in our environment and also plays a critical role in providing light for safety during night-time travel. In this scenario, when the street lights are in working functionality over the whole night that consumes a lot of energy and reduces the lifetime of the electrical equipment such as electric bulb etc.
- Especially in cities streetlights, it is a severe power consuming factor and also the most significant energy expenses for a city. In this regard, an intelligent lighting control system can decrease street lighting costs up to 70% [3] and increase the durability of the equipment.

1.4 MATERIAL REQUIREMENT:

1.4.1 HARDWARE REQUIREMENT:

- Arduino UNO
- Infrared (IR) sensor

- Liquid Cristal Display
- Light Depended Resistor (LDR) sensor
- Relay
- Potentiometer
- Resistor(100K)
- LED Bulbs

1.4.2 SOFTWARE REQUIREMENT:

- Arduino IDE
- PROTEUS software

1.5 PROCUMENT OF EQUIPMENT:

♦ We brought all the required hardware components in a electronic manufacturing shop and before assembling and interfacing hardware components to Arduino UNO micro-controller, we connected individual sensors to Arduino in Proteus Software and understood working of every individual sensor in Proteus Software.

♦ Now, we assembled all sensors and interfaced to Arduino UNO micro-controller using connecting wires through bread board and we wrote program to work all the hardware sensors in Arduino IDE, we understood the working of all sensors in a practical way.

♦ We observed that the LED bulb glows when the vehicle movement is detected and it stays in ON condition until sometime. It stays in OFF when there is light surroundings.

CHAPTER -2

LITERATURE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH AREA:

The idea of designing a new system for the streetlight that do not consume huge amount of electricity and illuminate large areas with the highest intensity of light is concerning each engineer working in this field. Providing street lighting is one of the most important and expensive responsibilities of a city. Lighting can account for 10-38% of the total energy bill in typical cities worldwide [1]. Street lighting is a particularly critical concern for public authorities in developing countries because of its strategic importance for economic and social stability. Inefficient lighting wastes significant financial resources every year, and poor lighting creates unsafe conditions. Energy efficient technologies and design mechanism can reduce cost of the street lighting drastically.

Manual control is prone to errors and leads to energy wastages and manually dimming during mid night is impracticable. Also, dynamically tracking the light level is manually impracticable. The current trend is the introduction of automation and remote management solutions to control street lighting [2].

There are various numbers of control strategy and methods in controlling the street light system such as design and implementation of CPLD based solar power saving system for street lights and automatic traffic controller [1], design and fabrication of automatic street light control system [3].

2.2 REVIEW ON RELATED LITERATURE:

In this system two kinds of sensors will be used which are light sensor and photoelectric sensor. The light sensor will detect darkness to activate the ON/OFF switch, so the streetlights will be ready to turn on and the photoelectric sensor will detect movement to activate the streetlights. LDR, which varies according to the amount of light falling on its surface, this gives an induction for whether it is a day-night time, the photoelectric sensors are placed on the side of the road, which can be controlled by microcontroller. The photoelectric will be activated only on the night time. If any object crosses the photoelectric beam, a particular light will be automatically ON. By using this as a basic principle, the intelligent system can be designed for the perfect usage of streetlights in any place.

S. Suganya et al [2] have proposed about Street Light Glow on detecting vehicle movement using sensor is a system that utilizes the latest technology for sources of light as LED lamps. It is also used to control the switching of street light automatically according to the light intensity to develop flow based dynamic control statistics using infrared detection technology and maintain wireless communication among lamppost and control terminal using ZigBee Wireless protocol. It also combines various technologies: a timer, a statistics of traffic flow magnitude, photodiodes, LED, power transistors.

K. Santha et al [3] have surveyed on Street Lighting System Based on Vehicle Movements. The system operates in the automatic mode which regulates the streetlight according to brightness and dimness algorithm and light intensity. The control can be made according to the seasonal variation. It includes a time cut-out function and an automatic control pattern for conserving more electricity. The whole project was implemented using a PIC microcontroller.

Srikanth et al [4] proposed a ZigBee based Remote Control Automatic Street Light System. The system is designed with the help of ZigBee modules that helps in detecting the faulty lights and control the light. It also discusses about an intelligent system that takes automatic decisions for ON/OFF/DIMMING considering the vehicle movement or pedestrian and also the surrounding environment. PIR motion sensor is used to detect movement of both living and non-living things.

M. Abhishek et al [5] have implemented design of traffic flow-based street light control system with effective utilization of solar energy in the year 2015. They used the renewable source of energy i.e., the solar power for street lighting. They have also used 8052 series microcontroller and is developed by replacing the normal bulbs with the LEDs due to which the power consumption is reduced by 3 times. Sensors are placed on either side of the road which senses the vehicle movement and sends the commands to the microcontroller to switch ON and OFF the lights. Here all the street lights remain switched off and it glows only when it senses the vehicle movement. Hence, because of the microcontroller, even when its night the lights are switched off.

Bhuvaneshwari et al [6] have analyzed the street light with auto tracking system by which one can increase the conversion efficiency of the solar power generation. Here, the sun tracking sensor is the sensing device which senses the position of the sun time to time and gives the output to the amplifier based on light density of the sun. Sun tracking sensor is LDR, amplifier unit is used to amplify the LDR signals which converts low level signals to high level signals and the output is given to comparator.

The LM324 IC is used as an amplifier. Comparator compares the signals and gives the command to AT89C51 microcontroller. Steve Chadwick [7] reports on the two-installation case studied in Scotland and Wales and explains the details and benefits of the technology. The system was called as MINOS that had a track record of over 100,000 units installed and working successfully.

Somchai Hiranvarodom [8] describes a comparative analysis of photovoltaic (PV) street lighting system in three different lamps. Namely, a low-pressure sodium lamp, a high-pressure sodium lamp and a fluorescent lamp have been used for installation in each mast to determine the suitable system to install in a typical rural area of Thailand. All three systems have been mounted with the same module type and wattage in different places within the Raja Mangala Institute of Technology, Thanya Buri district, Pathumthani province of Thailand. An operation of solar street lighting system can be divided into 2 period of time, namely, at 18.00-22.00 hours and 05.00-06.00 hours. The design of a control circuit was experimentally done in this work. The aim of this work is to determine the appropriate system to install in a typical rural area or a typical rural village of Thailand.

Radhi Priyasree [9] explains a system to reduce the power consumption of street lights by avoiding inefficient lighting which wastes significant financial resources each year. This is done by dimming the lights during less traffic hours. For this purpose, PIR sensor is used which detects any movement. This work also aims at reducing the fatal crashes and road accidents caused due to alcohol consumption. This is done using skin sensors placed in vehicle doors and also using breadth sensors inside the vehicle. By implementing this death rates due to drunk driving can be reduced to a great extent. The prototype has been implemented and works as expected and will prove to be very useful and will fulfill all the present constraints if implemented on a large scale. It also aims at detecting consumption of alcohol by the driver and if it exceeds certain level, it impairs the driver from entering into the Vehicle. This prevents occurrence of accidents or any fatal crashes. This initiative will help the government to save this energy and meet the domestic and industrial needs. From this literature survey, the methods each one has implemented and used is simple and easy to understand. These papers and journals have given many ideas to further implement a much efficient system and make things automated. The presentations are simple and clean with all the necessary information needed for a basic learner or reader.

CHAPTER-3

PROPOSED SYSTEM

3.1 OVERVIEW OF THE DESIGN:

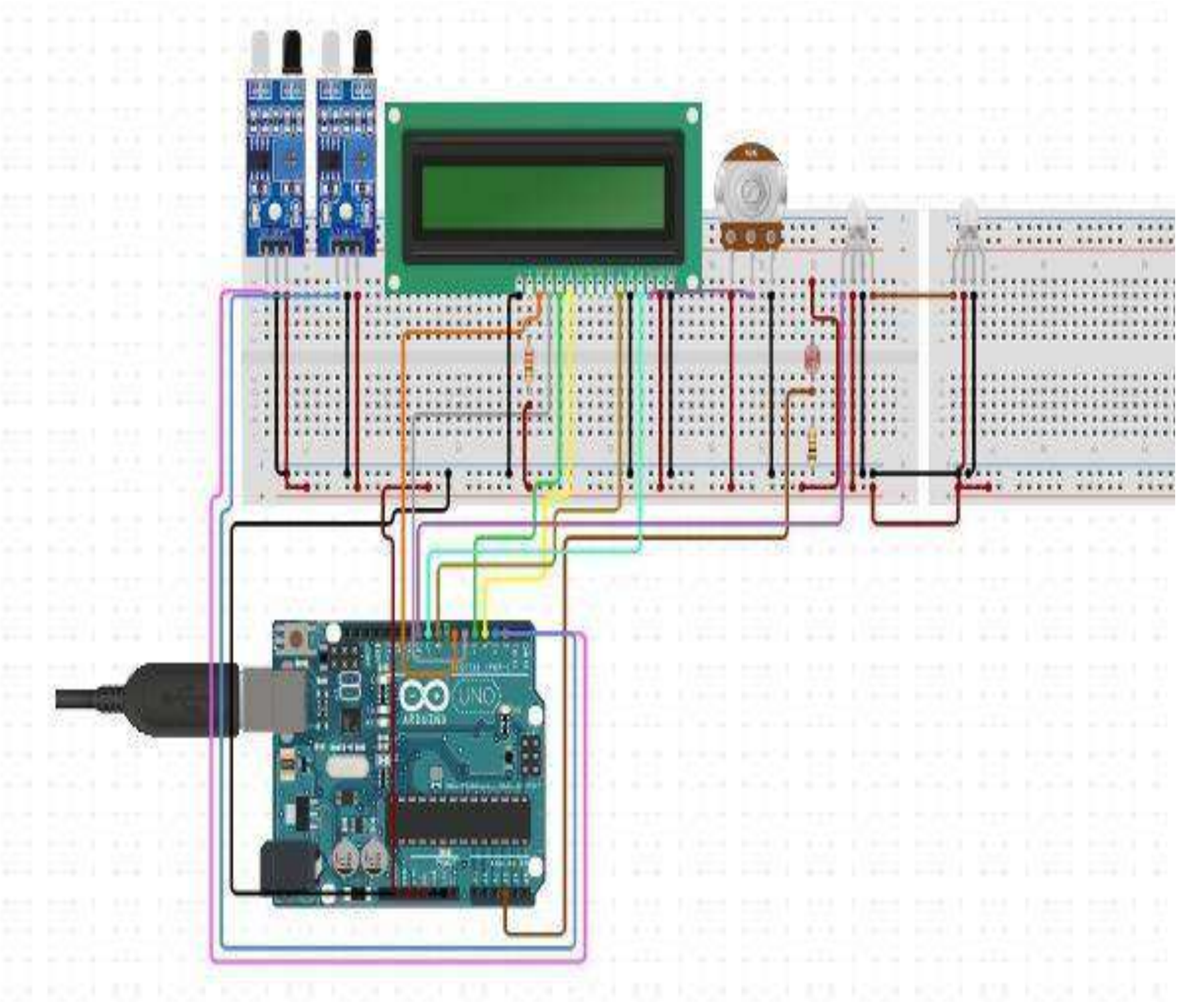


Fig: 3.1 – Schematic of System

3.2 ATMEGA328

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The ATmega8 microcontroller contains 32 general purpose working registers. As shown in the below figure these registers are directly connected to ALU. Two registers can carry one single instruction consequently in one clock cycle.

Specifications:

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 Ma
DC Current on 3.3V Pin	50 Ma
Flash Memory	32 KB (0.5 KB is used for Boot loader)
SRAM	2 KB
EEPROM	1 KB
Frequency(Clock Speed)	16Mhz

Table 3 Specifications

3.3 ARDUINO:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.

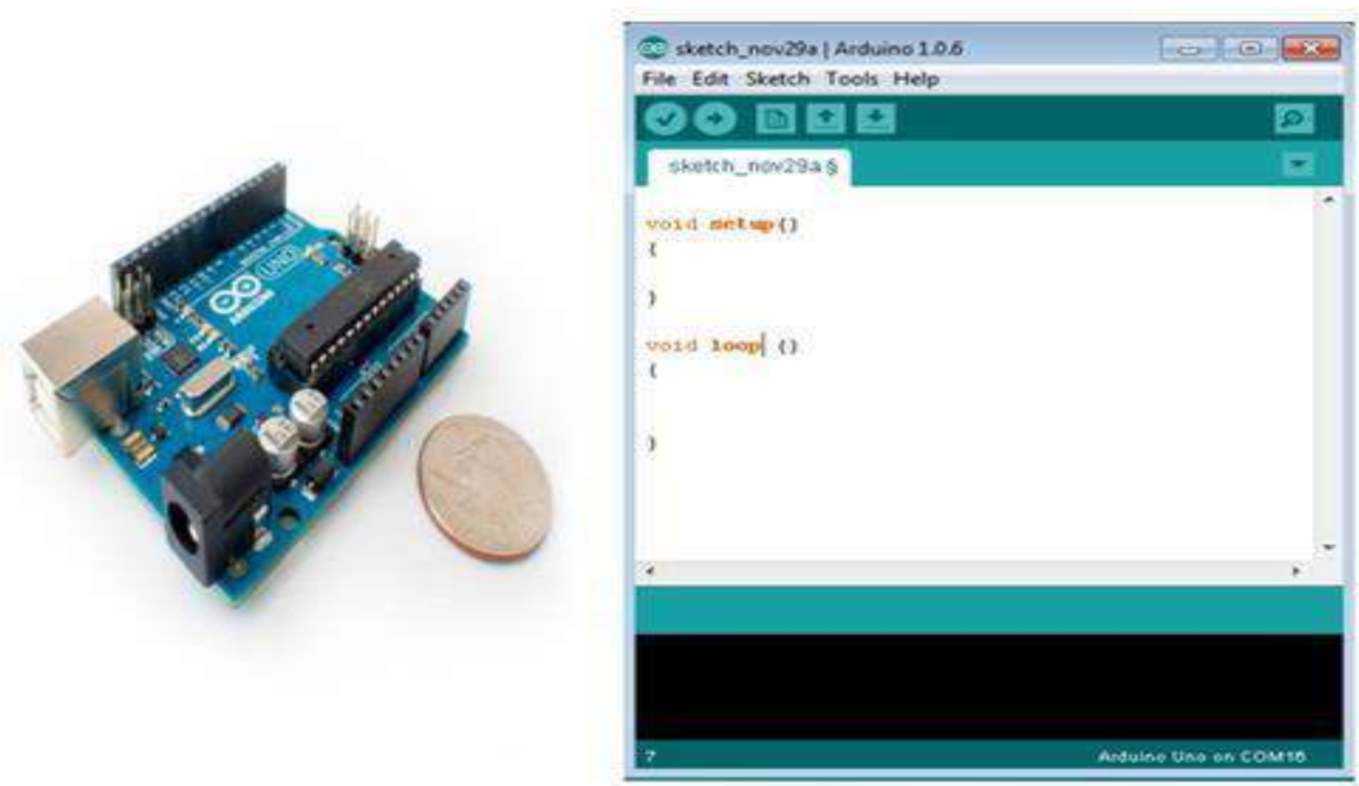


Fig 3.2 Arduino UNO

3.3.1 BOARD TYPES:

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega 16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega 16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega 16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega 16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega 16U2

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega 16U2

Board Name	**Operating Volt**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega 16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header

Board Name	**Operating Volt**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Lily Pad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compati
Lily Pad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.1 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
Lily Pad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 3.2 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16 U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Board Name	**Operating Volt**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table 3.3 Arduino boards based on AT91SAM3X8E microcontroller

3.3.2 BOARD DESCRIPTION:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

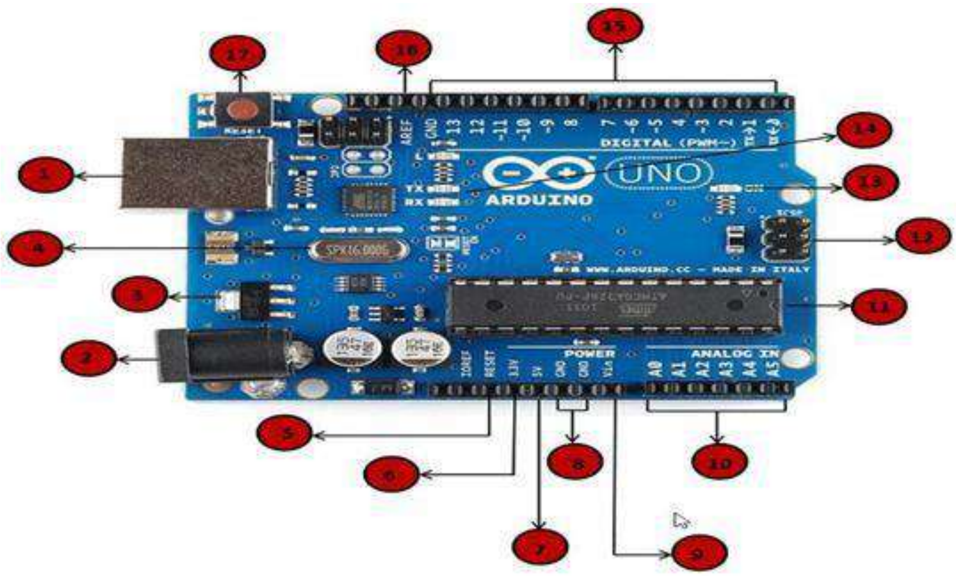


Fig: 3.3.1 – Arduino Pin Description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).

3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.

5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.

10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.

12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Table 3.4 Pin Description

3.4 ARDUINO FAMILY:

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project , check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino

Fig: 3.4.1 Arduino Family

3.5 SHIELDS:

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

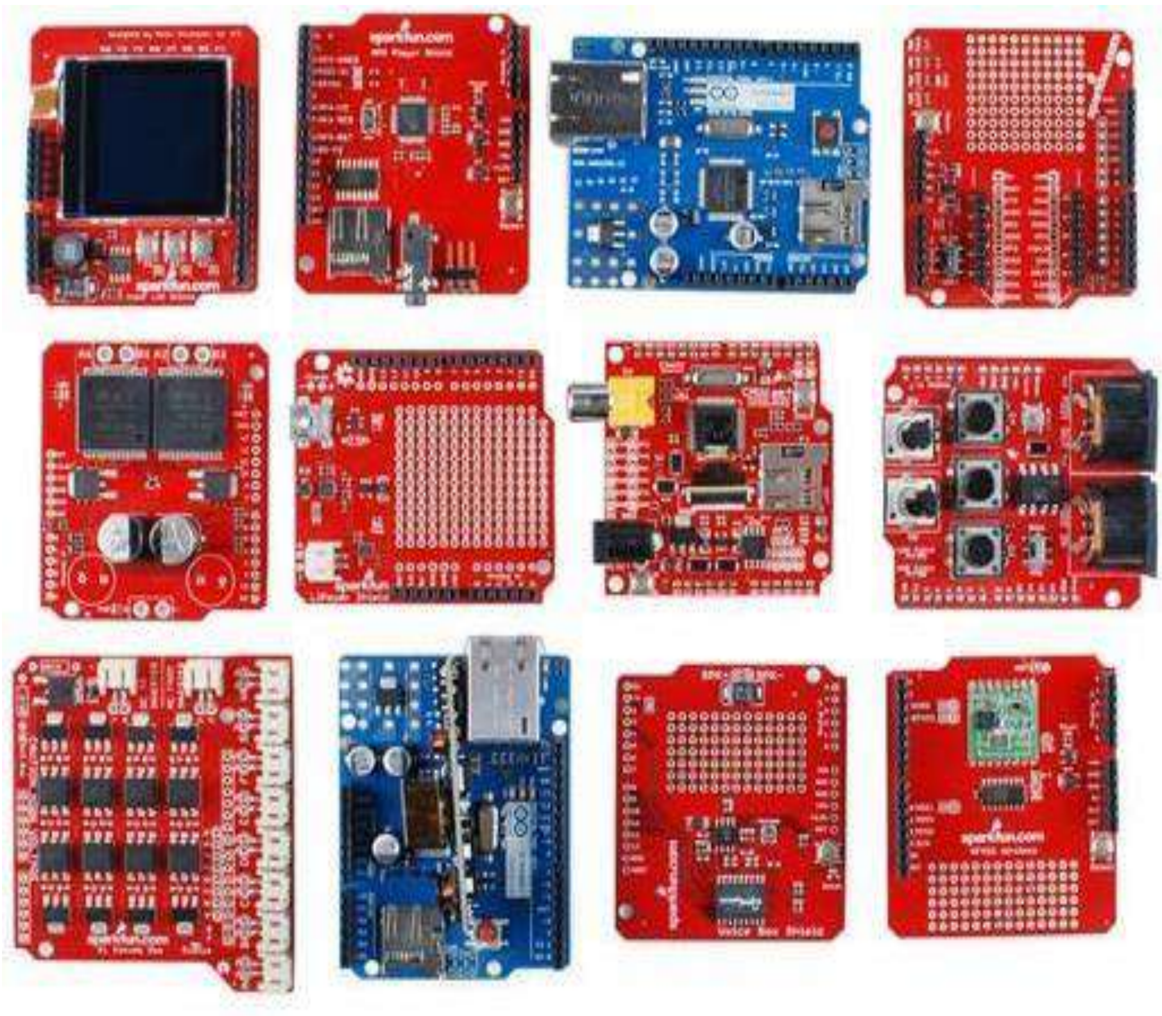


Fig.3.5.1: Arduino Shields

3.6 PIN DESCRIPTION OF ATMEGA328:

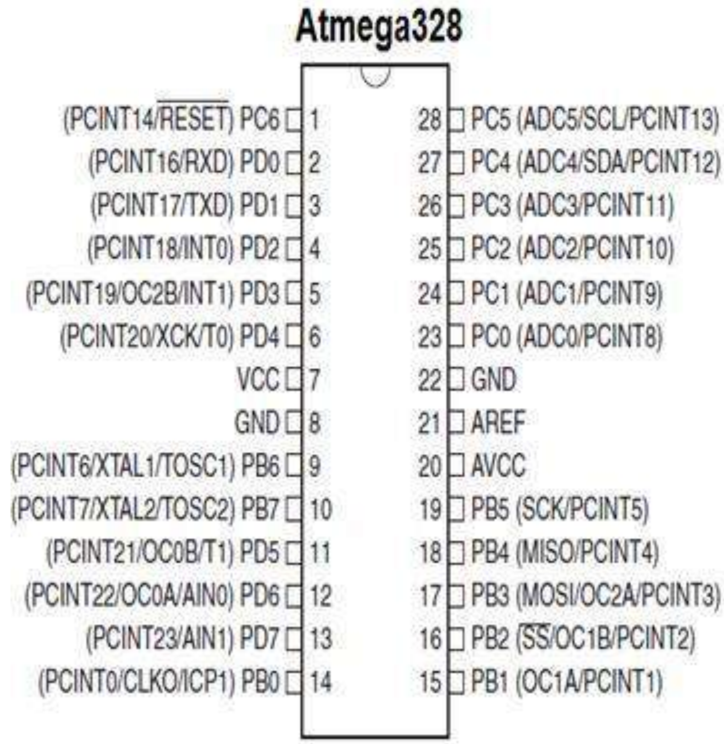


Fig: 3.6.1 Pin description of ATMEGA328

3.61 ADVANTAGES OF ARDUINO:

- It is cheap
- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
- For beginners, it is very simple to use.

3.6.2 APPLICATIONS:

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low-powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model.

3.7 RELAY

Relays are switches that open and close circuits electro mechanically or electronically. **Relays** control one electrical circuit by opening and closing contacts in another circuit. As **relay diagrams** show, when a **relay** contact is normally open (NO)

Fig: 3.7.1 RELAY

3.7.1 ELECTROMECHANICAL RELAYS:

Basic parts and functions of electromechanical relays include:

1. **Frame:** Heavy-duty frame that contains and supports the parts of the relay.
2. **Coil:** Wire is wound around a metal core. The coil of wire causes an electromagnetic field.
3. **Armature:** A relays moving part. The armature opens and closes the contacts. An attached spring returns the armature to its original position.
4. **Contacts:** The conducting part of the switch that makes (closes) or breaks (opens) a circuit.

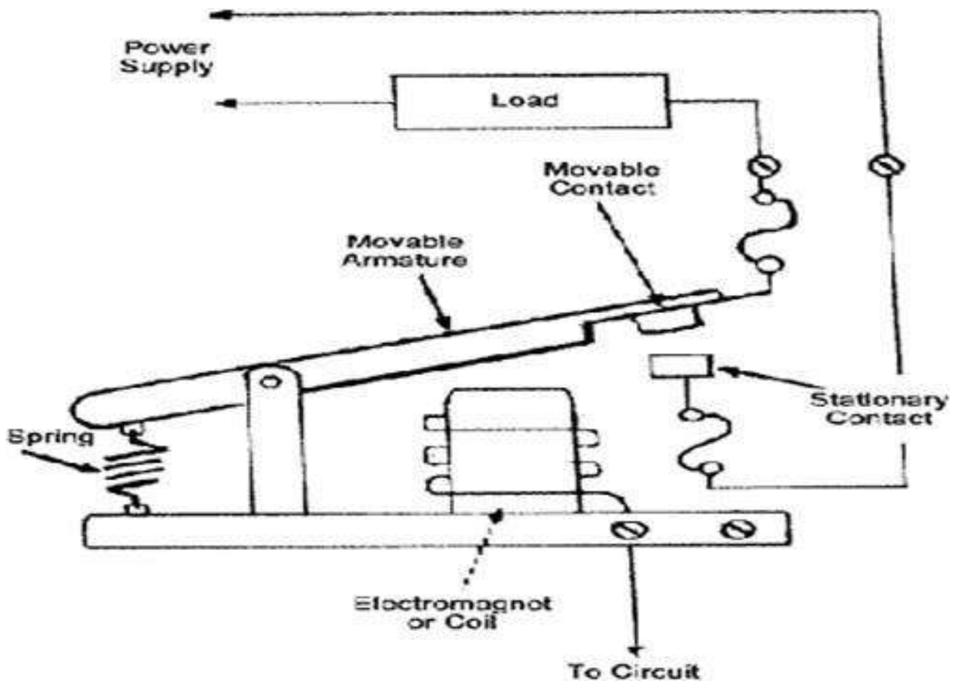


Fig:3.7.2 Internal Circuit of Relay.

Relays involve two circuits: the energizing circuit and the contact circuit. The coil is on the energizing side; and the relays contacts are on the contact side. When a relays coil is energized, current flow through the coil creates a magnetic field. Whether in a DC unit where the polarity is fixed, or in an AC unit where the polarity changes 120 times per second, the basic function remains the same: the magnetic coil attracts a ferrous plate, which is part of the armature. One end of the armature is attached to the metal frame, which is formed so that the armature can pivot, while the other end opens and closes the contacts. Contacts come in a number of different configurations, depending on the number of Breaks, poles and Throws that make up the relay. For instance, relays might be described as Single-Pole, Single-Throw (SPST), or Double-Pole, Single-Throw (DPST). These terms will give an instant indication of the design and function of different types of relays.

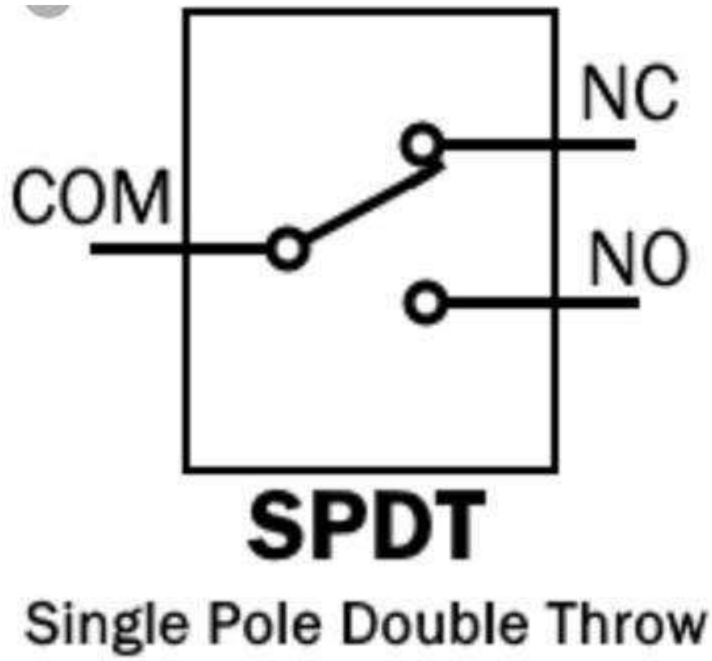


Fig:3.7.3 SPDT

- **Break** -This is the number of separate places or contacts that a switch uses to open or close a single electrical circuit. All contacts are either single break or double break. A single break (SB) contact breaks an electrical circuit in one place, while a double break (DB) contact breaks it in two places. Single break contacts are normally used when switching lower power devices such as indicating lights. Double break contacts are used when switching high-power devices such as solenoids.
- **Pole** -This is the number of completely isolated circuits that relays can pass through a switch. A single-pole contact (SP) can carry current through only one circuit at a time. A double-pole contact (DP) can carry current through two isolated circuits simultaneously. The maximum number of poles is 12, depending upon a relays design.
- **Throw** -This is the number of closed contact positions per pole that are available on a switch. A switch with a single throw contact can control only one circuit, while a double-throw contact can control two.

3.8 POWER SUPPLY UNIT

3.8.1 INTRODUCTION:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as "Regulated D.C Power Supply".

3.8.2 BLOCK DIAGRAM OF POWER SUPPLY:

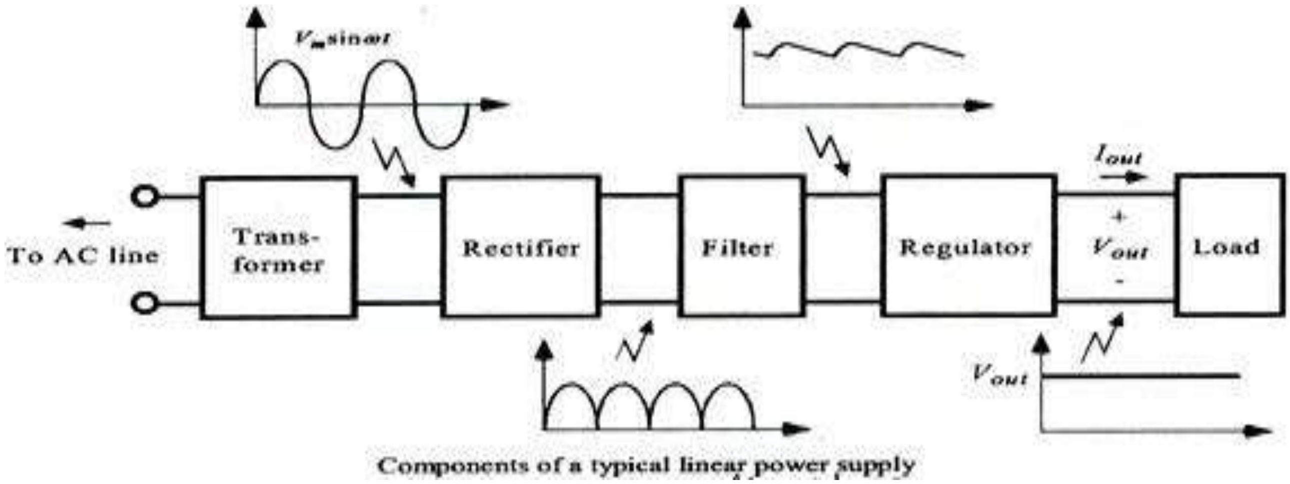


Fig: 3.8.1 Block Diagram of Power Supply

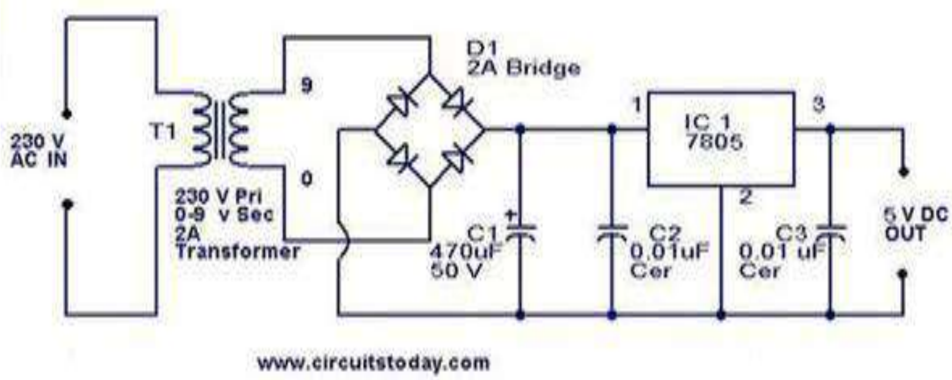


Fig: 3.8.2 Schematic Diagram of Power Supply

3.8.3 DESCRIPTION OF POWER SUPPLY:

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a a different form of energy – such as solar, mechanical, or chemical - into electrical energy.

A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered. For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

3.8.4 TRANSFORMER:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the secondary winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

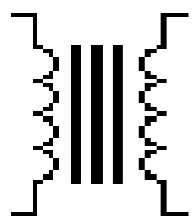


Fig: 3.8.3 Transformer Symbol

3.8.5 BASIC PRINCIPLE OF TRANSFORMER:

A transformer makes use of Faraday's_law and the ferromagnetic properties of an iron_core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

From Faraday's Law

$$\frac{V_S}{V_P} = \frac{N_S}{N_P}$$

For ideal transformer

The voltage ratio is equal to the turns ratio, and power in equals power out.

From conservation of energy

$$P_P = V_P I_P = V_S I_S = P_S$$

3.8.6 WORKING OF TRANSFORMER:

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

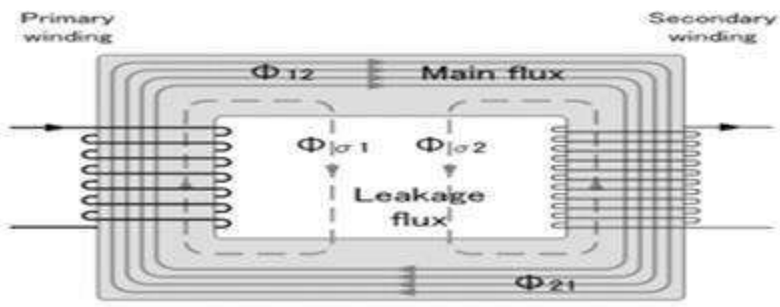


Fig:3.8.4 Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f. The iron core is laminated to prevent 'eddy currents' flowing in the core.

These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

- They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
- Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

3.8.7 CLASSIFICATION OF TRANSFORMER:

- Step-Up Transformer

- Step-Down Transformer

Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn's ratio of the two sets of windings determines the amount of voltage transformation.

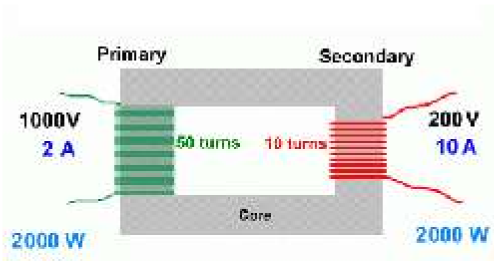


Fig:3.8.5 Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single phase steps down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase steps up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

Step-Up Transformer:

A step up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input. Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply. A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

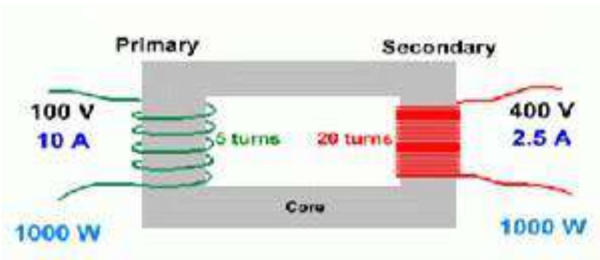


Fig: 3.8.6 Step-Up Transformer

3.8.8 DIODES:

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

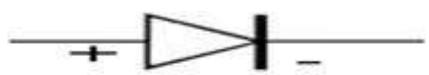


Fig: 3.8.7 Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

3.8.9 RECTIFIERS:

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

The Half-Wave Rectifier:

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure.

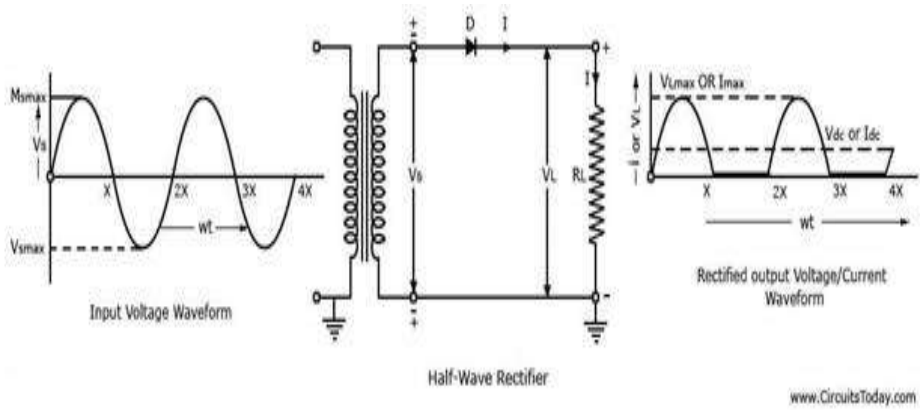


Fig: 3.8.8 Half Wave Rectifier

Figure shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

The Full-wave Rectifier:

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC.

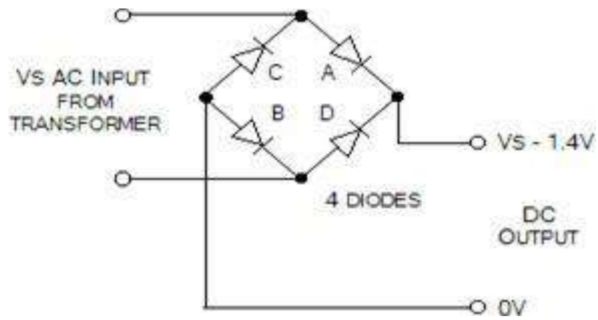


Fig:3.8.9 Full-Wave Rectifier

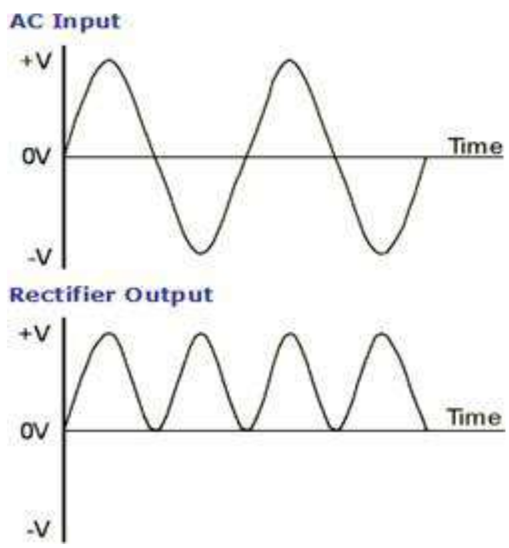


Fig:3.8.10 Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased. While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC.

3.8.10 CAPACITOR FILTER:

The capacitor-input filter, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

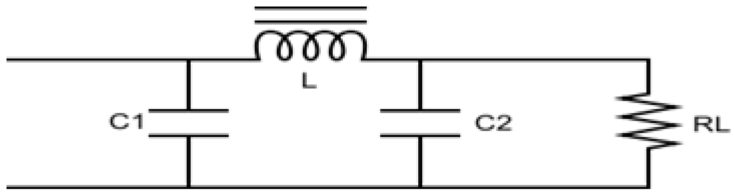


Fig:3.8.11 Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

- The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
- The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result, the DC component flows through the inductor while the AC component is blocked.
- The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result, only the DC component appears across the load RL.

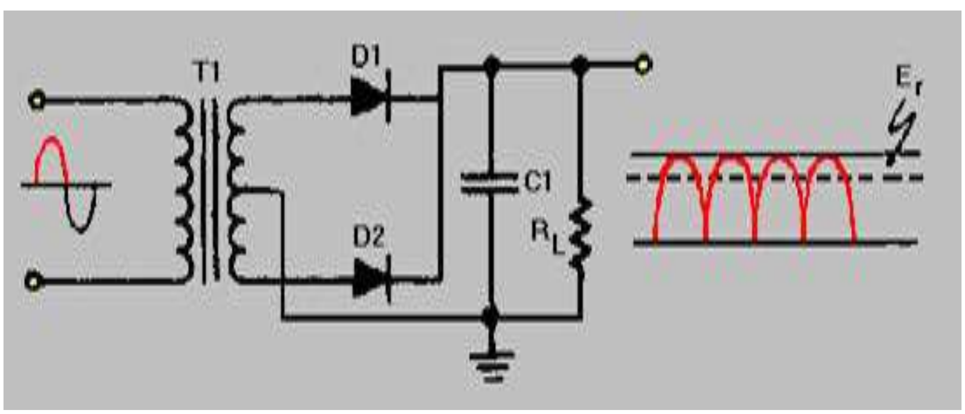


Fig:3.8.12 Centered Tapped Full-Wave Rectifier with a Capacitor Filter

3.8.11 VOLTAGE REGULATOR:

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx' indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx 'indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consist the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

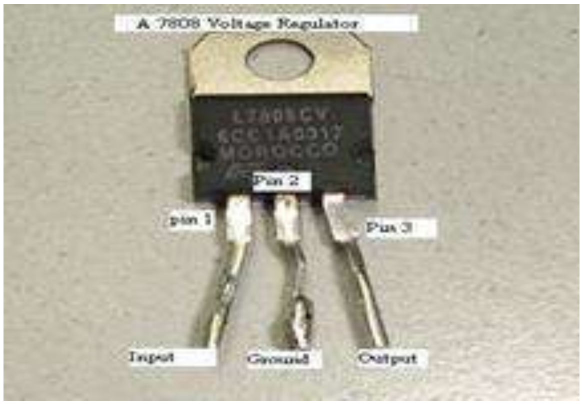


Fig:3.8.13 Regulator

3.9 HARDWARE COMPNENTS:

3.9.1 LDR SENSOR:

A **photoresistor** or **light dependent resistor** or **cadmium sulfide (CdS) cell** is a resistor whose resistance decreases with increasing incident light intensity. It can also be referred to as a **photoconductor**.

A photo resistor is made of a high resistance semiconductor. If light falling on the device is of high enough frequency, photons absorbed by the semiconductor give bound electrons enough energy to jump into the conduction band. The resulting free electron (and its hole partner) conduct electricity, there by lowering resistance.

A photoelectric device can be either intrinsic or extrinsic. An intrinsic semiconductor has its own charge carriers and is not an efficient semiconductor, e.g. silicon. In intrinsic devices the only available electrons are in the valence band, and hence the photon must have enough energy to excite the electron across the entire bandgap. Extrinsic devices have impurities, also called dopants, added whose ground state energy is closer to the conduction band; since the electrons do not have as far to jump, lower energy photons (i.e., longer wavelengths and lower frequencies) are sufficient to trigger the device. If a sample of silicon has some of its atoms replaced by phosphorus atoms (impurities), there will be extra electrons available for conduction. This is an example of an extrinsic semiconductor.

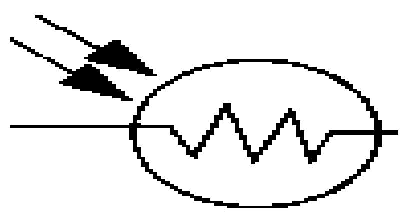


Fig: 3.9.1 The symbol for a photoresistor

Applications:

Photoresistors come in many different types. Inexpensive cadmium sulfide cells can be found in many consumer items such as camera light meters, street lights, clock radios, alarms, and outdoor clocks.

They are also used in some dynamic compressors together with a small incandescent lamp or light emitting diode to control gain reduction.

Lead sulfide (PbS) and indium antimonide (InSb) LDRs (light dependent resistor) are used for the mid infrared spectral region. Ge:Cu photoconductors are among the best far-infrared detectors available, and are used for infrared astronomy and infrared spectroscopy.

Transducers are used for changing energy types.

Fig: 3.9.2 A light dependent resistor

3.9.2 IR TRANSMITTER & RECEIVER:

- IR stands infrared red spectrum, by using this region we can detect number of obstacles passing through a region.
- IR consists of two sections one is transmitter section and other is receiver section.

- **Transmitter section:** it consists of a LED which continuously transmits a modulated signal of range(40 to 250khz).
- **Receiver section:** Receiver section consists of a photo diode it will continuously track and receive the IR rays from LED
- **Functionality:** The IR transmitter will continuously transmit IR rays and receiver (photo diode) will continuously receive the rays whenever some obstacle passes through the IR pair the transmission will be braked by which we come to know that an intruder have been detected. In this way this communication will help us in safeguarding and many other applications.

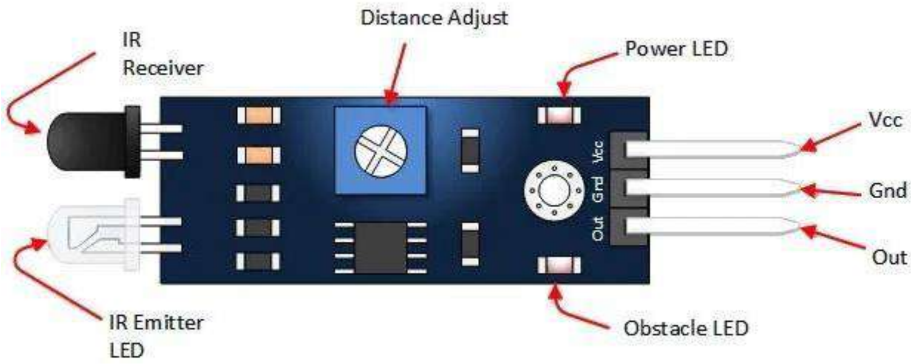


Fig: 3.9.3 IR Sensor

3.9.3 LED

Introduction:

A light-emitting diode (LED) is a semiconductor light source. LEDs are used as indicator lamps in many devices, and are increasingly used for lighting. Introduced as a practical electronic component in 1962, early LEDs emitted low-intensity red light, but modern versions are available across the visible, ultraviolet and infrared wavelengths, with very high brightness.

The LED is based on the semiconductor diode. When a diode is forward biased, electrons are able to recombine with holes within the device, releasing energy in the form of photons. This effect is called electroluminescence and the color of the light (corresponding to the energy of the photon) is determined by the energy gap of the semiconductor. An LED is usually small in area (less than 1 mm^2), and integrated optical components are used to shape its radiation pattern and assist in reflection. LEDs present many advantages over incandescent light sources including lower energy consumption, longer lifetime, improved robustness, smaller size, faster switching, and greater durability and reliability. However, they are relatively expensive and require more precise current and heat management than traditional light sources. Current LED products for general lighting are more expensive to buy than fluorescent lamp sources of comparable output.

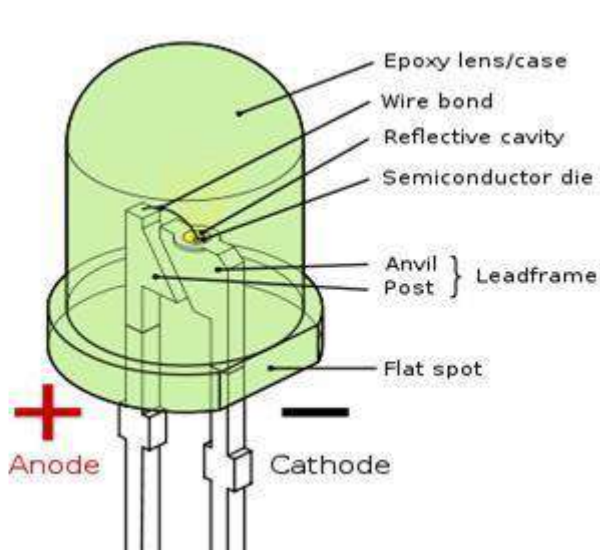


Fig: 3.9.4 LED

Working:

Charge-carriers—electrons and holes—flow into the junction from electrodes with different voltages. When an electron meets a hole, it falls into a lower energy level, and releases energy in the form of a photon. The wavelength of the light emitted, and therefore its color, depends on the band gap energy of the materials forming the p-n junction. In silicon or germanium diodes, the electrons and holes recombine by a non-radiative transition which produces no optical emission, because these are indirect band gap materials. The materials used for the LED have a direct band gap with energies corresponding to near-infrared, visible or near-

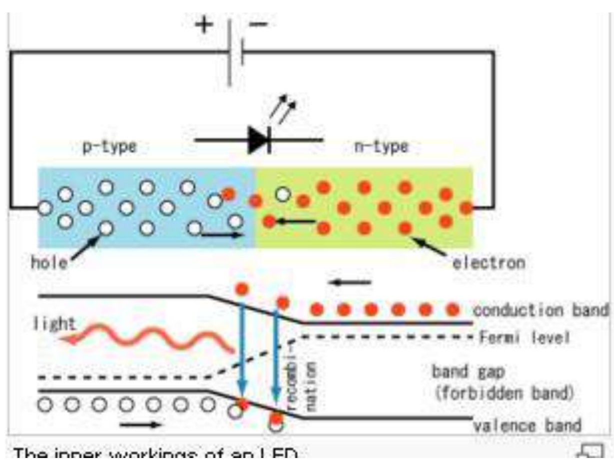


Fig: 3.9.5 Working Of LED

Colors and materials :

	Color	Wavelength (nm)	Voltage (V)	Semiconductor Material
	Infrared	$\lambda > 760$	$\Delta V < 1.9$	Gallium arsenide (GaAs) Aluminum gallium arsenide (AlGaAs)
	Red	$610 < \lambda < 760$	$1.63 < \Delta V < 2.03$	Aluminum gallium arsenide (AlGaAs) Gallium arsenide phosphide (GaAsP) Aluminium gallium indium phosphide (AlGaInP) Gallium(III) phosphide (GaP)
	Orange	$590 < \lambda < 610$	$2.03 < \Delta V < 2.10$	Gallium arsenide phosphide (GaAsP) Aluminum gallium indium phosphide (AlGaInP) Gallium(III) phosphide (GaP)
	Yellow	$570 < \lambda < 590$	$2.10 < \Delta V < 2.18$	Gallium arsenide phosphide (GaAsP) Aluminium gallium indium phosphide (AlGaInP) Gallium(III) phosphide (GaP)
	Green	$500 < \lambda < 570$	1.9[42] $< \Delta V < 4.0$	Indium gallium nitride (InGaN) / Gallium(III) nitride (GaN) Gallium(III) phosphide (GaP) Aluminium gallium indium phosphide

				(AlGaInP) Aluminium gallium phosphide (AlGaP)
	Blue	$450 < \lambda < 500$	$2.48 < \Delta V < 3.7$	Zinc selenide (ZnSe) Indium gallium nitride (InGaN) Silicon carbide (SiC) as substrate Silicon (Si) as substrate — (under development)
	Violet	$400 < \lambda < 450$	$2.76 < \Delta V < 4.0$	Indium gallium nitride (InGaN)
	Purple	multiple types	$2.48 < \Delta V < 3.7$	Dual blue/red LEDs, blue with red phosphor, or white with purple plastic
	Ultraviolet	$\lambda < 400$	$3.1 < \Delta V < 4.4$	Diamond (235 nm) Boron nitride (215 nm) Aluminium nitride (AlN) (210 nm) Aluminium gallium nitride (AlGaN) Aluminium gallium indium nitride (AlGaInN) — (down to 210 nm)
	White	Broad spectrum	$\Delta V = 3.5$	Blue/UV diode with yellow phosphor

Table 3.9.1 Colors and Materials

3.10 SOFTWARE REQUIREMENTS:

3.10.1 ARDUINO SOFTWARE:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals

- has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers make their own version
- Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money
- Getting Started with Arduino and Genuino products:-

Install the Arduino Software (IDE) on Windows PCs-

…This document explains how to install the Arduino Software (IDE) on Windows machines.

- ✓ Download the Arduino Software(IDE)
- ✓ Proceed with board specific instructions.

3.10.2 How to Download the Arduino Software (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

Installation:

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig: 3.10.1 USB Cable

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig: 3.10.2 A to Mini B Cable

Step 2 – Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

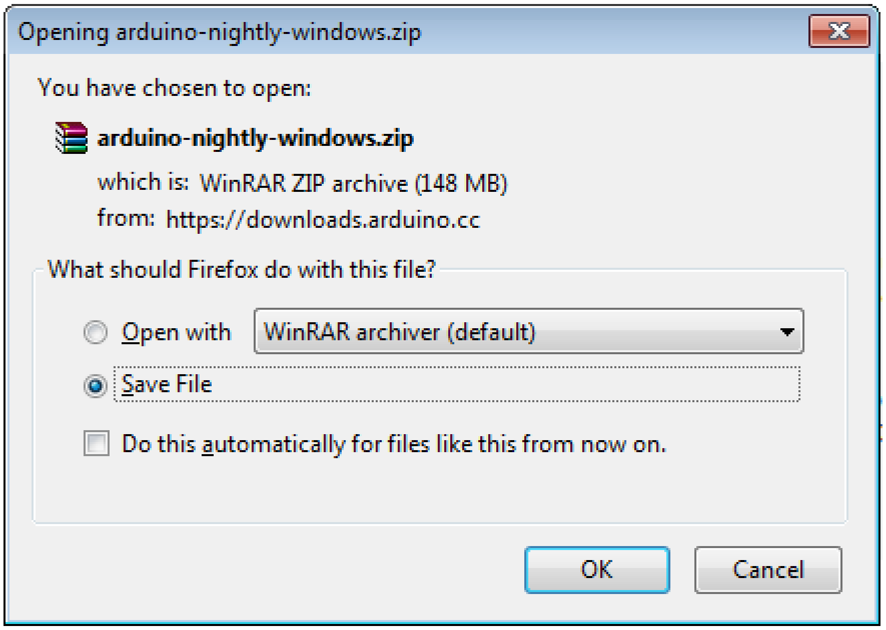


Fig: 3.10.3 Download Arduino IDE Software

Step 3 – Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 – Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

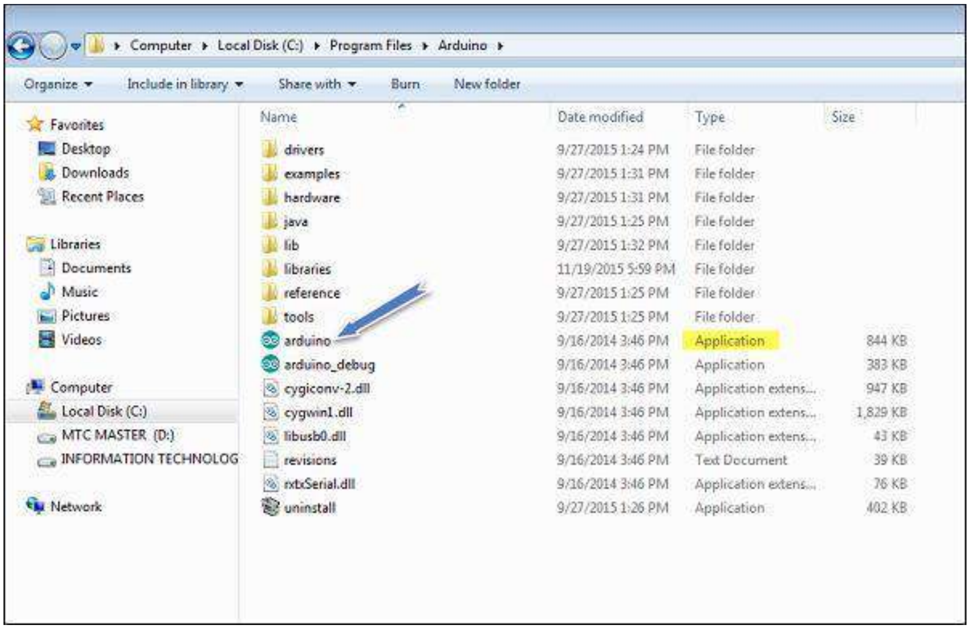


Fig: 3.10.4 Launch Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.

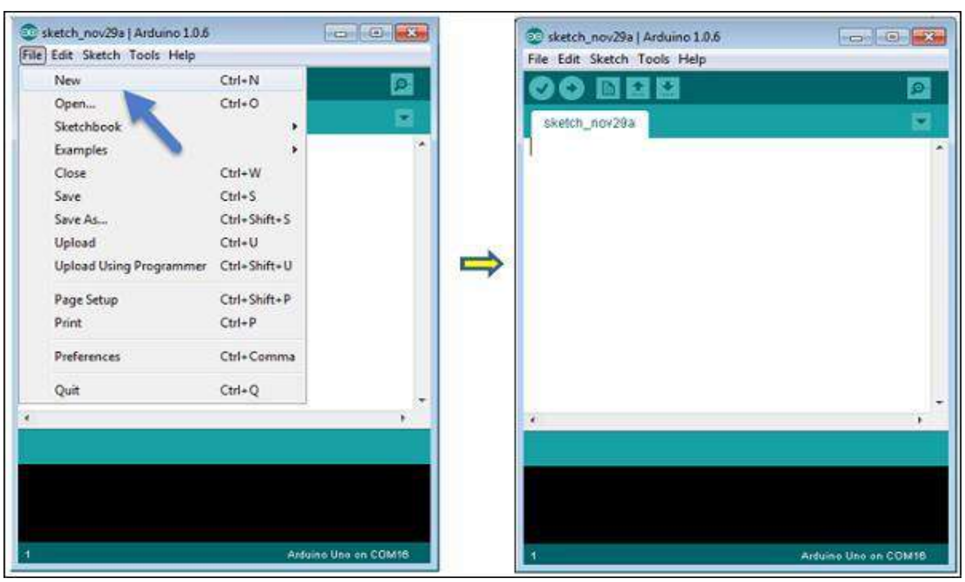


Fig: 3.10.5 Open Your First Project

To open an existing project example, select File → Example → Basics → Blink.

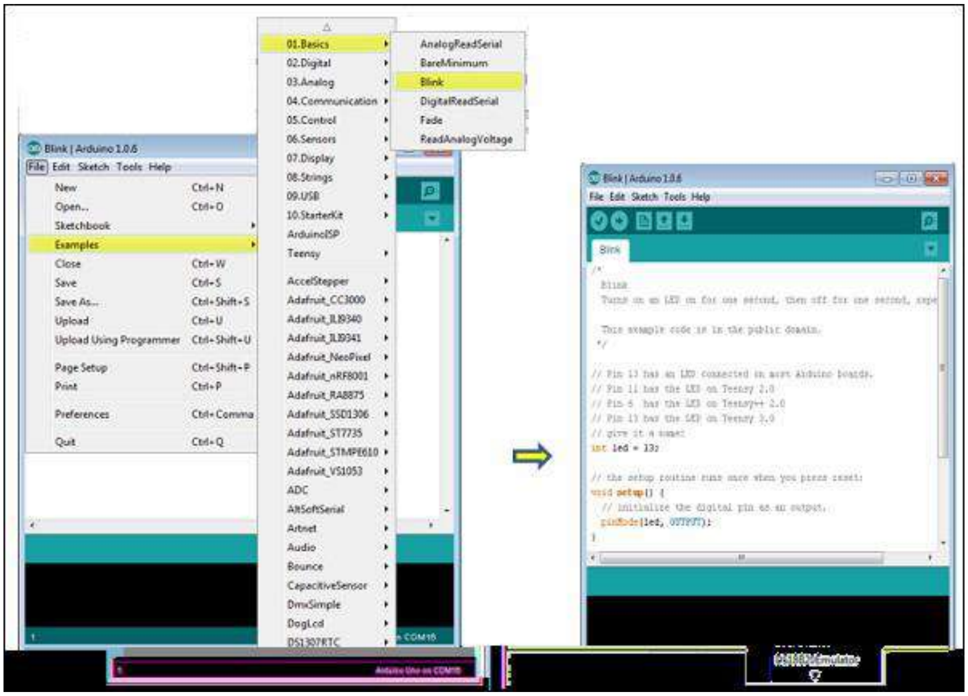


Fig: 3.10.6 Open Your First Program

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

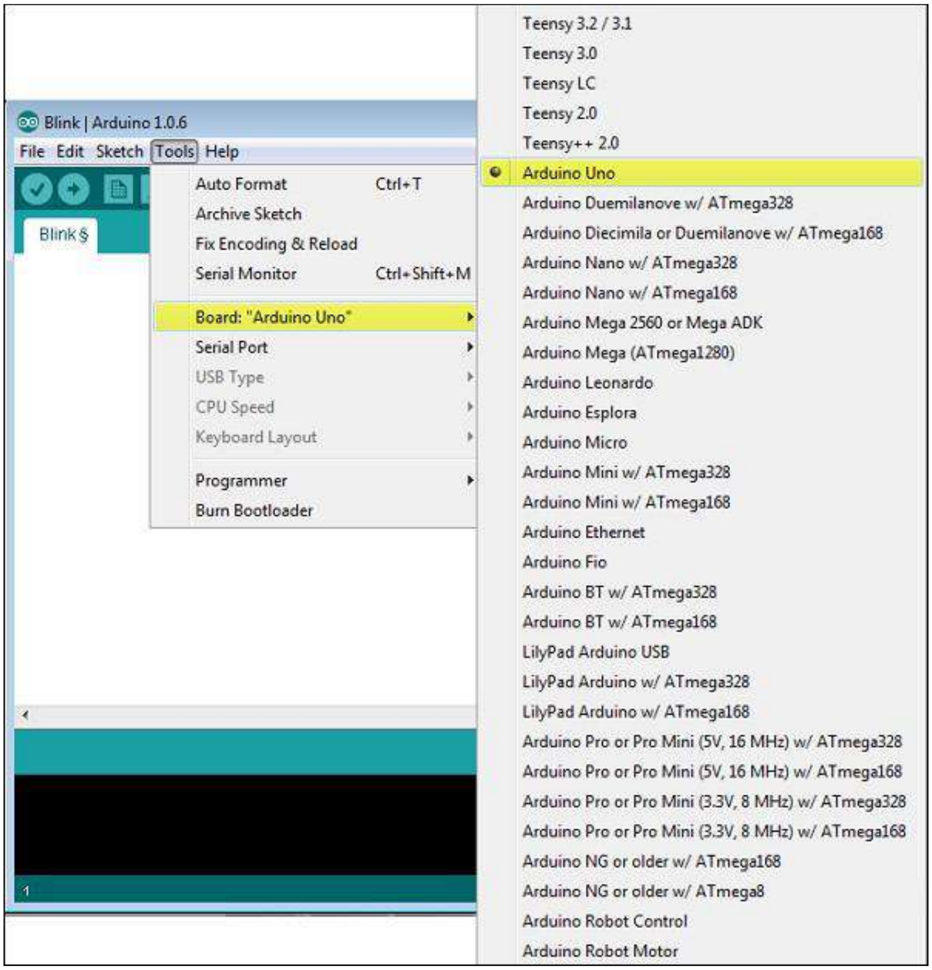


Fig: 3.10.7 Select Your Arduino Board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

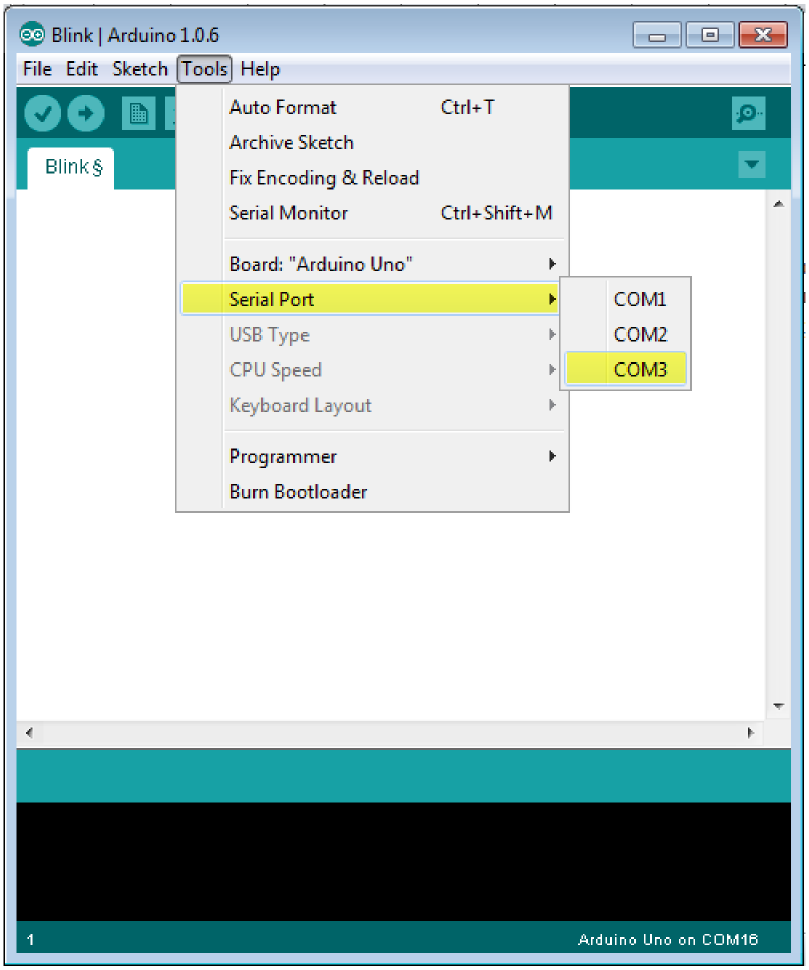


Fig: 3.10.8 Select Your Serial Port

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

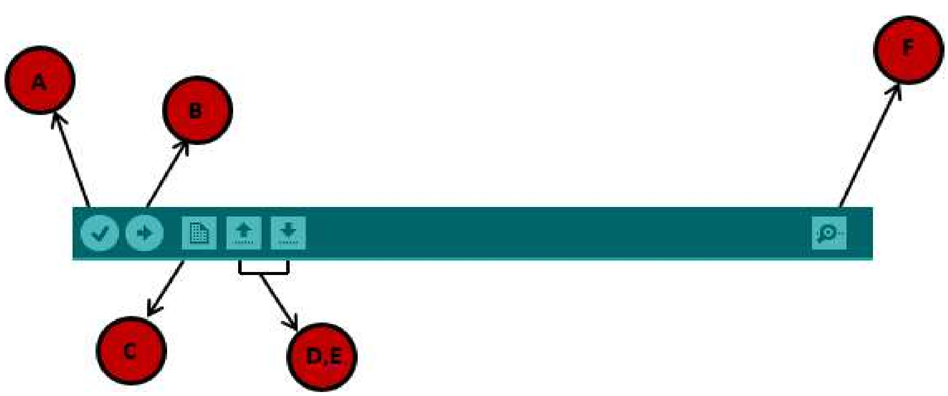


Fig: 3.10.9 Upload the Program To Your Board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24 V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

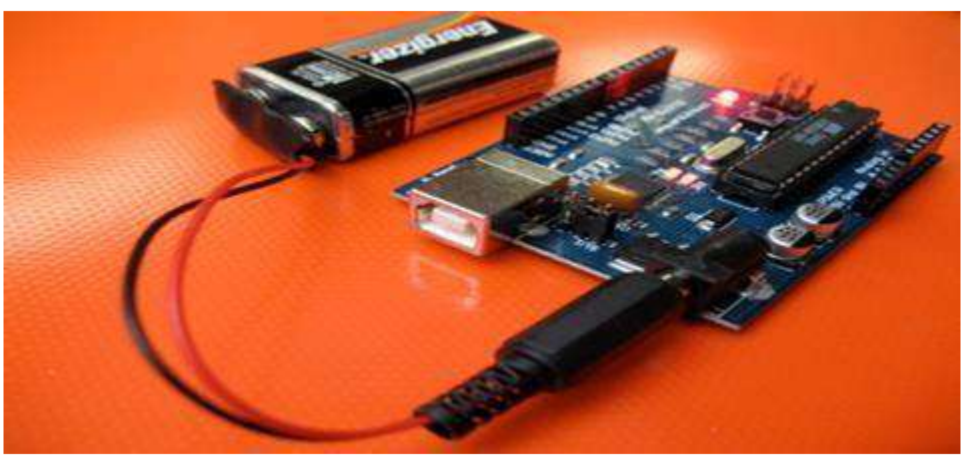


Fig:3.10.10 Arduino with battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC

.

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called “sketches”, but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

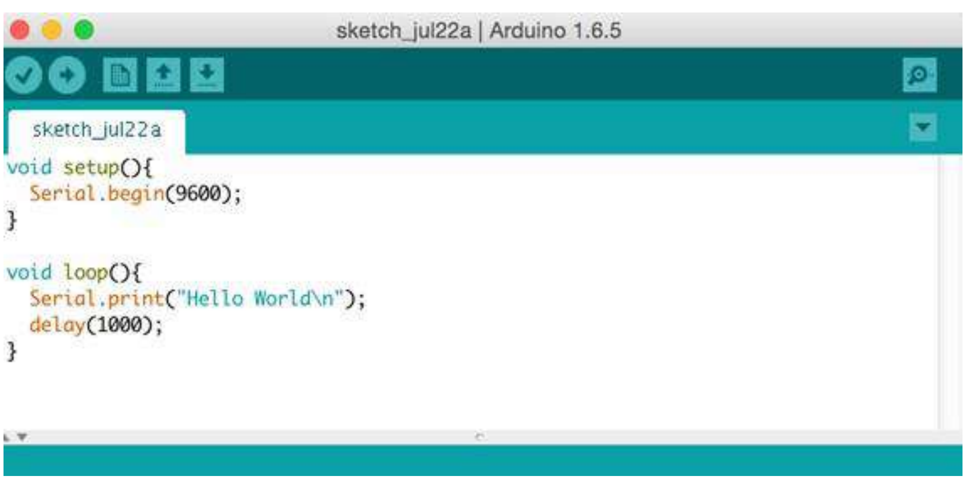


Fig: 3.10.11 Example Program

3.11 MC Programming Language: Embedded C

This is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements. Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

3.12 PROTEUS:

Proteus:

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing.

ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.

The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

3.12.1 Starting New Design

Step 1: Open ISIS software and select New design in File menu

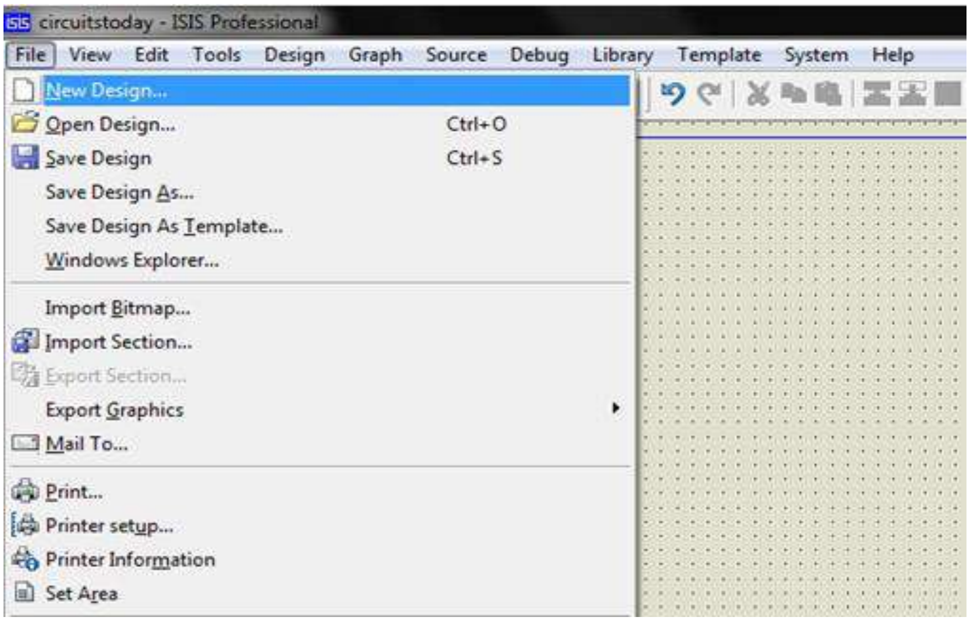


Fig: 3.12.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

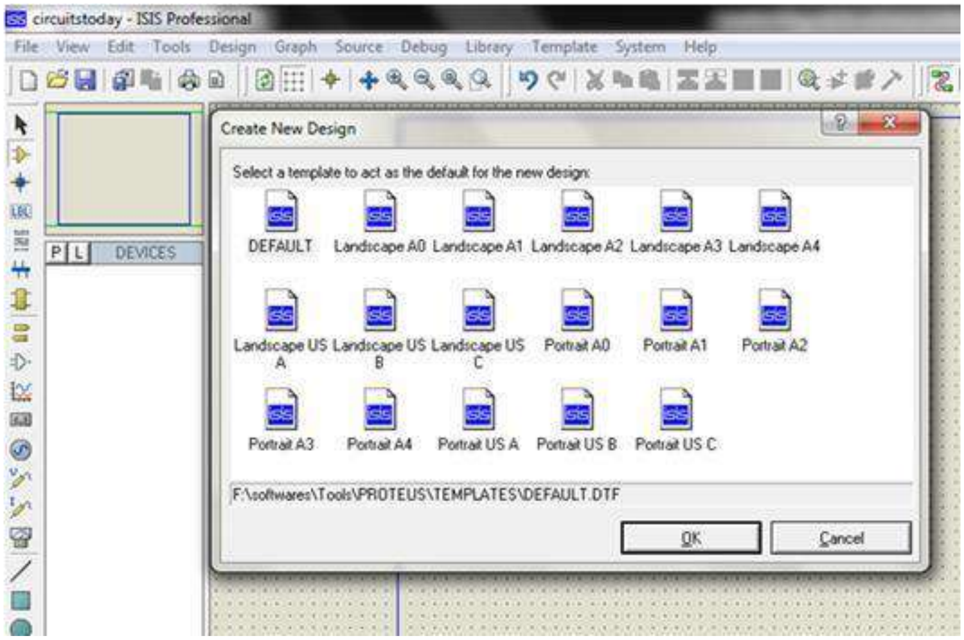


Fig:3.12.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

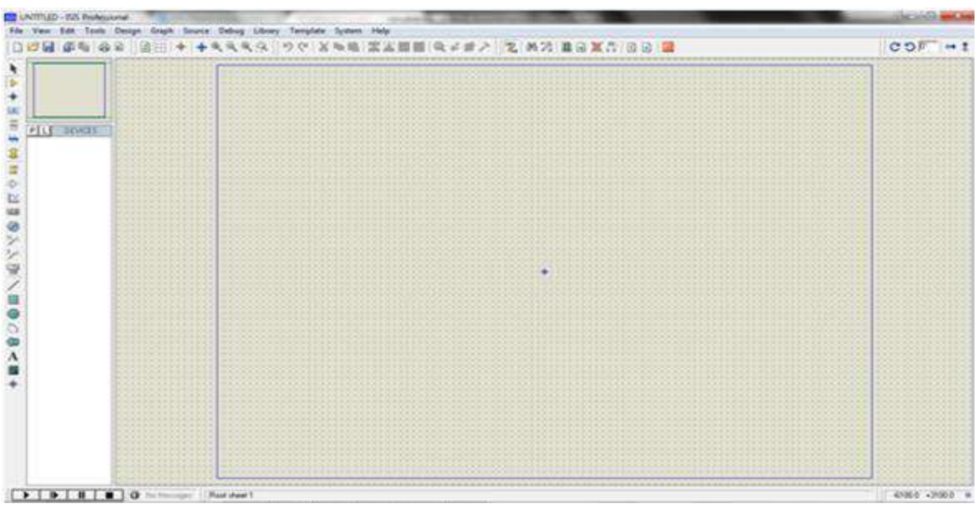


Fig:3.12.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

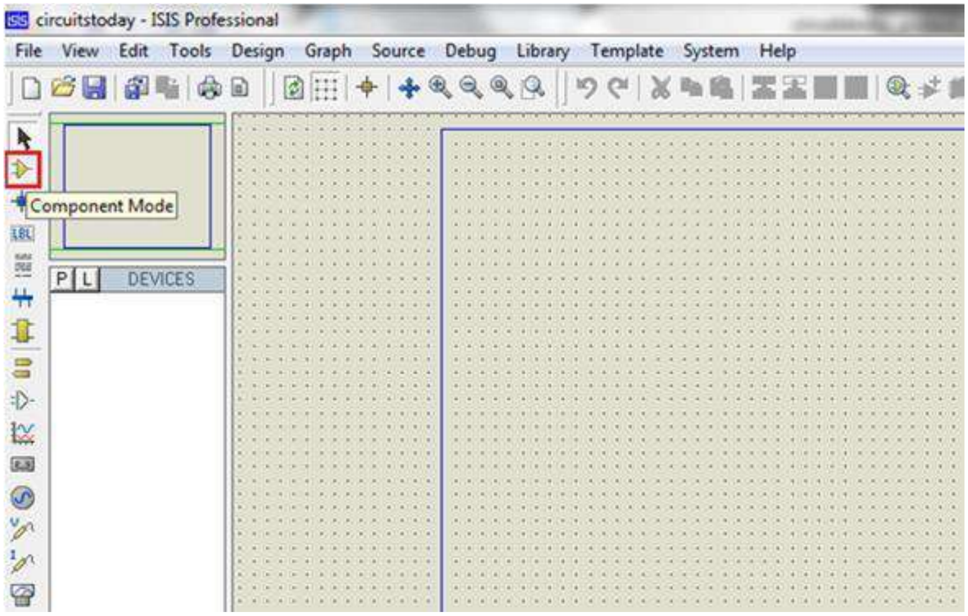


Fig: 3.12.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

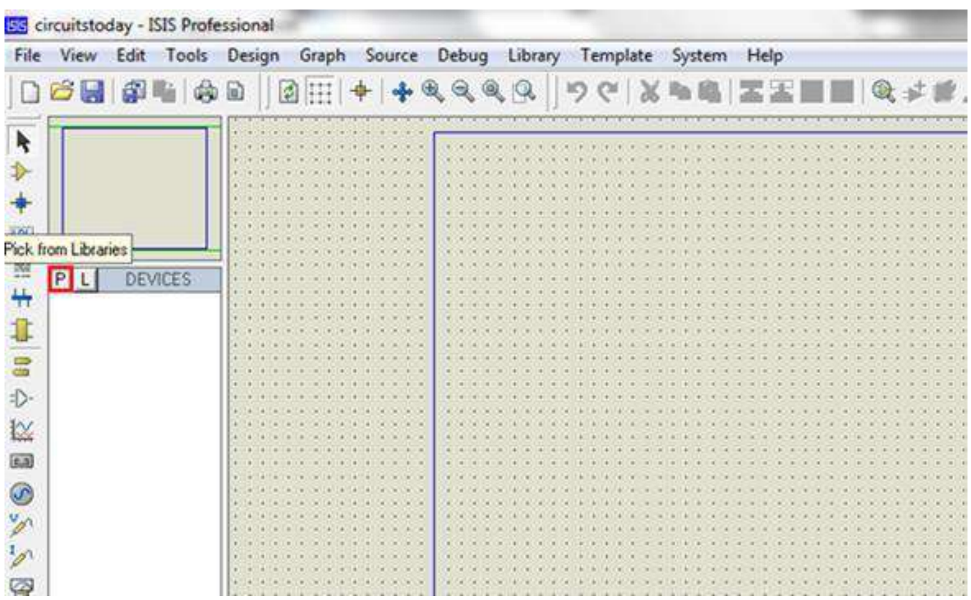


Fig: 3.12.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

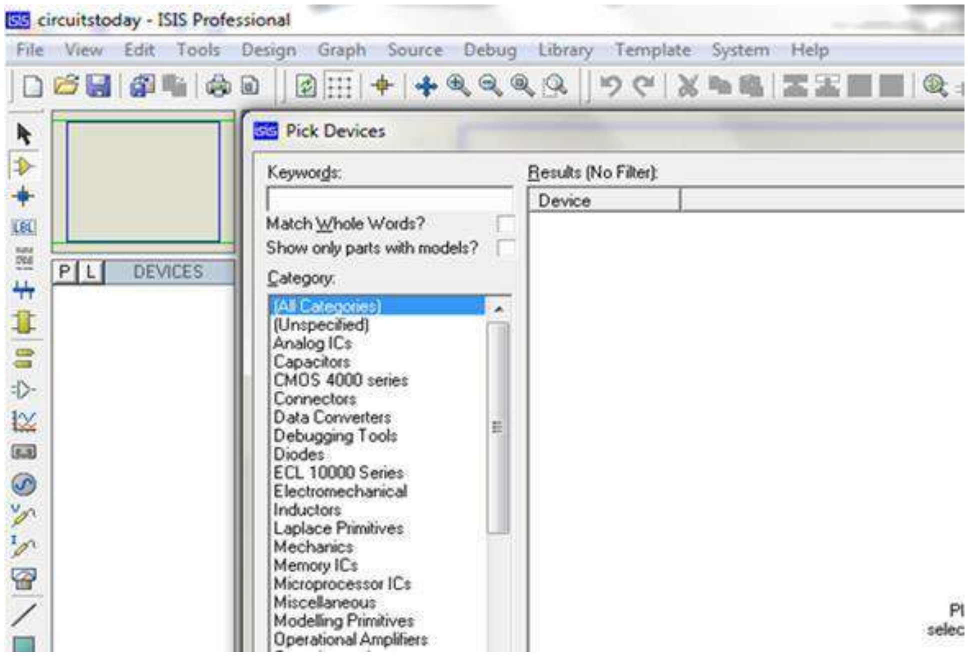


Fig: 3.12.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

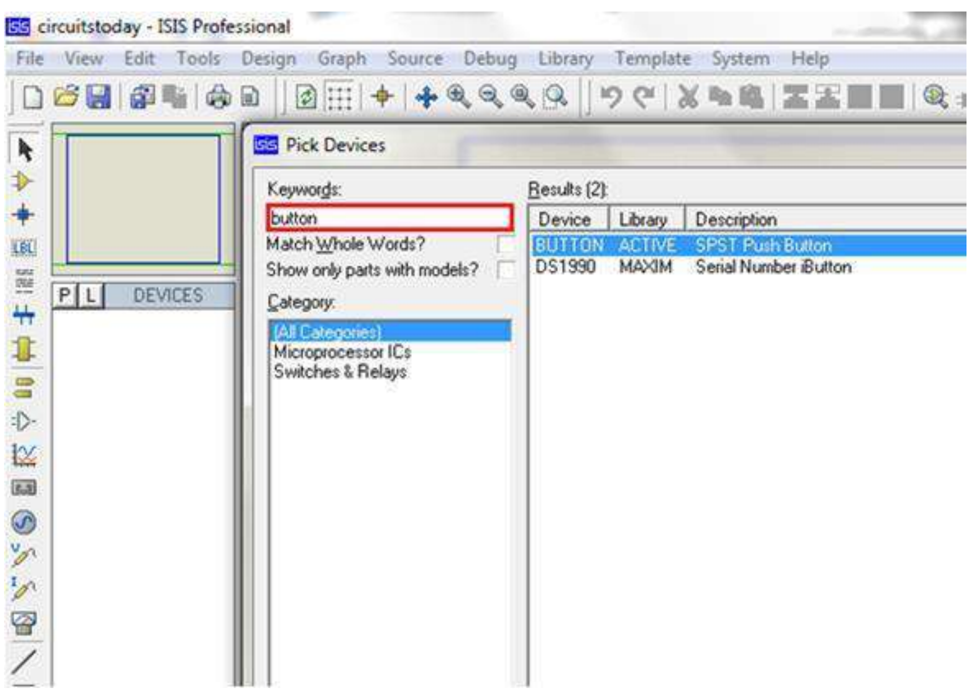


Fig: 3.12.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

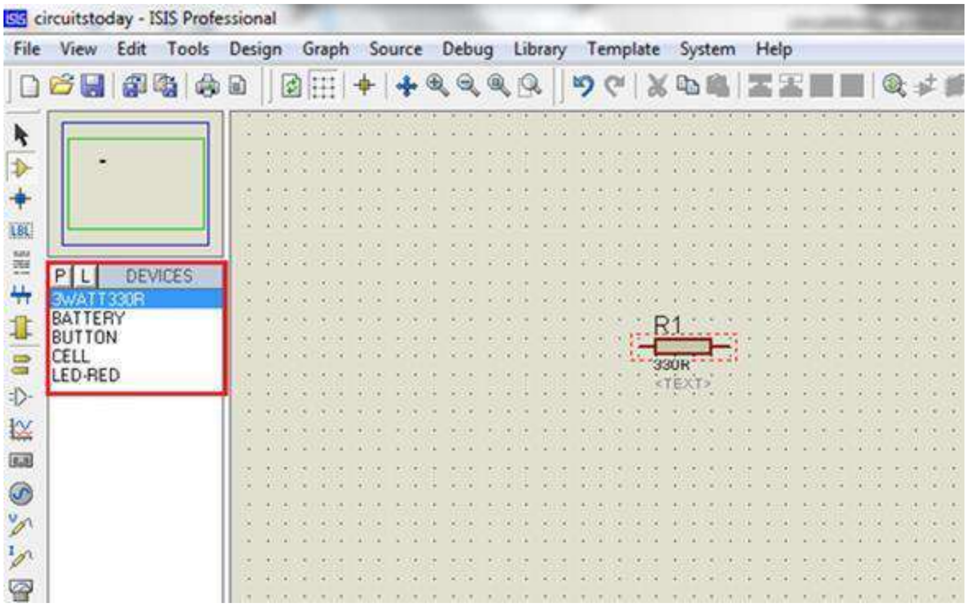


Fig: 3.12.8 Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

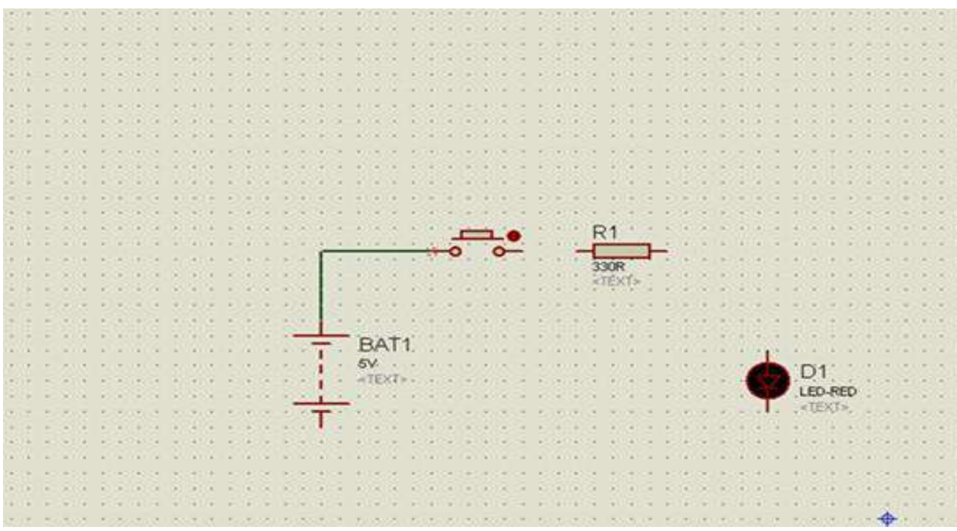


Fig:3.12.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

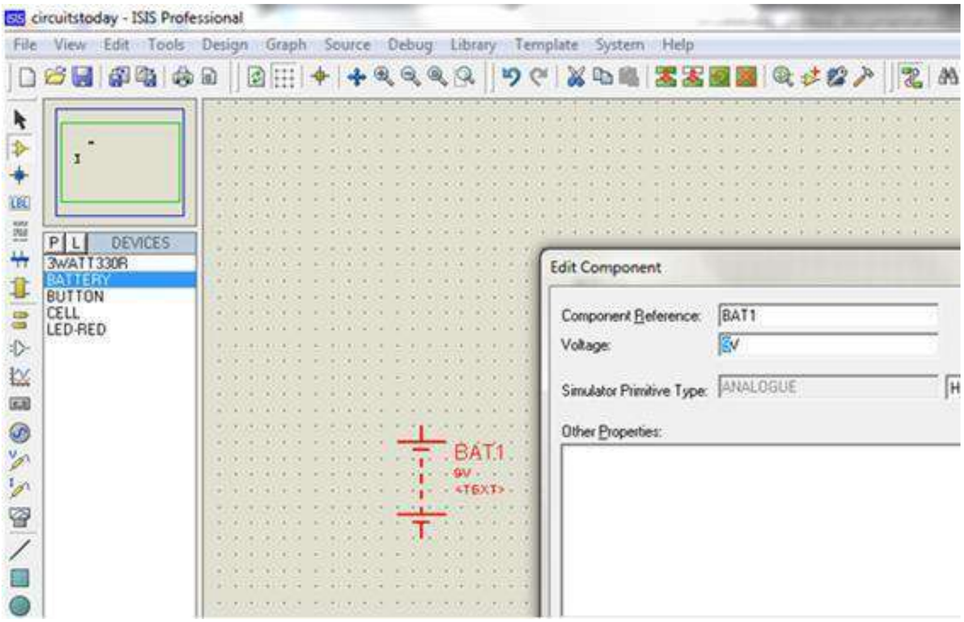


Fig: 3.12.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation

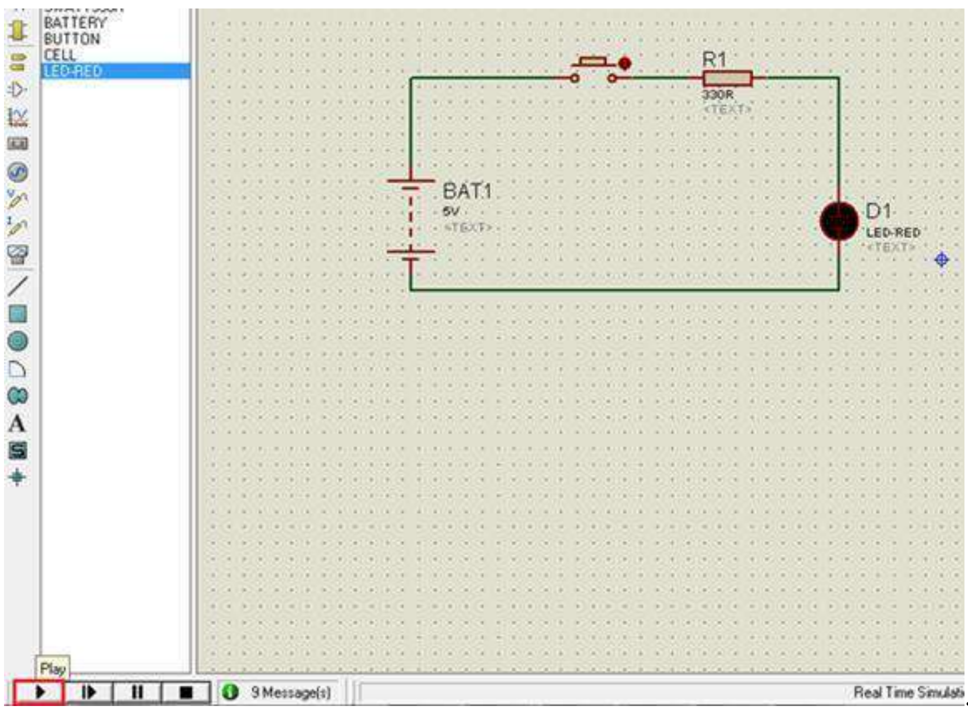


Fig: 3.12.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

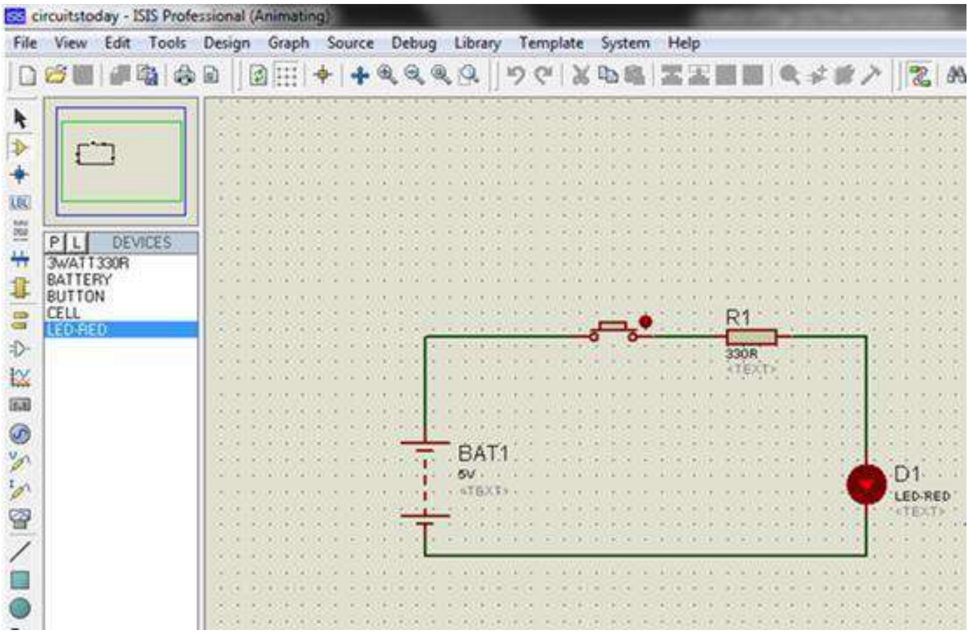


Fig:3.12.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

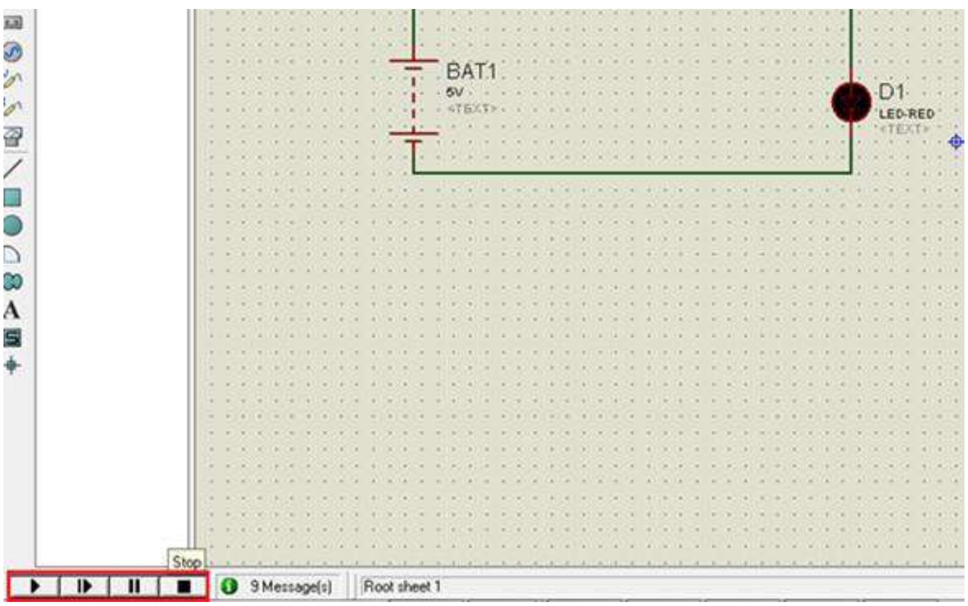


Fig: 3.12.13 Simulation Step-Pause-Stop Buttons

CHAPTER-4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM:

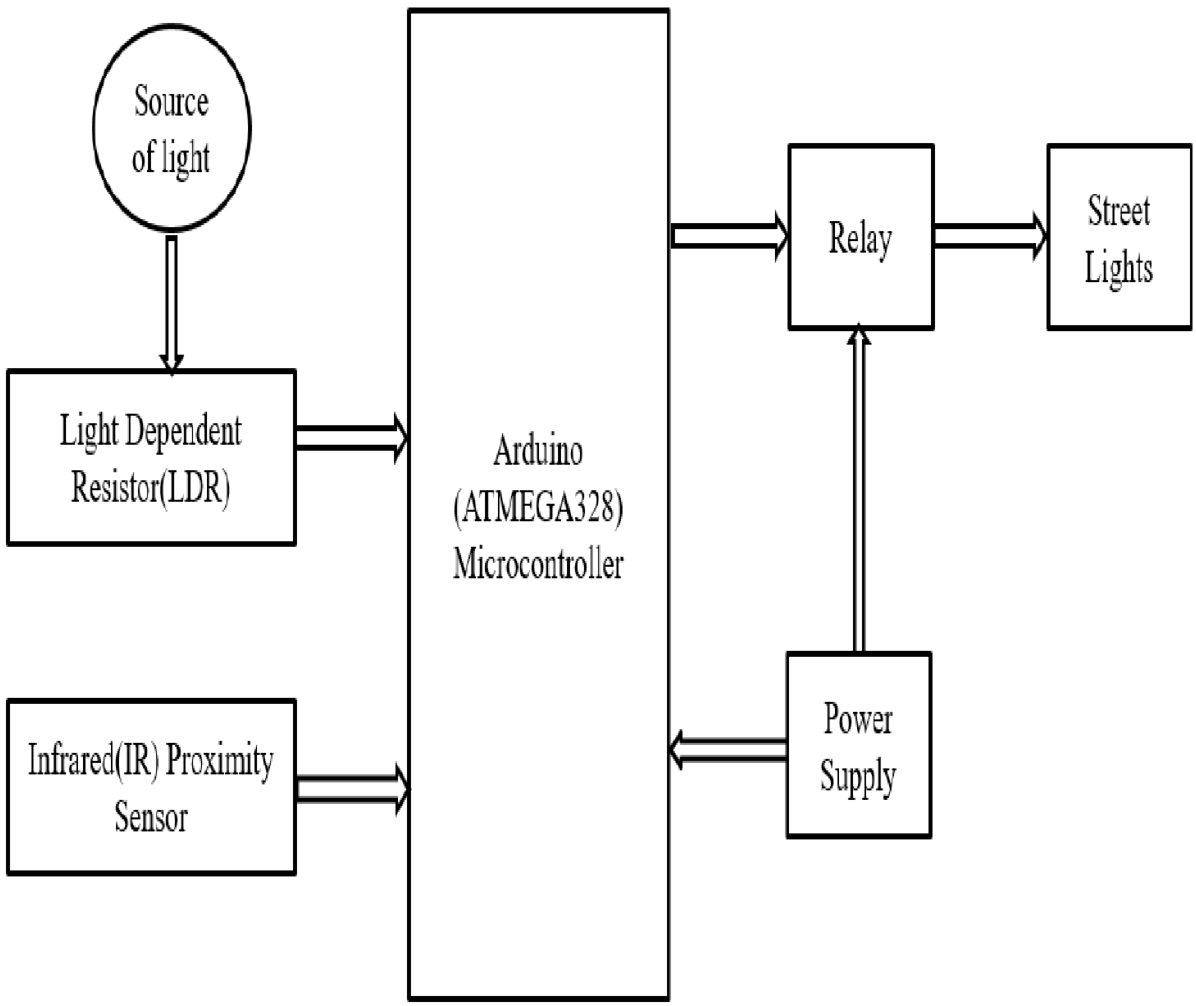


Fig: 4.1 Block Diagram Of The System

4.2 FLOW CHART:

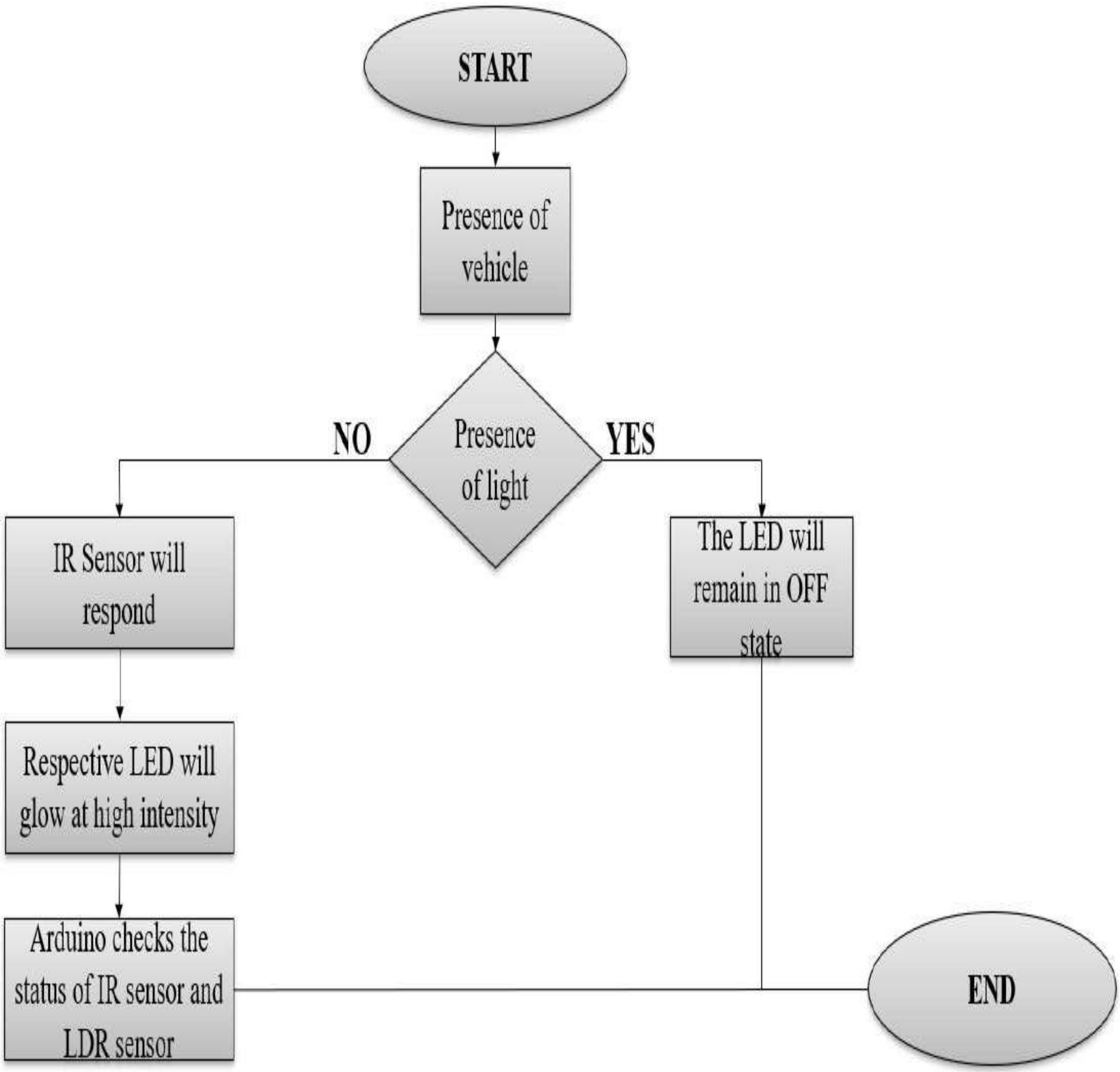


Fig: 4.2 Flow Chart

4.3 IMPLEMENTATION STAGES:

Stage 1:

When we switch on the power supply, the circuit will be activated and LCD displays.

Fig: 4.3.1 Starting Of The Operation

Stage 2:

When light is in surroundings, even when vehicle movement is detected or not, lights will be in OFF state.

Stage 3:

When it is dark, when vehicle movement is there ,the LED bulb corresponding to the movement glows for some time and then switches off.

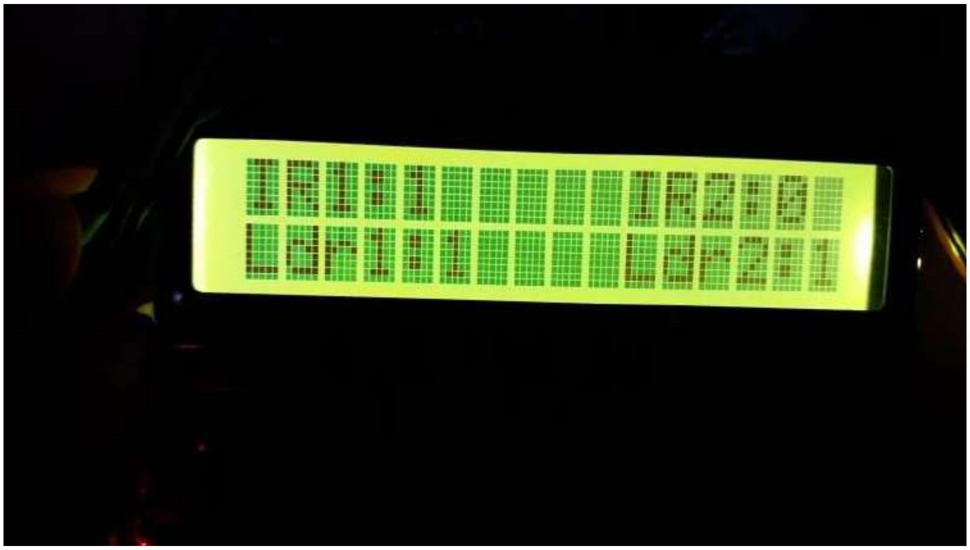


Fig: 4.3.2 Vehicle Movement Is Detected

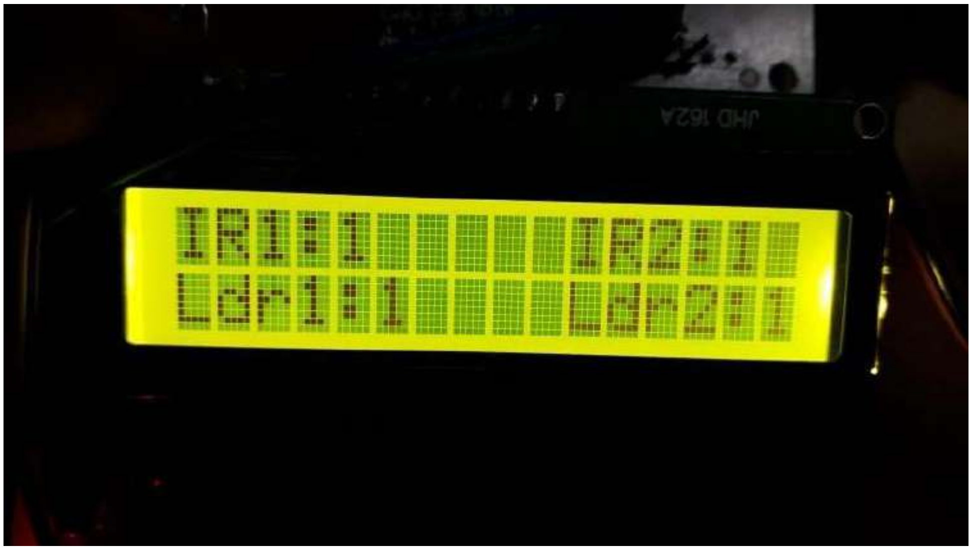


Fig: 4.3.3 Vehicle Movement Is Detected

Working:

Case i:

- Arduino is supplied with power.
- Arduino checks with IR sensor for presence of vehicle.
- If IR sensor doesn't detect any vehicle as an obstacle, then it sends information to Arduino.
- LDR sensor sends information to Arduino about lightening conditions.
- If there is light, then the LED remain in OFF state.

Case ii:

- Arduino checks with IR sensor for presence of vehicle.
- If IR sensor detects any vehicle as an obstacle, then it sends information to Arduino.
- LDR sensor sends information to Arduino about lightening conditions.
- If there is light, then the LED remain in OFF state.

Case iii:

- Arduino checks with IR sensor for presence of vehicle.
- If IR sensor doesn't any vehicle as an obstacle, then it sends information to Arduino.
- LDR sensor sends information to Arduino about lightening conditions.
- If it is dark, then the LED remain in OFF state

Case iv:

- Arduino checks with IR sensor for presence of vehicle.
- If IR sensor detects any vehicle as an obstacle, then it sends information to Arduino.
- LDR sensor sends information to Arduino about lightening conditions.
- If it is dark, then the LED remain in ON state.
- The respective led glows with high intensity until vehicle passes.
- Arduino checks the status of IR and LDR sensors.

4.4 PROJECT CODE:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd (13,12,11,10,9,8);  //RS 13,EN 12, D4 11 ,D5 10,D6 9,D7 8

int ir1 = A0;
int ir2 = A1;
int ldr1 = A2;
int ldr2 = A3;
int light1=6;
int light2=7;

void setup(void)
{
  Serial.begin(9600);
  lcd.begin(16,2);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("WELCOME");
  delay(1000);
 pinMode(light1,OUTPUT);
   pinMode(light2,OUTPUT);
   pinMode(ir1,INPUT);
```

```
  pinMode(ir2,INPUT);
  pinMode(ldr1,INPUT);
  pinMode(ldr2,INPUT);
  digitalWrite(light1,LOW);
  digitalWrite(light2,LOW);
}
void loop(void)
{
// int accident_sensor_data = digitalRead(accident_sensor);
 //Serial.println(accident_sensor_data);
 // delay(1000);
   int id1 = digitalRead(ir1);
   int id2 = digitalRead(ir2);
   int ld1 = digitalRead(ldr1);
   int ld2 = digitalRead(ldr2);
   delay(1000);
   lcd.clear();
   lcd.setCursor(0, 0);lcd.print("I1:");lcd.print(id1);lcd.print("     ");
   lcd.setCursor(8, 0);lcd.print("I2:");lcd.print(id2);lcd.print("     ");
   lcd.setCursor(0, 1);lcd.print("L1:");lcd.print(ld1);lcd.print("     ");
   lcd.setCursor(8, 1);lcd.print("L2:");lcd.print(ld2);lcd.print("     ");
   delay(1000);

   if(id1 == LOW || ld1 == LOW)
    {
      lcd.clear();lcd.setCursor(0, 1);lcd.print("LIGHT 1 ON");
      digitalWrite(light1,HIGH);
      delay(5000);
    }
```

```
    else
    {
      digitalWrite(light1,LOW);
    }
        if(id2 == LOW || ld2 == LOW)
    {
      lcd.clear();lcd.setCursor(0, 1);lcd.print("LIGHT 2 ON");
      digitalWrite(light2,HIGH);
      delay(5000);
    }
     else
    {
      digitalWrite(light2,LOW);
    }
}
```

4.5 FINAL OUTPUT:

- Since it is dark, LDR sensor decreases the resistance and allows conduction.
- When IR sensor detects and object, it sends information to Arduino.
- Arduino receives and analyzes the information and turns LED Bulb into ON state

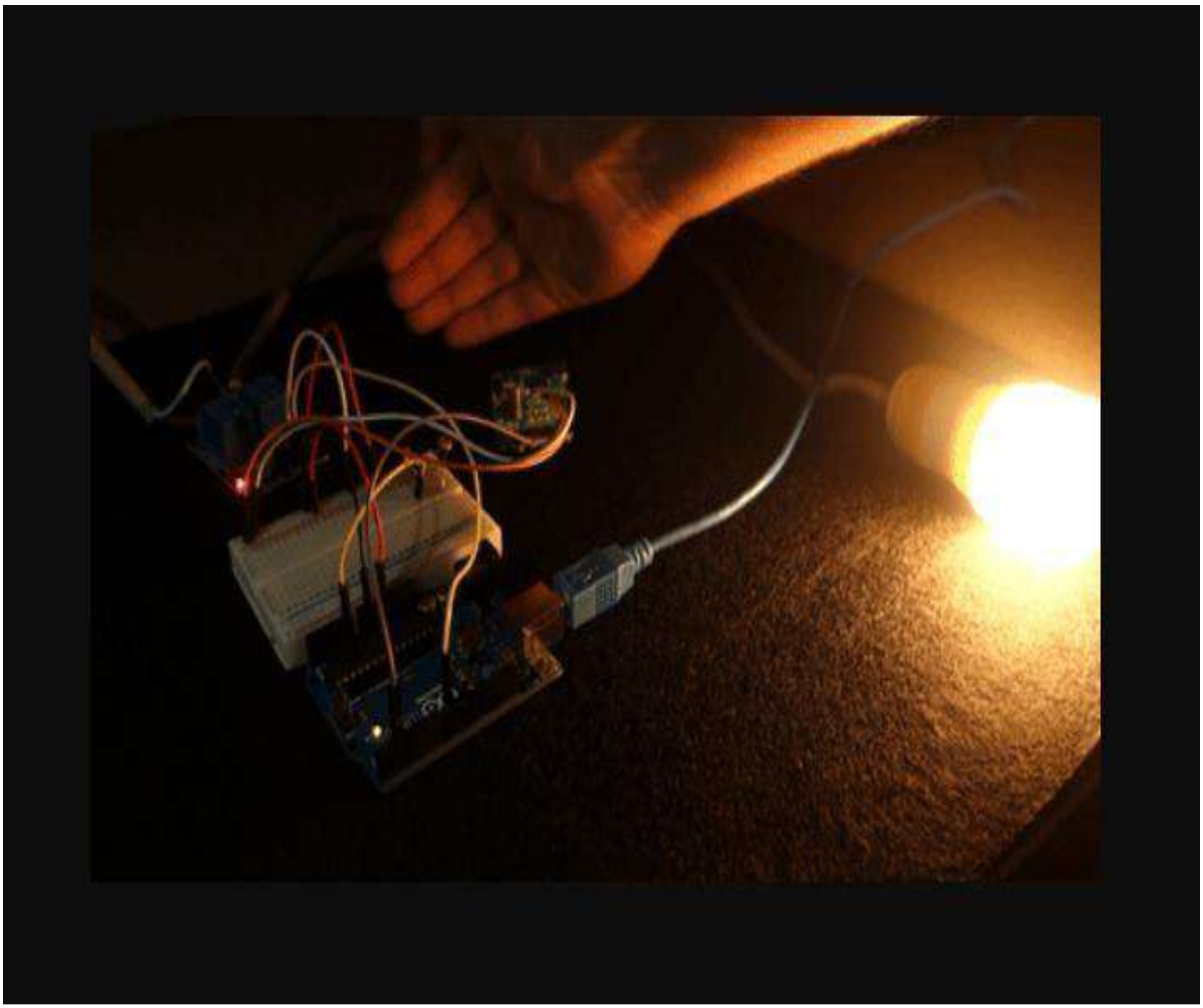


Fig: 4.5 Final Output

.

CHAPTER – 5
PROJECT TESTING

5.1 SOFTWARE TESTING:

Software Testing is a process of executing the application with an intent to find any software bugs. It is used to check whether the application met its expectations and all the functionalities of the application is working. The final goal of testing is to check whether the application is behaving in the way it is supposed to under specified conditions. All aspects of the code are examined to check the quality of application. The primary purpose of testing is to detect software failures so that defects may be uncovered and corrected. The test cases are designed in such way that scope of finding the bugs is maximum.

5.2 TESTING LEVELS:

There are various testing levels based on the specificity of test.

• **Unit testing:** Unit testing refers to tests conducted on a section of code in order to verify the functionality of that piece of code. This is done at the function level.

• **Integration Testing**: Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Its primary purpose is to expose the defects associated with the interfacing of modules.

• **System Testing:** System testing tests a completely integrated system to verify that the system meets its requirements.

• **Acceptance testing:** Acceptance testing tests the readiness of application, satisfying all requirements.

• **Performance testing:** Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or device such as response time or millions of instructions per second etc.

5.3 SYSTEM TEST CASES:

A test case is a set of test data, preconditions, expected results and post conditions, developed for a test scenario to verify compliance against a specific requirement. We have designed and executed a few test cases to check if the project meets the functional requirements.

- LED lighting in the street light provides a greater amount of power consumption in comparison to high pressure sodium lamp due to its various advantages. The system was designed for automatically turn OFF during the day time and automatically turn ON during the bad weather and night time.
- Manual operation for the street light specifically lighting street light at a particular time of the day and switching them off at another specific time consumes to be wastage of a lot of electric energy.
- The proposed system offers a street lightning system based on vehicle movement, i.e. the street lights will be switched on when they detect movement of any vehicle coming towards them. For this system, IR light as a sensor is used with Arduino microcontroller.
- IR sensors will be implanted at a certain distance away from the point where the streets lights are implanted and whenever any vehicle passes by the l sensors, the streets lights will be switched ON and the duration of their ON timings will be controlled by Arduino microcontroller.
- When there are no vehicles passing by, the street lights will be in a dim state and the intensity of the street lights can also be controlled and programmed via Arduino microcontroller.

CHAPTER-6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION:

- By using Smart Street light, one can save surplus amount of energy which is done by LED's and adding additional feature for security purposes.
- It prevents unnecessary wastage of electricity, caused due to manual switching of streetlights when it's not required.
- The system is now used only for one way traffic in highways, and can be extended to make the current system in two-way traffic.

6.2 FUTURE SCOPE:

- The number of channels can be increased to interface a greater number of light sensors which is possible by using advanced versions of microcontrollers.
- We can record number of vehicles and vehicle registration number by using internet of things. It detects the speed of vehicles passing by.
- It displays the status of the street and it helps to control. This way it helps decrease the calamities.
- The system can be driven by solar panel for more efficiency and less power consumption.
- Further the project can be enhanced by using appropriate sensors for detecting the failed street light and then sending an SMS to the control department via GSM modem for appropriate action.

Using this smart project, we can also estimate the speed of the vehicle, recognizing the number plate, recognizing the accidents took place on roads etc. This Smart Street light project not only helps in rural areas but also beneficial in urban areas too. As we are moving towards more advancement, we require more power so use of renewable resources is useful and advantageous. With this project, we can even add smart parking of vehicle and it is even useful for driverless cars. This project has a bright future not only to save power but also reduced the calamities and even reduced the crime rate.

PUBLICATION

Submitted Paper in the Conference ICSMEC – 21 with Paper ID (ICSMEC21-0012) and got Acceptance of paper.

REFERENCES

[1] Payal Rodi, Leena Chandrakar, Saylee Ginde Sivanantham S and Sivasankaran K, “Energy Conservation using Automatic Lighting System using FPGA”, International Conference on Green Engineering and Technologies, 2015.

[2] Deepak Kumar Rath,” Arduino Based: Smart Light Control System”, International Journal of Engineering Research and General Science Volume 4, March- April, 2016.

[3] Gouthami. C, Santosh. C, A. Pavan Kumar, Karthik. A, and Ramya.K.R, “Design and Implementation of Automatic Street Light Control System using Light Dependent Resistor”, International Journal of Engineering Trends and Technology (IJETT) – Volume 35 Number 10 - May 2016.

[4] Ahmed ShariqueAnees, Anupriya, Ayushy Chowdhary,Shalini Dubey and Shweta Verma, “Solar Powered Led Streetlight with Automatic Intensity Control”, International journal of innovative research in electrical, electronics, instrumentation and control engineering, Vol. 3, June 2015.

[5] Prof. K.Y.Rajput, Gargeyee Khatav, Monica Pujari and Priyanka Yadav, “Intelligent Street Lighting System Using Gsm”, International Journal of Engineering Science Invention, March, 2013.

[6] S. Suganya, R. Sinduja, T. Sowmiya and S. Senthilkumar, “Street light glow on detecting vehicle movement using sensor”, International journal for advance research in Engineering and technology, 2014.

[7] Sanal Malhotra1 and Shiv Taneja, “Automatic Brightness Control Using LDR Sensors”, Advanced Research in Electrical and Electronic Engineering, 2014.

[8] Ch Kavitha, Swathi, Satyaprakash and Vinitha, “Solar Powered LED Street Lighting with Auto Intensity Control”, International Journal of Scientific & Engineering Research, July, 2016.

[9] Parkash, Prabu V and Dandu Rajendra, “Internet of Things Based Intelligent Street Lighting System for Smart City”, International Journal of Innovative Research in Science Engineering and Technology, May, 2017

[10] ieeexplore.ieee.org/Smart Street light system based on image processing.

[11] eeexplore.ieee.org/A smart street-light intensity optimizer

[12]
https://www.researchgate.net/publication/326199352_Automatic_Street_Light_Control_by_Detecting_Vehicle_Movement

[13]
https://www.researchgate.net/publication/323280626_Internet_Of_Things_Based_Intelligent_Street_Lighting_System_for_Smart_City

[14] Schwab, R.N., Walton, N.E., Mounce, J.M., and Rosenbaum, M.J. (1982) Synthesis of Safety Research Related to Traffic Control and Roadway Elements Volume 2, Chapter 12: Highway Lighting. Report No. FHWA-TS-82-233. Federal Highway Administration. Imperial Journal of Interdisciplinary Research (IJIR) Vol-3, Issue-5, 2017 ISSN: 2454-1362, http://www.onlinejournal.in Imperial Journal of Interdisciplinary Research (IJIR) Page 708

[15] Elvik, R. (1995) "Meta-Analysis of Evaluations of Public Lighting as Accident Countermeasure." Transportation Research Record 1485, TRB, National Research Council, Washington, D.C., pp. 112-123. [3] Commission Internationaal de l'Éclairage. (1992) Road Lighting as an Accident Countermeasure1. Reinhard Müllner, Andreas Riener, (2011) "An energy efficient pedestrian aware Smart Street Lighting system", International Journal of Pervasive Computing and Communications, Vol.7 Iss: 2, pp.147 – 161 [4] Electrical Safety of Street Light Systems, Giuseppe Parise, Fellow, IEEE, Luigi Martirano, Senior Member, IEEE, and Massimo Mitolo, Senior Member, IEEE, VOL. 26, NO. 3, JULY 2011.

[16] Design and fabrication of automatic street light control system, M. A. Wazed, N. Nafis, M. T. Islam and, Vol. 5, No. 1, June 2010, pp 27-34

[17] "The Saracens themselves were the originators not only of algebra, chemistry, and geology, but of many of the so-called improvements or refinements of civilization, such as street lamps, window-panes, firework, stringed instruments, cultivated fruits, perfumes, spices, etc."

[18] Scott, S. P. (1904). History of the Moorish Empire in Europe. Philadelphia and London.: J. B. Lippincott Company. p. 3 vols. Artz, F. B. (1980). The Mind of the Middle Ages. Third edition revised. University of Chicago Press. pp. 148–50.

[19] "Sir Joseph Wilson Swan". home.frognet.net. Retrieved 16 October 2010.

[20] Electrical Safety of Street Light Systems, Giuseppe Parise, Fellow, IEEE, Luigi Martirano, Senior Member, IEEE, and Massimo Mitolo, Senior Member, IEEE, VOL. 26, NO. 3, JULY 2011.

[21] Energy Efficient Lighting Control System Design for Corridor illumination, Jayashri A. Bangali, Arvind D. Shaligram International Journal of Scientific & Engineering Research Volume 3, Issue 4, April-2012, ISSN 2229-5518.

[21] Wireless Streetlight Control System, Deepak Kapgate, G.H. Raisoni College of engineering, Nagpur University, International Journal of Computer Applications (0975 –8887), Volume 41– No.2, March 2012

[12] Design and fabrication of automatic street light control system, M. A. Wazed, N. Nafis, M. T. Islam and, Vol. 5, No. 1, June 2010, pp 27-34

[22] "The Saracens themselves were the originators not only of algebra, chemistry, and geology, but of many of the so-called improvements or refinements of civilization, such as street lamps, windowpanes, firework, stringed instruments, cultivated fruits, perfumes, spices, etc."

[23] Scott, S. P. (1904). History of the Moorish Empire in Europe. Philadelphia and London.: J. B. Lippincott Company. P. 3 vols. Artz, F. B. (1980). The Mind of the Middle Ages. Third edition revised. University of Chicago Press. Pp. 148– 50.

[24] "Sir Joseph Wilson Swan". Home.frognet.net. Retrieved 16 October 2010.

[25] Schwab, R.N., Walton, N.E., Mounce, J.M., and Rosenbaum, M.J. (1982) Synthesis of Safety Research Related to Traffic Control and Roadway Elements Volume 2, Chapter 12: Highway Lighting. Report No. FHWA-TS-82233.Federal Highway Administration.

BIBILOGRAPHY

http://www.efxkits.com/arduino-based-auto-intensity-control-of-street-lights[1]

http://www.ehow.com/how-does_5561845_do-ir-sensors-work.html#ixzz310syLD6I[2]

http://homepages.which.net/~paul.hills/Emc/BecBody.html[3]

Sanal Malhotra1 and Shiv Taneja, "Automatic Brightness Control Using LDR Sensors", Advanced Research in Electrical and Electronic Engineering, 2014.

Ch Kavitha, Swathi, Satyaprakash and Vinitha, "Solar Powered LED Street Lighting with Auto Intensity Control", International Journal of Scientific & Engineering Research, July, 2016.

http://en.wikipedia.org/wiki/Arduino[4]

http://www.galco.com/comp/prod/sensor.htm[5]

http://www.passmyexams.co.uk/GCSE/physics/LM324.html[6]

https://en.wikipedia.org/wiki/Internet_of_Things[7]

http://www.internetsociety.org/doc/IoT-overview[8]

https://www.firebase.com[9]

https://en.wikipedia.org/wiki/Firebase[10]

https://en.wikipedia.org/wiki/ULN2003A[11]

APPENDICES

This project report gives the detail study of the “Smart Street Light System”. The construction, working principle, implementation of the project is given throughout this report. Circuit meets expectations appropriately to turn road light ON/OFF depending on the movement of the vehicle. LD sensor senses the day or night time. The IR sensor senses the motion of the vehicle.

The results of our project work support our hypothesis that the Smart Street Light circuit would solve the power consumption problem. The final conclusion drawn from our project is that the circuit is very efficient and it can be used in street of India.

A

PROJECTREPORT

On

TOWARDS FASTER THAN NYQUIST TRANSMISSION FOR BEYOND 5G WIRELESS COMMUNICATION

Submitted by

1)Mr. B. Sai Santosh (17K81A04J3)
2)Mr. MD Muttaheer Ahmed (17K81A04M9)
3)Mr. S.T. Narsimha Charyulu (17K81A04P0)

In partial fulfillment for the award of the degree in

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Dr. A. Chaitanya Krishna

Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500100

NBA &NAAC A+ Accredited

JUNE 2021

St. MARTIN'S ENGINEERING COLLEGE
UGC AUTONOMOUS
Dhulapally, Secunderabad-500 100
NBA &NAAC A+ Accredited

BONAFIDE CERTIFICATE

This is to certify that the project entitled **"TOWARDS FASTER THAN NYQUIST TRANSMISSION FOR BEYOND 5G WIRELESS COMMUNICATION",** is being submitted by **1. Mr. B. Sai Santosh (17K81A04J3) 2. Mr. MD Muttaheer Ahmed (17K81A04M9) 3. S.T. Narsimha Charyulu (17K81A04P0)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS & COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Dr. A. Chaitanya Krishna
Department of ECE

Dr. B. Hari Krishna
Head of the Department

Internal Examiner

External Examiner

Place:

Date :

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **BALLA SAI SANTOSH** WITH ROLL NO.**17K81A04J3, MOHAMMED MUTTAHEER AHMED** WITH ROLL NO.**17K81A04M9**, **S T NARSIMHA CHARYULU** WITH ROLL NO.**17K81A04P0,** OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**TOWARDS FASTER-THAN-NYQUIST TRANSMISSION FOR BEYOND 5G WIRELESS COMMUNICATIONS**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of **'Bachelor of Technology in Department of Electronics & Communication Engineering'**, session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled **'TOWARDS FASTER THAN NYQUIST TRANSMISSION FOR BEYOND 5G WIRELESS COMMUNICATION'** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

B. Sai Santosh(17K81A04J3)
MD Muttaheer Ahmed(17K81A04M9)
S.T. Narsimha Charyulu(17K81A04P0)

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr. P. Santosh Kumar Patra**, **our beloved Principal, St. Martin's Engineering College** for his in valuable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his price less and meticulous supervision teach and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr. B. Hari Krishna**, **Head &Professor**, the Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our **Department project coordinator Dr. A. Anand Kumar** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project guide, **Dr. A. Chaitanya Krishna, Professor,** Department of ECE and our class project coordinator **Ms. P. Pushpa, Assistant Professor**, Department of ECE for guidance and encouragement in carrying out the project work

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr. Malayadri**, **Professor in IT** and **DR. R. Santosh Kumar, Associate professor in CSE** for his advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

1. B. Sai Santosh
2. MD Muttaheer Ahmed
3. S.T. Narsimha Charyulu

ABSTRACT

Faster-Than-Nyquist (FTN) is a technique that can improve the spectral efficiency of communication systems by making better use of available spectrum resources at the cost of inter-symbol interference (ISI) and inter-carrier interference (ICI). In this paper, we propose a hybrid signalling scheme for a practical application of FTN for MIMO transmission. We propose a new slot structure optimized for the hybrid signalling supporting both FTN signalling and orthogonal frequency division multiplexing (OFDM) signalling. Specifically, in the proposed slot structure, data transmission is based on the FTN signalling and the pilot transmission is based on the OFDM signalling. Numerical results confirm that the proposed signalling scheme has clear benefit over the systems employing only OFDM or FTN signalling.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND DEFINITIONS:

FTN	:	Faster than Nyquist Transmission
MIMO	:	Multi Input Multi Output
OFDM	:	Orthogonal Frequency Division Multiplexing
MSE	:	Mean Square Error
PAPR	:	Peak Average power Ratio
GUI	:	Graphical User Interface

GLOSSARY OF TERMS:

FTN : Faster than Nyquist Signalling scheme transmits signals faster than the Nyquist rate in order to improve efficiency

PAPR : Peak to Peak average ratio reduction scheme for OFDM system

OFDM : Multiplexing is a technique for transmitting large amounts of digital data over radio wave.

Chapter 1

INTRODUCTION

With the advent of 5G wireless communications, we are witnessing unprecedented services such as tactile internet, virtual augmented reality, autonomous driving, factory automations, and high-resolution video streaming, to name just a few [1], [2]. To support wide variety of services and enhanced user experience, we need to push the spectral efficiency of wireless systems to the limit. One viable approach to enhance the spectral efficiency is to transmit the symbol faster than the Nyquist rate. In doing so, symbol rate higher than two times of bandwidth can be achieved at the expense of the orthogonality violation among symbols. Such a transmission scheme, commonly called the faster-than-Nyquist (FTN) signaling [3], has received much attention in recent years as a means to improve the spectral efficiency of wireless systems. It has been shown that 25% faster symbol transmission is achieved by employing deliberately designed pulse shaping filters and binary symbols without virtual loss of performance [3]. It has also been shown in [4] that 42% faster symbol transmission can be achieved by the FTN signaling with square-root raised cosine pulse shaping filter.

Recently, multi-carrier systems employing the FTN signaling have been proposed [5]. Key feature of this scheme over the single carrier counterpart is to improve the bandwidth utilization by reducing the distance between adjacent subcarriers. Although multi-carrier FTN technology has received some research interests, commercial use of this scheme is rare due to the complication caused by the interference-carrier interference (ICI) control. In fact, due to the ICI caused by the overlap of pilot subcarriers, channel estimation quality would be deteriorated, resulting in the severe degradation of the symbol detection and decoding performance. In [6], effect of ICI on the channel estimation has been analyzed when the pilot signals are allocated using the FTN signaling. In [7], an approach to turn off part of subcarriers to ensure the orthogonality of pilot subcarriers has been suggested.

An aim of this paper is to propose an FTN signaling scheme suitable for multiple-input multiple-output (MIMO) transmission. Our work is motivated by the fact that the FTN-based pilot transmission degrades the channel estimation quality severely, in particular for MIMO systems. In order to support more symbols than the conventional OFDM systems can allow while ensuring the accuracy of the channel estimation quality, we propose a hybrid signaling scheme supporting both FTN signaling and OFDM signaling in the transmission time interval (TTI). Key feature of the proposed scheme is to support the data transmission using the FTN signaling and control and pilot signals using the conventional OFDM signaling. Since the pilot signals are transmitted by the OFDM signaling, orthogonality of pilot subcarriers is guaranteed, which ensures that the accurate channel estimation can be achieved even in the MIMO transmission. By controlling the ICI terms of data symbols using the nonlinear receiver techniques (e.g., BCJR [8]), we can achieve the substantial improvement in the spectral efficiency of the data transmission. In our simulations, we show that the proposed hybrid signaling achieves more than 20% gain in the spectral efficiency over the conventional OFDM systems.

1.1 MIMO-OFDM

MIMO-OFDM is the foundation for most advanced wireless local area network (Wireless LAN) and mobile broadband network standards because it achieves the greatest spectral efficiency and, therefore, delivers the highest capacity and data throughput. Greg Raleigh invented MIMO in 1996 when he showed that different data streams could be transmitted at the same time on the same frequency by taking advantage of the fact that signals transmitted through space bounce off objects (such as the ground) and take multiple paths to the receiver. That is, by using multiple antennas and precoding the data, different data streams could be sent over different paths. Raleigh suggested and later proved that the processing required by MIMO at higher speeds would be most manageable using OFDM modulation, because OFDM converts a high-speed data channel into a number of parallel, lower-speed channels.

Operations:

In modern usage, the term “MIMO” indicates more than just the presence of multiple transmit antennas (multiple input) and multiple receive antennas (multiple output). While multiple transmit antennas can be used for beamforming, and multiple receive antennas can be used for diversity, the word "MIMO" refers to the simultaneous transmission of multiple signals (spatial multiplexing) to multiply spectral efficiency (capacity).

Traditionally, radio engineers treated natural multipath propagation as an impairment to be mitigated. MIMO is the first radio technology that treats multipath propagation as a phenomenon to be exploited. MIMO multiplies the capacity of a radio link by transmitting multiple signals over multiple, co-located antennas. This is accomplished without the need for additional power or bandwidth. Space-time codes are employed to ensure that the signals transmitted over the different antennas are orthogonal to each other, making it easier for the receiver to distinguish one from another. Even when there is line of sight access between two stations, dual antenna polarization may be used to ensure that there is more than one robust path.

OFDM enables reliable broadband communications by distributing user data across a number of closely spaced, narrowband subchannels. This arrangement makes it possible to eliminate the biggest obstacle to reliable broadband communications, inter symbol interference (ISI). ISI occurs when the overlap between consecutive symbols is large compared to the symbols’ duration. Normally, high data rates require shorter duration symbols, increasing the risk of ISI. By dividing a high-rate data stream into numerous low-rate data streams, OFDM enables longer duration symbols. A cyclic prefix (CP) may be inserted to create a (time) guard interval that prevents ISI entirely. If the guard interval is longer than the delay spread—the difference in delays experienced by symbols transmitted over the channel—then there will be no overlap between adjacent symbols and consequently no inter symbol interference. Though the CP slightly reduces spectral capacity by consuming a small percentage of the available bandwidth, the elimination of ISI makes it an exceedingly worthwhile tradeoff.

A key advantage of OFDM is that fast Fourier transforms (FFTs) may be used to simplify implementation. Fourier transforms convert signals back and forth between the time domain and frequency domain. Consequently, Fourier transforms can exploit the fact that any complex waveform may be decomposed into a series of simple sinusoids. In signal processing applications, discrete Fourier transforms (DFTs) are used to operate on real-time signal samples. DFTs may be applied to composite OFDM signals, avoiding the need for the banks of oscillators and demodulators associated with individual subcarriers. Fast Fourier transforms are numerical algorithms used by computers to perform DFT calculations.

FFTs also enable OFDM to make efficient use of bandwidth. The subchannels must be spaced apart in frequency just enough to ensure that their time-domain waveforms are orthogonal to each other. In practice, this means that the subchannels are allowed to partially overlap in frequency.

MIMO-OFDM is a particularly powerful combination because MIMO does not attempt to mitigate multipath propagation and OFDM avoids the need for signal equalization. MIMO-OFDM can achieve very high spectral efficiency even when the transmitter does not possess channel state information (CSI). When the transmitter does possess CSI (which can be obtained through the use of training sequences), it is possible to approach the theoretical channel capacity. CSI may be used, for example, to allocate different size signal constellations to the individual subcarriers, making optimal use of the communications channel at any given moment of time.

More recent MIMO-OFDM developments include multi-user MIMO (MU-MIMO), higher order MIMO implementations (greater number of spatial streams), and research concerning “massive MIMO” and “Cooperative MIMO” for inclusion in coming 5G standards.

MU-MIMO is part of the IEEE 802.11ac standard, the first Wi-Fi standard to offer speeds in the gigabit per second range. MU-MIMO enables an access point (AP) to transmit to up to four client devices simultaneously. This eliminates contention delays, but requires frequent channel measurements to properly direct the signals. Each user may employ up to four of the available eight spatial streams. For example, an AP with eight antennas can talk to two client devices with four antennas, providing four spatial streams to each. Alternatively, the same AP can talk to four client devices with two antennas each, providing two spatial streams to each.

Multi-user MIMO beam forming even benefits single spatial stream devices. Prior to MU-MIMO beamforming, an access point communicating with multiple client devices could only transmit to one at a time. With MU-MIMO beamforming, the access point can transmit to up to four single stream devices at the same time on the same channel.

The 802.11ac standard also supports speeds up to 6.93 Gbit/s using eight spatial streams in single-user mode. The maximum data rate assumes use of the optional 160 MHz channel in the 5 GHz band and 256 QAM (quadrature amplitude modulation). Chipsets supporting six spatial streams have been introduced and chipsets supporting eight spatial streams are under development.

Massive MIMO consists of a large number of base station antennas operating in a MU-MIMO environment While LTE networks already support handsets using two spatial streams, and handset antenna designs capable of supporting four spatial streams have been tested, massive MIMO can deliver significant capacity gains even to single spatial stream handsets. Again, MU-MIMO beamforming is used to enable the base station to transmit independent data streams to multiple handsets on the same channel at the same time. However, one question still to be answered by research is: When is it best to add antennas to the base station and when is it best to add small cells?

Another focus of research for 5G wireless is Cooperative MIMO (CO-MIMO). In CO-MIMO, clusters of base stations work together to boost performance. This can be done using macro diversity for improved reception of signals from handsets or multi-cell multiplexing to achieve higher downlink data rates. However, CO-MIMO requires high-speed communication between the cooperating base stations.

1.2 Introduction and Background:

The OFDM strategy is based off a binary coding system called QPSK, which is a 2-bit per symbol format. The purpose of this system is to help detect errors and increase security by encrypting the data. We will use the following digital encoding system as an example of OFDM strategy:

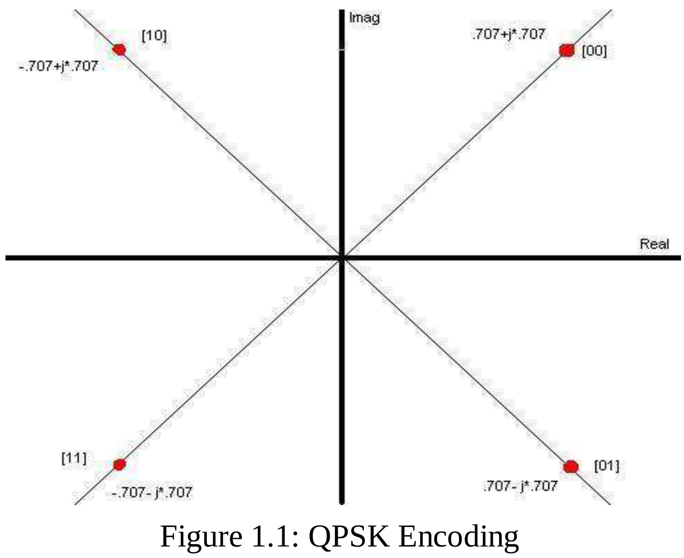


Figure 1.1: QPSK Encoding

The general full design of the OFDM system starts at an analog to digital converter (ADC) that converts analog data to digital if the system is for example a cellular telephone. If the source of data is a digital component, such as a computer, this step is skipped. Next, the digital data sorted, every two bits into its equivalent QPSK format as shown above in Figure 1. The data is sorted so it is encoded depending on which quadrant of the complex plane it lies in. For example, a bit pairing of [10] would correspond to -0.707+ j*0.707, [01] would correspond to 0.707-j*0.707 and so on.

This data is then grouped into 16 distinct, separate channels; the purpose of this will later be shown when noise figures are examined. Next the Inverse Fast Fourier Transform is computed, and the data is shifted out serially to an antenna, cable or other transmitting device.
The full system can be looked at in Figure 2:

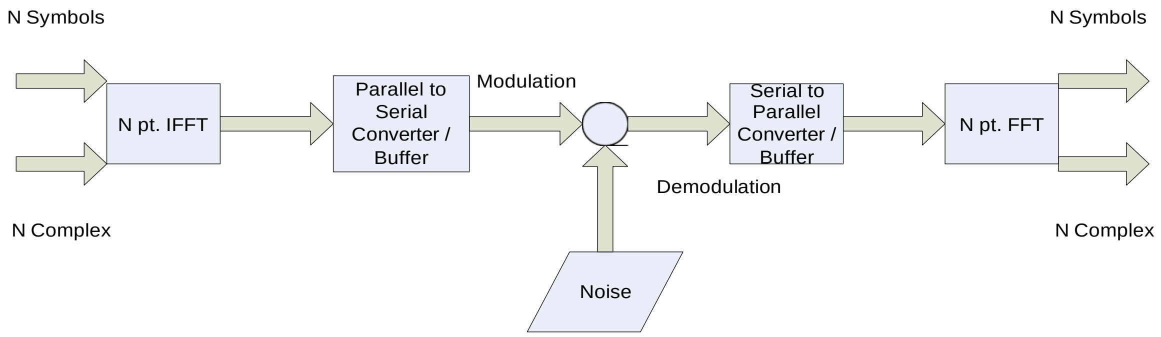


Figure 1.2: Full OFDM System

During the data's transmission, often noise is mixed in with the signal. So, at the receiver, the original data plus noise is entered into a Fast Fourier Transform after a serial to parallel conversion has taken place. The data resulting from the FFT would be complex in nature, and an effort will be made to reconstruct the original binary sequence of numbers for each channel.

Multiple Input Multiple Output (MIMO)-OFDM is widely recognized as a key technology for future wireless communications due to its high spectral efficiency and superior robustness to multipath fading channels. For MIMO-OFDM systems, accurate channel estimation is essential to guarantee the system performance. Generally, there are two categories of channel estimation scheme for MIMO-OFDM systems. The first one is nonparametric scheme, which adopts orthogonal frequency-domain pilots or orthogonal time-domain training sequences to convert the channel estimation in MIMO systems to that in single antenna systems. However, such scheme suffers from high pilot overhead when the number of transmit antennas increases. The second category is parametric channel estimation scheme, which exploits the sparsity of wireless channels to reduce the pilot overhead. The parametric scheme is more favorable for future wireless systems as it can achieve higher spectral efficiency. However, path delays of sparse channels are assumed to be located at the integer times of the sampling period, which is usually unrealistic in practice.

In this letter, a more practical sparse MIMO-OFDM channel estimation scheme based on spatial and temporal correlations of sparse wireless MIMO channels is proposed to deal with arbitrary path delays. The main contributions of this letter are summarized as follows. First, the proposed scheme can achieve super-resolution estimates of arbitrary path delays, which is more suitable for wireless channels in practice. Second, due to the small scale of the transmit and receive antenna arrays compared to the long signal transmission distance in typical MIMO antenna geometry, channel impulse responses (CIRs) of different transmit-receive antenna pairs share common path delays, which can be translated as a common sparse pattern of CIRs due to the spatial correlation of MIMO channels.

Meanwhile, such common sparse pattern is nearly unchanged along several adjacent OFDM symbols due to the temporal correlation of wireless channels. Compared with previous work which just simply extends the sparse channel estimation scheme in single antenna systems to that in MIMO by exploiting the spatial correlation of MIMO channels [5] or only considers the temporal correlation for single antenna systems, the proposed scheme exploits both spatial and temporal correlations to improve the channel estimation accuracy. Third, we reduce the pilot overhead by using the finite rate of innovation (FRI) theory, which can recover the analog sparse signal with very low sampling rate, as a result, the average pilot overhead per antenna only depends on the channel sparsity level instead of the channel length.

1.3 Sparse Channel Model:

Multipath signal propagation, the most salient feature of wireless channels, is a curse and a blessing from the viewpoint of capacity and reliability of such channels. On the one hand, multipath leads to signal fading – fluctuations in received signal strength – that severely impacts the reliability of such channels. On the other hand, knowledge of multipath structure can be exploited for diversity – multiple independent modes of communication – to increase the rate and/or reliability of communication. The impact of fading versus diversity on performance is governed by the amount of channel state information (CSI) known to the system. For example, if perfect CSI is available at the receiver (coherent communication), then the reliability of the fading channel converges to that of the AWGN as the level of diversity increases. Furthermore, the gap in the performance of coherent or non-coherent communication is generally quite significant. Technological advances in wideband multi-antenna RF frontends are enabling learning CSI at a finer resolution afforded by the increase in the spatial temporal signal space dimensions.

Accurate modeling of channel characteristics in time, frequency, and space, as a function of physical multipath characteristics, is thus critical for analyzing the impact of such emerging sophisticated RF front ends. In particular, while most existing models for wireless channels assume a rich multipath environment, there is growing experimental evidence that physical channel exhibit a sparse structure even with small number of antennas and especially at wide bandwidths we use a virtual representation of physical multipath channels that we have developed in the past several years to present a framework for modeling spare multipath channels and to study certain the implications of sparsity on channel learning and optimal communication. The virtual representation samples the multipath geometry in angle delay-Doppler at a resolution commensurate with the signal space dimensions and characterizes the statistically independent degrees of freedom (DOF) available for communication.

Sparse channels correspond to a sparse set of dominant non-vanishing virtual coefficients. A key implication of sparse multipath is that the DOF scale sub-linearly with the signal space dimensions in contrast to the linear scaling inherent in most existing models that implicitly assume a rich multipath. Sparsity of multipath in angle-delay Doppler leads to channel coherence in time, frequency and space that has significant implications for optimal communication in the low-SNR/wideband regime. In particular, we show that sparse multipath channels are perfectly learnable in the limit of large bandwidth and thus naturally bridge the gap between coherent and non-coherent extremes. From a spatial viewpoint, we argue that adapting the array configurations (antenna spacings) can dramatically increase MIMO capacity in sparse multipath in the low-SNR regime.

The following characteristics of MIMO are considered:

1) **ChannelSparsity**: LTE resolves the individual propagation paths from transmitters to receivers which results in a channel impulse response showing only a few peaks and many zeros which can be stated as a sparse signal. In an outdoor environment CIR is normally sparse due to many significant scatterers.

2) **SpatialCorrelation**: As compared to the long signal transmission distance, the scale of the transmit/receive antenna array is impeccably small, therefore channels of various transmit-receive antenna pairs share very similar scatterers. Meanwhile, for many communication systems, the path delay difference from the same scatterer is very much less than the system sampling period. Hence, CIRs of different transmit receive antenna pairs have a common sparse pattern, even though the corresponding path gains may be quite different.

3) **TemporalCorrelation**: Path delays of CIRs for several adjacent OFDM symbols are almost unchanged which can be equivalently stated as a common sparse pattern of CIRs because of the temporal correlation of MIMO channels. In wireless channels, the path gains vary continuously compared to the path delays. Hence, the channel sparse pattern is almost unchanged for several adjacent OFDM symbols as well as the path gains are also correlated.

SUPER RESOLUTION SPARSE MIMO-OFDM CHANNEL ESTIMATION

A pilot pattern is briefly introduced based on which a super resolution sparse MIMO-OFDM channel estimation is applied. Also, the required number of pilots is discussed under the framework of FRI theory. A. Pilot Pattern The pilot pattern used is shown in fig.3, P pilots are spaced equidistant to each other such that there is a pilot interval G between each other. In the meantime, each pilot is assigned with a pilot index I such that 0<I

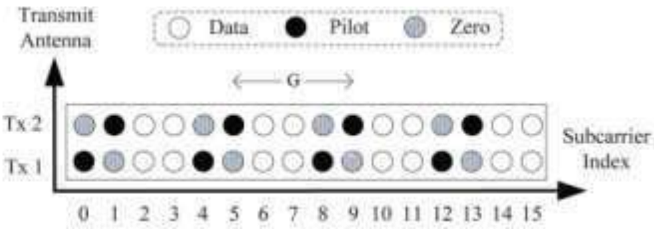


Figure 1.3: Pilot Pattern

B. SpatialandTemporalCorrelation: As the wireless channel is sparse by nature and the small scale of various transmit/receive antennas is negligible with respect to the large signal transmission distance, CIRs of multiple pairs of transmit-receive antennas have the common path delays, which is correspondingly interpreted as a common sparse pattern of CIRs owing to the spatial correlation of the MIMO channels. Hence, by utilizing such spatially common sparse pattern distributed among Mt transmit antennas and Mr. receiver antennas pertaining to the ith transmit antenna, the channel frequency response is given by H=VA+N ....(1) Where V is a Vander monde matrix of size P*P, A is path gain of size P*Mr. and N is additive white Gaussian noise. Comparing (1) with the traditional direction-of-arrival (DOA) problem it’s been observed that these two are mathematically equivalent. Explicitly, the classical DOA problem is to commonly estimate the DOAs of the M sources from a group of time-domain quantities, which are acquired from the P sensors outputs at Mt Mr discrete points in time.

On the contrary, the path delays of M multipaths in (1) are estimated from a group of frequency-domain quantities, which are obtained from P pilots of Mt Mr discrete pairs of antenna. The Total Least Square-Estimating Signal Parameters via Rotational Invariance Techniques (TLS-ESPRIT) algorithm is used to estimate path delays with arbitrary values by applying it to (1). Additionally, the temporal correlation of wireless channels is also exploited to enhance the precision of the channel estimation. Initially, path delays of CIRs for various adjacent OFDM symbols are almost unaltered which is similarly, made reference to as a common sparse pattern of CIRs because of the temporal correlation of MIMO channels. Furthermore, path gains throughout adjacent OFDM symbols are also correlated pertaining to the temporal continuity of the CIR. Also various adjacent OFDM symbols are also correlated. Hence, noise is decreased which in turn improves the accuracy of channel estimation.

C. PilotOverhead: Based on the FRI theory it is stipulated that CIRs of Mt Mr transmit-receive antenna pairs are like the Mt Mr. semi period sparse subspaces, and the P pilots are like the Tp multichannel filters. Hence, with respect to the FRI theory, the least required number of pilots for a transmit antenna is Tp=2Q(i.e., Q is the sparsity level) in a noiseless environment. On the contrary, for the non-parametric channel estimation techniques the necessary number of pilots highly depend upon channel length L, whereas the super resolution sparse parametric scheme only requires 2Q pilots where Q<<l

1.4 Code Division Multiple Access (CDMA)

Overview of CDMA:

Code division multiple access (CDMA) is a modulation and multiple-access scheme based on spread-spectrum communication. In this scheme, multiple users share the same frequency band at the same time, by spreading the spectrum of their transmitted signals, so that each user's signal is pseudo-orthogonal to the signals of the other users.

CDMA Signals:

In a CDMA system, each signal consists of a different pseudorandom binary sequence (called the spreading code) that modulates a carrier, spreading the spectrum of the waveform. A large number of CDMA signals share the same frequency spectrum. If CDMA is viewed in either the frequency or time domain, the multiple access signals overlap with each other. However, the use of statistically orthogonal spreading codes separates the various signals in the code space.

CDMA Receivers:

A CDMA receiver separates the signals by means of a correlator that uses the particular binary sequence to dispreads the signal and collects the energy of the desired signal. Other users signals, whose spreading codes do not match this sequence, are not dispreads in bandwidth and, as a result, contribute only to the noise. These signals represent a self-interference generated by the system. The output of the correlator is sent to a narrow-bandwidth filter. The filter allows all the desired signal's energy to pass through but reduces the interfering signal's energy by the ratio of the bandwidth before the correlator to the bandwidth after the correlator. This reduction greatly improves the signal-to-interference ratio of the desired signal. This ratio is also known as the processing gain. The signal-to-noise ratio is determined by the ratio of the desired signal power to the sum of all of the other signal powers. It is enhanced by the processing gain or the ratio of spread bandwidth to baseband data rate.

CDMA Channel Assignments

A CDMA digital cellular waveform design uses a pseudorandom noise (PN) sequence to spread the spectrum. The sample rate of the spreading sequence (called the chip rate) is chosen so that the bandwidth of the filtered signal is several times the bandwidth of the original signal.

A typical system might use multiple PN sequences. In addition, it might use repeated spreading codes of known lengths to ensure orthogonality between signals intended for different users. The channel assignment is essentially determined by the set of codes that are used for that link. Thus, the signal transmitted at any time in a logical channel is determined by: The frequency of operation for the base station The current symbol The specific orthogonal spreading code assigned for the logical channel The PN spreading code.

CDMA Signal Processing:

In the demodulation of CDMA signals, the different paths may be independently received, which greatly reduces the severity of the multipath fading. However, multipath fading is not eliminated because occasionally there may be multiple paths that cannot be independently processed by the demodulator.

Different users in CDMA employ signals that have very small cross-correlation. Thus, correlators can extract individual signals from a mixture of signals even though they are transmitted simultaneously in the same frequency band. CDMA systems employ wideband signals with good cross-correlation properties, which means the output of a filter matched to one user's signal is small when it receives a different user's signal as input. In direct-sequence spread-spectrum systems, a high-rate antipodal pseudorandom spreading sequence modulates the transmitted signal so that the bandwidth of the resulting signal is roughly equal to the rate of the spreading sequence. The cross-correlation of the signals is then largely determined by the cross-correlation properties of the spreading signals. Although CDMA signals overlap in both time and frequency domains, they can be separated, based on their spreading waveforms.

Spreading rates can be chosen to exceed the coherence bandwidth so that the channel becomes frequency selective. For instance, different spectral components are affected unequally by the channel, and only parts of the signal are affected by fades. Expressing the same observation in time domain terms, multipath components are resolvable at a resolution equal to the chip period and can be combined coherently, for example, by means of a rake receiver. Coherent combination of multipath components requires an estimate of the channel impulse response. Such an estimate can be calculated from a training sequence or by means of a pilot signal.

FUNDAMENTALS OF THE IS-95A CDMA SYSTEM:

This section describes the important features of the IS-95A CDMA system specifications. The CDMA Reference Block set is designed to help you develop models to simulate different components of the IS-95A CDMA system.

Channel Schematics:

The following figure illustrates an IS-95A forward channel. The transmitter section includes channel coding, modulation and spreading, and filtering. The receiver section includes filtering, dispreading and demodulation, and channel decoding

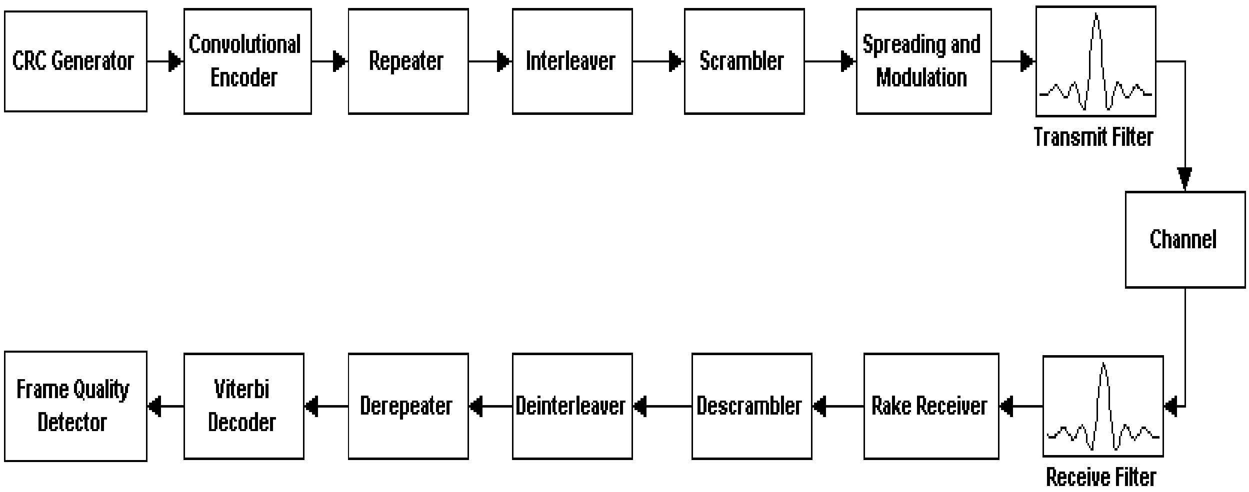


Fig.1.4. IS-95A Forward Channel Diagram

The following figure illustrates an IS-95A reverse channel. It includes many of the same operations that are in the forward channel, but the functionalities of the blocks correspond to the reverse channel specifications.

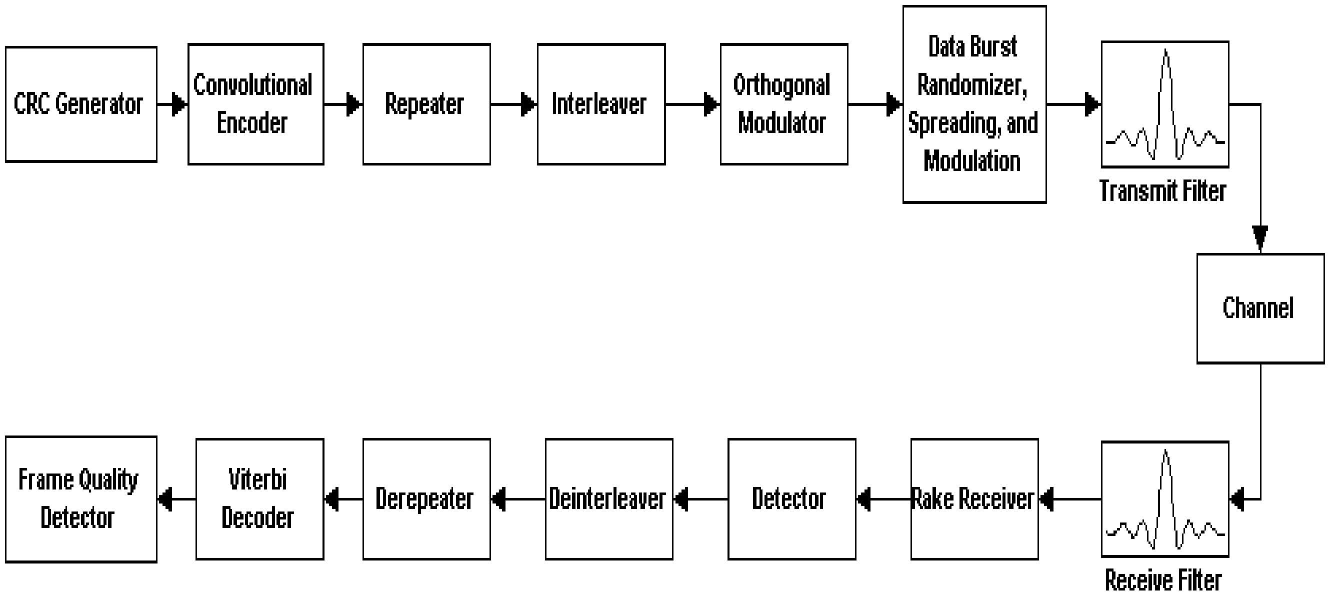


Fig.1.5. Reverse Channel diagram

Channel Assignment in IS-95A

The CDMA system requires spreading of the spectrum using a PN sequence. In IS-95A, the rate of this PN sequence (called the chip rate) is 1.2288 Mchips/s. This causes the resulting bandwidth of the spread signals to be about 1.25MHz, which is about one-tenth of the total bandwidth allocated to one cellular service carrier.

The IS-95A system uses two PN codes: The short PN code is a pair of periodic binary PN sequences with a period of 215. These sequences are used for spreading and dispreading signals into in-phase and quadrature components.

The same short PN code is used by multiple base stations using the same frequency band by using different timing offsets in the code cycle. The long PN code is a sequence with a period of 242-1, and this is used for spreading on the reverse link, as well as for data scrambling and power control burst randomization.

In addition to the PN codes, there is a set of length-64 mutually orthogonal codes called the Walsh code, which is used for ensuring orthogonality between the signals for different users receiving from the same base station. The Walsh code is also used for modulation for the reverse channel of IS-95A. Thus, the logical channel on the forward link is determined by the short PN code offset, the Walsh code assigned, and the assigned frequency of operation. On the reverse link, the logical channel is determined by the short PN code offset, the long code offset, and the assigned frequency of operation.

The IS-95A forward link uses several logical channels: The Pilot channel modulates a constant symbol and is used for channel estimation, which allows for coherent demodulation of the other channels that carry information bits. The Sync channel is used for providing synchronization and configuration information to the mobile stations. The Paging channels are used for control information and sending paging messages to the mobile station. The Traffic channel carries the speech or data.

Similarly, the reverse link has the Traffic and Access logical channels: The Access channel is meant for control information, and is used for originating requests, responding to paging and other messages, or providing other data to the base station. The Traffic channel carries the speech or data. The Traffic channel supports variable data rate operation. According to the J-STD-008 requirement, there are two sets of Traffic channel data rates. Rate Set I has a maximum data rate of 9.6 kbps, whereas Rate Set II has a maximum data rate of 14.4 kbps. Both rate sets support full, half, quarter, and eighth rates with respect to the maximum data rate.

Forward Channel Coding:

The IS-95A system uses the forward channels Sync, Paging, and Traffic to carry information from the base station to the mobile units. The channel coding operations in the forward CDMA operations use 20 ms frames for all channels, except for the Sync channel, which is coded using 26.666 ms frames. For error protection, a 1/2-rate convolutional code is used for all information channels. The receiver can use the Viterbi algorithm for optimal decoding of the encoded data. because the system uses variable data rates, the number of bits generated by the vocoder in one frame changes depending on the voice activity. To ensure that the symbol rate at the modulation stage is kept constant, the symbols are repeated for the lower data rate frames. For protection against bursts of errors, the frame data is interleaved prior to modulation. These error protection measures improve the overall bit error rate (BER) on the link.

Base Station Modulation and Spreading:

The IS-95A forward CDMA channel consists of the Pilot channel for coherent demodulation and the information channels. These channels are spread orthogonally by using a set of codes of length 64 called the Walsh codes. The combined signal is spread in quadrature by a pair of PN sequences with a fixed spreading rate.

The base station transmitter performs the encoding, the repetition, the interleaving, and the scrambling prior to spreading. The generated modulation symbols also need to carry power control commands to correct the mobile station transmit power. For this purpose, some of the symbols are replaced with command bits known as power control bits. The locations of these bits are randomized using a scheme based on a decimated long PN code. The details of these are in the IS-95A standard. The spread signal is filtered at baseband before transmission.

Base Station Transmitter Interface:

The base station transmitter interface combines the various channels in the CDMA forward channel. The forward channel contains the Pilot channel, the Sync channel, the Paging channels, and multiple Traffic channels. Each Traffic channel has a unique Walsh code assigned to it. These Walsh codes are orthogonal to each other, and the different symbol streams are spread by their respective Walsh code (in bipolar form) and added together in a manner such that the weight given to each corresponds to the intended power in that channel. The Traffic channel has a variable data rate. In the case of lower data rate frames, the symbols are repeated, and the Traffic channel power is reduced by the same repetition factor. This reduction ensures that the power transmitted for each information data bit (before coding and repetition) is the same.

Coherent Rake Receiver:

The coherent rake receiver demodulates the desired channel in the input signal by dispreading it with the corresponding Walsh code and the short PN code. The mobile user receives the signal transmitted from the serving CDMA base station through several paths with different propagation delays. The received signal, in addition to being corrupted by noise, is also distorted by the channel fading. For a basic receiver design, the delay-spread results in a loss of performance.

The rake receiver, on the other hand, uses the direct-sequence spreading of the coded signal to separate the components of the received signal corresponding to different propagation-delay paths. You can almost say that the rake receiver derives diversity gain from a potentially poor channel. After rake receiver dispreading, a demodulation routine detects the transmitted data from each delayed-path component and combines the results.

Reverse Channel Coding:

On the reverse CDMA channel in IS-95A, all the transmission uses 20 ms frames. The channels used in the reverse link are: The Traffic Channel and Access Channel. The Traffic channel carries data and speech. The data rate on the Traffic channel is variable; hence each frame may have a different rate. The Access channel carries control messages and requests to the base station.

The mobile station convolutionally encodes the data transmitted on the reverse channel. The station uses a 1/3-rate code when the Traffic channel uses Rate Set I, and a 1/2-rate code when the Traffic channel uses Rate Set II. The Access channel uses the 1/3-rate code. As in the case of the forward channel, the Viterbi algorithm provides optimal decoding at the receiver. Also, prior to modulation, symbols are repeated in the case of lower data rates, and the frame data is interleaved for protection against burst errors.

Mobile Station Walsh Modulation and Spreading:

The reverse CDMA channel is composed of the Access and Traffic channels. These channels share the same assigned CDMA frequency using direct-sequence CDMA techniques. Each of these channels is assigned a distinct user long PN code sequence, which is used for spreading the spectrum of the signal.

All transmission on the reverse CDMA channel in IS-95A uses 20-ms frames. The data frames are convolutionally encoded and interleaved for error protection. The modulation used is a 64-ary orthogonal modulation. Direct sequence spreading and filtering are used to obtain the modulation signal for transmission. Because the reverse traffic channel has a variable rate, randomized gating is used in the case of lower data rate frames to control the transmit power. Because the power control is done by gating, the power level of the transmitted symbols is kept constant, unlike the forward channel. The details of these transmitter tasks are in the IS-95A standard.

Non-coherent Rake Receiver:

There is no pilot available for the reverse link transmission, as there is for the forward link. A pilot is valuable for obtaining good carrier channel estimation, making it possible to perform coherent detection and combining of multipath components. The absence of carrier phase and amplitude estimation necessitates either non-coherent or differential coherent detection. Timing of all paths must also be acquired and tracked. This discussion assumes that timing is available, but phase and amplitude estimates are not.

The Walsh modulator collects log264 = 6 data bits and transmits one of 64 orthogonal Walsh functions. The demodulator tracks L independent paths. Assume that each path has a separate demodulator, but that their outputs are non-coherently combined. The optimum non-coherent demodulator of Walsh modulation is a bank of 64 orthogonal non-coherent correlators. Each of the 64 non-coherent correlators squares each of the two quadrature components and adds the results. For a single path, this demodulator makes its decision by selecting the largest of the 64 output magnitudes. In the case of L paths, the demodulator adds the individual non-coherent correlator outputs for each of the L independent paths before deciding.

In practice, it is possible to decide for each bit separately, rather than making decisions for all six bits at once. This may be done by finding the difference between the highest correlation values that correspond to the two possible values (0 and 1) of the bit of interest. Such an approach may be applied bit by bit and used to arrive at soft decisions (decisions whose difference value serves as a metric for decision reliability). Soft decisions can be used in the Viterbi decoder for the convolution code to improve performance.

1.5 Orthogonal Frequency Division Multiplexing (OFDM)

Orthogonal frequency-division multiplexing (OFDM), essentially identical to coded OFDM (COFDM) and discrete multi-tone modulation (DMT), is a frequency-division multiplexing (FDM) scheme used as a digital multi-carrier modulation method. Many closely spaced orthogonal sub-carriers are used to carry data. The data is divided into several parallel data streams or channels, one for each sub-carrier. Each sub-carrier is modulated with a conventional modulation scheme (such as quadrature amplitude modulation or phase-shift keying) at a low symbol rate, maintaining total data rates like conventional single-carrier modulation schemes in the same bandwidth.

OFDM has developed into a popular scheme for wideband digital communication, whether wireless or over copper wires, used in applications such as digital television and audio broadcasting, wireless networking and broadband internet access.

The primary advantage of OFDM over single-carrier schemes is its ability to cope with severe channel conditions (for example, attenuation of high frequencies in a long copper wire, narrowband interference and frequency-selective fading due to multipath) without complex equalization filters.

Channel equalization is simplified because OFDM may be viewed as using many slowly-modulated narrowband signals rather than one rapidly-modulated wideband signal. The low symbol rate makes the use of a guard interval between symbols affordable, making it possible to eliminate inter symbol interference (ISI) and utilize echoes and time-spreading (that shows up as ghosting on analogue TV) to achieve a diversity gain, i.e. a signal-to-noise ratio improvement. This mechanism also facilitates the design of single frequency networks (SFNs), where several adjacent transmitters send the same signal simultaneously at the same frequency, as the signals from multiple distant transmitters may be combined constructively, rather than interfering as would typically occur in a traditional single-carrier system.

Example of applications:

The following list is a summary of existing OFDM based standards and products. For further details, see the Usage section at the end of the article.

Wire

- ADSL and VDSL broadband access via POTS copper wiring.
- DVB-C2, an enhanced version of the DVB-Digital cable TV standard.
- Power line communication (PLC).
- ITU-TG.hn, a standard which provides high-speed local area networking over existing home wiring (power lines, phone lines and coaxial cables).
- Trail Blazer telephone line modems.
- Multimedia over Coax Alliance (MOCA) home networking.

Wireless

- The wireless LAN (WLAN) radio interfaces IEEE 802.11a, g, n and HIPERLAN/2.
- The digital radio systems DAB/EUREKA 147, DAB+, Digital Radio Mondale, HD Radio, T-DMB and ISDB-TSB.
- The terrestrial digital TV systems DVB-T and ISDB-T.
- The terrestrial mobile TV systems DVB-H, T-DMB, ISDB-T and Media FLO forward link.
- The wireless personal area network (PAN) ultra-wideband (UWB) IEEE 802.15.3a implementation suggested by Wi Media Alliance.

The OFDM based multiple access technology OFDMA is also used in several 4G and pre-4G cellular networks and mobile broadband standards:

- The mobility mode of the wireless MAN/broadband wireless access (BWA) standard IEEE 802.16e (or Mobile-Wi MAX).
- The mobile broadband wireless access (MBWA) standard IEEE 802.20.
- The downlink of the 3GPPLong Term Evolution (LTE) fourth generation mobile broadband standard. The radio interface was formerly named High Speed OFDM Packet Access (HSOPA), now named Evolved UMTS Terrestrial Radio Access (E-UTRA).

Key features:

The advantages and disadvantages listed below are further discussed in the Characteristics and principles of operation section below.

Summary of advantages:

- Can easily adapt to severe channel conditions without complex time-domain equalization.
- Robust against narrow-band co-channel interference.
- Robust against inter symbol interference (ISI) and fading caused by multipath propagation.

- High spectral efficiency as compared to conventional modulation schemes, spread spectrum, etc.
- Efficient implementation using Fast Fourier Transform (FFT).
- Low sensitivity to time synchronization errors.
- Tuned sub-channel receiver filters are not required (unlike conventional FDM).
- Facilitates single frequency networks (SFNs); i.e., transmitter macro diversity.

Summary of disadvantages:

- Sensitive to Doppler shift.
- Sensitive to frequency synchronization problems.
- High peak-to-average-power ratio (PAPR), requiring linear transmitter circuitry, which suffers from poor power efficiency.
- Loss of efficiency caused by cyclic prefix/guard interval.

Characteristics and principles of operation

ORTHOGONALITY

In OFDM, the sub-carrier frequencies are chosen so that the sub-carriers are orthogonal to each other, meaning that crosstalk between the sub-channels is eliminated and inter-carrier guard bands are not required. This greatly simplifies the design of both the transmitter and the receiver; unlike conventional FDM, a separate filter for each sub-channel is not required.

The orthogonality requires that the sub-carrier spacing is $\Delta f = \frac{k}{T_U}$ Hertz, where T_U seconds is the useful symbol duration (the receiver side window size), and k is a positive integer, typically equal to 1. Therefore, with N sub-carriers, the total passband bandwidth will be $B \approx N \cdot \Delta f$ (Hz).

The orthogonality also allows high spectral efficiency, with a total symbol rate near the Nyquist rate for the equivalent baseband signal (i.e., near half the Nyquist rate for the double-side band physical passband signal).

Almost the whole available frequency band can be utilized. OFDM generally has a nearly 'white' spectrum, giving it benign electromagnetic interference properties with respect to other co-channel users. A simple example: A useful symbol duration T_U = 1 ms would require a sub-carrier spacing of $\Delta f = \frac{1}{1\,\mathrm{ms}} = 1\,\mathrm{kHz}$ (or an integer multiple of that) for orthogonality. N = 1,000 sub-carriers would result in a total pass band bandwidth of $N\Delta f$ = 1 MHz For this symbol time, the required bandwidth in theory according to Nyquist is $N/2T_U$ = 0.5 MHz (i.e., half of the achieved bandwidth required by our scheme). If a guard interval is applied (see below), Nyquist bandwidth requirement would be even lower. The FFT would result in N = 1,000 samples per symbol. If no guard interval was applied, this would result in a base band complex valued signal with a sample rate of 1 MHz, which would require a baseband bandwidth of 0.5 MHz according to Nyquist. However, the passband RF signal is produced by multiplying the baseband signal with a carrier waveform (i.e., double-sideband quadrature amplitude-modulation) resulting in a passband bandwidth of 1 MHz A single-side band (SSB) or vestigial sideband (VSB) modulation scheme would achieve almost half that bandwidth for the same symbol rate (i.e., twice as high spectral efficiency for the same symbol alphabet length). It is however more sensitive to multipath interference.

OFDM requires very accurate frequency synchronization between the receiver and the transmitter; with frequency deviation the sub-carriers will no longer be orthogonal, causing inter-carrier interference (ICI) (i.e., cross-talk between the sub-carriers). Frequency offsets are typically caused by mismatched transmitter and receiver oscillators, or by Doppler shift due to movement. While Doppler shift alone may be compensated for by the receiver, the situation is worsened when combined with multipath, as reflections will appear at various frequency offsets, which is much harder to correct. This effect typically worsens as speed increases,[1] and is an important factor limiting the use of OFDM in high-speed vehicles. Several techniques for ICI suppression are suggested, but they may increase the receiver complexity.

Implementation using the FFT algorithm:

The orthogonality allows for efficient modulator and demodulator implementation using the FFT algorithm on the receiver side, and inverse FFT on the sender side. Although the principles and some of the benefits have been known since the 1960s, OFDM is popular for wideband communications today by way of low-cost digital signal processing components that can efficiently calculate the FFT.

Guard intervals for elimination of inter symbol interference:

One key principle of OFDM is that since low symbol rate modulation schemes (i.e., where the symbols are relatively long compared to the channel time characteristics) suffer less from inter symbol interference caused by multipath propagation; it is advantageous to transmit a number of low-rate streams in parallel instead of a single high-rate stream. Since the duration of each symbol is long, it is feasible to insert a guard interval between the OFDM symbols, thus eliminating the inter symbol interference.

The guard interval also eliminates the need for a pulse-shaping filter, and it reduces the sensitivity to time synchronization problems.

A simple example: If one sends a million symbols per second using conventional single-carrier modulation over a wireless channel, then the duration of each symbol would be one microsecond or less. This imposes severe constraints on synchronization and necessitates the removal of multipath interference. If the same million symbols per second are spread among one thousand sub-channels, the duration of each symbol can be longer by a factor of a thousand (i.e., one millisecond) for orthogonality with approximately the same bandwidth. Assume that a guard interval of 1/8 of the symbol length is inserted between each symbol. Intersymbol interference can be avoided if the multipath time-spreading (the time between the reception of the first and the last echo) is shorter than the guard interval (i.e., 125 microseconds). This corresponds to a maximum difference of 37.5 kilometers between the lengths of the paths.

The cyclic prefix, which is transmitted during the guard interval, consists of the end of the OFDM symbol copied into the guard interval, and the guard interval is transmitted followed by the OFDM symbol. The reason that the guard interval consists of a copy of the end of the OFDM symbol is so that the receiver will integrate over an integer number of sinusoid cycles for each of the multipaths when it performs OFDM demodulation with the FFT.

Simplified equalization:

The effects of frequency-selective channel conditions, for example fading caused by multipath propagation, can be considered as constant (flat) over an OFDM sub-channel if the sub-channel is sufficiently narrow-banded (i.e., if the number of sub-channels is sufficiently large). This makes frequency domain equalization possible at the receiver, which is far simpler than the time-domain equalization used in conventional single-carrier modulation. In OFDM, the equalizer only has to multiply each detected sub-carrier (each Fourier coefficient) in each OFDM symbol by a constant complex number, or a rarely changed value.

Our example: The OFDM equalization in the above numerical example would require one complex valued multiplication per subcarrier and symbol (i.e., $N = 1000$complex multiplications per OFDM symbol, i.e., one million multiplications per second, at the receiver).

The FFT algorithm requires $N \log_2 N = 10{,}000$complex-valued multiplications per OFDM symbol (i.e., 10 million multiplications per second), at both the receiver and transmitter side. This should be compared with the corresponding one million symbols/second single-carrier modulation case mentioned in theexample, where the equalization of 125 microseconds time-spreading using a FIR filter would require, in a naive implementation, 125 multiplications per symbol (i.e., 125 million multiplications per second). FFT techniques can be used to reduce the number of multiplications for an FIR filter-based time-domain equalizer to a number comparable with OFDM, at the cost of delay between reception and decoding which also becomes comparable with OFDM.

If differential modulation such as DPSK or DQPSK is applied to each sub-carrier, equalization can be completely omitted, since these non-coherent schemes are insensitive to slowly changing amplitude and phase distortion.

In a sense, improvements in FIR equalization using FFTs or partial FFTs leads mathematically closer to OFDM, but the OFDM technique is easier to understand and implement, and the sub-channels can be independently adapted in other ways than varying equalization coefficients, such as switching between different QAM constellation patterns and error-correction schemes to match individual sub-channel noise and interference characteristics.

Some of the sub-carriers in some of the OFDM symbols may carry pilot signals for measurement of the channel conditions (i.e., the equalizer gain and phase shift for each sub-carrier). Pilot signals and training symbols (preambles) may also be used for time synchronization (to avoid inter symbol interference, ISI) and frequency synchronization (to avoid inter-carrier interference, ICI, caused by Doppler shift).

OFDM was initially used for wired and stationary wireless communications. However, with an increasing number of applications operating in highly mobile environments, the effect of dispersive fading caused by a combination of multi-path propagation and Doppler shift is more significant. Over the last decade, research has been done on how to equalize OFDM transmission over doubly selective channels.

Channel coding and interleaving:

OFDM is invariably used in conjunction with channel coding (forward error correction), and almost always uses frequency and/or time interleaving.

Frequency (subcarrier) interleaving increases resistance to frequency-selective channel conditions such as fading. For example, when a part of the channel bandwidth fades, frequency interleaving ensures that the bit errors that would result from those subcarriers in the faded part of the bandwidth are spread out in the bit-stream rather than being concentrated. Similarly, time interleaving ensures that bits that are originally close together in the bit-stream are transmitted far apart in time, thus mitigating against severe fading as would happen when travelling at high speed.

However, time interleaving is of little benefit in slowly fading channels, such as for stationary reception, and frequency interleaving offers little to no benefit for narrowband channels that suffer from flat fading (where the whole channel bandwidth fades at the same time).

The reason why interleaving is used on OFDM is to attempt to spread the errors out in the bit-stream that is presented to the error correction decoder, because when such decoders are presented with a high concentration of errors the decoder is unable to correct all the bit errors, and a burst of uncorrected errors occurs. A similar design of audio data encoding makes compact disc (CD) playback robust.

A classical type of error correction coding used with OFDM-based systems is convolutional coding, often concatenated with Reed-Solomon coding. Usually, additional interleaving (on top of the time and frequency interleaving mentioned above) in between the two layers of coding is implemented.

The choice for Reed-Solomon coding as the outer error correction code is based on the observation that the Viterbi decoder used for inner convolutional decoding produces short errors bursts when there is a high concentration of errors, and Reed-Solomon codes are inherently well-suited to correcting bursts of errors.

Newer systems, however, usually now adopt near-optimal types of error correction codes that use the turbo decoding principle, where the decoder iterates towards the desired solution. Examples of such error correction coding types include turbo codes and LDPC codes, which perform close to the Shannon limit for the Additive White Gaussian Noise (AWGN) channel. Some systems that have implemented these codes have concatenated them with either Reed-Solomon (for example on the Media FLO system) or BCH codes (on the DVB-S2 system) to improve upon an error floor inherent to these codes at high signal-to-noise ratios.

Adaptive transmission:

The resilience to severe channel conditions can be further enhanced if information about the channel is sent over a return-channel. Based on this feedback information, adaptive modulation, channel coding and power allocation may be applied across all sub-carriers, or individually to each sub-carrier. In the latter case, if a particular range of frequencies suffers from interference or attenuation, the carriers within that range can be disabled or made to run slower by applying more robust modulation or error coding to those sub-carriers.

The term discrete multitone modulation (DMT) denotes OFDM based communication systems that adapt the transmission to the channel conditions individually for each sub-carrier, by means of so-called bit-loading. Examples are ADSL and VDSL.

The upstream and downstream speeds can be varied by allocating either more or fewer carriers for each purpose. Some forms of rate-adaptive DSL use this feature in real time, so that the bitrate is adapted to the co-channel interference and bandwidth is allocated to whichever subscriber needs it most.

OFDM extended with multiple access:

OFDM in its primary form is considered as a digital modulation technique, and not a multi-user channel access method, since it is utilized for transferring one bit stream over one communication channel using one sequence of OFDM symbols. However, OFDM can be combined with multiple accesses using time, frequency, or coding separation of the users.

In Orthogonal Frequency Division Multiple Access (OFDMA), frequency-division multiple access is achieved by assigning different OFDM sub-channels to different users. OFDMA supports differentiated quality of service by assigning different number of sub-carriers to different users in a similar fashion as in CDMA, and thus complex packet scheduling or Media Access Control schemes can be avoided. OFDMA is used in:

- the mobility mode of the IEEE 802.16Wireless MAN standard, commonly referred to as WiMAX,
- the IEEE 802.20 mobile Wireless MAN standard, commonly referred to as MBWA,

- The 3GPP Long Term Evolution (LTE) fourth generation mobile broadband standard downlink. The radio interface was formerly named High Speed OFDM Packet Access (HSOPA), now named Evolved UMTS Terrestrial Radio Access (E-UTRA).
- The now defunct Qualcomm/3GPP2Ultra Mobile Broadband (UMB) project, intended as a successor of CDMA2000, but replaced by LTE.

OFDMA is also a candidate access method for the IEEE 802.22Wireless Regional Area Networks (WRAN). The project aims at designing the first cognitive radio based standard operating in the VHF-low UHF spectrum (TV spectrum).

In Multi-carrier code division multiple access (MC-CDMA), also known as OFDM-CDMA, OFDM is combined with CDMA spread spectrum communication for coding separation of the users. Co-channel interference can be mitigated, meaning that manual fixed channel allocation (FCA) frequency planning is simplified, or complex dynamic channel allocation (DCA) schemes are avoided.

Space diversity:

In OFDM based wide area broadcasting, receivers can benefit from receiving signals from several spatially dispersed transmitters simultaneously, since transmitters will only destructively interfere with each other on a limited number of sub-carriers, whereas in general they will actually reinforce coverage over a wide area. This is very beneficial in many countries, as it permits the operation of national SFNs, where many transmitters send the same signal simultaneously over the same channel frequency. SFNs utilize the available spectrum more effectively than conventional multi-frequency broadcast networks (MFN), where program content is replicated on different carrier frequencies. SFNs also result in a diversity gain in receivers situated midway between the transmitters.

The coverage area is increased, and the outage probability decreased in comparison to an MFN, due to increased received signal strength averaged over all sub-carriers.

Although the guard interval only contains redundant data, which means that it reduces the capacity, some OFDM-based systems, such as some of the broadcasting systems, deliberately use a long guard interval in order to allow the transmitters to be spaced farther apart in an SFN, and longer guard intervals allow larger SFN cell-sizes.

A rule of thumb for the maximum distance between transmitters in an SFN is equal to the distance a signal travels during the guard interval — for instance, a guard interval of 200 microseconds would allow transmitters to be spaced 60 km apart.

A single frequency network is a form of transmitter macro diversity. The concept can be further utilized in dynamic single-frequency networks (DSFN), where the SFN grouping is changed from timeslot to timeslot.

OFDM may be combined with other forms of space diversity, for example antenna arrays and MIMO channels. This is done in the IEEE802.11nWireless LAN standard.

Linear transmitter power amplifier:

An OFDM signal exhibits a high peak-to-average power ratio (PAPR) because the independent phases of the sub-carriers mean that they will often combine constructively. Handling this high PAPR requires:

- a high-resolution digital-to-analogue converter (DAC) in the transmitter
- a high-resolution analogue-to-digital converter (ADC) in the receiver
- A linear signal chain.

Any non-linearity in the signal chain will cause inter modulation distortion that

- raises the noise floor
- may cause inter-carrier interference
- Generates out-of-band spurious radiation.

The linearity requirement is demanding, especially for transmitter RF output circuitry where amplifiers are often designed to be non-linear to minimize power consumption.

In practical OFDM systems a small amount of peak clipping is allowed to limit the PAPR in a judicious trade-off against the above consequences. However, the transmitter output filter which is required to reduce out-of-band spurs to legal levels has the effect of restoring peak levels that were clipped, so clipping is not an effective way to reduce PAPR.

Although the spectral efficiency of OFDM is attractive for terrestrial and space communications, the high PAPR requirements have so far limited OFDM applications to terrestrial system

Idealized System:

This section describes a simple idealized OFDM system model suitable for a time-invariant AWGN channel.

TRANSMITTER:

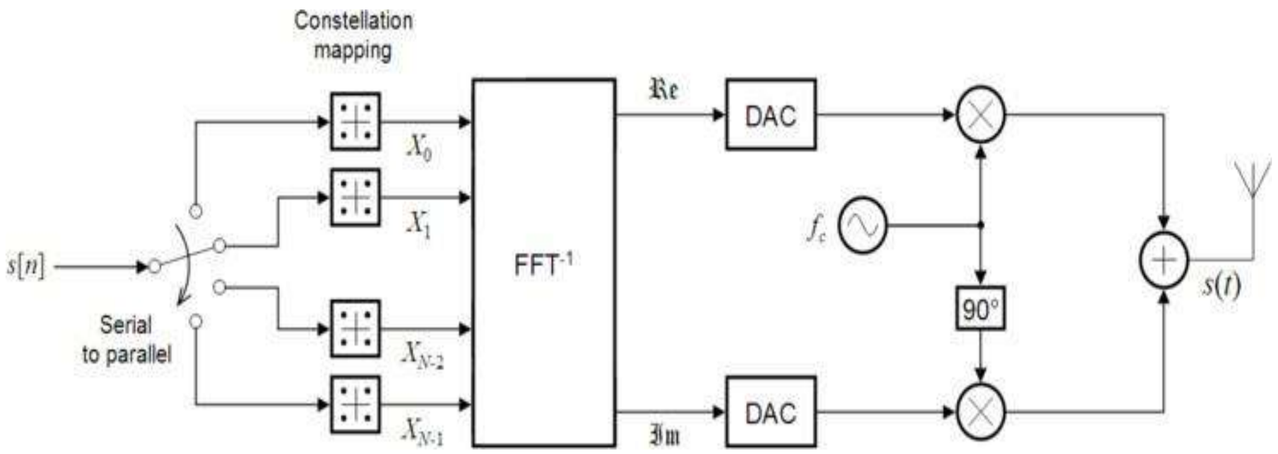


Fig.1.6

An OFDM carrier signal is the sum of a number of orthogonal sub-carriers, with baseband data on each sub-carrier being independently modulated commonly using some type of quadrature amplitude modulation (QAM) or phase-shift keying (PSK). This composite baseband signal is typically used to modulate a main RF carrier. $s[n]$ is a serial stream of binary digits. By inverse multiplexing, these are first de-multiplexed into N parallel streams, and each one mapped to a (possibly complex) symbol stream using some modulation constellation (QAM, PSK, etc.). Note that the constellations may be different, so some streams may carry a higher bit-rate than others.

An inverse FFT is computed on each set of symbols, giving a set of complex time-domain samples. These samples are then quadrature-mixed to passband in the standard way. The real and imaginary components are first converted to the analogue domain using digital-to-analogue converters (DACs); the analogue signals are then used to modulate cosine and sine waves at the carrier frequency, f_c respectively. These signals are then summed to give the transmission signal, $s(t)$.

RECEIVER:

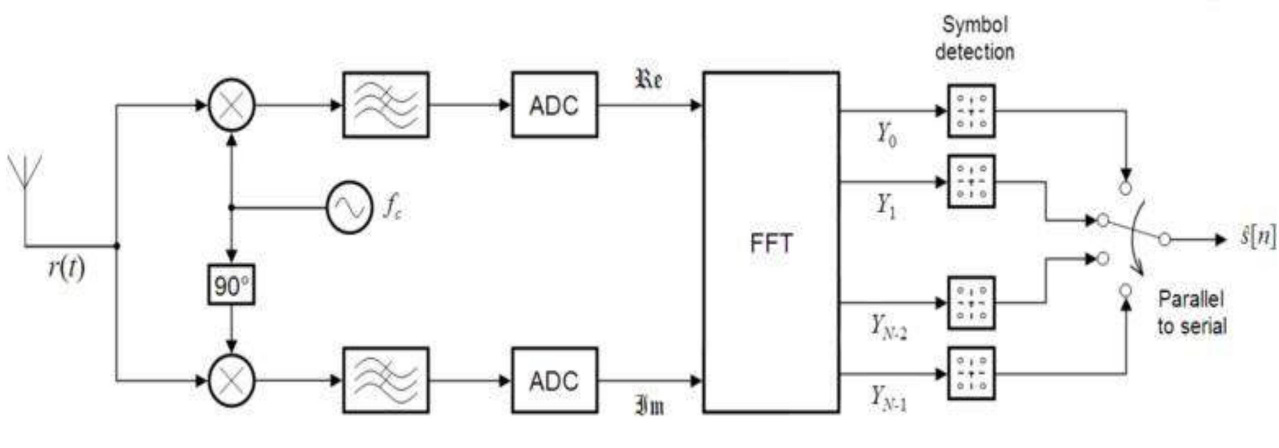


Fig.1.7

The receiver picks up the signal $r(t)$, which is then quadrature-mixed down to baseband using cosine and sine waves at the carrier frequency. This also creates signals centered on $2f_c$, so low-pass filters are used to reject these. The baseband signals are then sampled and digitized using analogue-to-digital converters (ADCs), and a forward FFT is used to convert back to the frequency domain.

This returns N parallel streams, each of which is converted to a binary stream using an appropriate symbol detector. These streams are then re-combined into a serial stream, $\hat{s}[n]$, which is an estimate of the original binary stream at the transmitter.

Mathematical description:

If N sub-carriers are used, and each sub-carrier is modulated using M alternative symbols, the OFDM symbol alphabet consists of M^N combined symbols.

The low-pass equivalent OFDM signal is expressed as:

$$\nu(t) = \sum_{k=0}^{N-1} X_k e^{j2\pi kt/T}, \quad 0 \le t < T,$$

Where $\{X_k\}$ are the data symbols, N is the number of sub-carriers, and T is the OFDM symbol time. The sub-carrier spacing of $\frac{1}{T}$ makes them orthogonal over each symbol period; this property is expressed as:

$$\frac{1}{T}\int_0^T \left(e^{j2\pi k_1 t/T}\right)^* \left(e^{j2\pi k_2 t/T}\right) dt$$
$$= \frac{1}{T}\int_0^T e^{j2\pi(k_2-k_1)t/T} dt = \delta_{k_1 k_2}$$

where $(\cdot)^*$ denotes the complex conjugate operator and δ is the Kronecker delta.

To avoid inter symbol interference in multipath fading channels, a guard interval of length T_g is inserted prior to the OFDM block. During this interval, a cyclic prefix is transmitted such that the signal in the interval $-T_g \le t < 0$ equals the signal in the interval $(T-T_g) \le t < T$. The OFDM signal with cyclic prefix is thus:

$$\nu(t) = \sum_{k=0}^{N-1} X_k e^{j2\pi kt/T}, \quad -T_g \le t < T$$

The low-pass signal above can be either real or complex-valued. Real-valued low-pass equivalent signals are typically transmitted at baseband wireline applications such as DSL use this approach. For wireless applications, the low-pass signal is typically complex-valued; in which case, the transmitted signal is up-converted to a carrier frequency f_c. In general, the transmitted signal can be represented as:

$$s(t) = \Re\left\{\nu(t)e^{j2\pi f_c t}\right\}$$
$$= \sum_{k=0}^{N-1} |X_k| \cos\left(2\pi[f_c + k/T]t + \arg[X_k]\right)$$

Chapter 2

LITERATURE SURVEY

In the study of signal processing, we define the Nyquist rate as the rate which is twice the bandwidth of a band limited signal or a channel. It can be conceptually studied relating the Nyquist rate in accordance with signaling and sampling as two different approaches. From the sampling point of view, it is defined that in order to have no aliasing effect, the minimum sampling rate should be equal to or greater than double the highest frequency component contained within the signal. This means that sampling rate must be higher than the Nyquist rate. Sampling process involves the discretization of the continuous time signals. These samples are to be taken in a manner that it represents the original signal significantly. The number of samples taken defines the signal rate for specific signal bandwidth.

Nyquist signaling or Nyquist pulse shaping criteria highlights the necessary and sufficient condition so as to achieve zero ISI scenario. In the Nyquist paper [22], it is described about the number of pulses that could be transmitted and received over a band limited channel and also the rate of transmission. In the paper, the Nyquist signaling rate is defined as the maximum possible pulses that could be transmitted through a given bandwidth limited channel. This gives an idea about the Nyquist signaling, which provides platform for the design of the modern communication systems.

Nyquist rate clarifies the concept defining the rate as lower bound for the sample rate and upper bound for the symbol rate in a bandwidth constrained baseband channel without ISI. The upper limit for the number of bits per signal change is defined by the channel characteristics which is also known as the Shannon limit [23]. In short, the sampling frequency greater than twice the bandwidth is the condition where no aliasing effect occurs indicating the fulfillment of the Nyquist criteria. This means that for the continuous function $x(t)$ sampled at fixed sampling rate, the function that is band limited to $1/2fs$ only satisfies the Nyquist condition, i.e., its Fourier transform

$X(f) = 0$ for all $|f| \geq 1/2fs$.

For $B \leq 1/2fs$, $2B$ and $1/2fs$ are the expressions defining Nyquist rate and frequency respectively, where the symbol $T = 1/fs$ represents the sample period, or the sampling interval and B is the bandwidth.

Necessary and sufficient conditions for Nyquist criteria:

In a wireless communication, when the linearly modulated symbols are transmitted, the channel impulse response or the frequency response spreads the transmitted signal 18 in time domain. Such consecutive symbol transmission can create aliasing effect as previously transmitted signal may overlap over the other later signal. Nyquist criterion indicates the condition which when fulfilled, provides the zero ISI during such transmission.

As seen from the derivation in section 2.3, we can define the Nyquist ISI criteria as, for the discrete signals, x[n] = x(nT s) and the channel impulse response h(t) = h(nT s), the necessary and sufficient Nyquist condition for zero ISI or pulse shaping criteria stipulate that for a given pulse h[n], it has to satisfy [2]

$$h[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

and its Fourier transform satisfy 1

$$\frac{1}{T_s} \sum_{k=-\infty}^{\infty} H(f + k/T_s) = 1$$

which means that the frequency shifted versions of H(f), the Fourier transform of h(t), is summed up to 1.

The relation gives information that the Nyquist criterion is the condition of a wireless channel threshold for designing transceiver structures to attain an ISI free transmission. Nyquist first criterion [24] explains that the contributions at the sampling instant (T, 2T, 3T...) of the received waveform must be zero. Equations (3.2) and (3.3) indicate the fact that the sum of delayed pulse spectra by k/T should sum up to T or a constant. Raised cosine pulses are such commonly used practical pulses that meet the Nyquist ISI criterion. This is discussed later in section

2.1 Pulse shaping filters:

Considering a situation for increasing data rate or transmission rate for a system with given (fixed) bandwidth without accuracy degradation, we see the bandwidth constraint to be constant. One possible way is to modify the pulses effectively such that it requires less bandwidth than the contemporary means. The pulse shape is desired to be smooth. The distorted shape may deteriorate the condition requiring larger bandwidth since such shapes contains the energy spectrum including lower frequency harmonic components. It is required that, at the perfect scenario, transmitted pulse do not interfere each other at the receiver with the successive symbols being received. The error occurred during this procedure is termed as ISI. So it is somewhat like a tradeoff between the gain and complexity. In order to have smooth shape for spectral efficiency, pulse shaping filter is selected such that it can transform the given waveform into the desired form.

The transmitted signals are the result of the data pulse passed through a pulse shaping filter which is responsible for defining the transmitter spectrum. They have a role of allocating the prescribed signal in a defined bandwidth minimizing the spectral leakage and also to maintain the high transmission rate and low error rate for a signal. In a system with band limited channels, during the increased modulation rate of a transmitted signal, it is highly probable to have distortion or the ISI due to bandwidth limitation. So, it requires fixing the transmitted waveform into the shape such that signal remains 19 in the prescribed bandwidth. Ideal filters undergo a proper sampling procedure to ensure the minimum ISI and high stop band attenuation to reduce the inter channel interference (ICI) [25].

Usually, a rectangular ideal pulse is desired over an interval [0, T] but the pulses having different functional forms for, e.g., Sinc pulses have the spectrum with side lobes decreasing with frequency. This causes an unintentional overlapping of the symbols from the adjacent channels or the signals. Ideally, we expect the pulse with peak at $t = 0$ and zero at other sampling instants $t = T, 2T, 3T$. In time domain it must satisfy that its impulse response has periodic zero and zeros must occur at other sampling times, i.e., $h[nT] = 0$, where T is the data sample period. Pulses bearing such characteristics are referred as Nyquist pulses and the corresponding filters bearing such impulse responses are called Nyquist filters [26]. The pulse shaping filters are designed to satisfy the Nyquist ISI criterion.

When the band pass filtering is done prior to matched filtering, these filters reduce the cost of hardware designs as well. Pulse shaping filters implemented in frequency domain make it easy to recover the information embedded in the carrier signal by exploiting the frequency diversity at the receiver end using the signal decomposition or recombination [27].

For optimal filter design, the delta function $\delta(t)$ with pulse information is sent from the transmitter end. The full signal recovery ideal output then should be $\delta(1 + \sigma)$, where σ is the time delay of the causal system. Consequently, at the receiver end, there should be ideal output $\delta(t)$ [28].

Usually, these pulse shaping filters at the transmitter is used in reference to the corresponding matched filter at the receiver unit for optimal performance. Many pulse shaping filters like Sinc shaped filter, rectangular pulse filter, Raised-cosine filter, Gaussian filter are in practice.

Here, in the thesis work, raised cosine filters are used for its best suitability satisfying Nyquist criteria and the spectral efficiency.

2.2 Roll-off factor:

Roll-off factor gives the measure of the excess bandwidth occupied by any digital filter as compared to that of theoretical minimum Nyquist bandwidth. Bandwidth utilization is very important in wireless communication. In real time communication, the signals usually tend to occupy the bandwidth more than specified by Nyquist bandwidth. This excess value is referred as the roll-off factor. For, e.g., any digital filter with 100 MHz Nyquist bandwidth if occupies 130 MHz, is said to have roll-off factor 0.3 which means the filter excess bandwidth is 0.3 times the Nyquist bandwidth. It can also be defined as the steepness measure in the transition between the stop band and pass band of the filters. It directly relates to the bandwidth saving strategy.

Bandwidth saving between two spectral roll-off factors can be calculated as [29]

Bandwidth saving (%) = 100 − (($\beta 1/\beta 2$) × 100).

Here, $\beta 1$ is referred from the lower value between the two roll-off factors. Mathematically roll-off factor can be related as, for the Nyquist bandwidth of 1/2T, if the excess bandwidth as compared to this Nyquist bandwidth be Δf, then the roll-off fac- 20 tor (β) is given by

$\beta = \Delta f\ 1/2T = \Delta f\ Rs/2 = 2T\ \Delta f$,

where 1/T is the symbol rate. The value of β lies between 0 and 1. It is an important modulation parameter to determine the signal bandwidth and spectral efficiency.

Variation in the values of roll-off factor causes a change in the characteristics of modulated signal which directly affects the impact of the transmission signal. Higher roll-off means low ISI, since it reduces the change rate of the amplitude of output pulse at zero crossings. But it makes the signal bandwidth wider consequently consuming larger spectrum. On the other hand, lower the value of roll-off factors designed, higher is the spectral efficiency achieved. However, the rate of signal decay is slowest in the time domain. This makes the baseband signal more complex. In short, comparative lower roll-off values means a corresponding decrease in bandwidth usage and increase in symbol rate and overall throughput. This highlights that the proper selection of roll-off factor plays an important role on the performance of the system designed. The maximum symbol rate Rs attainable as studied in [30] can be related from rolloff factor for the given bandwidth as

$Rs = B(1 + \beta)$,

where B is the bandwidth of the system and β be the roll-off factor.

Raised cosine pulse:

Raised cosine (RC) filters are known for their ability to reduce the inter symbol interference of the signal. In communication systems, they are used as pulse shaping filters to shape the pulses. Basically, the name raised cosine stands for non-zero portion of the frequency spectrum with roll-off factor equal to 1 which defines the cosine function raised above the horizontal axis. It is the representation of pulse spectrum. For the channel with bandwidth B, if T > 1/2B then the pulse spectrum for this condition is called the raised cosine spectrum where 1/T is the sampling rate of the signal.

Raised cosine pulse shaping modulation is bandwidth efficient, i.e., more bits/sec/Hz and also power efficient, i.e., it requires comparatively low Eb/N0 for same bit error rate. These pulses stand as a balance between the rectangular pulse and the Sinc pulses. Rectangular pulses have sharp edges in the time domain indicating the large bandwidth in the frequency domain. So it requires infinite bandwidth. On the other hand, Sinc pulses have an infinite length of time domain sequence. Raised cosine pulses are therefore preferred as they exhibit the balanced nature between these two pulses. These pulses do not show such infinite long nature in both the time and frequency domains. When the passband frequency of these filters is half the data rate, it satisfies the Nyquist criterion such that T = NTs [31], where T is the data period and N is any integer. 21 Frequency response characteristic of a raised cosine function is given by [2] Xrc(f) =

$$X_{rc}(f) = \begin{cases} T & for \quad 0 \leq |f| \leq \frac{1-\beta}{2T} \\ \frac{T}{2}\{1 + \cos(\pi T/\beta(|f| - \frac{1-\beta}{2T}))\} & for \quad \frac{1-\beta}{2T} \leq |f| \leq \frac{1+\beta}{2T} \\ 0 & for \quad |f| > \frac{1+\beta}{2T}. \end{cases}$$

β is the roll-off factor and T is the reciprocal of symbol rate. Impulse response of the spectrum for raised cosine function is given by [2]

$$\begin{aligned} x(t) &= \frac{sin(\pi t/T)}{\pi t/T} \frac{cos(\pi\beta t/T)}{1 - 4\beta^2 t^2/T^2} \\ &= Sinc(\pi t/T) \frac{cos(\pi\beta t/T)}{1 - 4\beta^2 t^2/T^2}. \end{aligned}$$

the raised cosine pulse takes the form of a Since pulse shapes. The bandwidth of the pulse and the rate of decay are determined by the value of roll-off factor (β). Lower the value of β, sharper the spectrum. With roll-off value 1, it gives the pure raised cosine spectrum removing the flat portion. The tails of the signal x(t) is decayed as 1/t3 for roll-off factor greater than zero. This is the reason behind the formation of ISI which is formed when the timing mismatch occurs during sampling. Since the impulse response of RC filters has indefinite structure, it needs to be truncated for practical applications. In frequency domain, this truncation yields a non-zero side lobe. However, since pulse response delay is proportional to 1/t3 , the result might vary from that of theoretical approach [26].

Ideal response of a raised cosine filter has a unity gain in the region of low frequency and total attenuation at the high frequency. In the middle of these two frequencies, responses have a raised cosine function, size of which is determined by the roll-off factor. So, they are also called low pass filters. This raised cosine function meets the requirement of a pulse shaping filter, i.e., low pass response in frequency domain, zero ISI case and finite extent in time domain [26].

For an ideal channel,

Xrc(f) = Gt(f)Gr(f),

Where Gt(f) and Gr(f) are the frequency responses of the transmitter filter and receiver filter respectively [2]. These filters are used to analyze the overall desired frequency response.

For the matched condition of transmitter and receiver filter, Gt(f) = G∗ r (f) which yields,

$$G_t(f) = \sqrt{|X_{rc}(f)|}e^{-j2\pi f t_0}$$

Here, t0 is the nominal delay that is introduced to ensure the physical realizability of the filter [2].

2.3 Matched Filters:

Matched filters are the optimum linear filters used for enhancing the signal power or SNR. Basically, these filters are implemented using a known signal as a reference signal and comparing the unknown signal with the known referenced signal. This is done by taking the convolution of unknown signal with the time reversed form of referenced known signal. It can also be referred as convolution of unknown signal with conjugated time reversed version of known signal. The pulse compression where impulse response is matched to input signals, radar signaling are the examples of matched filtering. They are the linear time invariant filters that increases the SNR and decreases the error probability.

Since the main principles for matched filter were developed by Dwight North in 1943, reproduced in 1963 by Lamont Blake, it is also known as the North filter or conjugate filter [37]. Matched filter has its unique feature that it gives maximum possible instantaneous SNR at the output for the input of signal plus additive white noise.

The received signal consists of the transmitted pulses distorted by an additive channel noise. These noises are usually considered as additive Gaussian white noise. These received noisy signals are to be filtered out to receive the original transmitted pulse without noise components. Minimizing the noise effect out of the received signal is done in an optimum manner. Output signal power is desired to be considerably higher than the instantaneous noise power which is defined as maximizing the signal to noise power ratio. These filters give sharp peak response to the desired pulse at its input. This peak helps to determine the location t0 and the amplitude of the pulse from received signal. The general matched filter representation is shown in Figure.

For a transmitted signal x(t) and noise n(t), 26 Signal x(t) Noise n(t) Received signal s(t) Matched Filter H(w) Output y(t)

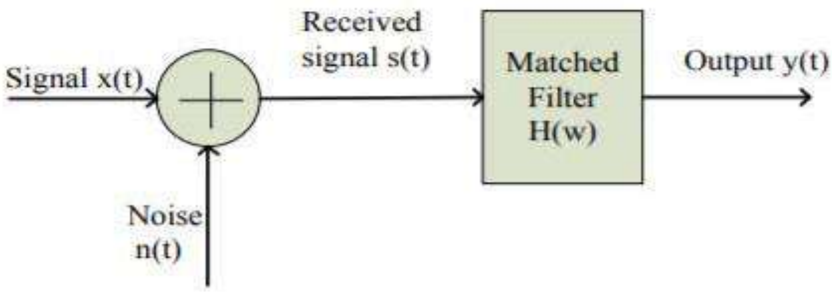


Fig.2.1 Basic representation of a matched filter.

$$s(t) = x(t) + n(t)$$

filter output, $$y(t) = s(t) * h(t).$$

Since the filter is linear, output can be written as,

$$y(t) = x(t) * h(t) + n(t) * h(t) = x0(t) + n0(t).$$

Now the matched filter is the linear filter h(t) or equivalently H(ω) that maximizes the SNR. Let the input signal for matched filter be represented by

$$x(t) = Csi(t - t1) + ni(t),$$

where t1 is the unknown time delay, ni(t) is the input white noise and C is any constant value. Now for the white input noise, its autocorrelation and power spectral density (PSD) functions can be written as

$$Rni(t) = N0\ 2\ \delta(t)\ Sni(\omega) = N0/2,$$

where N0 is a constant value. Output can then be written as, [38]

$$y(t) = Cso(t - t1) + no(t)$$

$$so(t) = si(t) * h(t)\ no(t) = ni(t) * h(t).$$

Here h(t) is the impulse response and * denotes convolution operation.

SNR of the matched filter depends on the ratio of signal energy to the PSD of the white noise at filter output. The signal component with maximum energy occurs at the sampling instant t = T. In matched filter, output is proportional to the shifted version of the auto correlation function of input signal [39].

Matched filters are used considering the noise as white noise. However, if the noise is colored, the problem is solved by using the filter, known as whitening filter, which filters the input signal noise to make it white.

Matched filters are the optimal equalizers used to maximize the SNR at each beam former output beam when the impulse response is known and the noise is additive white noise. Since each different signal requires different filters, it is called matched filtering as it has to match all the waveforms. These filters, however, do not play role in minimizing ISI. Since the optimization procedure in matched filter assumes only one signal transmitted over a channel, it does not focus much with ISI [40]. It enhances the SNR by improving the received signal and not the noise.

2.4 Whitening filters:

In any communication systems, noise is always an unavoidable parameter, which is present in the received signal that deteriorates the quality of signal power. Since it degrades the throughput of the system, it needs to be minimized. White noises are the ones with flat power spectrum having constant power at all the frequencies, i.e., they are random signal with constant power spectral density. They can also be considered as a sequence of serially uncorrelated random variables having zero mean and constant variance. Unlike this, colored noise is the one with varying power spectrum. Such noises increase inefficiency of the system. So it is imperative to whiten the colored noise. The filters that implement this process of whitening the noises are called whitening filters.

Whitening transformation is basically inferred as the decorrelation transformation of random variables with known covariance matrix into new set of random variables with identity matrix as its new covariance. This transforms the noise vector into a white noise vector, hence the name whitening filter. They are uncorrelated and have the variance equal to unity. Whitening filters can help minimize the interference due to other users in spread spectrum communication systems. The process is also known as digital whitening. During whitening, first the interfering signals are estimated and then the coefficients to be used in transversal filter are computed which is then followed by the filtering procedures for the received signal. In Figure, r(t) is the received signal

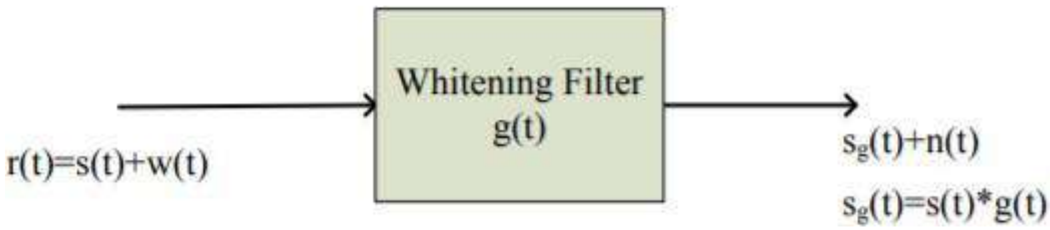


Fig.2.2 Basic flow for Whitening Filter.

comprising of signal s(t) with colored noise w(t), is passed through a whitening filter to get the whitened output signal. The term g(t) indicates the whitening filter and n(t) is the white noise having unit spectral density.

These filters can also be considered as an inverse of the system that distorts it. If the frequency response of the system is H(Z), then the white filter response is 1/H(Z). In 28 whitened matched filter model, noise whitening filter is introduced after the sampled matched filter. To be useful in practice, the whitening filter should be stable and causal as well. If the filter g(t) is used to whiten the noise in order to produce the output noise having power spectral density S 0 n (f) = N0/2 , then the filter should have [41]

$$S_n(f)|G(f)|^2 = N_0/2$$
$$i.e. \qquad |G(f)|^2 = N_0/2 \cdot 1/S_n(f)$$

where g(t) is the impulse response of filter, G(f) is its transfer function which change the power spectral density Sn(f) of the Gaussian noise process with noise N(t).

2.5 Equalizers:

Wireless communications is very likely to face different frequency dependant phase and amplitude distortions due to multipath signal transmissions and fading. In signal transmission, the received pulses come in a distorted form than the original transmitted pulse. The transmission characteristics of a band limited channel or channel noise gives rise to an ISI which is the cause for the distortion of the signals. Design of the inverse filter to counterfeit the channel impulse response so as to compensate the error and retain the signals as a replica of original signal is the main task of an equalizer.

Usually the signals are given a pattern through a pulse shaping filter. Proper sampling of the signals with no addition of the noise yields no ISI scenario. However, if it is not sampled at the sampling interval or against the Nyquist condition, the signal reconstruction is not exact. Moreover, the addition of channel noise causes a distortion in signal regenerating process at the receiver. To overcome this situation, the distorted signal is passed through a filter which is basically a multiplicative inverse in spectral domain of the frequency response of a channel. This is called equalization. Thus the equalized output is considered as reliable output.

Generally for short range transmissions, preset equalizers are used since in such case channel characteristics is not an important factor that needs to be considered. This is because the channel parameters are more or less time invariant. For the long range communications, the equalizers must be adaptive such that equalizers adapt to the time varying characteristics of the channel. The equalizer adjustment algorithms are mainly based on either peak distortion criteria or mean square distortion criteria [42].

In a communication system, considered linear time invariant, output signal can be modeled

as

$$x[n] = X\infty\ k=-\infty\ h[k]s(n-k) + v(n),$$

where s[n] is the transmitted symbol, h[k] the impulse response and v[n] is additive white gaussian noise (AWGN). Now equation can be simplified as,

$$x[n] = h[0] + s[n] + X\infty\ k=-\infty,k6=0\ h[k]s(n-k) + v(n).$$

In equation the second part of the equation refers to the delayed version of the transmitted symbols, which are known as inter symbol interferences. The equalizer is focused to eliminate this ISI to the minimum level possible. Various equalization methods are implemented as demanded by the situation of signal transmitting and receiving procedures. In the thesis work, MMSE equalization method is effectuated.

2.6 MMSE equalization

In digital communications, equalizers come along with matched filters where equalizers are responsible to minimize ISI and the matched filters reduce the channel noise. Equalizers can be classified based on their structure, algorithms used to adapt the coefficients or their optimization criteria. MMSE algorithm makes use of the joint minimization of noise and ISI parameters.

MMSE equalization is considered as a measure of estimator quality. It is based on MMSE filtering. The linear filter which is designed to minimize the error between the transmitted signal and the received output such that original signal can be estimated out of it are the MMSE filters. The error consists of an ISI in addition to the noise. The MMSE criterion ensures an optimum tradeoff between the residual ISI in the received signal and noise enhancement. In such equalization, since the minimization of mean square error (MSE) is done, the noise amplification is reduced. This is why it outperforms the zero forcing equalizers.

The main objective of the MMSE equalizer is to minimize the variance of the error signal or the the mean square value of the error. The error signal is calculated by the help of estimated signals. It is the difference of transmitted signal and the estimate of the received signal. At low to moderate SNR values, these equalizers give remarkably lower bit errors as compared to zero forcing equalizers

$$el = sl - Gl,l\hat{R}\ l$$

$$Jl = E|e|\ \hat{}2$$

In the relation (3.23), Jl is mean square error, Gl,l is equalizer coefficient, sl is the transmitted signal, $\hat{R}$ l the estimate of received signal and el is the error. Minimizing the mean square error can be done using the orthogonality principle which states that mean square error Jl is minimum if the equalizer coefficient Gl,l is selected in a way that received signal Rl and error el are orthogonal to each other [43]

$$E[elR* \ l\ ] = 0.$$

Now the equalizer coefficient for the MMSE equalization is calculated by the equation

$$Gl,l = H*l,l\ |Hl,l|\ 2 + \sigma\ 2$$

where H represents the channel matrix with the diagonal components Hl,l and σ 2 represents the noise variance. Equation infers that MMSE computation needs the value of noise variance (σ 2), which is determined during the system design that helps MMSE equalizer give the optimal performance.

CHAPTER 3

SOFTWARE INTRODUCTION:

3.1. Introduction to MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

3.2 The MATLAB system:

The MATLAB system consists of five main parts.

- **Development Environment**:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library**:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language**:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both “programming in the small” to rapidly create quick and dirty throw-away programs, and “programming in the large” to create large and complex application programs.

- **Graphics:**

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

3.3 GRAPHICAL USER INTERFACE (GUI):

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.) and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension. fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.

- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type, guide filename, Where filename is the name of an existing FIG-file on the current path. If filename is omitted,
GUIDE opens a new (i.e., blank) window.

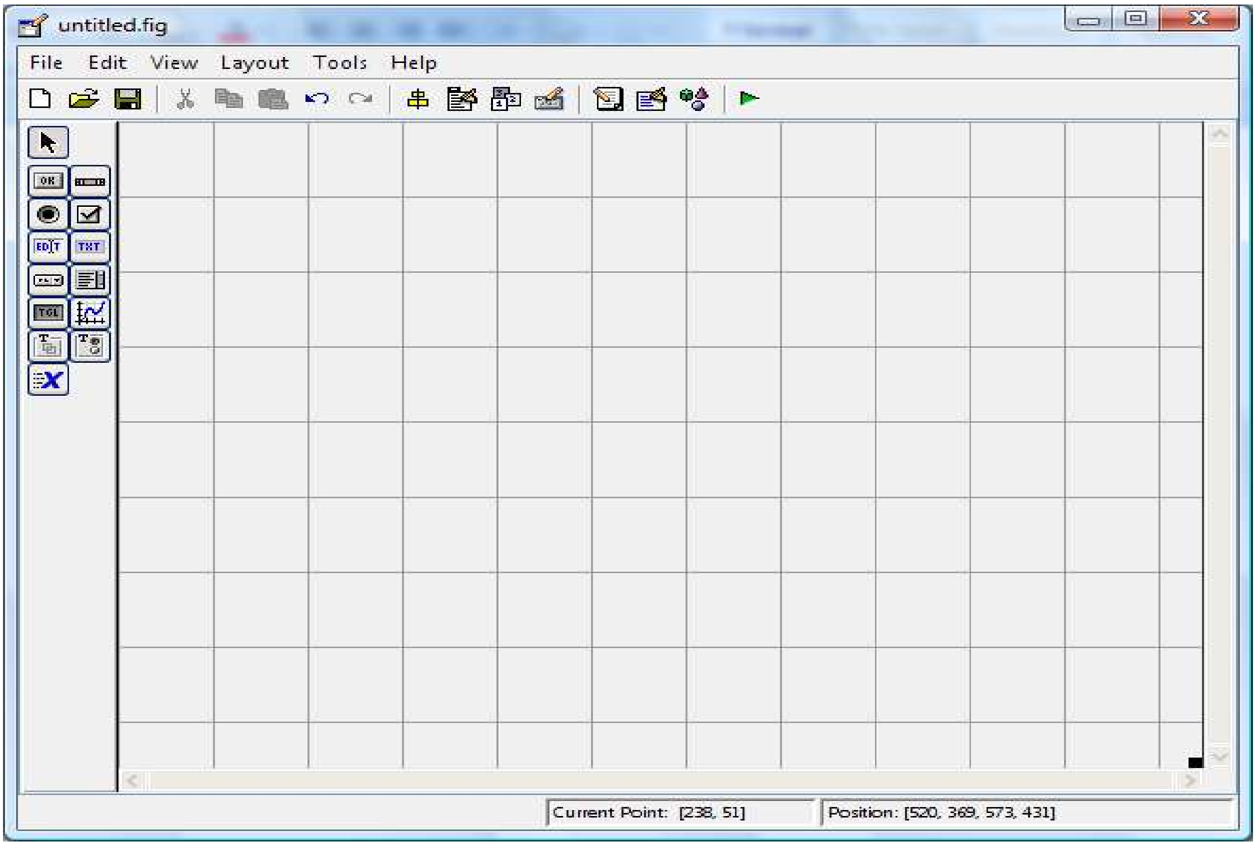


Fig.3.1

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

3.4 Getting Started

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

3.4.1 Introduction - describes the components of the MATLAB system.

3.4.2 Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

3.4.3 Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

3.4.4 Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

3.4.5 Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

3.5 DEVELOPMENT ENVIRONMENT

3.5.1 Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

3.5.2 Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop. On a UNIX platform, to start MATLAB, type MATLAB at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

3.5.3 Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish. M script.

3.5.4 MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB. The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

3.5.5 Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example, maces magic's invokes an editor called emacs for a file named magic's. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type help browser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whose.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a. mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to run MATLAB functions. The Editor/Debugger provides a graphical user interface for basic textediting, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type of function.

3.6 MANIPULATING MATRICES

3.6.1 Entering Matrices.

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Durer's matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Durer's matrix, simply type in the Command Window

A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]

MATLAB displays the matrix you just entered.

```
A =
   16    3    2   13
    5   10   11    8
    9    6    7   12
    4   15   14    1
```

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

3.6.2 Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables

- Numbers
- Operators
- Functions

Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

3.6.3 Operators

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)

^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

3.6.4 Functions

MATLAB provides many standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfin For a list of more advanced mathematical and matrix functions, type help spec fun help elm at

Some of the functions, like sqrt and sin, are built in. They are part of the MATLAB core, so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sin, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.14159265...
I	Imaginary unit, $\sqrt{-1}$
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realm in	Smallest floating-point number, 2^{-1022}
Realm ax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NAN	Not-a-number

3.7 GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

3.7.1 Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of UI controlled objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

3.7.2 GUI Development Environment

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of its children (UI controls and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the callbacks, which are defined as subfunctions. This M-file is referred to as the application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

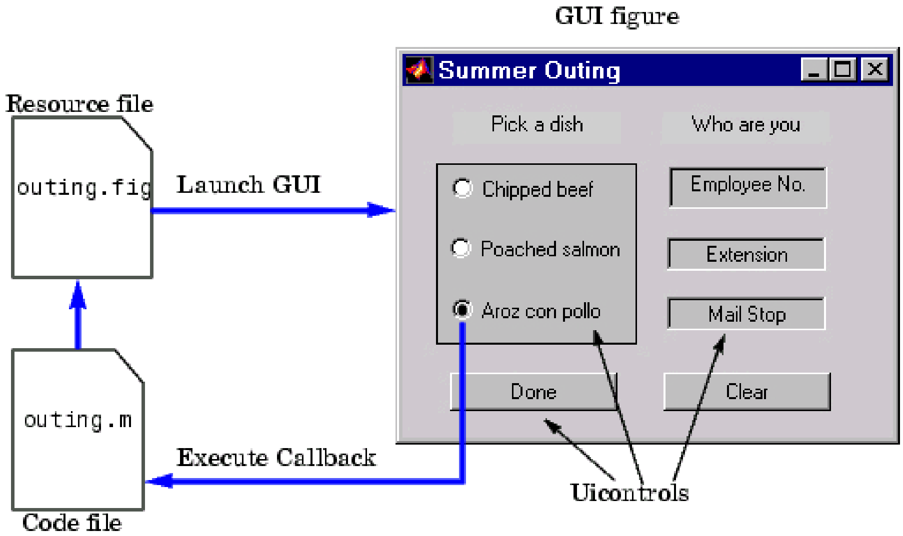


FIG 3.2 graphical user blocks

3.7.3 Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no UI control creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

3.7.4 Beginning the Implementation Process:

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques used in the application M-file.

Application Examples - a collection of examples that illustrate techniques which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

3.7.5 User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB UI control objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- List boxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h, event data, handles, varargin)
button_state = get(h,'Value');
if button_state == get(h,'Max')
   % toggle button is pressed
elseif button_state == get(h,'Min')
   % toggle button is not pressed
end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a true color image. For example, the array a defines 16-by-128 true color image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);
a(:,:,2) = rand(16,128);
a(:,:,3) = rand(16,128);
set(h,'CData',a)
```

Radio Buttons

Radio buttons are similar to checkboxes but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)
set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)
off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];
mutual_exclude(off)
% Continue with callback
 .
 .
 .
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with several independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)
if (get(h,'Value') == get(h,'Max'))
   % then checkbox is checked-take approriate action
else
   % checkbox is not checked-take approriate action
end
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)
user_string = get(h,'string');
% proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)
user_entry = str2double(get(h,'string'));
if isnan(user_entry)
errordlg('You must enter a numeric value','BadInput','modal')
end
% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menu bar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menu bar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only UI controls can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The List box Top property defines which string in the list displays as the topmost item when the list box is not large enough to display all list entries. List box Top is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Non integer values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

List boxes differentiate between single and double clicks on an item and set the figure Selection Type property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections.In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure Selection Type property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)
val = get(h,'Value');
switch val
case 1
% The user selected the first item
case 2
% The user selected the second item
% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)
val = get(h,'Value');
string_list = get(h,'String');
selected_string = string_list{val}; % convert from cell array to string
% etc.
```

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure Window Button Down Fcn callback executes. Then the control's Button Down Fcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not UIcontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes Button Down Fcn property to define the callback.

3.7.6 Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

```
axes(handles.axes1)
```

makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axes. See GUI with Multiple Axes for and example that uses two axes.

Figure

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control.

Chapter 4

PROJECT IMPLIMENTATION

PROPOSED FTN-BASED HYBRID SIGNALING

In this section, we present the proposed hybrid signaling scheme employing the FTN signaling for the data transmission and the OFDM signaling for the ICI-free pilot transmission. Key features of the proposed scheme are as follows:

• A slot, the unit of data scheduling in time domain, is generated by the time division multiplexing (TDM) of OFDM and FTN symbols (see Fig. 3).

• Control and pilot signals are transmitted by the OFDM signaling to guarantee the accurate channel estimation quality. Whereas, data transmission is served by the FTN signaling for better spectral efficiency.

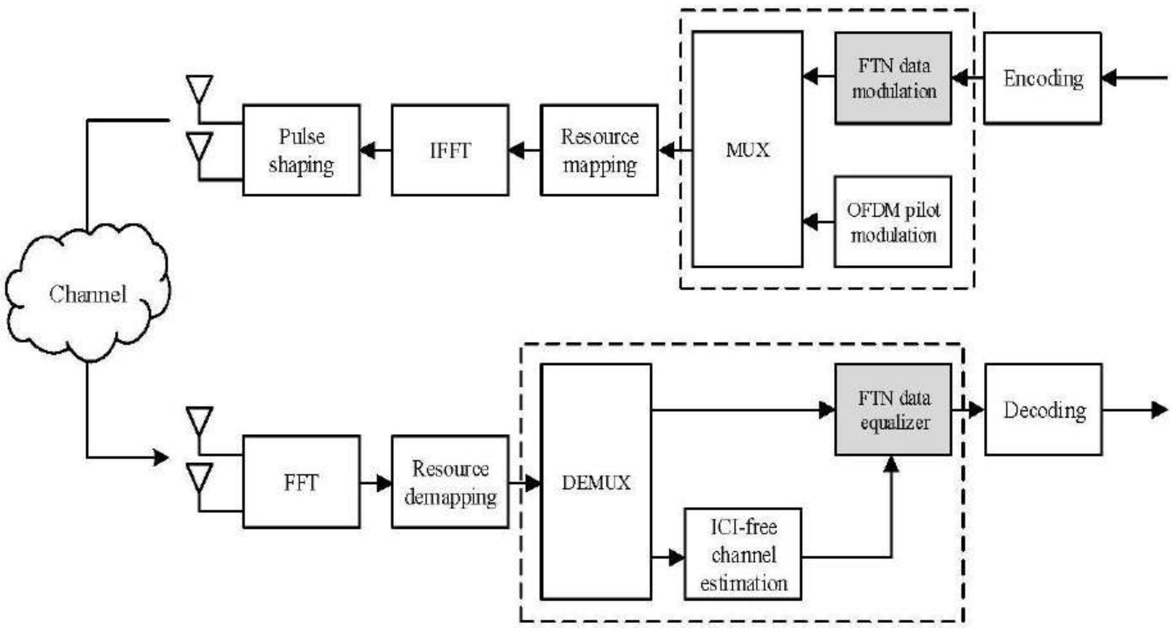


Fig. 4.1 Proposed Block Diagram

4.1 Proposed Slot Structure

In 4G LTE or 5G NR systems, user data is transmitted on resource units [12], [13].4 In the proposed scheme, we set the duration of OFDM and FTN symbols to be the same (i.e., TOFDM = TFTN), which simplifies the scheduling operation in the basestation as well as the buffering operation at the user terminal. Similarly, alignment of resource block (RB)

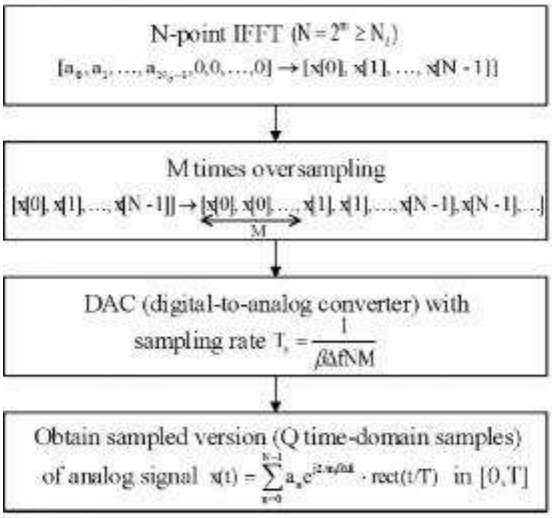


Fig: Multi-carrier signal generation procedure.

between OFDM and FTN symbols can also simplify the system operation by enabling the transceiver processing per RB. Towards this end, the squeezing factor β should be chosen such that the number of subcarriers for FTN and OFDM in one RB satisfies NRB,OFDM = βNRB,FTN(NRB,FTN ∈ N), given that ΔfFTN = βΔfOFDM.

Fig shows the case where each RB consists of 12 sub-carriers (for the OFDM signaling) and 15 and 20 subcarriers (for the FTN signaling). In general, downlink slot consists of control channel, pilot signal, and data. In case each slot consists of 14 symbols as in 5G NR system [1], two symbols are used for the control channel, one symbol is used for the pilot signaling, and the rest of 11 symbols are used for the data transmission. When compared to the OFDM systems, the proposed FTN transmission scheme can achieve up to 20% spectral efficiency gain for β = 0.8. In case the downlink slot consists only of pilot signals and data, the spectral efficiency gain is increased up to 23%.

Practical Consideration on the Proposed Hybrid Signaling

In this subsection, we discuss the sampling rate alignment between OFDM and FTN symbols used to facilitate the implementation of the proposed FTN-based hybrid signaling. By the alignment of the sampling rate, complicated timing switch between OFDM and FTN signaling can be prevented. Fig. depicts the multi-carrier signal generation procedure at the transmitter. Without loss of generality, the IFFT size is set to be larger than or equal to the number of data subcarriers Nd (i.e., N = 2m ≥ Nd (m ∈ Z)). After the IFFT operation, oversampling (with factor of M) and digital-to-analog conversion (DAC) with the sampling rate Ts are performed. Finally, under given symbol duration T, pulse shaping (e.g., square-root raised cosine) is applied to obtain the time-domain samples in the analog domain.

Two factors determining the alignment of the sampling rate Ts between OFDM and FTN symbols are the IFFT size N and oversampling factor M (see Fig. 4). First, in order to avoid the change of the IFFT size in the same slot, the IFFT size of the OFDM signaling should be set such that N ≥ Nd,FTN (Nd,FTN>Nd,OFDM) where Nd,FTN and Nd,OFDM are the

TABLE.4.1 Number of subcarriers for the system bandwidth supported by the 5G system(β = 0.8).

		Bandwidth [MHz]					
	SCS [kHz]	5	10	15	20	25	40
OFDM	15	300	624	948	1272	1596	2592
	30	132	288	456	612	780	1272
	60	N/A	132	216	288	372	612
FTN	12	375	780	1185	1590	1995	3240
	24	165	360	570	765	975	1590
	48	N/A	165	270	360	465	765

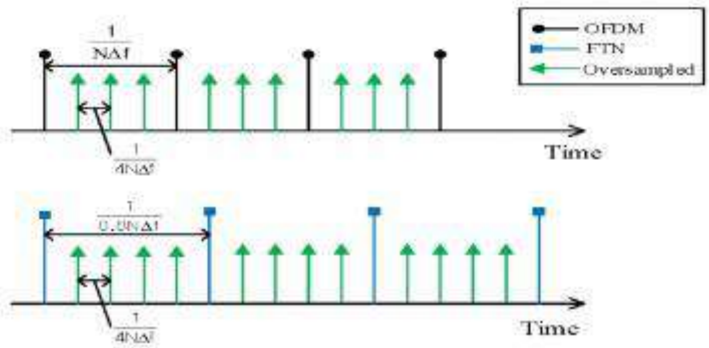


Fig.4.2

Time domain samples of OFDM and FTN signaling when β = 0.8.number of data subcarriers supported by the FTN and OFDM signaling, respectively. For example, as observed from Table I, when the system bandwidth is 15 MHz and Δf = 15 kHz, OFDM can use 1024 IFFT size for Nd,OFDM = 948, but FTN uses 2048 IFFT size for Nd,FTN = 1185. By choosing the IFFT size N = 2048, we can employ the same IFFT size in the transmission time interval.

We next investigate the oversampling factor M with the same IFFT size N for both OFDM and FTN signaling. Note that the sampling rates of the OFDM and FTN symbols are expressed as Ts,OFDM =1/ΔfNMOFDM and Ts,FTN =1/βΔfNMFTN, respectively. In order to avoid the switching of the sampling rate in the same slot, we need to choose the oversampling factor such that βMFTN = MOFDM. As shown in Fig. 5, by choosing the oversampling factors MOFDM = 4 and MFTN = 5 when β = 0.8, we can satisfy the alignment condition βMFTN = MOFDM.

In summary, two conditions to align the sampling timing of OFDM and FTN symbols are as follows:

• The IFFT size N for both OFDM and FTN signaling should satisfy N ≥ Nd,FTN.

• The oversampling factors for OFDM and FTN should satisfy βMFTN = MOFDM.

4.2 PAPR Analysis

In this subsection, we investigate the peak to average power ratio (PAPR) of the proposed FTN-based hybrid signaling scheme. Note that the PAPR is an important metric in the waveform design since the high PAPR brings on the signal distortion in the nonlinear region of the high power amplifier [14]. The PAPR of continuous-time signal x(t) is defined as

$$PAPR = \frac{\max |x(t)|^2}{E[|x(t)|^2]},$$

,

where max |x(t)|^2 is the peak power of the signal x(t) and E[|x(t)|^2] is the average power. The discrete version of PAPR can be expressed as

$$PAPR = \frac{\max_{k\in\{0,\ldots,Q-1\}} |x[k]|^2}{E[|x[k]|^2]}$$

where Q is the number of time-domain samples at a given symbol duration T (see (3)). When the number of subcarriers is sufficiently large, the distribution of x[k] approaches to the complex Gaussian and thus |x[k]|^2 can be readily approximated to χ 2-random variable with 2 degrees of freedom. Thus, the PDF is given by pX(x) = 1 2σ2 e –x 2σ2 and the PDF of Y = |x0[k]|^2 =|x[k]|^2 E[|x[k]| 2] =X 2σ2 is pY (y) = 2σ 2pX(2σ 2y) = e−y Since the CDF of Y is given by Pr(Y ≤ y) = R Y0pY (y) =1 − e−γ, the CDF of PAPR in (8) is

$$\begin{aligned} Pr(PAPR \le \gamma) &= Pr\left(\bigcap_{i=0}^{Q-1} \left(|x_0[k]|^2 \le \gamma\right)\right) \\ &= \prod_{i=0}^{Q-1} Pr\left(\left(|x_0[k]|^2 \le \gamma\right)\right) = \prod_{i=0}^{Q-1} (1 - e^{-\gamma}) \end{aligned}$$

and hence the complementary CDF of the PAPR is

$$Pr(PAPR > \gamma) = 1 - \prod_{i=0}^{Q-1} (1 - e^{-\gamma}) = 1 - \left(1 - e^{-\gamma}\right)^Q$$

When the IFFT sizes of OFDM and FTN symbols are the same and the oversampling factors satisfy βMF T N = MOFDM, the number of time domain samples Q in the same symbol duration T equals for both OFDM and FTN symbols. It is not hard to convince from this discussion that when the OFDM and FTN symbols have the same number of time domain samples Q, similar PAPR performance can be achieved for both OFDM and FTN symbols.

.

4.3 Transmission Capacity

In this subsection, we compare the capacity of the conventional OFDM signaling and FTN-based signaling schemes. The achievable capacity for the OFDM signaling and FTN-based signaling under SISO AWGN channel can be expressed as

$$C_{OFDM} \approx W \log2(1 + PS/PN),$$
$$C_{FTN} \approx W/\beta \log2(1 + PS/PN + PICI),$$

where W is the signal bandwidth, PS is the signal power, PN is the noise power, and PICI is the ICI power. One can easily see that the maximum achievable capacity of the FTN-based signaling is higher than that of the OFDM signaling when ICI is controlled properly. When we set β = 0.8, performance gap of the proposed scheme over the OFDM signaling with perfect channel state information is negligible [4]. In Fig, we observe that the capacity gain of the FTN-based signaling over the OFDM signaling is about 23%, which is close to the maximum achievable capacity gain (i.e., 25%) for β = 0.8.

4.4 Code Implementation

```
%Towards Faster-Than-Nyquist Transmission for
%Beyond 5G Wireless Communications
clc;
clear all;
close all
% Setting parameters
   numOfBlk = 1e4;     % number of blocks of data to be transmitted
   qamOrder = 16;      % the QAM modulation order
   SNRdB    = 6:1:30;
   p=1;c=2;h=0.5;packet_size=10;h_h=100;tranimitters=4;
   receivers=2;t=0.1;k=0.2;
   linColor = 'b'; %
   linSym  = 'o'; %
errRate = zeros(size(SNRdB));

for i = 1 : length(SNRdB)
% Main Program
   txData = randint(numOfBlk*2,1,qamOrder);
   temp = reshape(txData,numOfBlk,2);
   y=p*c*h+packet_size;     %%%%%%%%%%
```

```
    % QAM Modulation of transmite data
    temp = qammod(temp,qamOrder);
    H = 1/sqrt(2) * (randn(numOfBlk,2) + sqrt(-1)*randn(numOfBlk,2));
    txMod(:,1) =  H(:,1).* 1/sqrt(2).*temp(:,1)     + H(:,2).* 1/sqrt(2).*temp(:,2)     ;
    txMod(:,2) = -H(:,1).*(1/sqrt(2).*temp(:,2)').' + H(:,2).*(1/sqrt(2).*temp(:,1)').' ;
    s=p*c*y  %%%%%%%%5

    % adding noise
    txMod = awgn(txMod,SNRdB(i),'measured');
    m=s*(tranimitters*((h-h_h)*(h-h_h)));    %%%%%%%%%11
```

% receiving the data

```
    temp(:,1) = sqrt(2)*(H(:,1)'.' .* txMod(:,1) + H(:,2) .* txMod(:,2)'.')./(abs(H(:,1)).^2 +
abs(H(:,2)).^2);
    temp(:,2) = sqrt(2)*(H(:,2)'.' .* txMod(:,1) - H(:,1) .* txMod(:,2)'.')./(abs(H(:,1)).^2 +
abs(H(:,2)).^2);

    rxData(:,1) = qamdemod(temp(:,1),qamOrder);
    rxData(:,2) = qamdemod(temp(:,2),qamOrder);
   [numErr errRate(i)] = symerr(rxData,reshape(txData,numOfBlk,2));
end
```

%% No space time coding

```
for i = 1 : length(SNRdB)
    temp = qammod(txData,qamOrder);
    q=real(receivers*(t))/s*real(receivers*(t));     %%%%%%%7
    % Channel Definition
    H  = 1/sqrt(2) * (randn(numOfBlk*2,1) + sqrt(-1)*randn(numOfBlk*2,1));
    % passing through channel
    txMod = H.*temp;
    % adding noise
    txMod = awgn(txMod,SNRdB(i),'measured');
    % decoding
    temp = txMod./H;
    rxData = qamdemod(temp,qamOrder);
    [numErr errRate(i)] = symerr(rxData,txData);
end
```

Chapter 5

PROJECT TESTING AND RESULTS

5.1 OVERVIEW OF TESTING METHODS:

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

5.2 RESULTS:

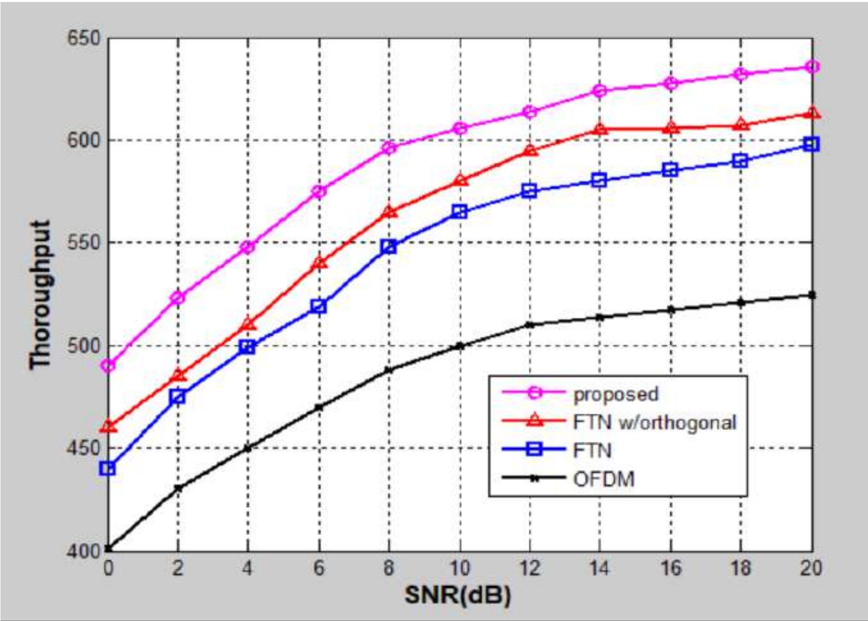


Fig. 5.1 Throughput performance for hybrid FTN and OFDM

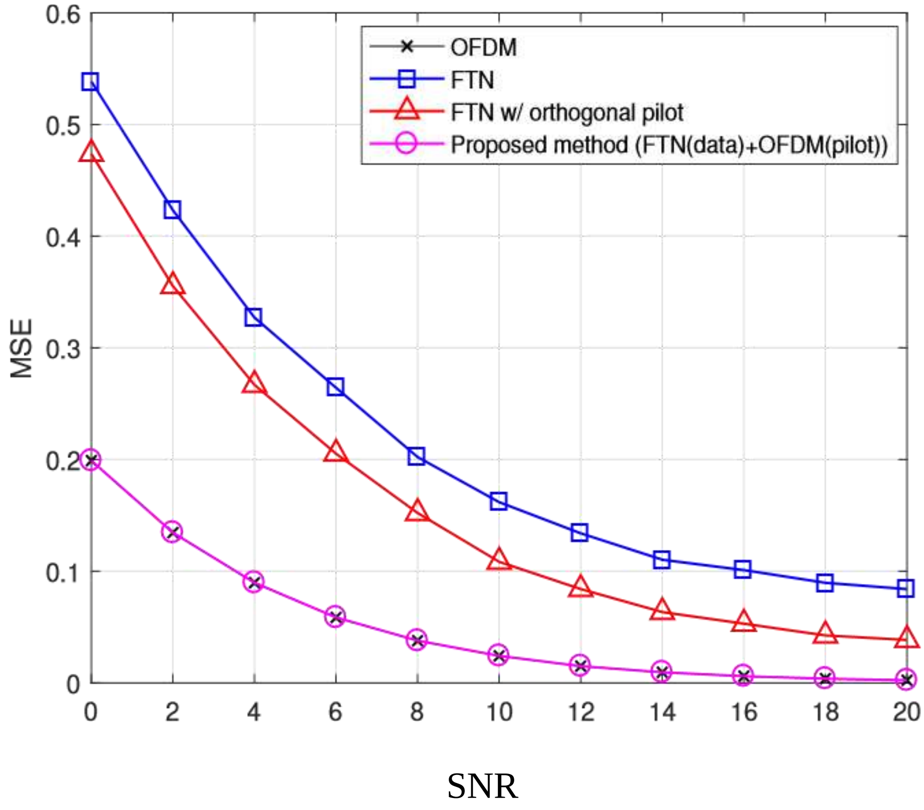


Fig. 5.2 MSE performance for hybrid FTN and OFDM

Chapter 6

CONCLUSION AND FUTURE ENHANCEMENT

In this paper, we proposed the FTN signaling scheme suitable for the MIMO transmission. Our work was motivated by the observation that the FTN-based pilot transmission degrades the channel estimation quality severely. Key features of the proposed scheme are 1) channel estimation using the OFDM signaling and 2) data reception using the FTN signaling. We observed from numerical evaluations that the proposed hybrid scheme achieves a substantial gain in the spectral efficiency over the OFDM and FTN systems. One future direction is to improve the channel estimation quality of the proposed FTN signaling by recycling the detected symbols as pilots.

REFERENCES

[1] 3GPP TS 38.211, “Physical channels and modulation (Release 15),” V15.2.0, Jun. 2018.

[2] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, “Ultra-reliable and low-latency communications in 5G downlink: physical layer aspects”, IEEE Wireless Comm., vol. 25, no. 3, pp. 124-130, Jun. 2018.

[3] A. D. Liveris and C. N. Georghiades “Exploiting faster-than-Nyquist signaling,” IEEE Trans. Commun., vol. 51, no. 9, pp. 1502-1511, Sep. 2003.

[4] F. Rusek and J. B. Anderson “Non-binary and precoded faster-than- Nyquist signaling,” IEEE Trans. Commun., vol. 56, no. 5, pp. 808-817, May 2008.

[5] T. Xu and I. Darwazeh, “Spectrally efficient FDM: Spectrum saving technique for 5G,” Proc. 1st Int. Conf. 5G Ubiquitous Connectivity (5GU), Nov. 2014, pp. 273-278.

[6] A. Chorti, I. Kanaras, M. Rodrigus and I. Darwazeh, ”Joint channel equalization and detection of spectrally efficient FDM signals”, in Proc. of IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), 2010.

[7] S. Isam and I. Darwazeh, ”Robust channel estimation for spectrally efficient FDM system”, in Proc. of 19th International Conference on Telecommunications (ICT), 2012.

[8] S. Li, B. Bai, J. Zhou, P. Chen, and Z. Yu, “Reduced-complexity equalization for Faster-than-Nyquist signaling: new methods based on Ungerboeck observation model,” IEEE Trans. Commun., vol. 66, no. 3, pp. 1190-1204, Mar. 2018.

[9] J. G. Proakis and M. Salehi, Digital Communications, 5th ed. McGraw-Hill, 2008.

[10] M. McGuire and M. Sima, “Discrete time faster-than-Nyquist signaling,” in Proc. IEEE Global Telecommun. Conf., Miami, FL, USA, Dec. 2010, pp. 1-5.

[11] J. Choi and D. J. Love, “Bounds on eigenvalues of a spatial correlation matrix,” IEEE Comm. Lett., vol. 18, no. 8, pp. 1391-1394, Aug. 2014.

[12] C. Lim, T. Yoo, B. Clerckx, B. Lee, and B. Shim, “Recent trend of multiuser MIMO in LTE-Advanced,” IEEE Comm. Mag., vol. 51, no. 3, pp. 127-135, Mar. 2013.

[13] H. Ji, Y. Kim, J. Lee, E. Onggosanusi, Y. Nam, J. Zhang, B. Lee, and B. Shim, “Overview of Full-Dimension MIMO in LTE-Advanced Pro,” IEEE Comm. Mag., vol. 55, no. 2, pp. 176-184, Feb. 2017.

[14] C. Siclet, D. Roque, A. Marquet, and L. Ros, “Improving spectral efficiency while reducing PAPR using faster-than-Nyquist multicarrier signaling,” in Proc. Int. Workshop. Commun Technol. Veh., pp. 2132, 2017.

[15] S. Park, B. Shim, and J. Choi, “Iterative channel estimation using virtual pilot signals for MIMO-OFDM systems,” IEEE Trans. Sig. Proc., vol. 63, no. 12, pp. 3032-3045, June 2015.

A

MAJOR PROJECT REPORT

On

SOLDIER HEALTH AND POSITION TRACKING SYSTEM

Submitted by

1)Ms. Milaliyan Samantaray (17K81A04M8) 2) Mr.Gurram Rakesh(17K81A04K9)

3) Ms. Preethi Roshan (17K81A04N8)

in partial fulfillment for the award of the degree

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Esteemed Guidance of

Mr. P. JOEL JOSEPHSON

Associate Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

An Autonomous Institute

Dhulapally, Secunderabad – 500 10

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)
An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the major-project work entitled **"Soldier Health and Position Tracking System"** is a bonafide work carried out by **Gurram Rakesh (17K81A04K9), Milaliyan Samantaray (17K81A04M8), Preethi Roshan (17K81A04N8)** in partial fulfilment of the requirements for the degree of **Bachelor of Technology in Electronics & Communication Engineering** by the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2020-21.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree.

PROJECT GUIDE	**HEAD OF DEPARTMENT**
P.Joel Josephson **Associate Professor** **Department of ECE**	**Dr. B. Hari Krishna,** **Professor** **Department of ECE**
Internal Examiner	External Examiner

Place:

Date:

LASYA INFOTECH

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT MILALIYAN SAMANTARAY WITH ROLL NO.17K81A04M8, GURRAM RAKESH WITH ROLL NO.17K81A04K9, PREETHI ROSHAN WITH ROLL NO.17K81A04N8, OF B. TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "SOLDIER HEALTH AND POSITION TRACKING SYSTEM" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT

DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014

Email : contact@lasyainfotech.com, ov@lasyainfotech.com

Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/8

DECLARATION

We, the students of **Bachelor of Technology** in Department of ELECTRONICS AND COMMUNICATION ENGINEERING, session: <2017 – 2021>, St. Martin’s Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled “SOLDIER HEALTH AND POSITION TRACKING SYSTEM” is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

MILALIYAN SAMANTARAY GURRAM RAKESH PREETHI ROSHAN

(17K81A04M8) (17K81A04K9) (17K81A04N8)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr.P.SANTOSH KUMAR PATRA**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr.B.HARI KRISHNA**, **Head &Professor**, Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in our memory for years to come.

We would like to express our gratitude to our dept. project coordinator **Dr.A.ANAND, Professor** for his involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time to come.

Finally, we are highly grateful and obliged to our project guide **Mr.P.JOEL JOSEPHSON, Associate Professor,** for his advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

1.MILALIYAN SAMANTARY (17K81A04M8)
2.GURRAM RAKESH (17K81A04K9)
3.PREETHI ROSHAN (17K81A04N8)

ABSTRACT

The soldier Health and Position Tracking System allows military to track the current GPS position of soldier and checks the health status including body temperature and heartbeats of soldier. The System also consists extra feature with the help of that soldier can ask for help manually or send a distress signal to military if he is in need. The GPS modem sends the latitude and longitude position with link pattern with the help of that military can track the current position of the soldier. The system is very helpful for getting health status information of soldier and providing them instant help.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The nation's security is monitored and kept by army, navy, and air-force. The important and vital role is of soldiers who sacrifice their life for their country. There are many concerns regarding the safety of the soldier. Soldiers entering the enemy lines often lose their lives due to lack of connectivity, it is very vital for the army base station to know the location as well as health status of all soldiers. India has already lost so many soldiers in war-fields as there was no proper health backup and connectivity between the soldiers on the war-fields and the officials at the army base stations. Recently on 29 September 2016, a military confrontation between India and Pakistan began, Indian soldiers conducted a surgical strike against militant launch pads across the line of control in Pakistani-administered Azad Kashmir and inflicted "significant casualties". Indian soldiers are mainly known for their courage, despite scarce ammunitions and safety measures, they have many triumphs to their credits. All must be really concerned about the safety of the soldiers, so we have decided to build a project which will efficiently keep a check on the health status of the soldier, and his precise location to equip him with necessary medical treatments as soon as possible. Soldier's tracking is done using GPS and GSM is used to provide wireless communication system. For monitoring the health parameters of soldiers, we are using biomedical sensors such as temperature sensors and heartbeat sensors.

An oxygen level sensor is used to monitor atmospheric oxygen so if there are any climatic changes the soldiers will be equipped accordingly.
The infantry soldier of tomorrow promises to be one of the most technologically advanced modern warfare has ever seen. Around the world, various research programs are currently being conducted, such as the United States' Future Force Warrior (FFW) and the United Kingdom's Future Infantry Soldier Technology (FIST), with the aim of creating fully integrated combat systems .Alongside vast improvements in protective and weaponry subsystems, another major aspect of this technology will be the ability to provide information superiority at the operational edge of military networks by equipping the dismounted soldier with advanced visual, voice, and data communications.

Helmet mounted visors, capable of displaying maps and real-time video from other squad members, ranges of physiological sensors display the heartbeat, body temperature, atmosphere pressure, surrounding oxygen level etc. These devices will improve awareness for collateral military personnel as well as who will exchange information using wireless networks along with hosts. The challenge was to integrate these piecemeal components into a lightweight package that could achieve the desired result without being too bulky and cumbersome or requiring too much power. Communicating with the base (control room) station become the fundamental challenges in military operations. The proper navigation between soldier's organizations plays an important role for careful planning and coordination.

1.2 OBJECTIVES OF THE STUDY

So, this project focuses on tracking the location of soldiers from GPS, which is useful for control room stations to know the exact location of soldiers and accordingly they will guide them. Also High-speed, short-range, soldier-to-soldier wireless communications to relay information on situational awareness, such as Biomedical sensors, GPS navigation, Wireless communication.

1.3 SCOPE OF THE STUDY

- Soldier health and position tracking system is such a type of a system that provides the health and position status of a soldier fast with the help of gps and gsm modules.

- Soldiers are important assets of our country. They workday and night to protect our country. Sometimes even they sacrifice their life for the country.

- And soldiers at borders often lose their lives defending the country.

- To overcome this situation this system helps us to track the position of the soldiers and even information regarding health is produced by this system. And will efficiently keep a check on the health status of the soldier, and his precise location to equip him with necessary medical treatments as soon as possible.

- Here we have proposed a system that is called Soldier health and position tracking system, which is so much efficient, reliable, and friendly to use. In this system soldier's tracking is done using GPS and GSM is used to provide wireless communication systems. For monitoring the health parameters of soldiers, we are using biomedical sensors such as temperature sensors and heartbeat sensors.

1.4 MATERIAL REQUIREMENT:

1.4.1 HARDWARE REQUIREMENT:

- Arduino uno
- 12V Battery
- Temperature Sensor
- Heartbeat sensor
- Push Button
- LCD
- GSM
- GPS
- Buzzer

1.4.2 SOFTWARE REQUIREMENT

- Proteus Software
- Arduino Software

1.5 PROCUMENT OF EQUIPMENT

We brought all the required hardware components in an electronic manufacturing shop and before assembling and interfacing hardware components to Arduino UNO micro-controller, we connected individual components to Arduino in Proteus Software and understood working of every individual component in Proteus Software.

Now, we assembled all components and interfaced to Arduino UNO micro-controller using connecting wires through bread board and we wrote program to work all the hardware components in Arduino IDE we understood the working of all components in a practical way.

We were able to monitor the health and position of soldiers better than before we successfully got the output in mobile and LCD screen. We would like to improve the project in future for further developments.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH AREA

Nowadays all nations keep its security at high priority. Wars are being fought for land, water and acquiring the position of most powerful nation. A country's arm forces consist of three professional uniformed services: the army, the navy, and the air force. Soldiers being the backbone of any armed force usually lose their lives due to lack of medical help when in emergency, also soldiers who are involved in missions or in special operations get straggled on war fields and lose contact with the authorities. To overcome these concerns, we had built this project which, using wireless body area sensor network (WBANS) such as temperature sensor, heartbeat sensor etc. will monitor the health status of the soldier whenever required. Also using GPS, we can track the soldier's exact location whenever required. Using oxygen level sensor, we can also monitor the environmental condition, so authorities can provide essential aids. The communication is established between the soldiers and authorities via GSM. Any abnormalities in the readings of wireless body area sensor network (WBASNs) are considered as a trigger for GSM to establish the connection between the soldier and base unit and send current location and health status to the receiver. By using all this equipment's, we had tried to implement the basic guarding system for the soldier in low cost, light weighted, portable and precise device.

In enemy territory soldiers not only have to deal with the physical threat, but also with stress and fatigue caused by protracted operations or lack of sleep. So, for the security purpose we need a tool for remote soldier performance and health monitoring. So, in this project a tool is implemented using biosensors for health monitoring purpose. Also, a GPS system is used to track the location of soldier. Additionally, a GSM modem is also used to make the system wirelessly compatible.

In today's world, warfare is an important factor in any nation's security. One of the important and vital roles is played by the army soldiers. There are many concerns regarding the safety of soldiers. So, for their security purpose, many instruments are mounted on them to view their health status as well as their real time location. Bio-sensor systems comprise various types of small physiological sensors, transmission modules and processing capabilities, and can thus facilitate low-cost wearable unobtrusive solutions for health monitoring. This paper gives an ability to track the location and monitor health of the soldiers in real time who become lost and get injured in the battlefield.

2.2 REVIEW ON RELATED LITERATURE

In [1], the author has worked on the safety of the soldier by tracking his health condition during the war which provides the control room to plan the war strategies and tracks the location if by chance the soldier is lost. When the control room notices the soldier is lost in the battlefield then they guide the soldier in the right path. The soldier's health information is transmitted to the control room and keeps track of the injured soldier and take necessary action to save the life. This system mainly focuses on the soldier who have involved in the warfare and tracking the health of the injured soldier. In [2], the author has focused to improve the communication of the soldier with the control room people and control plane operation continuously. He has also focused to track and guide the position, direction, and the surrounding temperature of the soldier by using the wristwatch mountaineers and by providing headphones to guide the soldier to the right path. He has also used the technology like the Radio collar strapped to ankles, so the movement of the soldier can be tracked and displays the current location at the base station. Microcontroller is used to record the body parameters of the soldier's and transmitted to the base station. In [3], the author has focused to provide the embedded wireless system for the soldier and to minimize the time to track the location of the soldier, rescue and search operation and their health status using GPS module and wireless body sensors, the data are collected from the GPS and sensors are transmitted using the ZigBee technology to the base station/control room. It also allows the soldier to communicate with the other fellow soldiers within the wireless transmission range, they can also request help from the control room. In [4], the author has worked on the location tracking system using GPS with the Google Maps based monitoring for vehicle. This system provides tracking irrespective of weather condition. It gives the shortest route to track. This idea is deployed in the proposed system to track the current location and movement of the soldier along with the shortest path to him, which is need for the rescue operation. In [5], here the author focus is on monitoring the ill patient, continuously and reports the changes spontaneously to the concerned person using the GPS, 4 Chaitra R L, Mamatha V which helps in protecting the life of the individual. In this paper only, the ill patient is considered for monitoring. Once the patient is discharged from the hospital, the body sensor is mounted on them to monitor their health condition and take necessary action in case of emergency by reporting to the guardian. In [6], the author has worked to track the location of the soldier and monitor the health status using an Arduino board. For the transmitting of data to control room he has used GPS module for connectivity. In [7], the author focus is to provide a good connectivity between the control room and the soldier by using Arduino and GSM module.

2.3 CONCLUSION ON REVIEWS

From the above implementation we have concluded that the communication hurdles between the soldiers and authorities at the base unit is overcome using GSM, the precise location and health parameters are known using GPS and wireless body area sensor network (WBASNs) respectively and with the GSM modem all information is send to the base station so that field commander will take necessary action. Base station gets location of soldier from GPS. The base station can access the status of the soldier which is displayed on the phone with the help of GSM and hence appropriate actions can be found.

CHAPTER 3

PROJECT DESIGN

3.1 OVERVIEW OF THE DESIGN

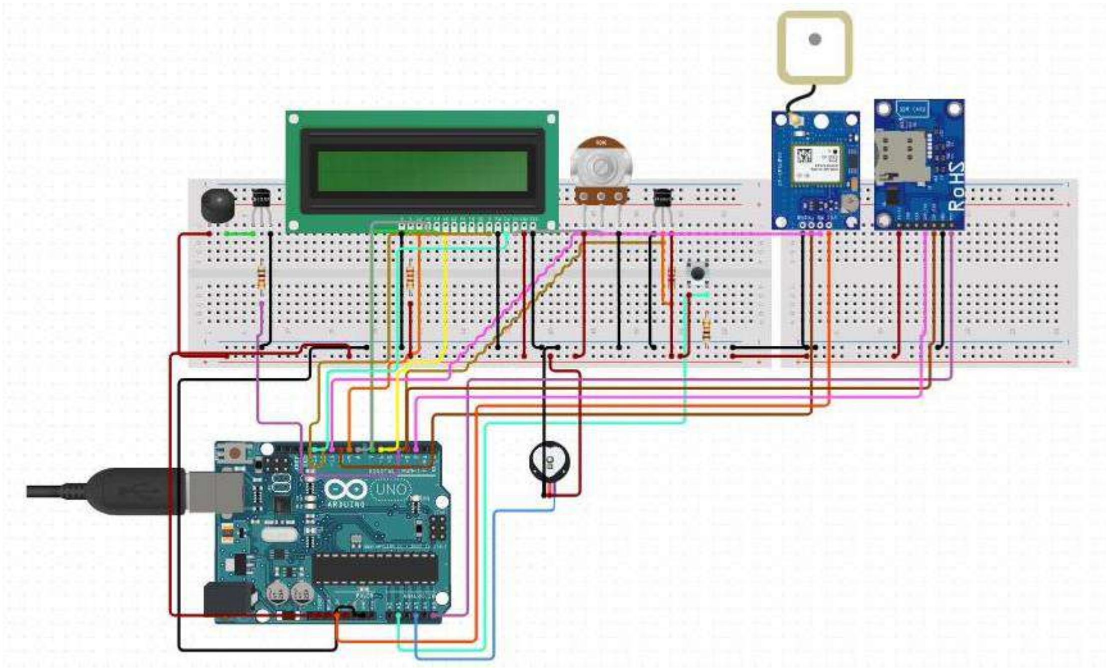


Fig.3.1 Overview of the design

3.2 EQUIPMENT ANALYSIS

3.2.1 ARDUINO

3.2.1.1 INTRODUCTION

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).

- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

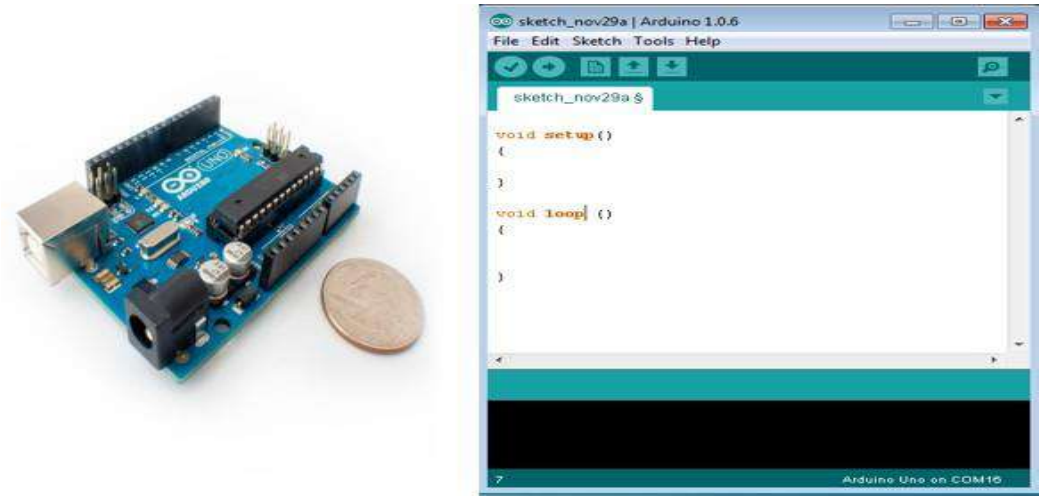


Fig 3.2.1.1 Arduino Uno

3.2.1.2 BOARD TYPES

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface

Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini-05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini-3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header

Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Lilypad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compati
Lilypad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.2.1.2.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 3.2.1.2.2 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface

Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 3.2.1.2.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table 3.2.1.2.4 Arduino boards based on AT91SAM3X8E microcontroller

3.2.1.3 BOARD DESCRIPTION

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

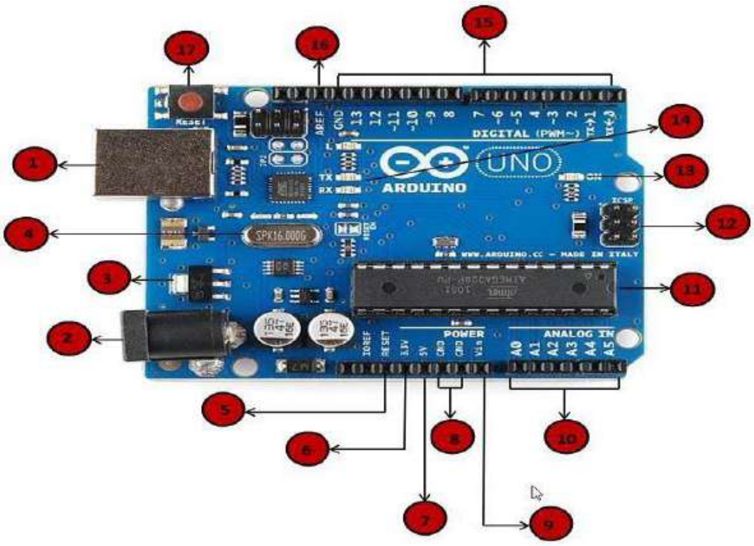


Fig.3.2.1.3 Board Description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).

6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volts. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in

	two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Table 3.2.1.3 Board description of Arduino UNO

3.2.1.4 ARDUINO FAMILY

Arduino makes several different boards, each with different capabilities. In addition, part of being open-source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you are not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

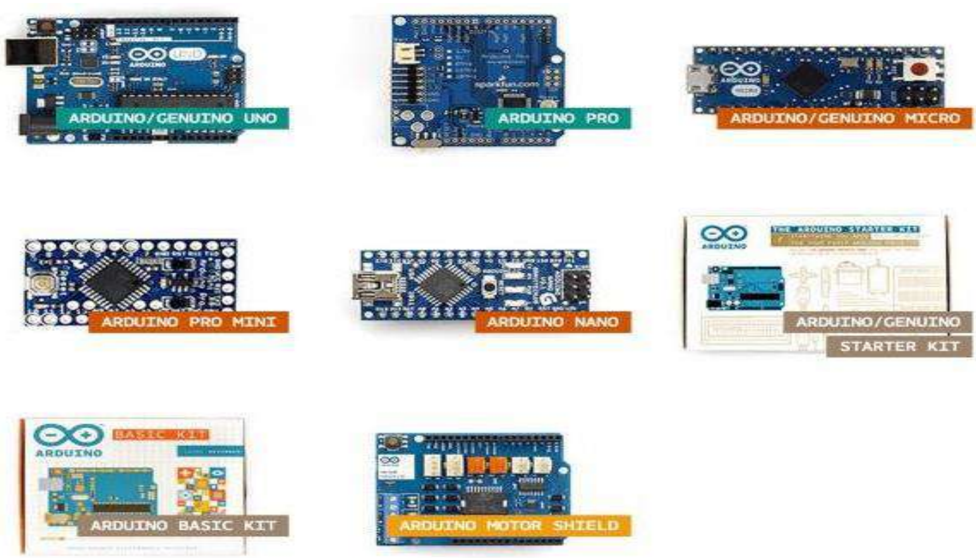


Fig.3.2.1.4 Arduino Family

3.2.1.5 SHIELDS

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

Fig.3.2.1.5 Arduino Shields

3.2.1.6 PIN DESCRIPTION OF ATMEGA328

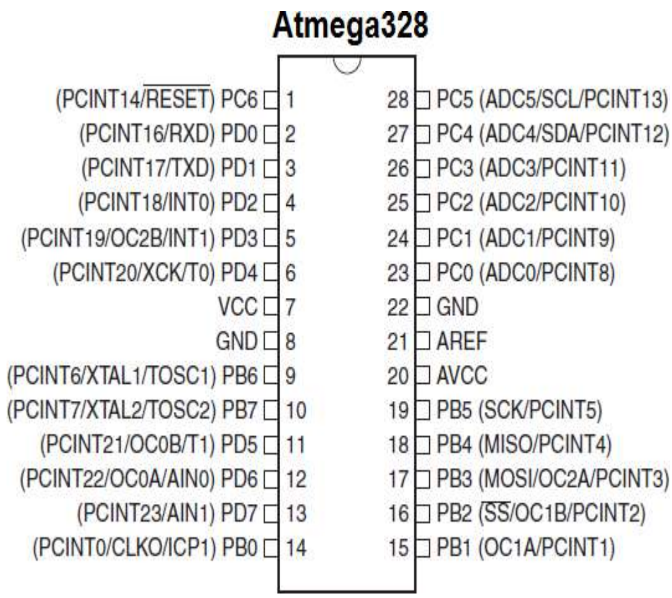


Fig.3.2.1.6 Pin description of ATMEGA328

3.2.1.7 APPLICATIONS

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

3.2.2 16*2 LCD SCREEN DISPLAY

3.2.2.1 INTRODUCTION

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

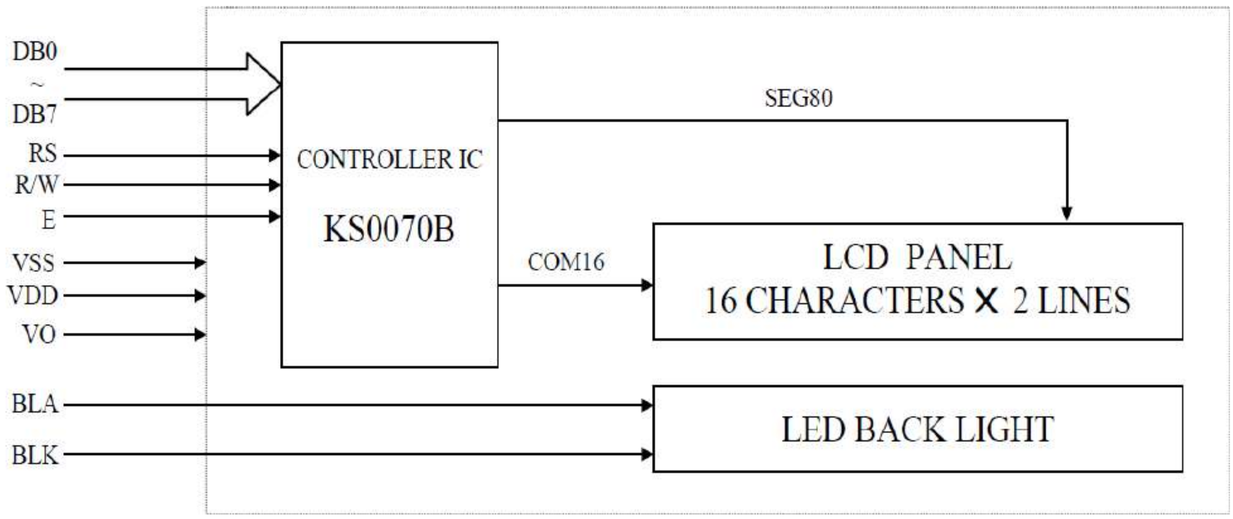


Fig.3.2.2.1 Lcd block diagram

3.2.2.2 DISPLAY DATA RAM:

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.

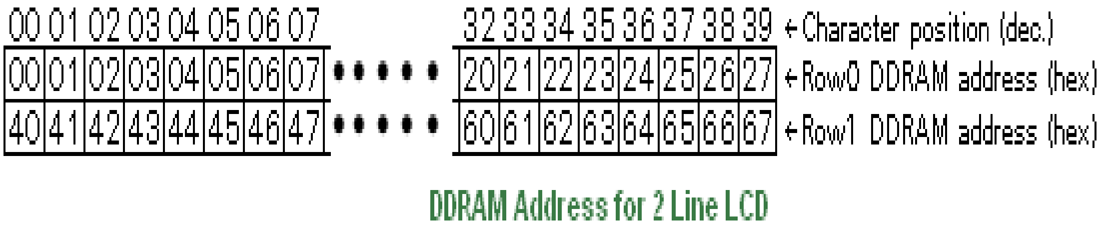


Fig.3.2.2.2 DDRAM addresses of 2 Line LCD

3.2.2.3 CHARACTER GENERATOR ROM

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM.

3.2.2.4 BUS FLAG

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e. BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

3.2.2.5 INSTRUCTION REGISTER (IR) AND DATA REGISTER (DR)

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g. LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD.

Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

3.2.2.6 16 X 2 ALPHANUMERIC LCD MODULE FEATURES:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

3.2.2.7 CIRCUIT DESCRIPTION

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore, by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore, we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result, we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly. The 2-line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

3.2.3 HOW TO CHOOSE A BUZZER

There are many kinds of buzzer to choose, first we need to know a few parameters, such as voltage, current, drive method, dimension, mounting type, and the most important thing is how much SPL and frequency we want.

Operating voltage: Normally, the operating voltage for a magnetic_buzzer is from 1.5V to 24V, for a piezo buzzer is from 3V to 220V. However, in order to get enough SPL, we suggest giving at least 9V to drive a piezo buzzer.

Consumption current: According to the different voltage, the consumption current of a magnetic buzzer is from dozens to hundreds of mill amperes; oppositely, the piezo type saves much more electricity, only needs a few mill amperes, and consumes three times current when the buzzer start to work.

Driving method: Both magnetic and piezo buzzer have self-drive type to choose. Because of the internal set drive circuit, the self-drive buzzer can emit sound if connecting with the direct current. Due to the different work principle, the magnetic buzzer need to be driven by 1/2 square waves, and the piezo buzzer need square waves to get better sound output.

Dimension: The dimension of the buzzer affects its SPL and the frequency, the dimension of the magnetic buzzer is from 7 mm to 25 mm; the piezo buzzer is from 12 mm to 50 mm, or even bigger.

Connecting way: Dip type, Wire type, SMD type, and screwed type for big piezo buzzer are usually seen.

Sound Pressure Level (SPL): Buzzer is usually tested the SPL at the distance of 10 cm, if distance double, the SPL will decay about 6 dB; oppositely, the SPL will increase 6 dB when the distance is shortened by one time. The SPL of the magnetic buzzer can reach to around 85 dB/ 10 cm; the piezo buzzer can be designed to emit very loud sound, for example, the common siren, are mostly made of piezo buzzer.

3.2.3.1 INTRODUCTION OF PIEZO BUZZER

Piezo buzzers are simple devices that can generate basic beeps and tones. They work by using a piezo crystal, a special material that changes shape when voltage is applied to it. If the crystal pushes against a diaphragm, like a tiny speaker cone, it can generate a pressure wave which the human ear picks up as sound. Simple change the frequency of the voltage sent to the piezo and it will start generating sounds by changing shape very quickly!

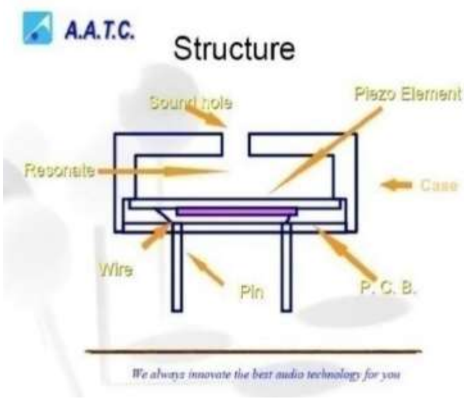


Fig.3.2.3.1 piezo buzzer

3.2.3.2 SPECIFICATIONS

Rated Voltage: A piezo buzzer is driven by square waves (Vp-p).
Operating Voltage: For normal operating. But it is not guaranteed to make the minimum SPL under the rated voltage.
Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.

Capacitance: A piezo buzzer can make higher SPL with higher capacitance, but it consumes more electricity.

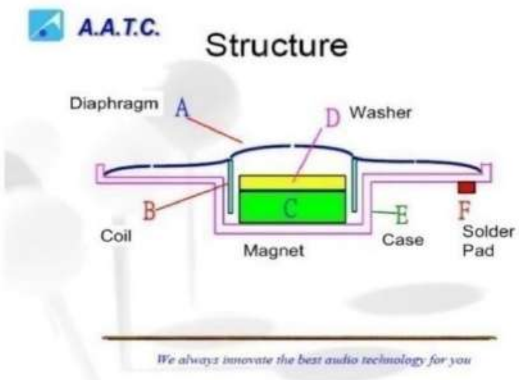


Fig. 3.2.3.2 structure of piezo buzzer

Sound Output: The sound output is measured by decibel meter.
Applying rated voltage and square waves, and the distance of 10 cm.
Rated Frequency: A buzzer can make sound on any frequencies, but we suggest that the highest

and the most stable SPL comes from the rated frequency.

Operating Temp.: Keep working well between -30℃ and +70℃.

3.2.4 TEMPERATURE SENSOR

Temperature Sensor which converts temperature value into electrical signals. We used IC called LM 35 as a temperature sensor. LM35 series sensors are precision integrated-circuit temperature sensors whose output voltage is linearly proportional to the Celsius temperature. The LM35 requires no external calibration since it is internally calibrated. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°Cover a full −55 to +150°C temperature range.

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteed (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±1⁄4°C typical
- Low impedance output, 0.1 W for 1 mA load

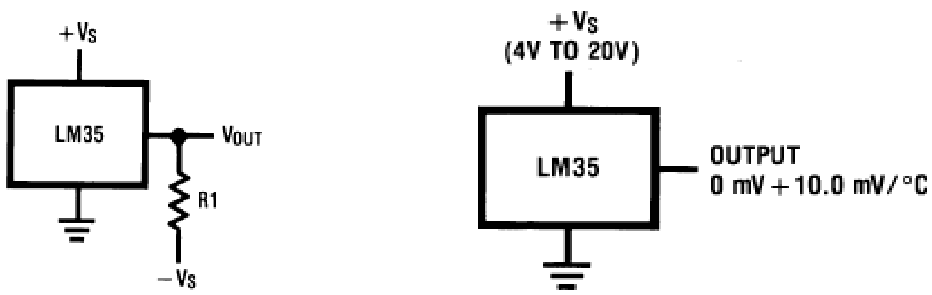


Figure. 3.2.4 Temperature Sensor

3.2.4.1 TEMPERATURE SENSING CIRCUIT

The methods of temperature measurement may be divided into two main classes according as the exchange of heat between the testing body and the hot system takes place by contact, or by radiation across a space. In the contact methods, thermometers or thermocouples are used and they are immersed in solids or liquids. The thermodynamic equilibrium between the hot body and the testing body is established by material contact. In the non-contact methods, the thermodynamic equilibrium is established by the radiation emitted as excited atom and molecules in the hot body return to the ground state.

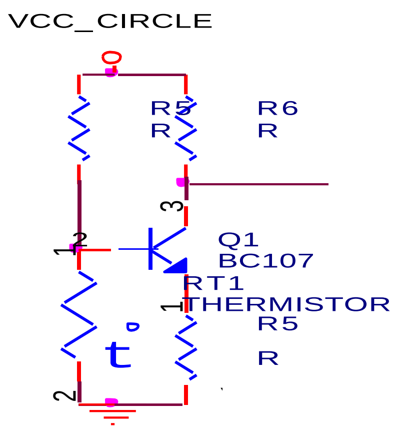


Figure. 3.2.4.1 Temperature Sensing Circuit

3.2.5 BATTERY

Batteries are a collection of one or more cells whose chemical reactions create a flow of electrons in a circuit. All batteries are made up of three basic components: an anode (the '-' side), a cathode (the '+' side), and electrolyte (a substance that chemically reacts with the anode and cathode).

When the anode and cathode of a battery is connected to a circuit, a chemical reaction takes place between the anode and the electrolyte. This reaction causes electrons to flow through the circuit and back into the cathode where another chemical reaction takes place. When the material in the cathode or anode is consumed or no longer able to be used in the reaction, the battery is unable to produce electricity. At that point, your battery is "dead."

Batteries that must be thrown away after use are known as **primary batteries**. Batteries that can be recharged are called **secondary batteries**.

Fig.3.2.5.1 Lithium Battery

Lithium polymer batteries, for example, can be recharged without batteries, your quadcopter would have to be tethered to the wall, you would have to hand crank your car, and your Xbox controller would have to plugged in all the time (like in the good old days). Batteries offer a way to store electrical potential energy in a portable container.

Fig.3.2.5.2 Batteries come in a variety of shapes, sizes, and chemistries

3.2.5.1 THE TERM BATTERY

Historically, the word "battery" was used to describe a "series of similar objects grouped together to perform a function," as in a battery of artillery. In 1749, Benjamin Franklin first used the term to describe a series of capacitors he had linked together for his electricity experiments. Later, the term would be used for any electrochemical cells linked together for the purpose of providing electric power.

Fig.3.2.5.1 The term battery

Battery of Leyden Jar "capacitors" linked together (Image courtesy of Alvinrune of Wikimedia Commons)

3.2.6 SOFTWARE EXPLANATION

- Arduino software
- Proteus simulation
-

3.2.6.1 ARDUINO SOFTWARE

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards can read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

3.2.6.2 WHY ARDUINO?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low-cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step-by-step instructions of a kit or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.Wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for

beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

- Open source and extensible software - The Arduino software is published as open-source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money
- Getting Started with Arduino and Genuino products: -

Install the Arduino Software (IDE) on Windows PCs-

This document explains how to install the Arduino Software (IDE) on Windows machines.

✓ Download the Arduino Software (IDE)

✓ Proceed with board specific instructions.

3.2.6.3 HOW TO DOWNLOAD THE ARDUINO SOFTWARE (IDE)

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

3.2.6.4 INSTALLATION

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.3.2.6.4.1 Standard USB Cable (A plug to B plug)

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig. 3.2.6.4.2 Standard USB Cable (an A to Mini-B plug)

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

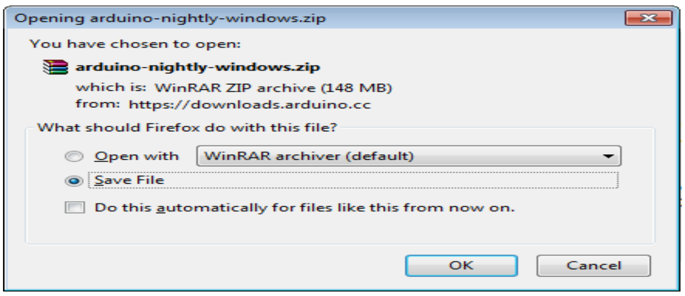


Fig. 3.2.6.4.3 Download Arduino IDE Software

Step 3 − Power up your board

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

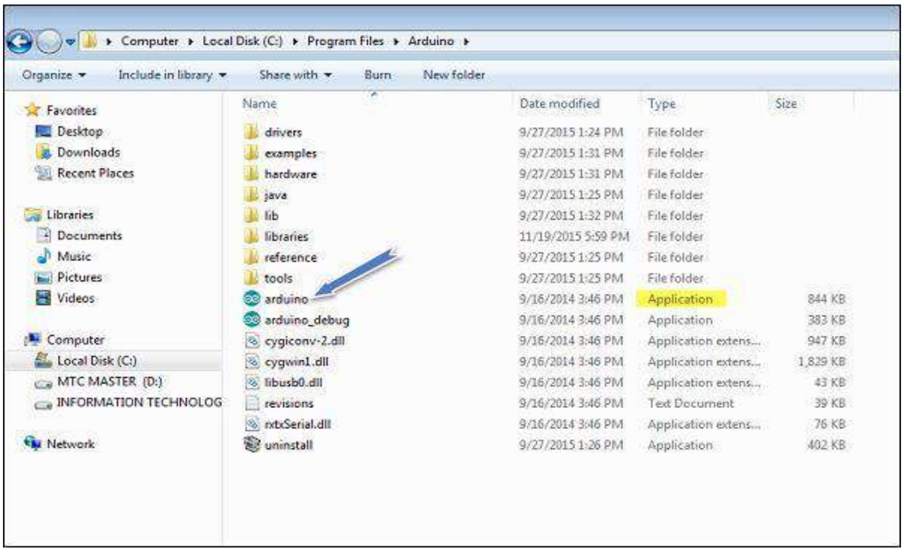


Fig. 3.2.6.4.4 Launch Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**

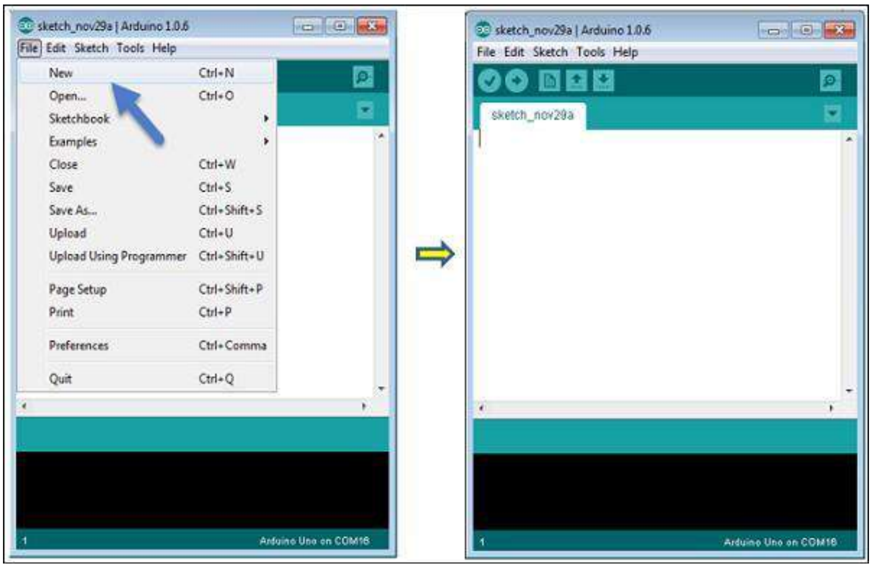


Fig. 3.2.6.4.5 Open your first project.

To open an existing project example, select File → Example → Basics → Blink.

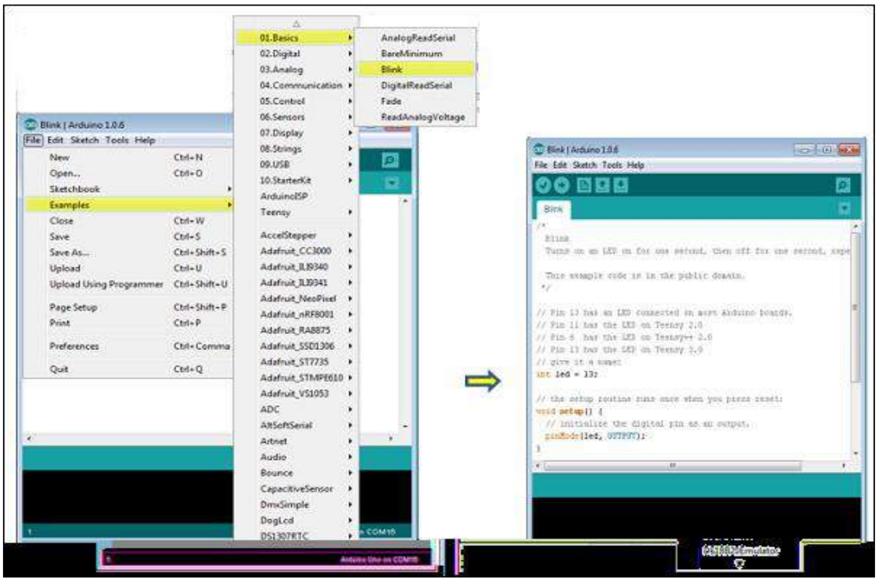


Fig. 3.2.6.4.6 Open your first program.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

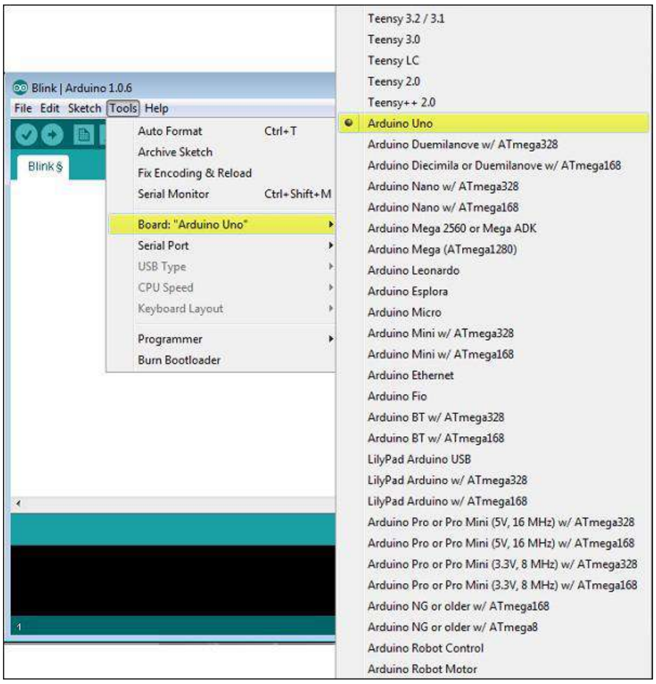


Fig. 3.2.6.4.7 Select your Arduino board.

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

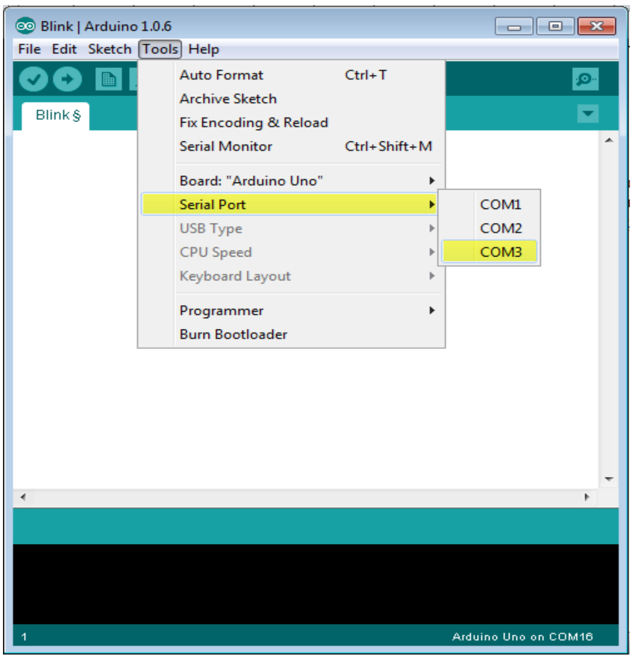


Fig. 3.2.6.4.8 Select your serial port.

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

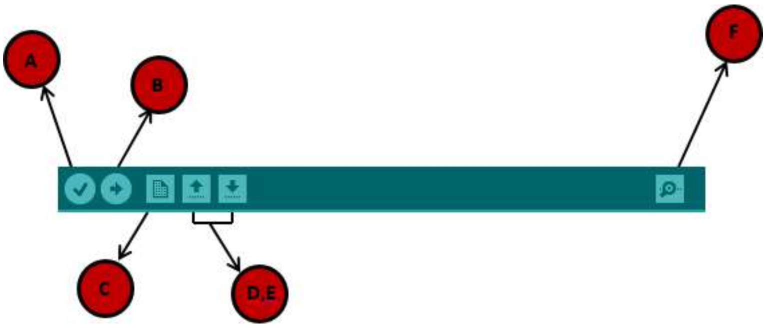


Fig. 3.2.6.4.9 Upload the program to your board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketches.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from board & send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

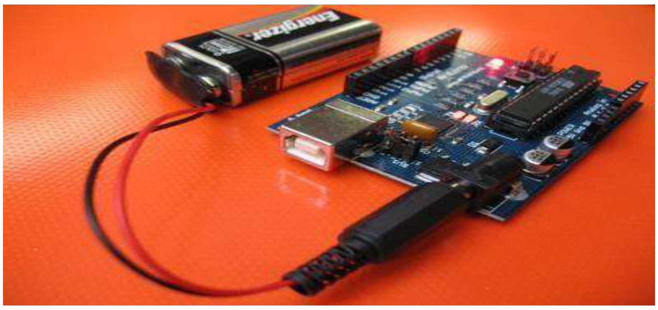


Fig. 3.2.6.4.10 Connecting a Battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called "sketches", but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

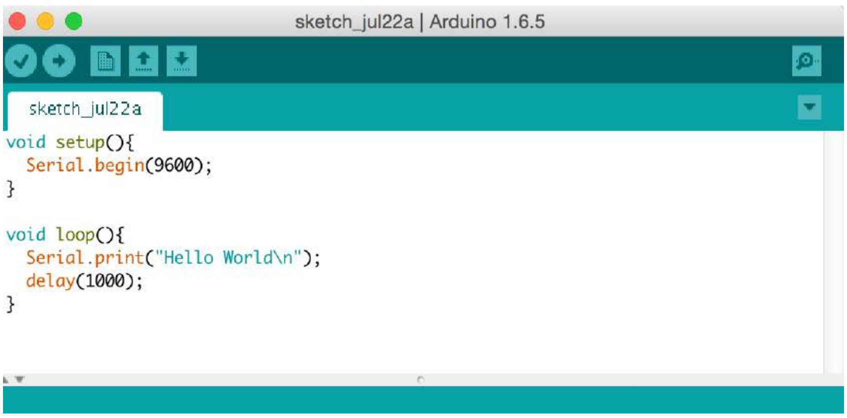


Fig. 3.2.6.4.11 Example Program

3.2.7 PROTEUS

PROTEUS

Proteus is a simulation and design software tool developed by Lab center Electronics for Electrical and Electronic circuit design. It also possesses 2D CAD drawing feature. It deserves to bear the tagline "From concept to completion".

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the footprints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design:

Step 1: Open ISIS software and select new design in File menu

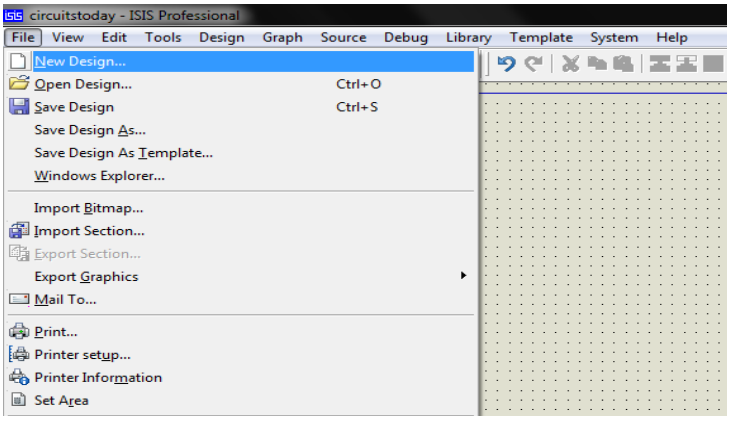


Fig. 3.2.7.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

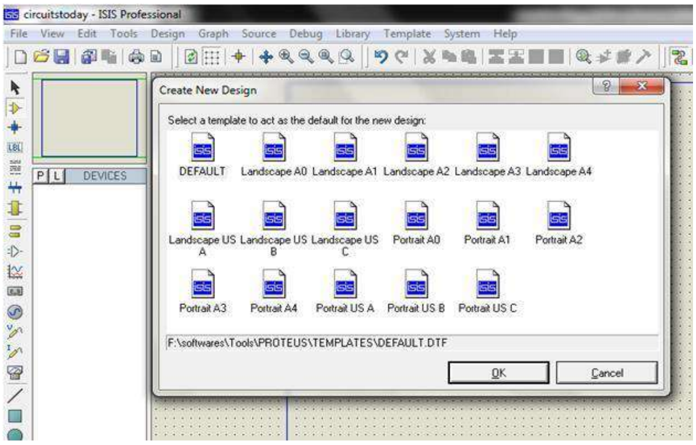


Fig. 3.2.7.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

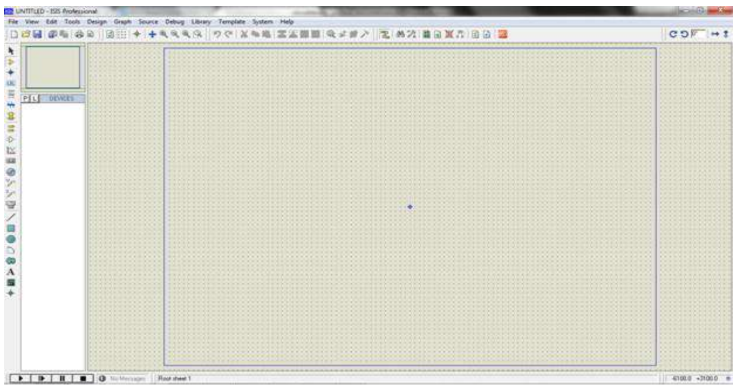


Fig. 3.2.7.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

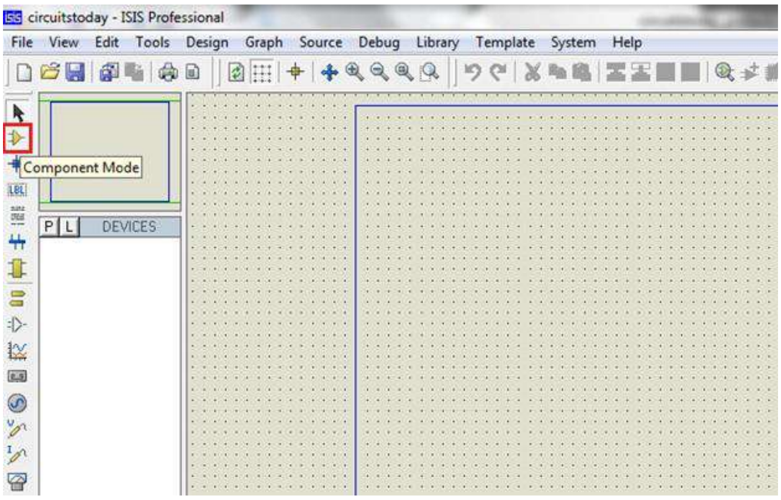


Fig. 3.2.7.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

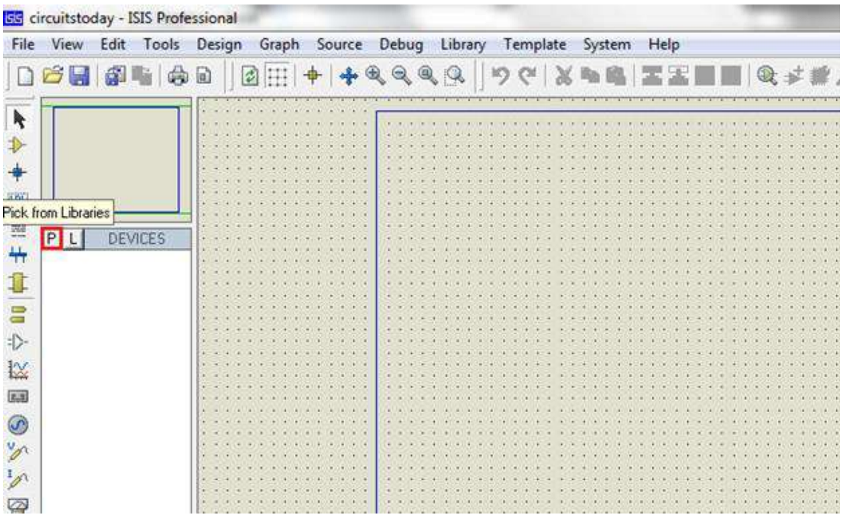


Fig. 3.2.7.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keyword's text box.

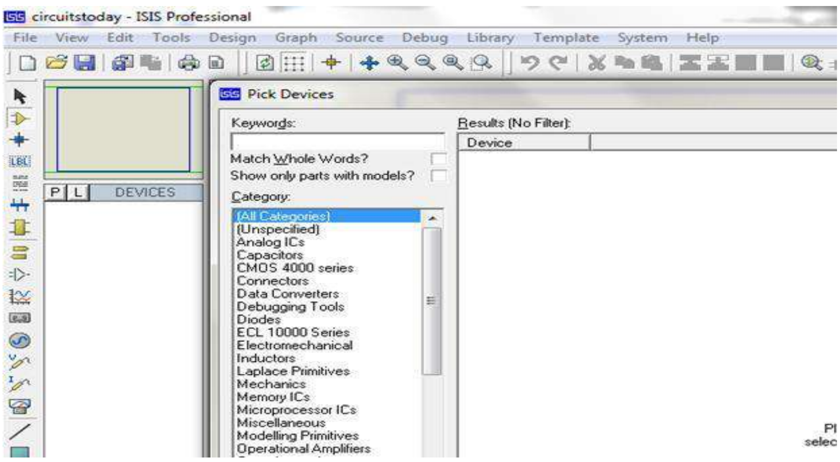


Fig. 3.2.7.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

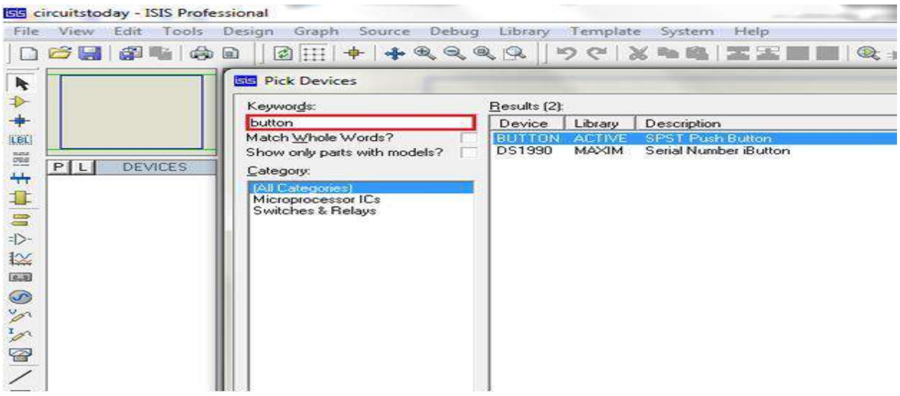


Fig. 3.2.7.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

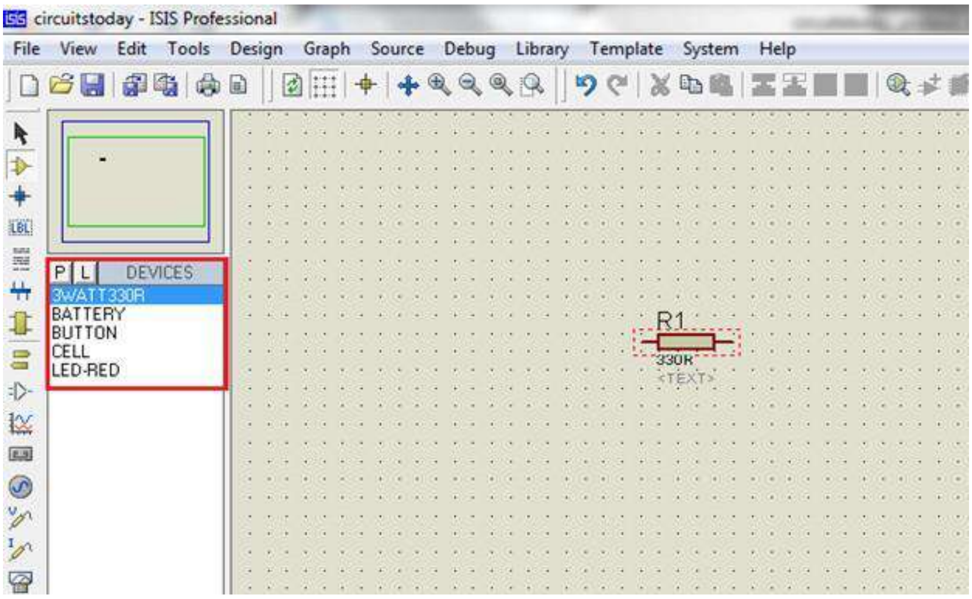


Fig. 3.2.7.8 Component Selection

Place all the required components and route the wires i.e., make connections. Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

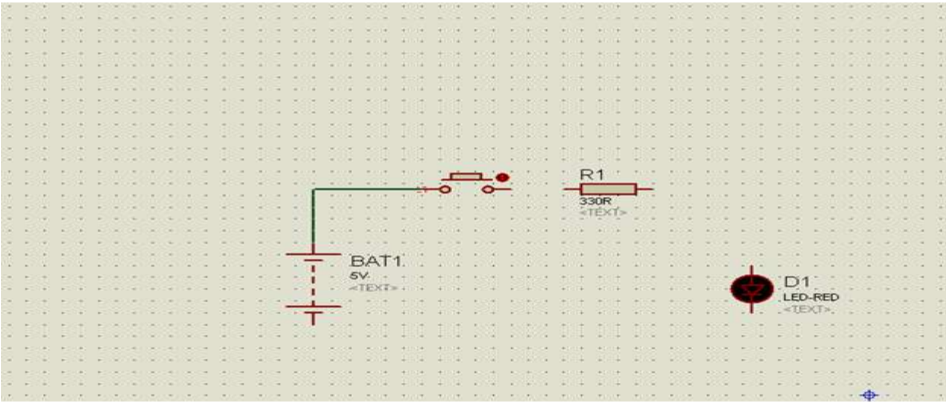


Fig. 3.2.7.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

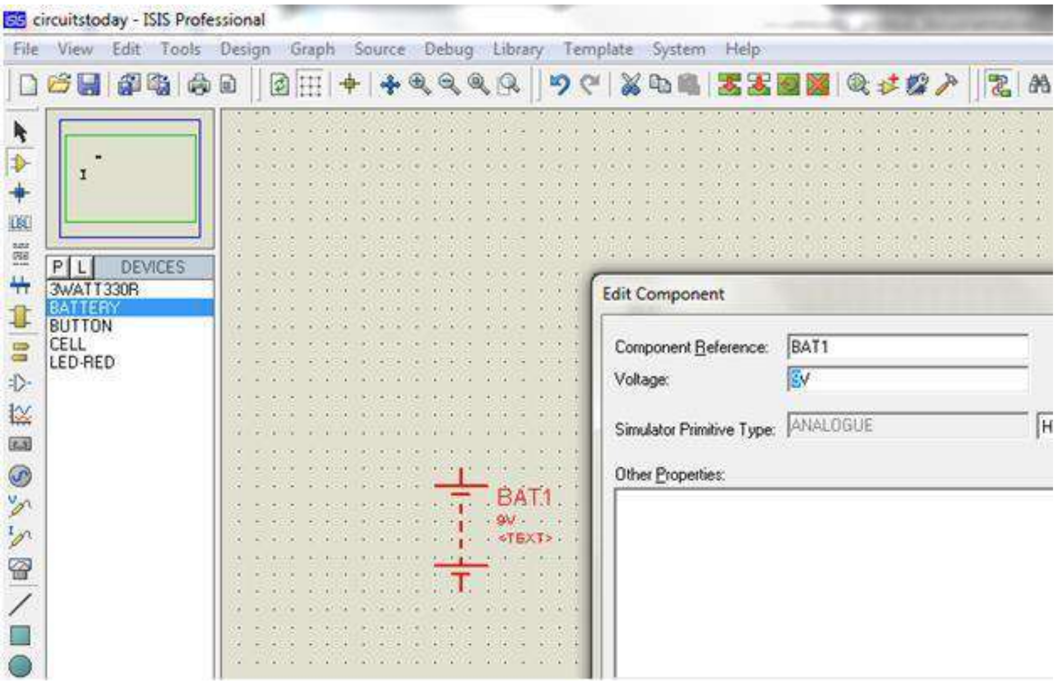


Fig. 3.2.7.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

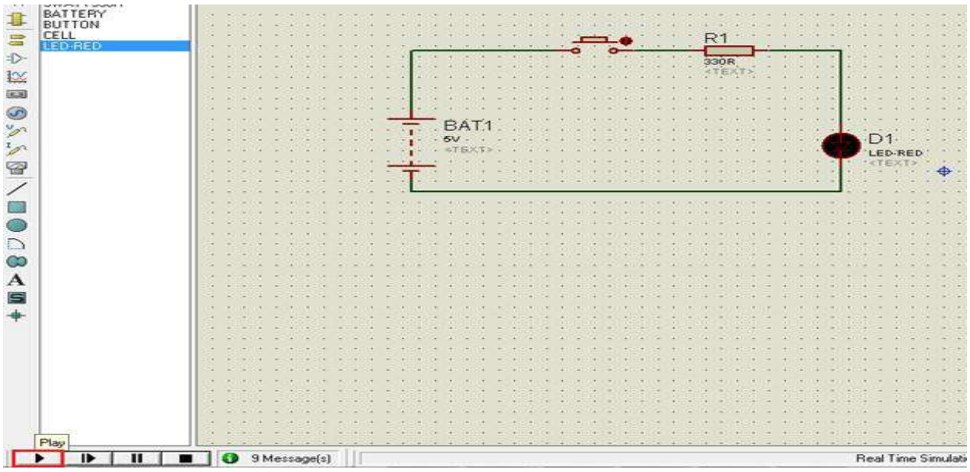


Fig. 3.2.7.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

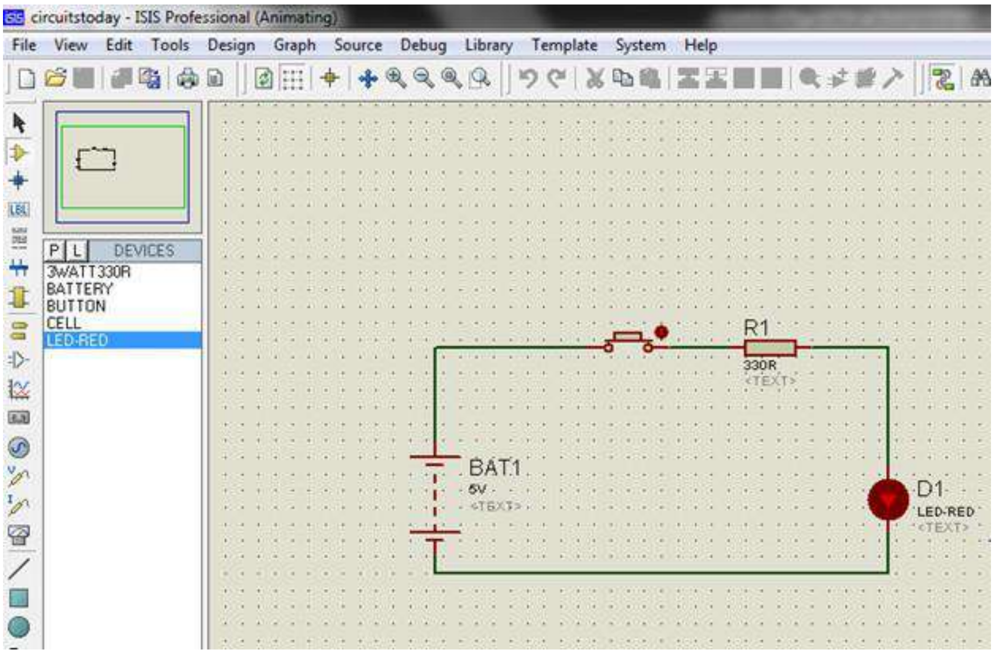


Fig. 3.2.7.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

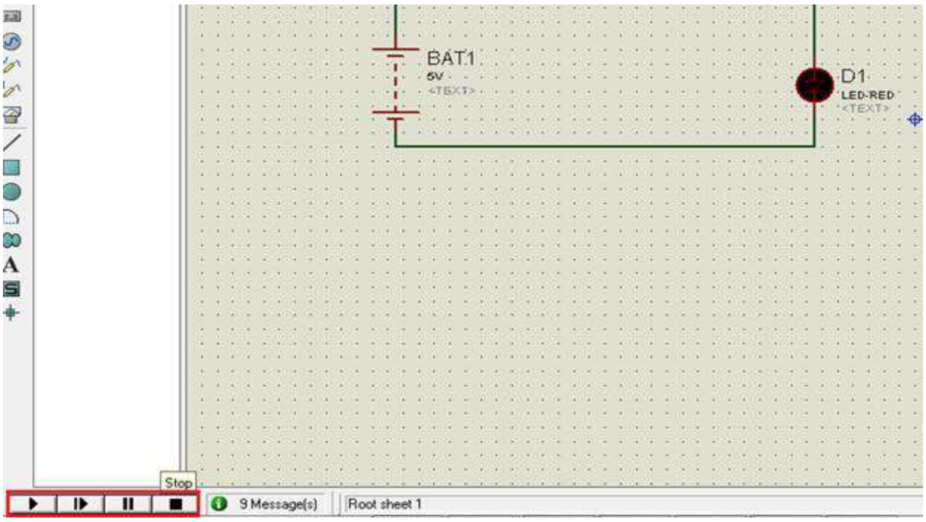


Fig. 3.2.7.13 Simulation Step-Pause-Stop Buttons

According to its developers, Things Speak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. Things Speak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates.

Things Speak was originally launched by IOT Bridge in 2010 as a service in support of IoT applications.

Things Speak has integrated support from the numerical computing software MATLAB from MathWorks, allowing Things Speak users to analyse and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license from Math works. [citation needed].

Things Speak has a close relationship with Math works, Inc. In fact, all of the Things Speak documentation is incorporated into the Math works' MATLAB documentation site and even enabling registered Math works user accounts as valid login credentials on the Things Speak website. The terms of service and privacy policy of ThingSpeak.com are between the agreeing user and Math works, Inc.

Setup Things Speak

Create a free MathWorks account or sign into Things Speak using an existing account.

Select the Things Speak channel you want your data to stream into. See Collect Data in a New Channel for help creating a new channel.

Record the following for the selected channel:

Channel ID, which is listed at the top of the channel view.

Write API key, which can be found on the API Keys tab of your channel view.

3.3 DEFINE THE MODULES

3.3.1 GLOBAL SYSTEM FOR MOBILE COMMUNICATION (GSM)

GSM (Global System for Mobile communications) is an open, digital cellular technology used for transmitting mobile voice and data services. GSM (Global System for Mobile communication) is a digital mobile telephone system that is widely used in Europe and other parts of the world. GSM uses a variation of Time Division Multiple Access (TDMA) and is the most widely used of the three digital wireless telephone technologies (TDMA, GSM, and CDMA). GSM digitizes and compresses data, then sends it down a channel with two other streams of user data, each in its own time slot. It operates at either the 900 MHz or 1,800 MHz frequency band. It supports voice calls and data transfer speeds of up to 9.6 kbit/s, together with the transmission of SMS (Short Message Service).

3.3.1.1 INTRODUCTION TO GSM

The **GSM** (*Global System for Mobile communications*) network is at the start of the 21^{st} century, the most used mobile telephony standard in Europe. It is called as Second Generation

(2G) standard because communications occur in an entirely digital mode, unlike the first generation of portable telephones. When it was first standardized in 1982, it was called as **Group Special Mobile** and later, it became an international standard called **"Global System for Mobile communications"** in 1991.

In Europe, the GSM standard uses the 900 MHz and 1800 MHz frequency bands. In the United States, however, the frequency band used is the 1900 MHz band. For this reason, portable telephones that can operate in both Europe and the United States are called **tri-band** while those that operate only in Europe are called **bi-band**.

The GSM standard allows a maximum throughput of 9.6 kbps which allows transmission of voice and low-volume digital data like text messages (**SMS**, for *Short Message Service*) or multimedia messages (**MMS**, for *Multimedia Message Service*).

3.3.1.2 GSM STANDARDS

GSM uses narrowband TDMA, which allows eight simultaneous calls on the same radio frequency. There are three basic principles in multiple access, FDMA (Frequency Division Multiple Access), TDMA (Time Division Multiple Access), and CDMA (Code Division Multiple Access). All three principles allow multiple users to share the same physical channel. But the two competing technologies differ in the way user sharing the common resource.

TDMA allows the users to share the same frequency channel by dividing the signal into different time slots. Each user takes turn in a round robin fashion for transmitting and receiving over the channel. Here, users can only transmit in their respective time slot.

CDMA uses a spread spectrum technology that is it spreads the information contained in a particular signal of interest over a much greater bandwidth than the original signal. Unlike TDMA, in CDMA several users can transmit over the channel at the same time.

3.3.1.3 TDMA IN BRIEF

In late1980's, as a search to convert the existing analog network to digital to improve capacity, the cellular telecommunications industry association chose TDMA over FDMA. Time Division Multiple Access is a type of multiplexing where two or more channels of information are transmitted over the same link by allocating a different time interval for the transmission of each channel. The most complex implementation using TDMA principle is of GSM's (Global System for Mobile communication). To reduce the effect of co-channel interference, fading and multipath, the GSM technology can use frequency hopping, where a call jumps from one channel to another channel in a short interval.

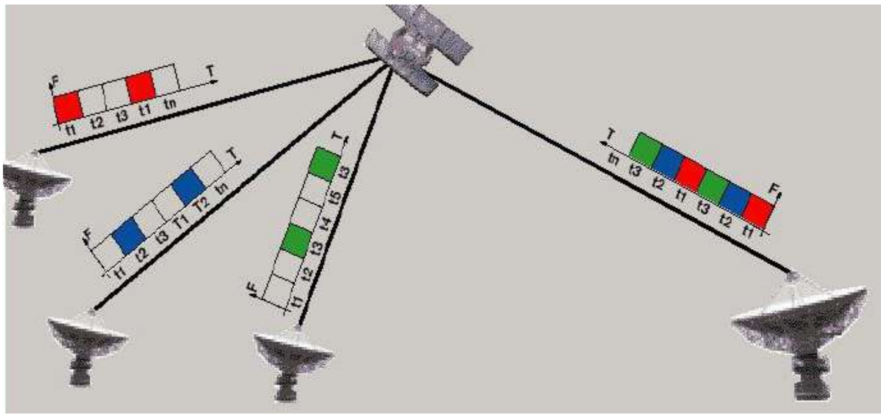


Fig.3.3.1.3 Time Division Multiplexing Access

TDMA systems still rely on switch to determine when to perform a handoff. Handoff occurs when a call is switched from one cell site to another while travelling. The TDMA handset constantly monitors the signals coming from other sites and reports it to the switch without caller's awareness. The switch then uses this information for making better choices for handoff at appropriate times. TDMA handset performs hard handoff, i.e., whenever the user moves from one site to another, it breaks the connection and then provides a new connection with the new site.

3.3.1.4 THE CONCEPT OF CELLULAR NETWORK

Mobile telephone networks are based on the concept of **cells**, circular zones that overlap to cover a geographical area.

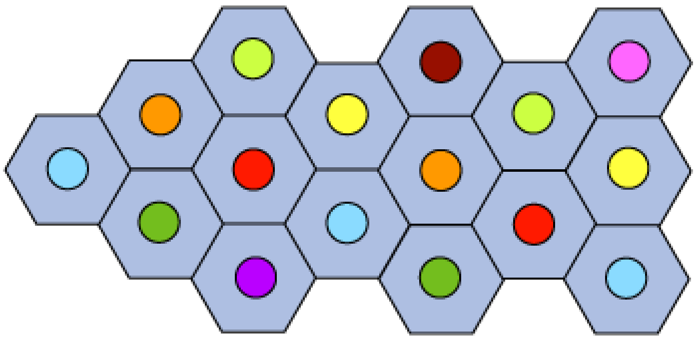


Fig.3.3.1.4 The Concept of cellular network

Cellular networks are based on the use of a central transmitter-receiver in each cell, called a "**base station**" (or *Base Transceiver Station*, written **BTS**). The smaller the radius of a cell, the higher is the available bandwidth. So, in highly populated urban areas, there are cells with a radius of a few hundred meters, while huge cells of up to 30 kilometers provide coverage in rural areas.

In a cellular network, each cell is surrounded by 6 neighboring cells (thus a cell is generally drawn as a hexagon). To avoid interference, adjacent cells cannot use the same frequency. In practice, two cells using the same frequency range must be separated by two to three times the diameter of the cell.

3.3.1.5 GSM MODEM

A GSM modem is a wireless modem that works with a GSM wireless network. A wireless modem behaves like a dial-up modem. The main difference between them is that a dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves.

Fig.3.3.1.5.1 GSM/GPRS Modem

A GSM modem can be an external device or a PC Card / PCMCIA Card. Typically, an external GSM modem is connected to a computer through a serial cable or a USB cable. A GSM modem in the form of a PC Card / PCMCIA Card is designed for use with a laptop computer. It should be inserted into one of the PC Card / PCMCIA Card slots of a laptop computer. Like a GSM mobile phone, a GSM modem requires a SIM card from a wireless carrier in order to operate.

A SIM card contains the following information:

- Subscriber telephone number (MSISDN)
- International subscriber number (**IMSI, International Mobile Subscriber Identity)**
- State of the SIM card
- Service code (operator)
- Authentication key
- PIN (***Personal Identification Code***)
- PUK (***Personal Unlock Code***)

Computers use AT commands to control modems. Both GSM modems and dial-up modems support a common set of standard AT commands. In addition to the standard AT commands, GSM modems support an extended set of AT commands. These extended AT commands are defined in the GSM standards. With the extended AT commands, the following operations can be performed:

- Reading, writing, and deleting SMS messages.
- Sending SMS messages.
- Monitoring the signal strength.
- Monitoring the charging status and charge level of the battery.
- Reading, writing, and searching phone book entries.

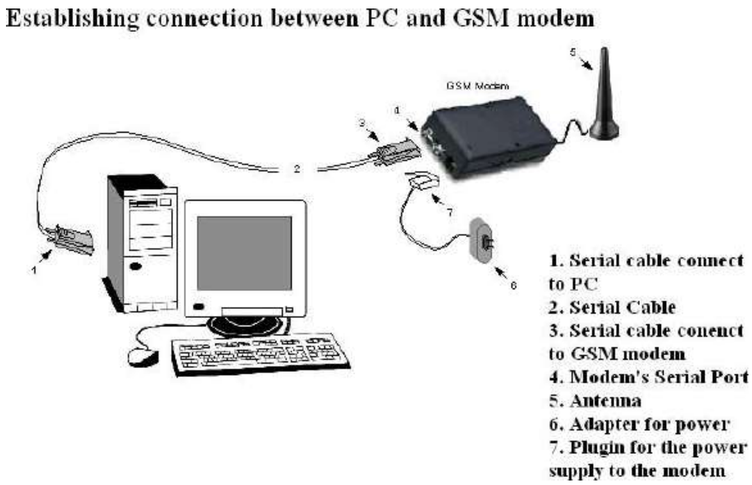


Fig.3.3.1.5.2 Establishing connection between PC and GSM modem

The number of SMS messages that can be processed by a GSM modem per minute is very low i.e., about 6 to 10 SMS messages per minute.

3.3.1.6 INRODUCTION TO AT COMMANDS

AT commands are instructions used to control a modem. AT is the abbreviation of Attention. Every command line starts with "AT" or "at". That is the reason, modem commands are called AT commands. Many of the commands that are used to control wired dial-up modems, such as ATD (Dial), ATA (Answer), ATH (Hook control) and ATO (Return to online data state) are also supported by GSM modems and mobile phones.

Besides this common AT command set, GSM modems and mobile phones support an AT command set that is specific to the GSM technology, which includes SMS-related commands like AT+CMGS (Send SMS message), AT+CMSS (Send SMS message from storage), AT+CMGL (List SMS messages) and AT+CMGR (Read SMS messages).

It should be noted that the starting "AT" is the prefix that informs the modem about the start of a command line. It is not part of the AT command name. For example, D is the actual AT command name in ATD and +CMGS is the actual AT command name in AT+CMGS.

Some of the tasks that can be done using AT commands with a GSM modem or mobile phone are listed below:

Get basic information about the mobile phone or GSM modem. For example, name of manufacturer (AT+CGMI), model number (AT+CGMM), IMEI number (International Mobile Equipment Identity) (AT+CGSN) and software version (AT+CGMR).

Get basic information about the subscriber. For example, MSISDN (AT+CNUM) and IMSI number (International Mobile Subscriber Identity) (AT+CIMI).

Get the status of the mobile phone or GSM/GPRS modem. For example, mobile phone activity status (AT+CPAS), mobile network registration status (AT+CREG), radio signal strength (AT+CSQ), battery charge level and battery charging status (AT+CBC).

Establish a data connection or voice connection to a remote modem (ATD, ATA, etc.).

Send and receive fax (ATD, ATA, AT+F*).

Send (AT+CMGS, AT+CMSS), read (AT+CMGR, AT+CMGL), write (AT+CMGW) or delete (AT+CMGD) SMS messages and obtain notifications of newly received SMS messages (AT+CNMI).

Read (AT+CPBR), write (AT+CPBW) or search (AT+CPBF) phonebook entries.

Perform security-related tasks, such as opening or closing facility locks (AT+CLCK), checking whether a facility is locked (AT+CLCK) and changing passwords (AT+CPWD). (Facility lock examples: SIM lock [a password must be given to the SIM card every time the mobile phone is switched on] and PH-SIM lock [a certain SIM card is associated with the mobile phone. To use other SIM cards with the mobile phone, a password must be entered.])

Control the presentation of result codes / error messages of AT commands. For example, the user can control whether to enable certain error messages (AT+CMEE) and whether error messages should be displayed in numeric format or verbose format (AT+CMEE=1 or AT+CMEE=2).

Get or change the configurations of the mobile phone or GSM/GPRS modem. For example, change the GSM network (AT+COPS), bearer service type (AT+CBST), radio link protocol parameters (AT+CRLP), SMS center address (AT+CSCA) and storage of SMS messages (AT+CPMS).

Save and restore configurations of the mobile phone or GSM/GPRS modem. For example, save (AT+CSAS) and restore (AT+CRES) settings related to SMS messaging such as the SMS center address.

It should be noted that the mobile phone manufacturers usually do not implement all AT commands, command parameters and parameter values in their mobile phones. Also, the behavior of the implemented AT commands may be different from that defined in the standard. In general, GSM modems, designed for wireless applications, have better support of AT commands than ordinary mobile phones.

3.3.2 GLOBAL POSITIONING SYSTEM

3.3.2.1 INTRODUCYION TO GPS

Fig 3.3.2.1 Interface ublox NEO-6M GPS Module with Arduino

Give your next Arduino project ability to sense locations with NEO-6M GPS Module that can track up to 22 satellites and identifies locations anywhere in the world. It may serve as a great launch pad for anyone looking to get into the world of GPS.

They are low power (suitable for battery powered devices), inexpensive, easy to interface with and are insanely popular among hobbyists.

3.3.2.2 HARDWARE OVERVIEW OF NEO-6M GPS MODULE

NEO-6M GPS Chip

At the heart of the module is a NEO-6M GPS chip from u-blox. The chip measures less than the size of a postage stamp but packs a surprising number of features into its little frame.

Fig.3.3.2.2 Overview of NEO-6M GPS Module

It can track up to 22 satellites on 50 channels and achieves the industry's highest level of sensitivity i.e., -161 dB tracking, while consuming only 45mA supply current.

Unlike other GPS modules, it can do up to 5 location updates a second with 2.5m Horizontal position accuracy. The u-blox 6 positioning engine also boasts a Time-To-First Fix (TTFF) of under 1 second.

One of the best features the chip provides is Power Save Mode (PSM). It allows a reduction in system power consumption by selectively switching parts of the receiver ON and OFF. This dramatically reduces power consumption of the module to just 11mA making it suitable for power sensitive applications like GPS wristwatch.

3.3.2.3 POSITION FIX LED INDICATOR

The necessary data pins of NEO-6M GPS chip are broken out to 0.1″ pitch headers. This includes pins required for communication with a microcontroller over UART. The module supports baud rate from 4800bps to 230400bps with default baud of 9600.

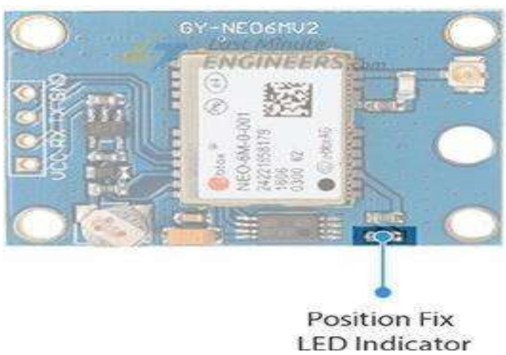


Fig.3.3.2.3 Position Fix LED Indicator

There is an LED on the NEO-6M GPS Module which indicates the status of Position Fix. It will blink at various rates depending on what state it's in:

- No Blinking – It is searching for satellites.
- Blink every 1s – Position Fix is found (The module can see enough satellites).

3.3.2.4 3.3V LDO REGULATOR

Fig.3.3.2.4 3.3V LDO Regulator

The operating voltage of the NEO-6M chip is from 2.7 to 3.6V. But the good news is that the module comes with MIC5205 ultra-low dropout 3V3 regulator from MICREL.

The logic pins are also 5-volt tolerant, so we can easily connect it to an Arduino or any 5V logic microcontroller without using any logic level converter.

3.3.2.5 BATTERY & EEPROM

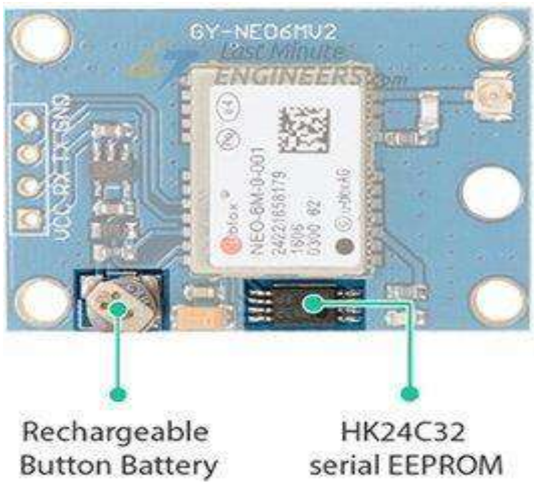


Fig. 3.3.2.5 Battery & EEPROM

The module is equipped with an HK24C32 two wire serial EEPROM. It is 4KB in size and connected to the NEO-6M chip via I2C.

The module also contains a rechargeable button battery which acts as a super-capacitor.

An EEPROM together with battery helps retain the battery backed RAM (BBR). The BBR contains clock data, latest position data (GNSS orbit data) and module configuration. But it’s not meant for permanent data storage.

As the battery retains clock and last position, time to first fix (TTFF) significantly reduces to 1s. This allows much faster position locks.

Without the battery the GPS always cold start so the initial GPS lock takes more time.

The battery is automatically charged when power is applied and maintains data for up to two weeks without power.

3.3.2.6 ANTENNA

An antenna is required to use the module for any kind of communication. So, the module comes with a patch antenna having -161 dBm sensitivity.

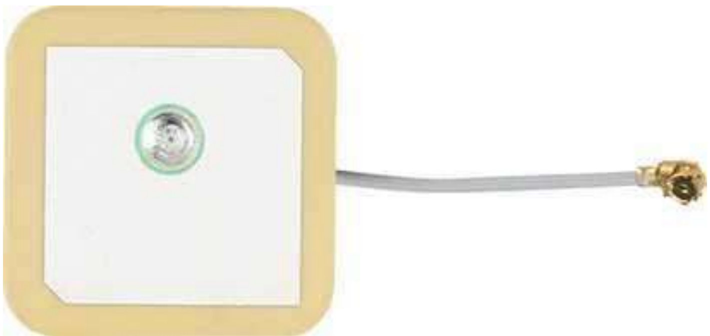


Fig. 3.3.2.6 Antenna

3.3.2.7 UFL CONNECTOR

You can snap-fit this antenna to small U.FL connector located on the module.

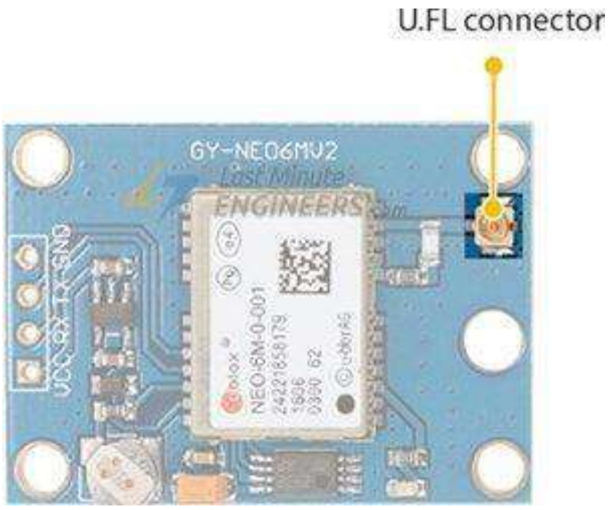


Fig.3.3.2.7 UFL Connector

Patch antenna is great for most projects. But if you want to achieve more sensitivity or put your module inside a metal case, you can also snap on any 3V active GPS antenna via the U.FL connector.

3.3.2.8 NEO-6M GPS MODULE PINOUT

The NEO-6M GPS module has total 4 pins that interface it to the outside world. The connections are as follows:

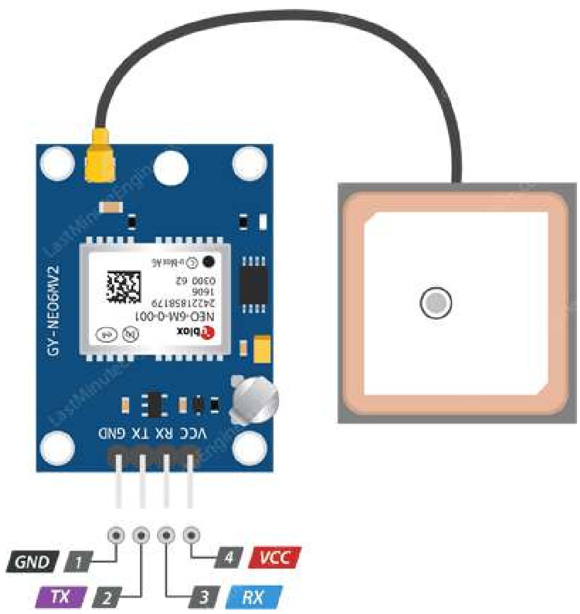


Fig.3.3.2.8 NEO-6M GPS MODULE PINOUT

GND is the Ground Pin and needs to be connected to GND pin on the Arduino.

TxD (Transmitter) pin is used for serial communication.

RxD (Receiver) pin is used for serial communication.

VCC supplies power for the module. You can directly connect it to the 5V pin on the Aíduino.

Wiring NEO-6M GPS module with Arduino UNO

Now that we know everything about the module, we can begin hooking it up to our Arduino!

Start by connecting the patch antenna to the U.FL connector. Remember to thread the U.FL cable through one of the mounting holes for robust connection.

The module usually comes with header pins unsoldered. So, you'll need to solder them.

Now, connect Tx and Rx pin on module to digital pin#2 and #3 respectively on Arduino; as we'll be using software serial to talk to the module.

Next, connect VCC pin to the 5V pin on the Arduino and GND to ground.

CHAPTER 4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM OF PROJECT

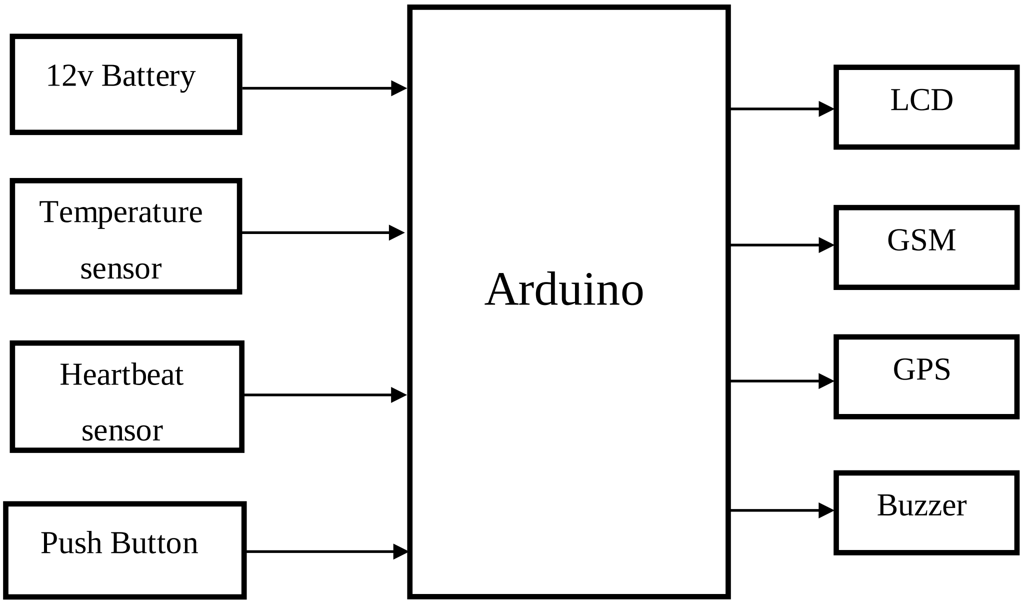


Fig.4.1 Block diagram of the Project.

4.2 FLOW CHART

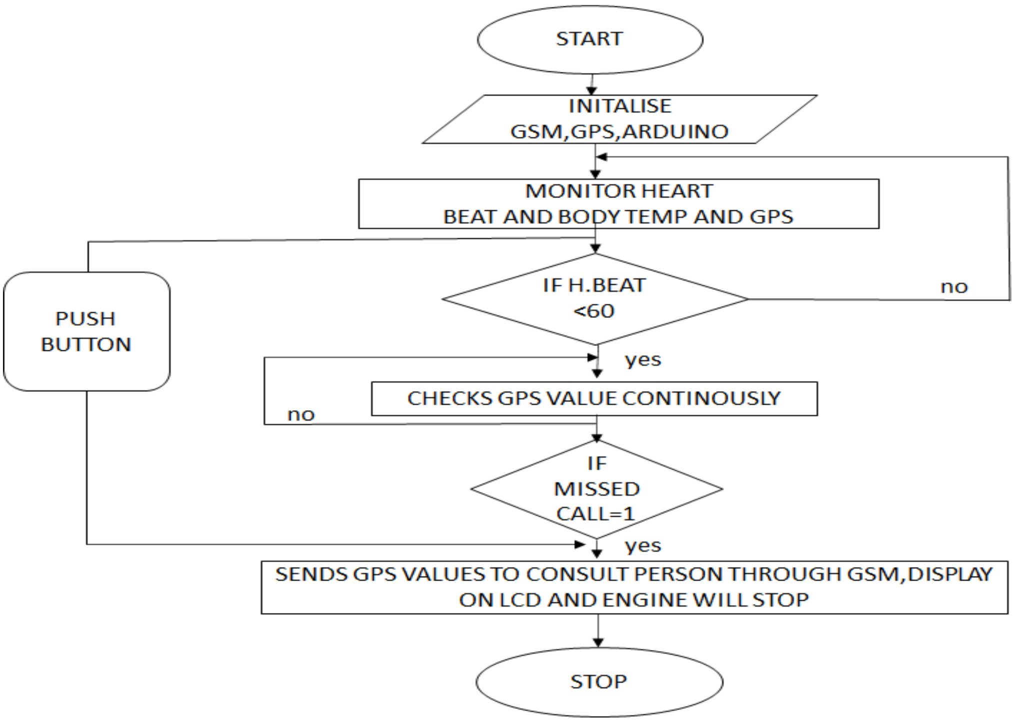


Fig.4.2 Flow Chart

4.3 IMPLEMENTATION STAGES

Stage-1: First on lcd we will get a message displaying “Welcome”.

Fig.4.3.1 Welcome Message

Step 2: It displaying biomedical sensors output and status of switch.

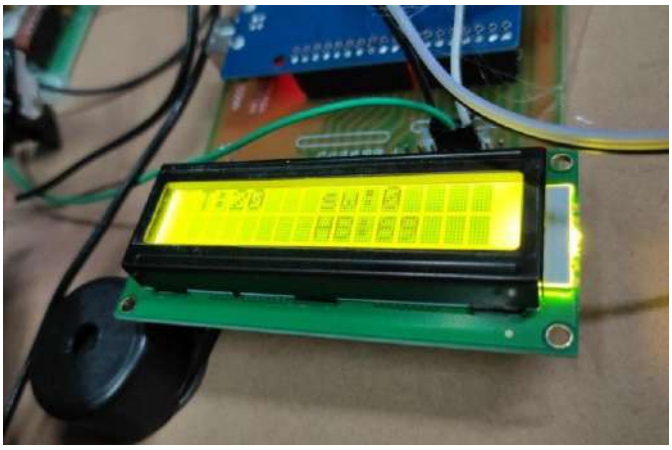


Fig.4.3.2 Biomedical sensors output and status of switch.

Step 3: Lcd displaying high temperature when threshold value is stricken.

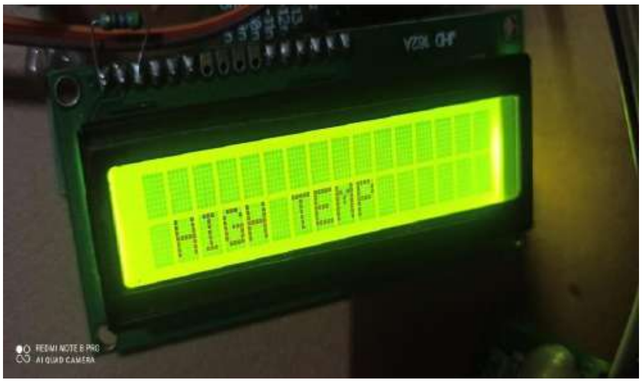


Fig.4.3.3 Lcd displaying high temperature

4.4 PROJECT CODE

```
#include <SoftwareSerial.h>
//SoftwareSerial mySerial(2,3);
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2,3);//rx , tx
#include <LiquidCrystal.h>
LiquidCrystal lcd (13,12,11,10,9,8);

unsigned int hbpin = 4;
int hcount = 0, hbval = 0;
int temp=A4;
int buz=7;
int sw=A5;
void serialFlush(){
while (Serial.available() > 0) {
char tt = Serial.read();
 }
}
void sendmsg(char *num,char * msg) //gsm
{
mySerial.print("AT+CMGS=\"");
mySerial.print(num);
mySerial.println("\""); delay (800);
mySerial.println(msg); delay (800);
mySerial.write(0x1a); delay (2000);
}

const char* number = "0123456789\0";/*Give the required mobile num to send the msg*/


char t;
```

```
 void setup ()
{
 char ret;
 pinMode(hbpin,  INPUT);
 pinMode(temp,   INPUT);
 pinMode(buz, OUTPUT);
 pinMode(sw, INPUT);

 mySerial.begin(9600);
 Serial.begin(9600);
 lcd.begin(16,2);
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("Welcome");
 Serial.println("WELCOME1");


 mySerial.println("AT"); delay (1000);
 mySerial.println("AT+CMGF=1"); delay (1000);
 mySerial.println("AT+CNMI=1,2,0,0"); delay (1000);
 mySerial.println("AT+CSMP=17,167,0,16"); delay (1000);
}

 char buff [200], buff2[100];
 char bf2[50];

  unsigned long duration = 0;
  int hbeat = 0;

 void loop ()
 {

 Serial.print("temperature = ");
 int td = analogRead(temp)/3;
 Serial.print(td);

 duration = pulseIn(hbpin, LOW, 5000000) / 1000;
 if (duration == 0)
 hbeat = 0;
 else
 hbeat = 64 + duration % 18;
 delay (1000);
 lcd.clear();
 lcd.setCursor(0, 1);
 hbval = hbeat;
 Serial.println(hbval);
```

```
 lcd.clear();
 //lcd.setCursor(0, 0);lcd.print(" T:");lcd.print(temp);
 lcd.setCursor(7, 0);lcd.print(" sw:");lcd.print(sw);
 lcd.setCursor(8, 1); lcd.print("HB:"); lcd.print(hbval); //lcd.print(" ");
delay (1000);



if (hbval > 78)
   {
     lcd.clear(); lcd.setCursor(0, 1); lcd.print("HIGH HEARTBEAT ");
     sendmsg(number,"HIGH HEARTBEAT ");
     digitalWrite(buz,LOW);
     delay (1000);
   }

if (hbval>10 &&hbval < 60)
   {
    lcd.clear(); lcd.setCursor(0, 1); lcd.print("LOW HEARTBEAT  ");
    sendmsg(number,"LOW HEARTBEAT ");
    digitalWrite(buz,LOW);
     delay (1000);
   }

   if(temp>35)
  {
   lcd.clear();lcd.setCursor(0, 0);lcd.print("temp high");
   sendmsg(number,"HIGH TEMPERATUR ");
   delay (1000);
   }

 if(sw==1)
  {
    lcd.clear();lcd.setCursor(0, 0);lcd.print("NEED HELP");
    sendmsg(number,"NEED HELP ");
    delay (1000);

   }

   }
```

4.5 RESULTS

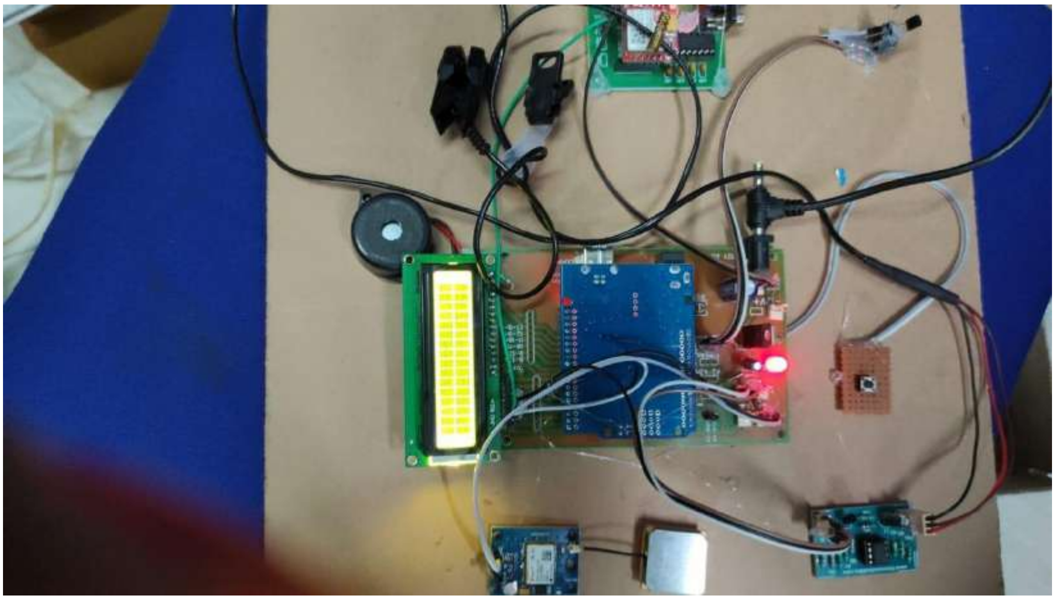


Fig 4.5 Final Output

CHAPTER 5

PROJECT TESTING

5.1 OVERVIEW OF TESTING METHODS

The soldier Health and Position Tracking System allows the military to track the current GPS position of a soldier and checks the health status including body temperature and heartbeats of the soldier. The System also consists of extra features with the help that a soldier can ask for help manually or send a distress signal to the military if he is in need. The GPS modem sends the latitude and longitude position with a link pattern with the help of that the military can track the current position of the soldier. The system is very helpful for getting health status information of soldiers and providing them instant help. Here are some techniques used to track current position and health condition. In today's era enemy warfare is a very important factor in any nation's security. The national security mainly depends on the defense force. In this system we have come up with an idea of tracking soldiers as well as giving their status of their health condition and location during the war.

So, in implementation of the project, we encountered few problems such as the entire project depends on the wireless communication i.e., it depends on GSM module and signals. So, there are few problems we encountered and solved them.

- Signal problem in hill regions.
- Temperature differences in body that take place due to climatic changes.
- Rise of heartbeat in hill region when climbing.

1. Signal Problem:

   So, to overcome signal problem initially we had used helical antenna which perform well in open area and where there are no obstacles between antenna and signal towers which has many disadvantages, so it was replaced with another antenna which work works well in any environment and has proper bandwidth than helical antenna which is called as 3dBi.
   The 3dBi GSM antenna along with a U. FL to SMA adapter which can be obtained online for less than $3. You can snap-fit this antenna to small u.fl connector located on the top-left corner of the module. This type of antenna has a better performance and allows putting your module inside a metal case – as long the antenna is outside.

2. Temperature differences in body that take place due to climatic changes.

   Here we can change the value in program with respect to climate in outdoors which makes it flexible to any kind of climates can change the threshold values according to region of the soldier's mission.

3. Rise of heartbeat in hill region when climbing.

   Here depending on region in which mission is given to soldier's the threshold value can be varied.

CHAPTER-6

CONCLUSION AND FUTURESCOPE

6.1 CONCLUSION

- Above system, when completed, would make it easy to help in determining the health status of soldiers with measures of heart beats and body temperature.
- It would also help in tracking his position by using GPS modem and with GSM modem it can send all information to the base station so that further necessary action could be taken.

6.2 FUTURE SCOPE

- We can add a graphical display section to this project which helps to display a digital map which shows the position of all soldiers in the unit as they surround a block of buildings and launch their attacks.
- Welling to make live tracking possible.
- Helmet attached screens, accomplished by presenting information from maps and video using varieties of physiological sensors.

PUBLICATION

Submitted Paper in the Conference ICISECT - 21 with Paper ID (ICISECT21-0003) and got Acceptance for the Paper.

REFERENCES

[1]. P.chakravarth, S.Natarajan, M.Anto Bennete “GSM based soldier tracking system and monitoring using wireless communication” Department of Electronics and Communication, published in September 1st ,2017.

[2]. Akshita v.Armarkar, Deepika J, Punekar, Mrunali, p.Kapse, Shweta Kumari, Jayshree Shelka “Soldier health and position tracking system” Department of ETC engineering , volume 7 issue no..3

[3]. Patil Akshay , Shelake Balaji, Pinjari Raju, Mirajkar P.P “GPS based soldier tracking and health monitoring”, Department of ETC engineering, volume 40 issue:03 March 2017.

[4]. Shruthi Nikam, Supriya Patil, Prajkata Powar, V.S.Bendre “GPS based soldier tracking and health indication system”, Department of E&TC , PIMPRI CHINCHWA COLLEGE OF ENGINEERING, volume 2, issue 3, March 2013.

[5]. Govindiraj A, Dr.S.Sindhurja Banu “GPS based soldier tracking and health indication system with environmental analysis”, Department of Electronics and Communicaton Engineering volume 2, issue 12, Decemember 2013.

[6]. Zeeshan Raza,Kamaran Liaquat,Shahazad Ashraf “Monitoring of soldier health and transmition of secret codes”,NFC instutation of engineering volume 8 no 2,june 2017.

[7]. Shubhangi Gupta, Shivani Kulshrestha, Divya Singh, Ashish Kumar, Er.Hitendra Singh “GPS and GSM based soldier health monitoring and tracking system” issue 3 volume 6, March 2017.

[8]. JouniRantakoko,JoakimRydell and peter Sromback, “Accurate and Realiable soldier and first responder Poasitioning :Multisensor System and co-operative localization” April-2011

[9]. RavidraB.Sathe ,A.S.Bhide. “Gps-based soldier tracking and health monitoring system” conference on Advances in communication and computing April 2011-12.

[10]. VincentPereira,AudreyGiremus,andericgrivel “Modeling of multipath environment using copulas For particle filtering based GPS Navigation” June-2012

[11]. Shruti Nikam1, Supriya Patil2, Prajkta Powar3, V.S.Bendre-“ Gpsbased soldier tracking and health indication system”International Journal of Advanced

Research in Electrical, Electronics and Instrumentation Engineering.Vol. 2, Issue 3, March 2013.

[12]. Akshita v.Armarkar, Deepika J, Punekar, Mrunali, p.Kapse, Shweta Kumari, Jayshree Shelka “Soldier health and position tracking system” Department of ETC engineering , volume 7 issue no..3

[13]. Patil Akshay , Shelake Balaji, Pinjari Raju, Mirajkar P.P “GPS based soldier tracking and health monitoring”, Department of ETC engineering, volume 40 issue:03 March 2017.

BIBILOGRAPHY

[1]. JouniRantakoko,JoakimRydell and peter Sromback, “Accurate and Realiable soldier and first responder Poasitioning :Multisensor System and co-operative localization” April-2011

[2]. RavidraB.Sathe ,A.S.Bhide. “Gps-based soldier tracking and health monitoring system” conference on Advances in communication and computing April 2011-12.

[3]. VincentPereira,AudreyGiremus,andericgrivel “Modeling of multipath environment using copulas For particle filtering based GPS Navigation” June-2012

[4]. Shruti Nikam1, Supriya Patil2, Prajkta Powar3, V.S.Bendre-“ Gpsbased soldier tracking and health indication system”International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering.Vol. 2, Issue 3, March 2013.

[5]. Akshita v.Armarkar, Deepika J, Punekar, Mrunali, p.Kapse, Shweta Kumari, Jayshree Shelka “Soldier health and position tracking system” Department of ETC engineering , volume 7 issue no..3

[6]. Patil Akshay , Shelake Balaji, Pinjari Raju, Mirajkar P.P “GPS based soldier tracking and health monitoring”, Department of ETC engineering, volume 40 issue:03 March 2017.

APPENDICES

The project shows that the health monitoring and tracking system for soldiers. The proposed system can be mounted on the soldier's body to track their health status and current location using GPS. This information will be transmitted to the control room through IoT. The proposed system comprises of tiny wearable physiological equipment's, sensors, transmission modules. Hence, with the use of the proposed equipment, it is possible to implement a low-cost mechanism to protect the valuable human life on the battlefield.

A

MAJOR PROJECT REPORT

On

SOLDIER HEALTH AND POSITION TRACKING SYSTEM

Submitted by

1)Ms. Milaliyan Samantaray (17K81A04M8) 2) Mr.Gurram Rakesh(17K81A04K9)

3) Ms. Preethi Roshan (17K81A04N8)

in partial fulfillment for the award of the degree

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Esteemed Guidance of

Mr. P. JOEL JOSEPHSON

Associate Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

An Autonomous Institute

Dhulapally, Secunderabad – 500 10

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the major-project work entitled **"Soldier Health and Position Tracking System"** is a bonafide work carried out by **Gurram Rakesh (17K81A04K9), Milaliyan Samantaray (17K81A04M8), Preethi Roshan (17K81A04N8)** in partial fulfilment of the requirements for the degree of **Bachelor of Technology in Electronics & Communication Engineering** by the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2020-21.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree.

PROJECT GUIDE

P.Joel Josephson
Associate Professor
Department of ECE

HEAD OF DEPARTMENT

Dr. B. Hari Krishna,
Professor
Department of ECE

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT MILALIYAN SAMANTARAY WITH ROLL NO.17K81A04M8, GURRAM RAKESH WITH ROLL NO.17K81A04K9, PREETHI ROSHAN WITH ROLL NO.17K81A04N8, OF B. TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "SOLDIER HEALTH AND POSITION TRACKING SYSTEM" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT

DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

DECLARATION

We, the students of **Bachelor of Technology** in Department of ELECTRONICS AND COMMUNICATION ENGINEERING, session: <2017 – 2021>, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled "SOLDIER HEALTH AND POSITION TRACKING SYSTEM" is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

MILALIYAN SAMANTARAY GURRAM RAKESH PREETHI ROSHAN
(17K81A04M8) (17K81A04K9) (17K81A04N8)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr.P.SANTOSH KUMAR PATRA**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr.B.HARI KRISHNA**, **Head &Professor**, Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in our memory for years to come.

We would like to express our gratitude to our dept. project coordinator **Dr.A.ANAND, Professor** for his involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time to come.

Finally, we are highly grateful and obliged to our project guide **Mr.P.JOEL JOSEPHSON, Associate Professor,** for his advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

1.MILALIYAN SAMANTARY (17K81A04M8)
2.GURRAM RAKESH (17K81A04K9)
3.PREETHI ROSHAN (17K81A04N8)

ABSTRACT

The soldier Health and Position Tracking System allows military to track the current GPS position of soldier and checks the health status including body temperature and heartbeats of soldier. The System also consists extra feature with the help of that soldier can ask for help manually or send a distress signal to military if he is in need. The GPS modem sends the latitude and longitude position with link pattern with the help of that military can track the current position of the soldier. The system is very helpful for getting health status information of soldier and providing them instant help.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The nation's security is monitored and kept by army, navy, and air-force. The important and vital role is of soldiers who sacrifice their life for their country. There are many concerns regarding the safety of the soldier. Soldiers entering the enemy lines often lose their lives due to lack of connectivity, it is very vital for the army base station to know the location as well as health status of all soldiers. India has already lost so many soldiers in war-fields as there was no proper health backup and connectivity between the soldiers on the war-fields and the officials at the army base stations. Recently on 29 September 2016, a military confrontation between India and Pakistan began, Indian soldiers conducted a surgical strike against militant launch pads across the line of control in Pakistani-administered Azad Kashmir and inflicted "significant casualties". Indian soldiers are mainly known for their courage, despite scarce ammunitions and safety measures, they have many triumphs to their credits. All must be really concerned about the safety of the soldiers, so we have decided to build a project which will efficiently keep a check on the health status of the soldier, and his precise location to equip him with necessary medical treatments as soon as possible. Soldier's tracking is done using GPS and GSM is used to provide wireless communication system. For monitoring the health parameters of soldiers, we are using biomedical sensors such as temperature sensors and heartbeat sensors.

An oxygen level sensor is used to monitor atmospheric oxygen so if there are any climatic changes the soldiers will be equipped accordingly.
The infantry soldier of tomorrow promises to be one of the most technologically advanced modern warfare has ever seen. Around the world, various research programs are currently being conducted, such as the United States' Future Force Warrior (FFW) and the United Kingdom's Future Infantry Soldier Technology (FIST), with the aim of creating fully integrated combat systems .Alongside vast improvements in protective and weaponry subsystems, another major aspect of this technology will be the ability to provide information superiority at the operational edge of military networks by equipping the dismounted soldier with advanced visual, voice, and data communications.

Helmet mounted visors, capable of displaying maps and real-time video from other squad members, ranges of physiological sensors display the heartbeat, body temperature, atmosphere pressure, surrounding oxygen level etc. These devices will improve awareness for collateral military personnel as well as who will exchange information using wireless networks along with hosts. The challenge was to integrate these piecemeal components into a lightweight package that could achieve the desired result without being too bulky and cumbersome or requiring too much power. Communicating with the base (control room) station become the fundamental challenges in military operations. The proper navigation between soldier's organizations plays an important role for careful planning and coordination.

1.2 OBJECTIVES OF THE STUDY

So, this project focuses on tracking the location of soldiers from GPS, which is useful for control room stations to know the exact location of soldiers and accordingly they will guide them. Also High-speed, short-range, soldier-to-soldier wireless communications to relay information on situational awareness, such as Biomedical sensors, GPS navigation, Wireless communication.

1.3 SCOPE OF THE STUDY

- Soldier health and position tracking system is such a type of a system that provides the health and position status of a soldier fast with the help of gps and gsm modules.

- Soldiers are important assets of our country. They workday and night to protect our country. Sometimes even they sacrifice their life for the country.

- And soldiers at borders often lose their lives defending the country.

- To overcome this situation this system helps us to track the position of the soldiers and even information regarding health is produced by this system. And will efficiently keep a check on the health status of the soldier, and his precise location to equip him with necessary medical treatments as soon as possible.

- Here we have proposed a system that is called Soldier health and position tracking system, which is so much efficient, reliable, and friendly to use. In this system soldier's tracking is done using GPS and GSM is used to provide wireless communication systems. For monitoring the health parameters of soldiers, we are using biomedical sensors such as temperature sensors and heartbeat sensors.

1.4 MATERIAL REQUIREMENT:

1.4.1 HARDWARE REQUIREMENT:

- Arduino uno
- 12V Battery
- Temperature Sensor
- Heartbeat sensor
- Push Button
- LCD
- GSM
- GPS
- Buzzer

1.4.2 SOFTWARE REQUIREMENT

- Proteus Software
- Arduino Software

1.5 PROCUMENT OF EQUIPMENT

We brought all the required hardware components in an electronic manufacturing shop and before assembling and interfacing hardware components to Arduino UNO micro-controller, we connected individual components to Arduino in Proteus Software and understood working of every individual component in Proteus Software.

Now, we assembled all components and interfaced to Arduino UNO micro-controller using connecting wires through bread board and we wrote program to work all the hardware components in Arduino IDE we understood the working of all components in a practical way.

We were able to monitor the health and position of soldiers better than before we successfully got the output in mobile and LCD screen. We would like to improve the project in future for further developments.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH AREA

Nowadays all nations keep its security at high priority. Wars are being fought for land, water and acquiring the position of most powerful nation. A country's arm forces consist of three professional uniformed services: the army, the navy, and the air force. Soldiers being the backbone of any armed force usually lose their lives due to lack of medical help when in emergency, also soldiers who are involved in missions or in special operations get straggled on war fields and lose contact with the authorities. To overcome these concerns, we had built this project which, using wireless body area sensor network (WBANS) such as temperature sensor, heartbeat sensor etc. will monitor the health status of the soldier whenever required. Also using GPS, we can track the soldier's exact location whenever required. Using oxygen level sensor, we can also monitor the environmental condition, so authorities can provide essential aids. The communication is established between the soldiers and authorities via GSM. Any abnormalities in the readings of wireless body area sensor network (WBASNs) are considered as a trigger for GSM to establish the connection between the soldier and base unit and send current location and health status to the receiver. By using all this equipment's, we had tried to implement the basic guarding system for the soldier in low cost, light weighted, portable and precise device.

In enemy territory soldiers not only have to deal with the physical threat, but also with stress and fatigue caused by protracted operations or lack of sleep. So, for the security purpose we need a tool for remote soldier performance and health monitoring. So, in this project a tool is implemented using biosensors for health monitoring purpose. Also, a GPS system is used to track the location of soldier. Additionally, a GSM modem is also used to make the system wirelessly compatible.

In today's world, warfare is an important factor in any nation's security. One of the important and vital roles is played by the army soldiers. There are many concerns regarding the safety of soldiers. So, for their security purpose, many instruments are mounted on them to view their health status as well as their real time location. Bio-sensor systems comprise various types of small physiological sensors, transmission modules and processing capabilities, and can thus facilitate low-cost wearable unobtrusive solutions for health monitoring. This paper gives an ability to track the location and monitor health of the soldiers in real time who become lost and get injured in the battlefield.

2.2 REVIEW ON RELATED LITERATURE

In [1], the author has worked on the safety of the soldier by tracking his health condition during the war which provides the control room to plan the war strategies and tracks the location if by chance the soldier is lost. When the control room notices the soldier is lost in the battlefield then they guide the soldier in the right path. The soldier's health information is transmitted to the control room and keeps track of the injured soldier and take necessary action to save the life. This system mainly focuses on the soldier who have involved in the warfare and tracking the health of the injured soldier. In [2], the author has focused to improve the communication of the soldier with the control room people and control plane operation continuously. He has also focused to track and guide the position, direction, and the surrounding temperature of the soldier by using the wristwatch mountaineers and by providing headphones to guide the soldier to the right path. He has also used the technology like the Radio collar strapped to ankles, so the movement of the soldier can be tracked and displays the current location at the base station. Microcontroller is used to record the body parameters of the soldier's and transmitted to the base station. In [3], the author has focused to provide the embedded wireless system for the soldier and to minimize the time to track the location of the soldier, rescue and search operation and their health status using GPS module and wireless body sensors, the data are collected from the GPS and sensors are transmitted using the ZigBee technology to the base station/control room. It also allows the soldier to communicate with the other fellow soldiers within the wireless transmission range, they can also request help from the control room. In [4], the author has worked on the location tracking system using GPS with the Google Maps based monitoring for vehicle. This system provides tracking irrespective of weather condition. It gives the shortest route to track. This idea is deployed in the proposed system to track the current location and movement of the soldier along with the shortest path to him, which is need for the rescue operation. In [5], here the author focus is on monitoring the ill patient, continuously and reports the changes spontaneously to the concerned person using the GPS, 4 Chaitra R L, Mamatha V which helps in protecting the life of the individual. In this paper only, the ill patient is considered for monitoring. Once the patient is discharged from the hospital, the body sensor is mounted on them to monitor their health condition and take necessary action in case of emergency by reporting to the guardian. In [6], the author has worked to track the location of the soldier and monitor the health status using an Arduino board. For the transmitting of data to control room he has used GPS module for connectivity. In [7], the author focus is to provide a good connectivity between the control room and the soldier by using Arduino and GSM module.

2.3 CONCLUSION ON REVIEWS

From the above implementation we have concluded that the communication hurdles between the soldiers and authorities at the base unit is overcome using GSM, the precise location and health parameters are known using GPS and wireless body area sensor network (WBASNs) respectively and with the GSM modem all information is send to the base station so that field commander will take necessary action. Base station gets location of soldier from GPS. The base station can access the status of the soldier which is displayed on the phone with the help of GSM and hence appropriate actions can be found.

CHAPTER 3

PROJECT DESIGN

3.1 OVERVIEW OF THE DESIGN

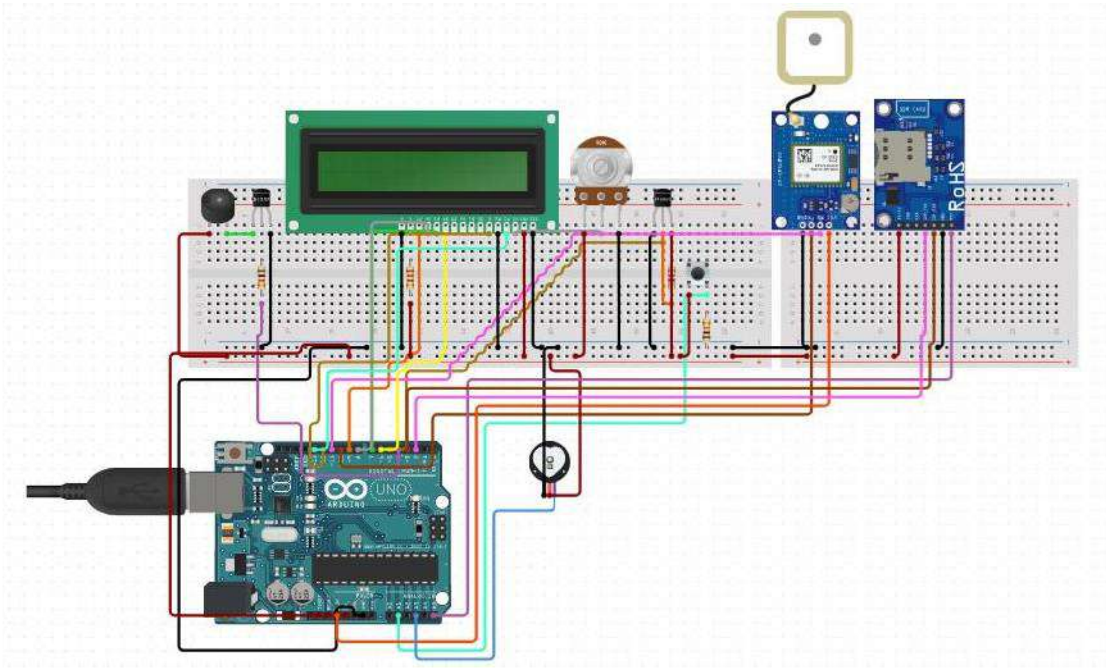


Fig.3.1 Overview of the design

3.2 EQUIPMENT ANALYSIS

3.2.1 ARDUINO

3.2.1.1 INTRODUCTION

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).

- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

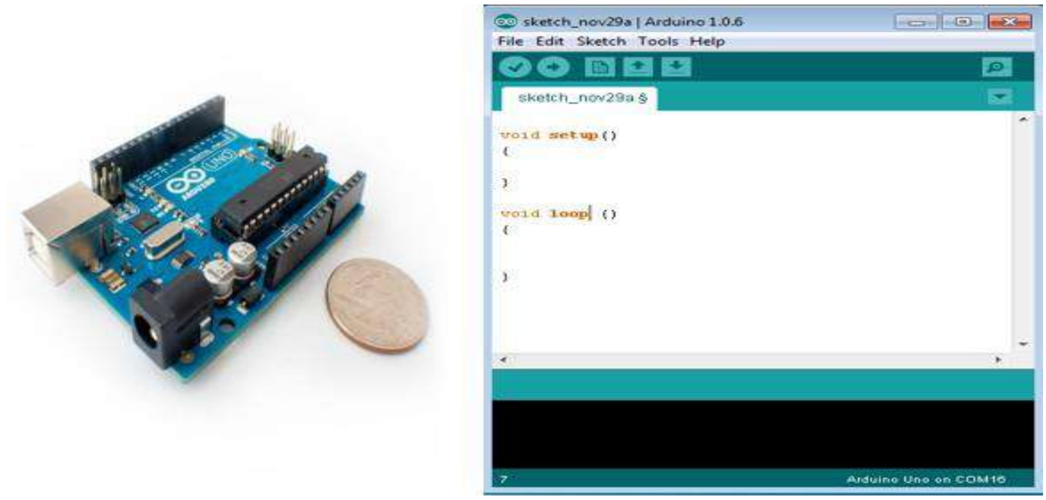


Fig 3.2.1.1 Arduino Uno

3.2.1.2 BOARD TYPES

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface

Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini-05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini-3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header

Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Lilypad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compati
Lilypad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.2.1.2.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 3.2.1.2.2 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface

Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 3.2.1.2.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table 3.2.1.2.4 Arduino boards based on AT91SAM3X8E microcontroller

3.2.1.3 BOARD DESCRIPTION

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

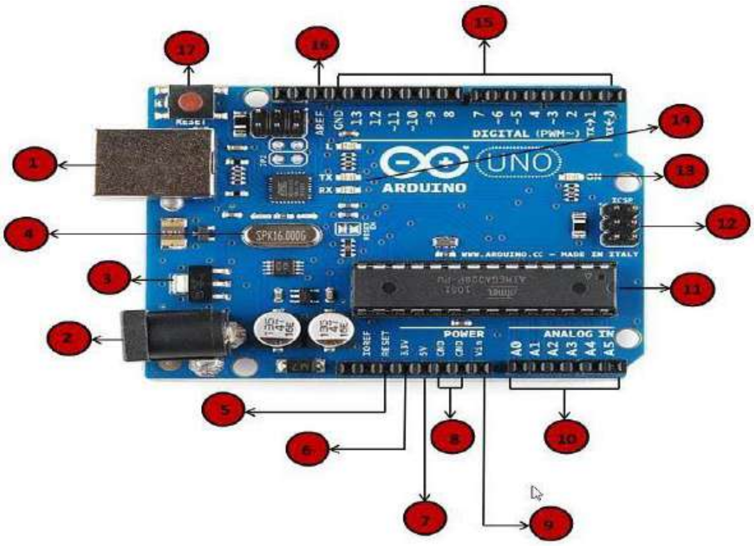


Fig.3.2.1.3 Board Description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).

6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volts. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in

	two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Table 3.2.1.3 Board description of Arduino UNO

3.2.1.4 ARDUINO FAMILY

Arduino makes several different boards, each with different capabilities. In addition, part of being open-source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you are not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

Fig.3.2.1.4 Arduino Family

3.2.1.5 SHIELDS

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

Fig.3.2.1.5 Arduino Shields

3.2.1.6 PIN DESCRIPTION OF ATMEGA328

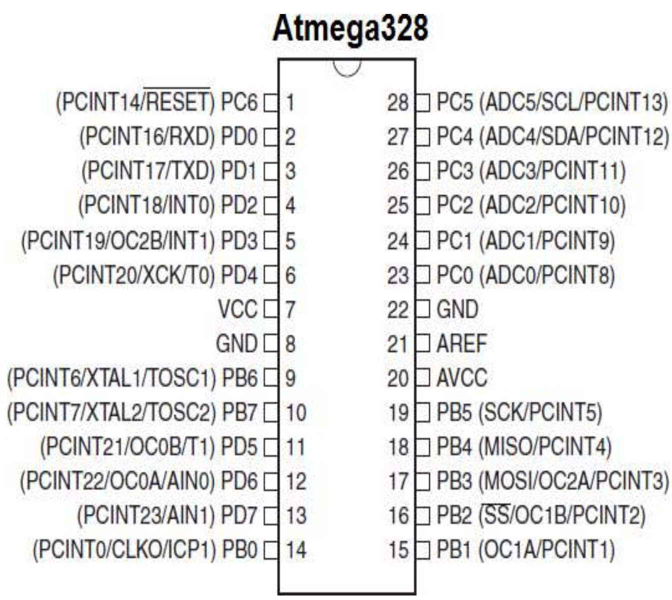


Fig.3.2.1.6 Pin description of ATMEGA328

3.2.1.7 APPLICATIONS

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

3.2.2 16*2 LCD SCREEN DISPLAY

3.2.2.1 INTRODUCTION

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

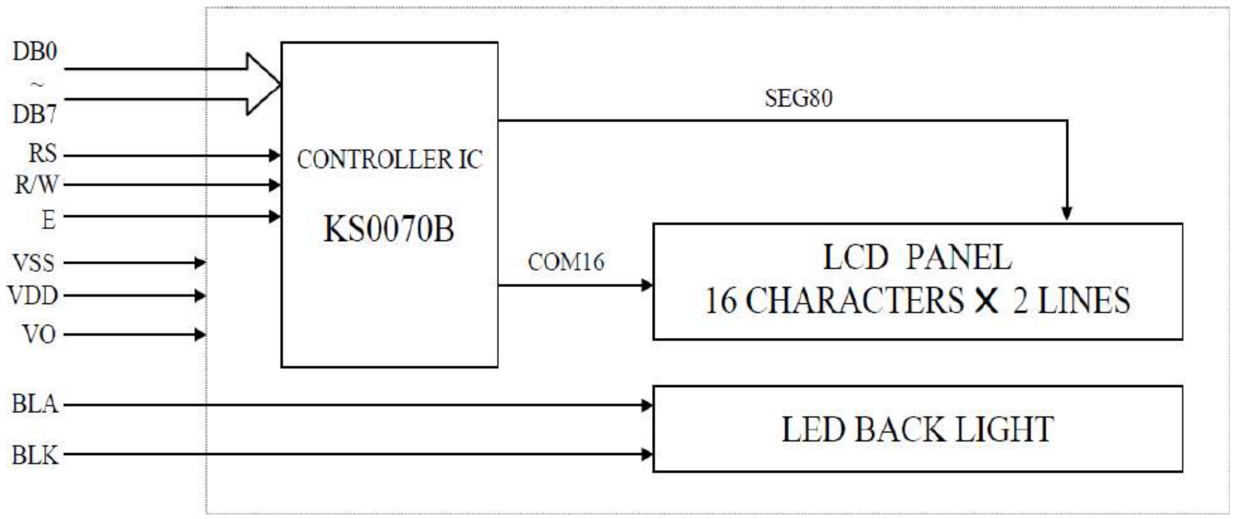


Fig.3.2.2.1 Lcd block diagram

3.2.2.2 DISPLAY DATA RAM:

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.

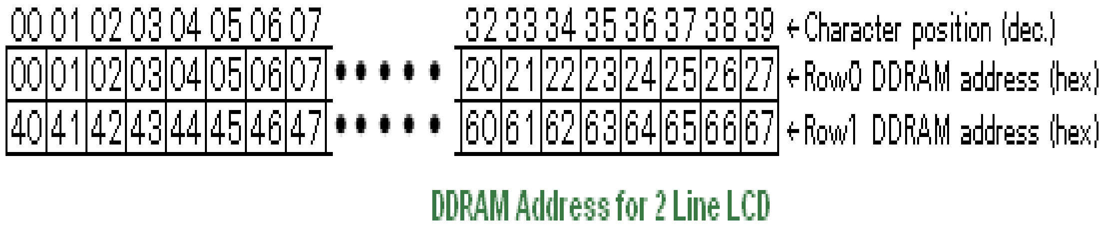


Fig.3.2.2.2 DDRAM addresses of 2 Line LCD

3.2.2.3 CHARACTER GENERATOR ROM

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM.

3.2.2.4 BUS FLAG

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e. BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

3.2.2.5 INSTRUCTION REGISTER (IR) AND DATA REGISTER (DR)

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g. LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD.

Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

3.2.2.6 16 X 2 ALPHANUMERIC LCD MODULE FEATURES:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

3.2.2.7 CIRCUIT DESCRIPTION

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore, by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore, we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result, we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly. The 2-line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

3.2.3 HOW TO CHOOSE A BUZZER

There are many kinds of buzzer to choose, first we need to know a few parameters, such as voltage, current, drive method, dimension, mounting type, and the most important thing is how much SPL and frequency we want.

Operating voltage: Normally, the operating voltage for a magnetic_buzzer is from 1.5V to 24V, for a piezo buzzer is from 3V to 220V. However, in order to get enough SPL, we suggest giving at least 9V to drive a piezo buzzer.

Consumption current: According to the different voltage, the consumption current of a magnetic buzzer is from dozens to hundreds of mill amperes; oppositely, the piezo type saves much more electricity, only needs a few mill amperes, and consumes three times current when the buzzer start to work.

Driving method: Both magnetic and piezo buzzer have self-drive type to choose. Because of the internal set drive circuit, the self-drive buzzer can emit sound if connecting with the direct current. Due to the different work principle, the magnetic buzzer need to be driven by 1/2 square waves, and the piezo buzzer need square waves to get better sound output.

Dimension: The dimension of the buzzer affects its SPL and the frequency, the dimension of the magnetic buzzer is from 7 mm to 25 mm; the piezo buzzer is from 12 mm to 50 mm, or even bigger.

Connecting way: Dip type, Wire type, SMD type, and screwed type for big piezo buzzer are usually seen.

Sound Pressure Level (SPL): Buzzer is usually tested the SPL at the distance of 10 cm, if distance double, the SPL will decay about 6 dB; oppositely, the SPL will increase 6 dB when the distance is shortened by one time. The SPL of the magnetic buzzer can reach to around 85 dB/ 10 cm; the piezo buzzer can be designed to emit very loud sound, for example, the common siren, are mostly made of piezo buzzer.

3.2.3.1 INTRODUCTION OF PIEZO BUZZER

Piezo buzzers are simple devices that can generate basic beeps and tones. They work by using a piezo crystal, a special material that changes shape when voltage is applied to it. If the crystal pushes against a diaphragm, like a tiny speaker cone, it can generate a pressure wave which the human ear picks up as sound. Simple change the frequency of the voltage sent to the piezo and it will start generating sounds by changing shape very quickly!

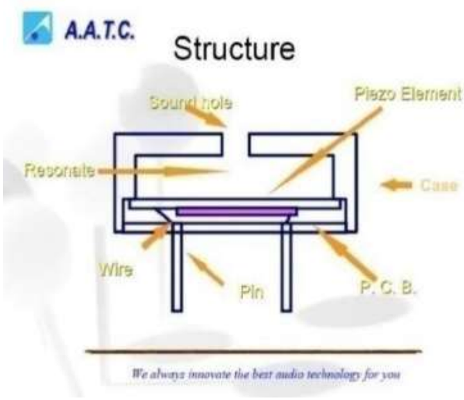


Fig.3.2.3.1 piezo buzzer

3.2.3.2 SPECIFICATIONS

Rated Voltage: A piezo buzzer is driven by square waves (Vp-p).
Operating Voltage: For normal operating. But it is not guaranteed to make the minimum SPL under the rated voltage.
Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.

Capacitance: A piezo buzzer can make higher SPL with higher capacitance, but it consumes more electricity.

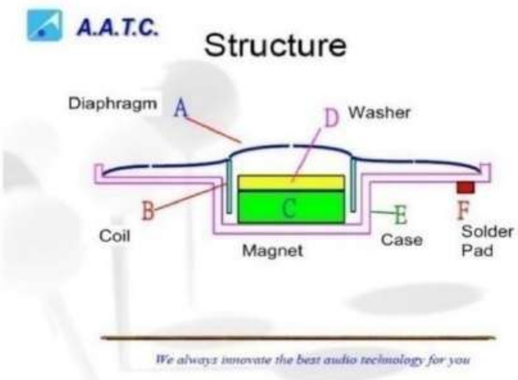


Fig. 3.2.3.2 structure of piezo buzzer

Sound Output: The sound output is measured by decibel meter.
Applying rated voltage and square waves, and the distance of 10 cm.
Rated Frequency: A buzzer can make sound on any frequencies, but we suggest that the highest

and the most stable SPL comes from the rated frequency.

Operating Temp.: Keep working well between -30℃ and +70℃.

3.2.4 TEMPERATURE SENSOR

Temperature Sensor which converts temperature value into electrical signals. We used IC called LM 35 as a temperature sensor. LM35 series sensors are precision integrated-circuit temperature sensors whose output voltage is linearly proportional to the Celsius temperature. The LM35 requires no external calibration since it is internally calibrated. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°Cover a full −55 to +150°C temperature range.

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteed (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±1⁄4°C typical
- Low impedance output, 0.1 W for 1 mA load

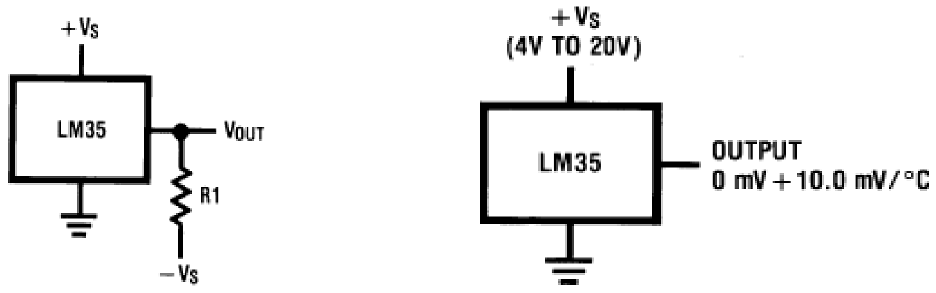


Figure. 3.2.4 Temperature Sensor

3.2.4.1 TEMPERATURE SENSING CIRCUIT

The methods of temperature measurement may be divided into two main classes according as the exchange of heat between the testing body and the hot system takes place by contact, or by radiation across a space. In the contact methods, thermometers or thermocouples are used and they are immersed in solids or liquids. The thermodynamic equilibrium between the hot body and the testing body is established by material contact. In the non-contact methods, the thermodynamic equilibrium is established by the radiation emitted as excited atom and molecules in the hot body return to the ground state.

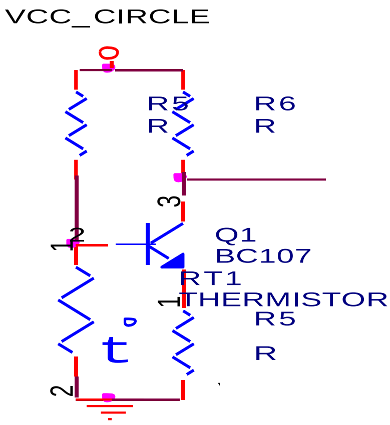


Figure. 3.2.4.1 Temperature Sensing Circuit

3.2.5 BATTERY

Batteries are a collection of one or more cells whose chemical reactions create a flow of electrons in a circuit. All batteries are made up of three basic components: an anode (the '-' side), a cathode (the '+' side), and electrolyte (a substance that chemically reacts with the anode and cathode).

When the anode and cathode of a battery is connected to a circuit, a chemical reaction takes place between the anode and the electrolyte. This reaction causes electrons to flow through the circuit and back into the cathode where another chemical reaction takes place. When the material in the cathode or anode is consumed or no longer able to be used in the reaction, the battery is unable to produce electricity. At that point, your battery is "dead."

Batteries that must be thrown away after use are known as **primary batteries**. Batteries that can be recharged are called **secondary batteries**.

Fig.3.2.5.1 Lithium Battery

Lithium polymer batteries, for example, can be recharged without batteries, your quadcopter would have to be tethered to the wall, you would have to hand crank your car, and your Xbox controller would have to plugged in all the time (like in the good old days). Batteries offer a way to store electrical potential energy in a portable container.

Fig.3.2.5.2 Batteries come in a variety of shapes, sizes, and chemistries

3.2.5.1 THE TERM BATTERY

Historically, the word "battery" was used to describe a "series of similar objects grouped together to perform a function," as in a battery of artillery. In 1749, Benjamin Franklin first used the term to describe a series of capacitors he had linked together for his electricity experiments. Later, the term would be used for any electrochemical cells linked together for the purpose of providing electric power.

Fig.3.2.5.1 The term battery

Battery of Leyden Jar "capacitors" linked together (Image courtesy of Alvinrune of Wikimedia Commons)

3.2.6 SOFTWARE EXPLANATION

- Arduino software
- Proteus simulation
-

3.2.6.1 ARDUINO SOFTWARE

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards can read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

3.2.6.2 WHY ARDUINO?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low-cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step-by-step instructions of a kit or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.Wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for

beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

- Open source and extensible software - The Arduino software is published as open-source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money
- Getting Started with Arduino and Genuino products: -

Install the Arduino Software (IDE) on Windows PCs-

This document explains how to install the Arduino Software (IDE) on Windows machines.

✓ Download the Arduino Software (IDE)

✓ Proceed with board specific instructions.

3.2.6.3 HOW TO DOWNLOAD THE ARDUINO SOFTWARE (IDE)

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

3.2.6.4 INSTALLATION

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.3.2.6.4.1 Standard USB Cable (A plug to B plug)

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig. 3.2.6.4.2 Standard USB Cable (an A to Mini-B plug)

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

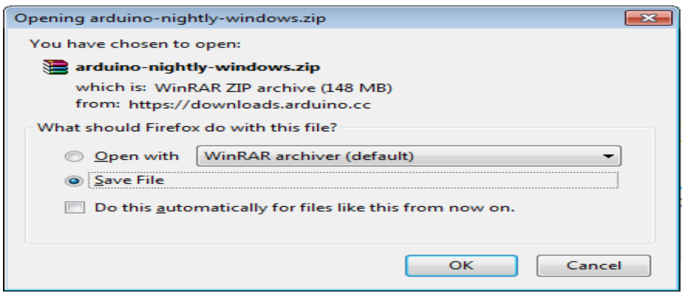


Fig. 3.2.6.4.3 Download Arduino IDE Software

Step 3 − Power up your board

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

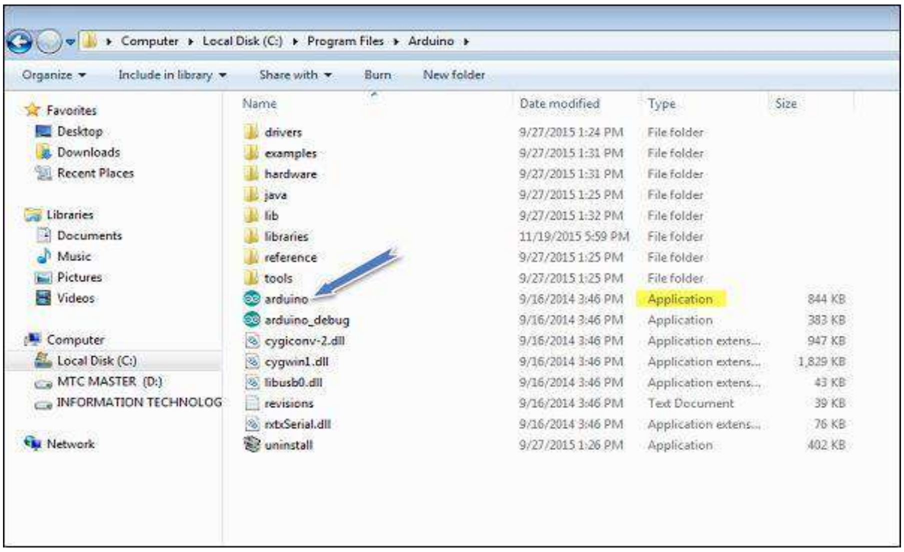


Fig. 3.2.6.4.4 Launch Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**

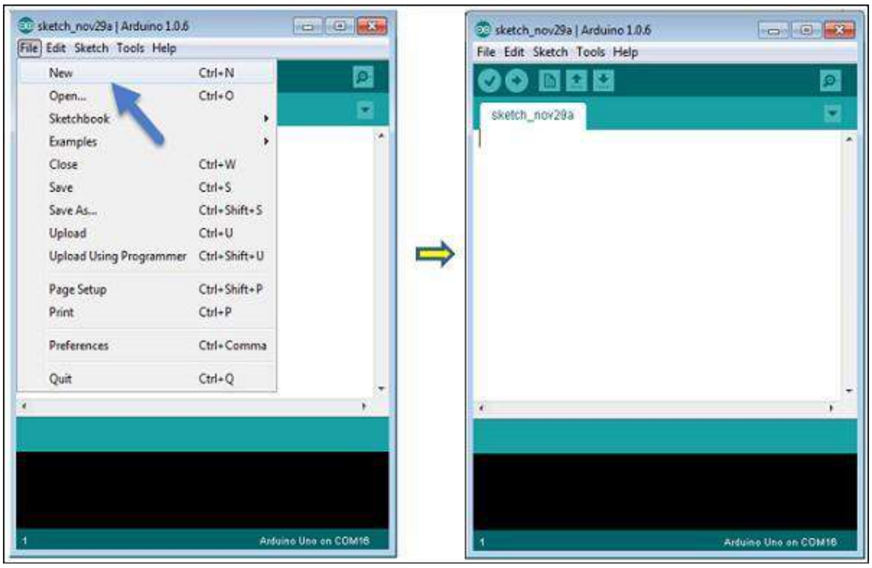


Fig. 3.2.6.4.5 Open your first project.

To open an existing project example, select File → Example → Basics → Blink.

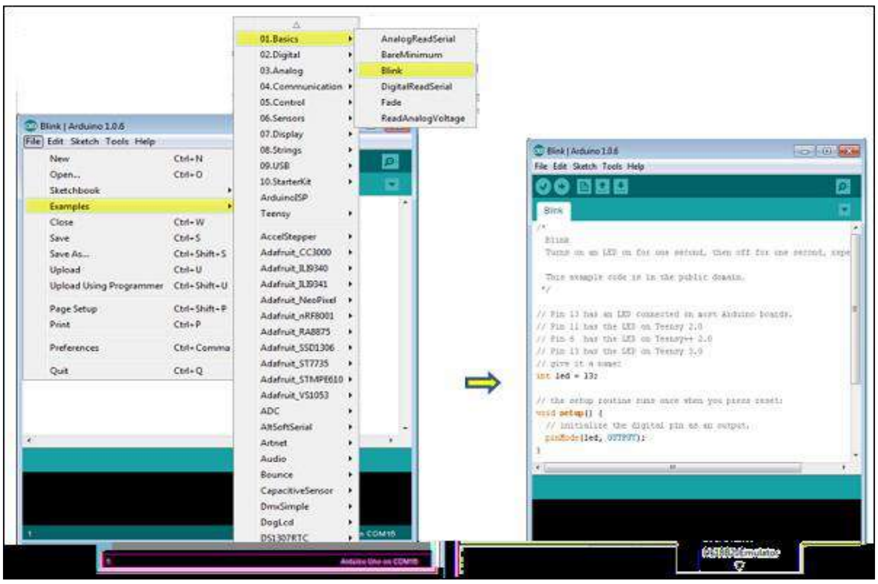


Fig. 3.2.6.4.6 Open your first program.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

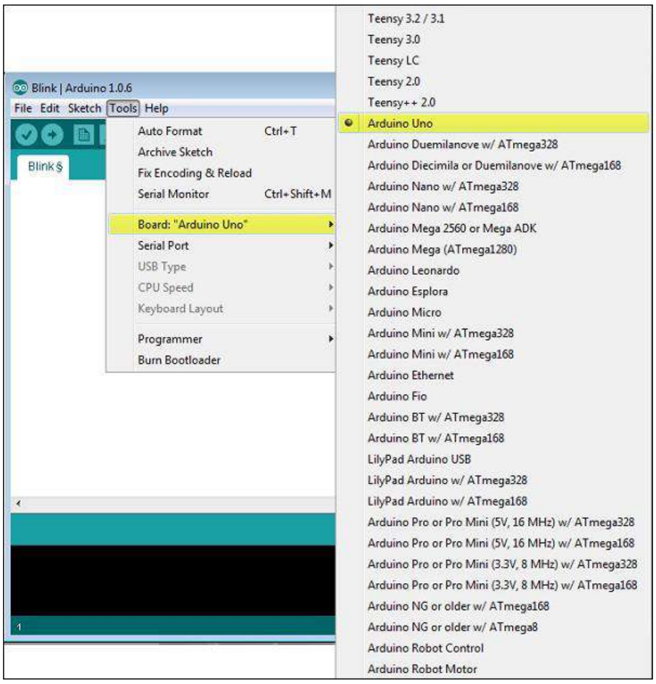


Fig. 3.2.6.4.7 Select your Arduino board.

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

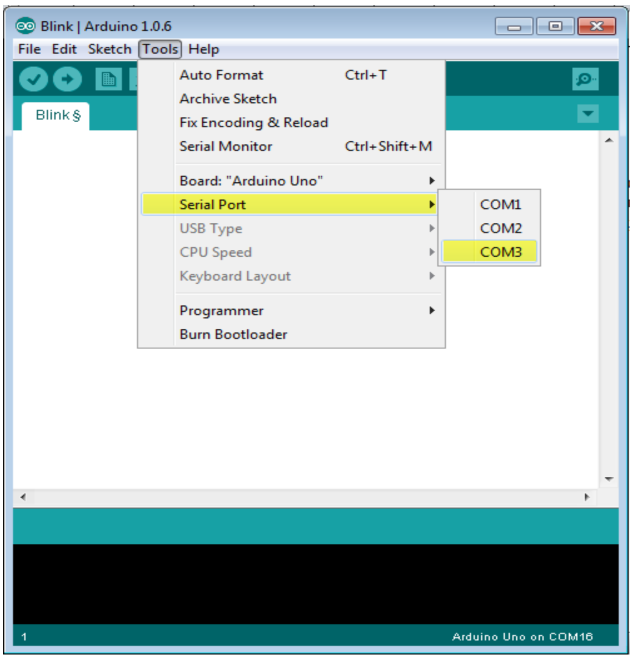


Fig. 3.2.6.4.8 Select your serial port.

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

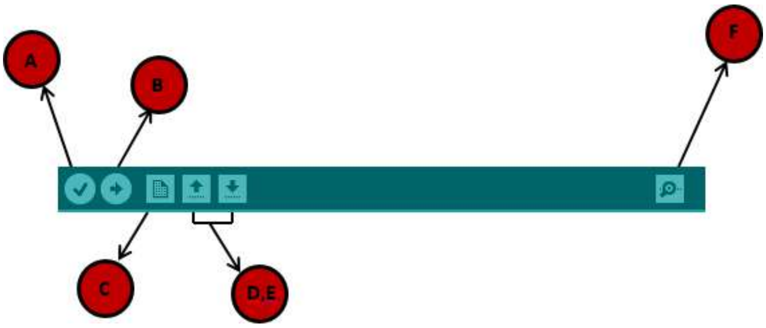


Fig. 3.2.6.4.9 Upload the program to your board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketches.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from board & send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

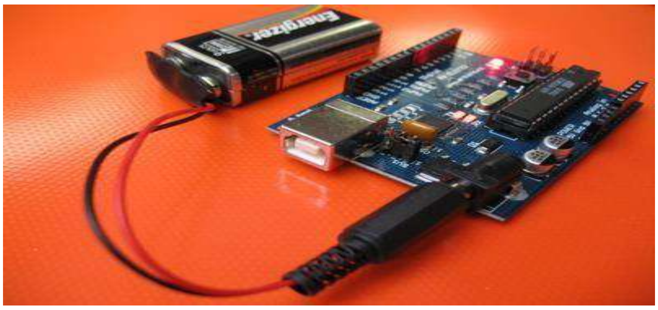


Fig. 3.2.6.4.10 Connecting a Battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called "sketches", but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

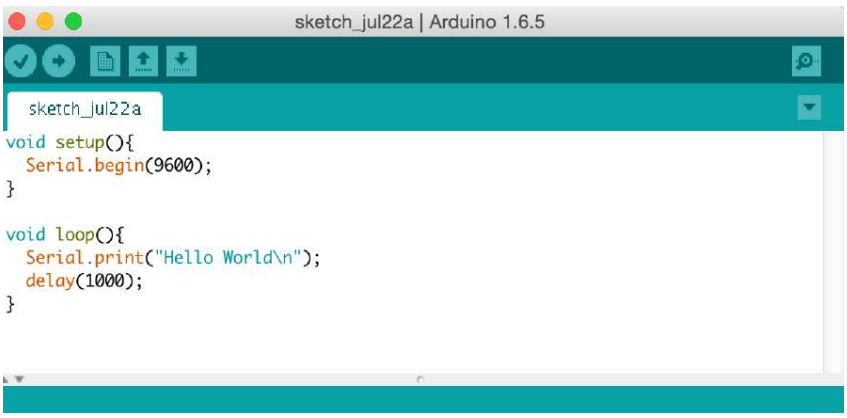


Fig. 3.2.6.4.11 Example Program

3.2.7 PROTEUS

PROTEUS

Proteus is a simulation and design software tool developed by Lab center Electronics for Electrical and Electronic circuit design. It also possesses 2D CAD drawing feature. It deserves to bear the tagline "From concept to completion".

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the footprints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design:

Step 1: Open ISIS software and select new design in File menu

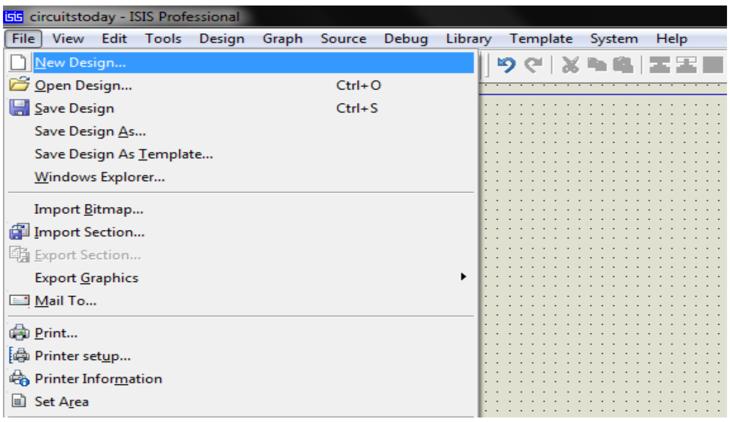


Fig. 3.2.7.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

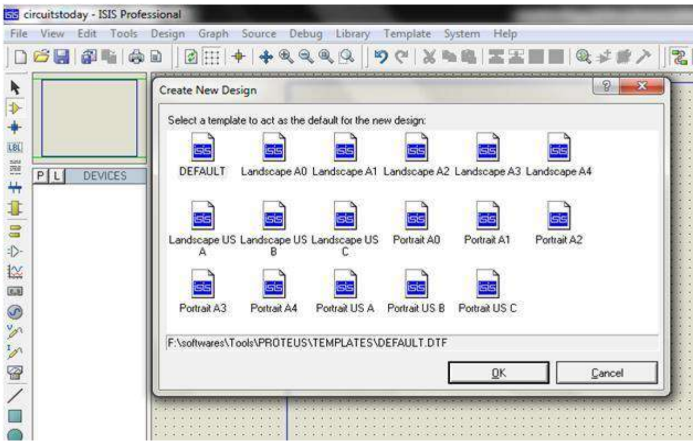


Fig. 3.2.7.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

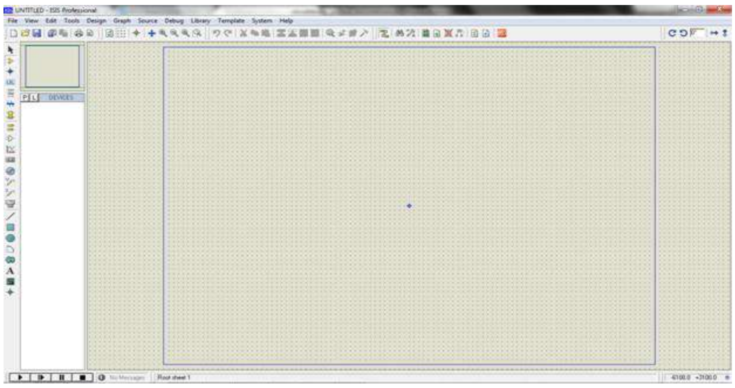


Fig. 3.2.7.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

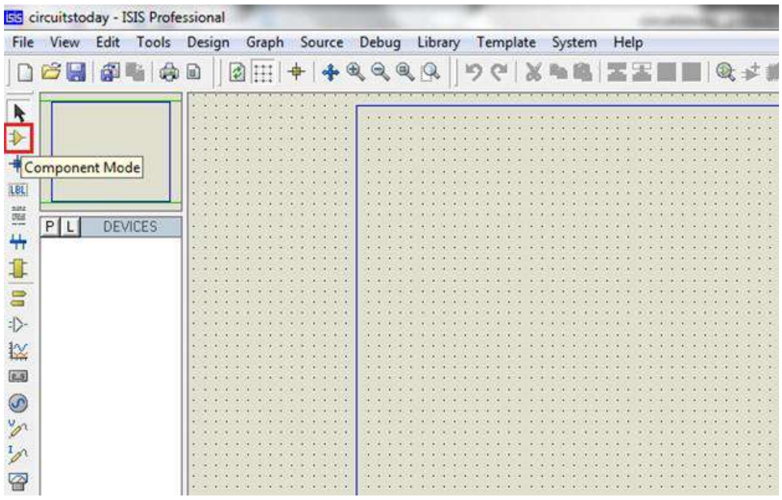


Fig. 3.2.7.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

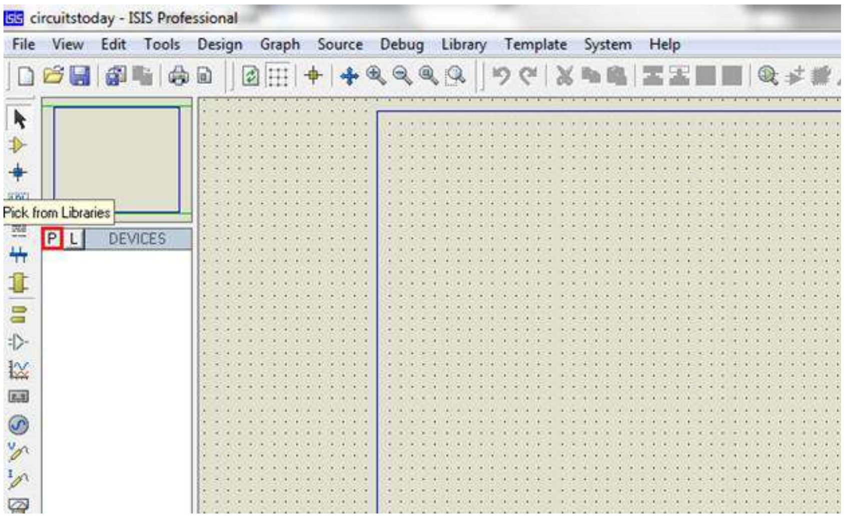


Fig. 3.2.7.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keyword's text box.

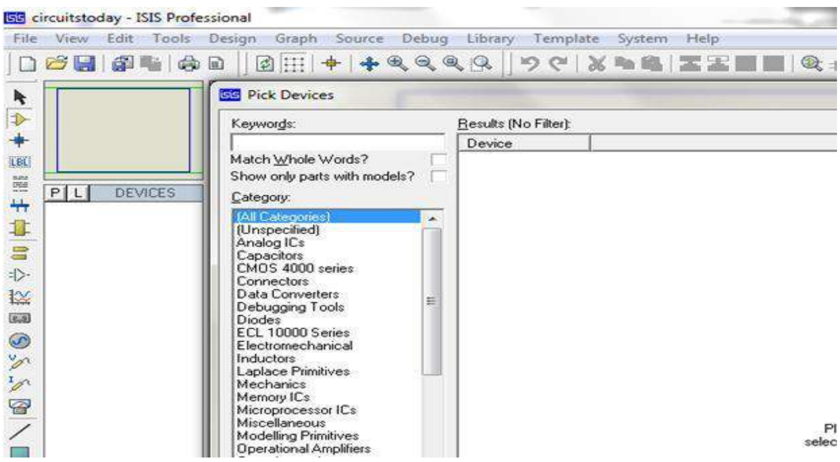


Fig. 3.2.7.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

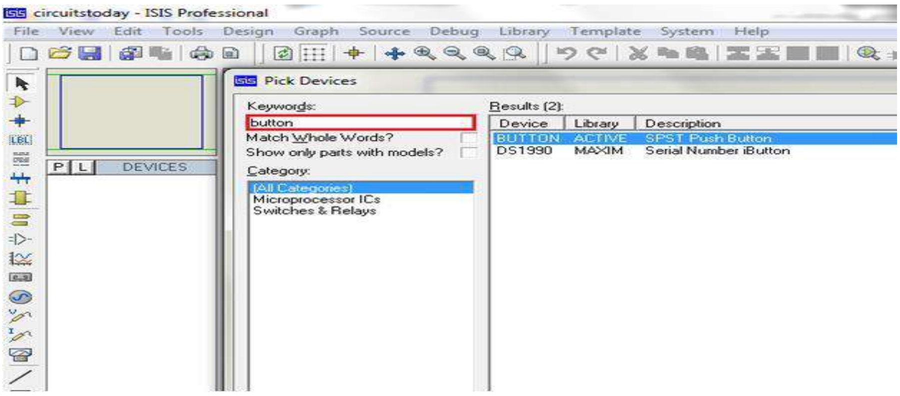


Fig. 3.2.7.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

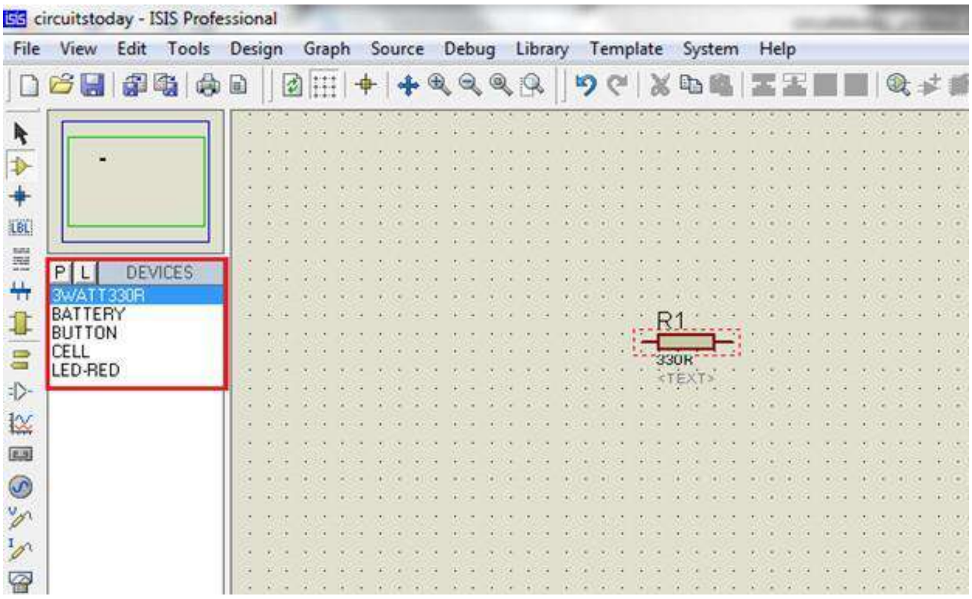


Fig. 3.2.7.8 Component Selection

Place all the required components and route the wires i.e., make connections. Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

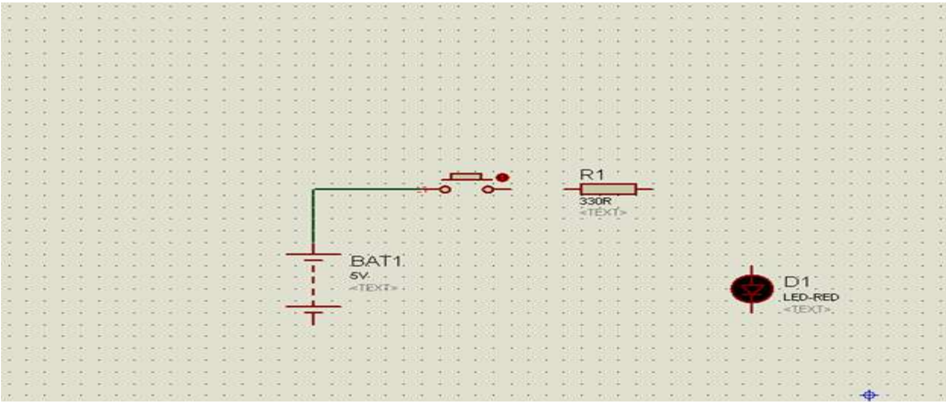


Fig. 3.2.7.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

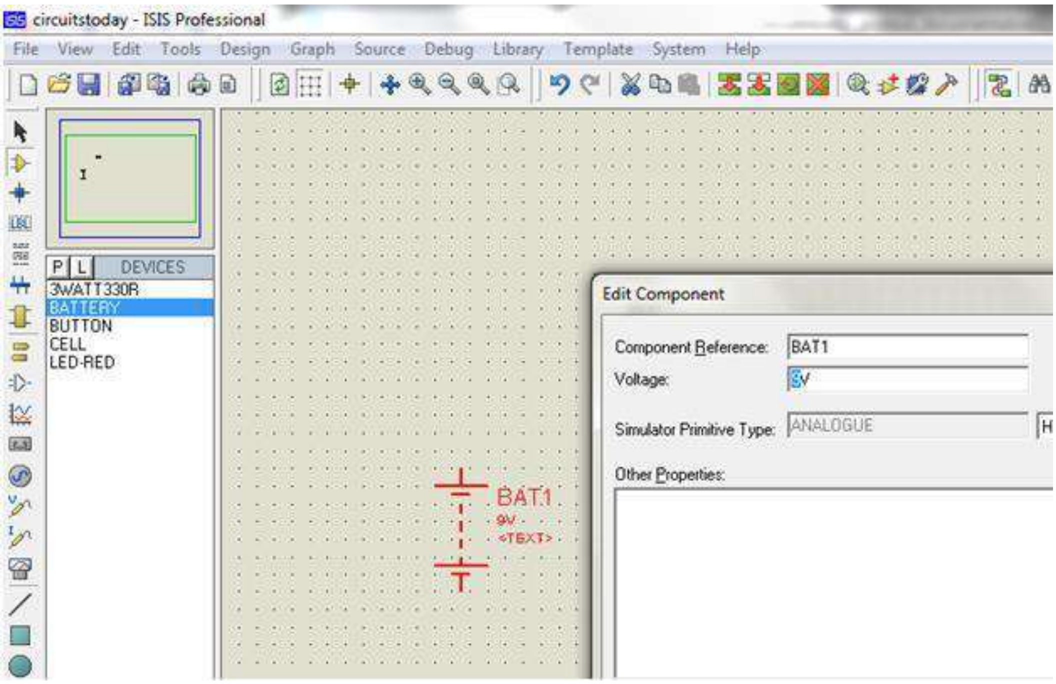


Fig. 3.2.7.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

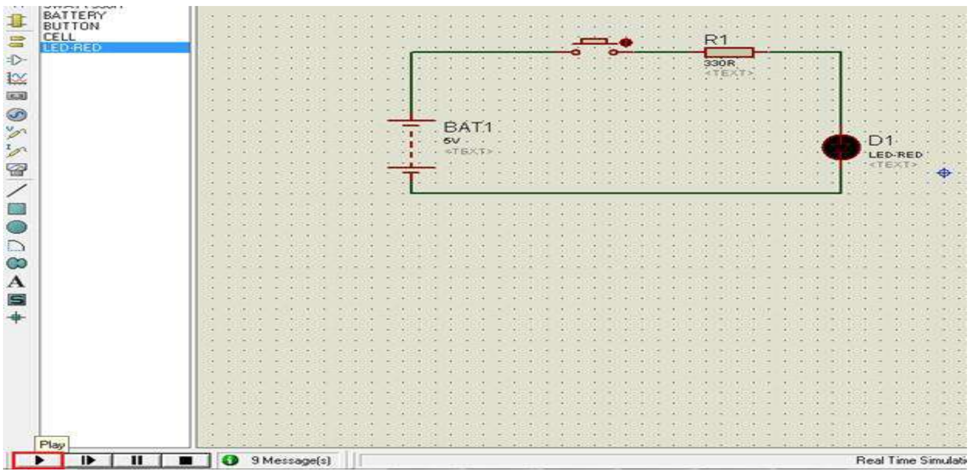


Fig. 3.2.7.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

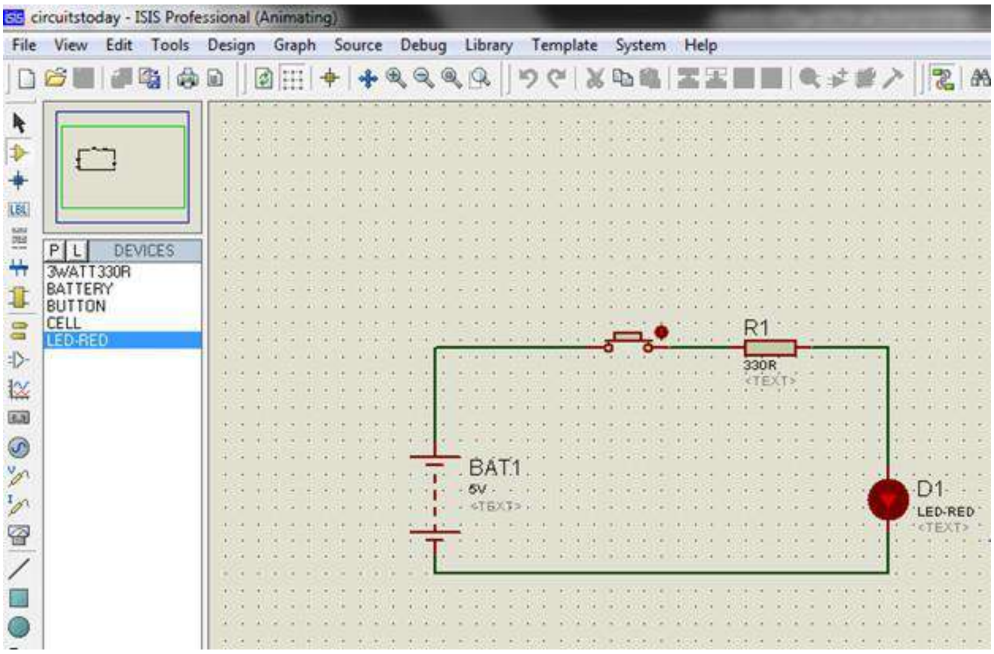


Fig. 3.2.7.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

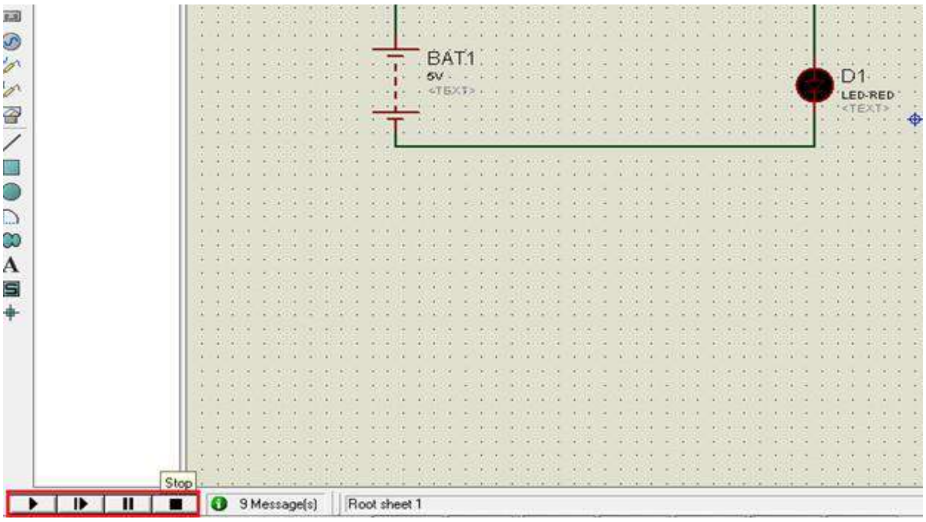


Fig. 3.2.7.13 Simulation Step-Pause-Stop Buttons

According to its developers, Things Speak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. Things Speak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates.

Things Speak was originally launched by IOT Bridge in 2010 as a service in support of IoT applications.

Things Speak has integrated support from the numerical computing software MATLAB from MathWorks, allowing Things Speak users to analyse and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license from Math works. [citation needed].

Things Speak has a close relationship with Math works, Inc. In fact, all of the Things Speak documentation is incorporated into the Math works' MATLAB documentation site and even enabling registered Math works user accounts as valid login credentials on the Things Speak website. The terms of service and privacy policy of ThingSpeak.com are between the agreeing user and Math works, Inc.

Setup Things Speak

Create a free MathWorks account or sign into Things Speak using an existing account.

Select the Things Speak channel you want your data to stream into. See Collect Data in a New Channel for help creating a new channel.

Record the following for the selected channel:

Channel ID, which is listed at the top of the channel view.

Write API key, which can be found on the API Keys tab of your channel view.

3.3 DEFINE THE MODULES

3.3.1 GLOBAL SYSTEM FOR MOBILE COMMUNICATION (GSM)

GSM (Global System for Mobile communications) is an open, digital cellular technology used for transmitting mobile voice and data services. GSM (Global System for Mobile communication) is a digital mobile telephone system that is widely used in Europe and other parts of the world. GSM uses a variation of Time Division Multiple Access (TDMA) and is the most widely used of the three digital wireless telephone technologies (TDMA, GSM, and CDMA). GSM digitizes and compresses data, then sends it down a channel with two other streams of user data, each in its own time slot. It operates at either the 900 MHz or 1,800 MHz frequency band. It supports voice calls and data transfer speeds of up to 9.6 kbit/s, together with the transmission of SMS (Short Message Service).

3.3.1.1 INTRODUCTION TO GSM

The **GSM** (*Global System for Mobile communications*) network is at the start of the 21st century, the most used mobile telephony standard in Europe. It is called as Second Generation

(2G) standard because communications occur in an entirely digital mode, unlike the first generation of portable telephones. When it was first standardized in 1982, it was called as **Group Special Mobile** and later, it became an international standard called **"Global System for Mobile communications"** in 1991.

In Europe, the GSM standard uses the 900 MHz and 1800 MHz frequency bands. In the United States, however, the frequency band used is the 1900 MHz band. For this reason, portable telephones that can operate in both Europe and the United States are called **tri-band** while those that operate only in Europe are called **bi-band**.

The GSM standard allows a maximum throughput of 9.6 kbps which allows transmission of voice and low-volume digital data like text messages (**SMS**, for *Short Message Service*) or multimedia messages (**MMS**, for *Multimedia Message Service*).

3.3.1.2 GSM STANDARDS

GSM uses narrowband TDMA, which allows eight simultaneous calls on the same radio frequency. There are three basic principles in multiple access, FDMA (Frequency Division Multiple Access), TDMA (Time Division Multiple Access), and CDMA (Code Division Multiple Access). All three principles allow multiple users to share the same physical channel. But the two competing technologies differ in the way user sharing the common resource.

TDMA allows the users to share the same frequency channel by dividing the signal into different time slots. Each user takes turn in a round robin fashion for transmitting and receiving over the channel. Here, users can only transmit in their respective time slot.

CDMA uses a spread spectrum technology that is it spreads the information contained in a particular signal of interest over a much greater bandwidth than the original signal. Unlike TDMA, in CDMA several users can transmit over the channel at the same time.

3.3.1.3 TDMA IN BRIEF

In late1980's, as a search to convert the existing analog network to digital to improve capacity, the cellular telecommunications industry association chose TDMA over FDMA. Time Division Multiple Access is a type of multiplexing where two or more channels of information are transmitted over the same link by allocating a different time interval for the transmission of each channel. The most complex implementation using TDMA principle is of GSM's (Global System for Mobile communication). To reduce the effect of co-channel interference, fading and multipath, the GSM technology can use frequency hopping, where a call jumps from one channel to another channel in a short interval.

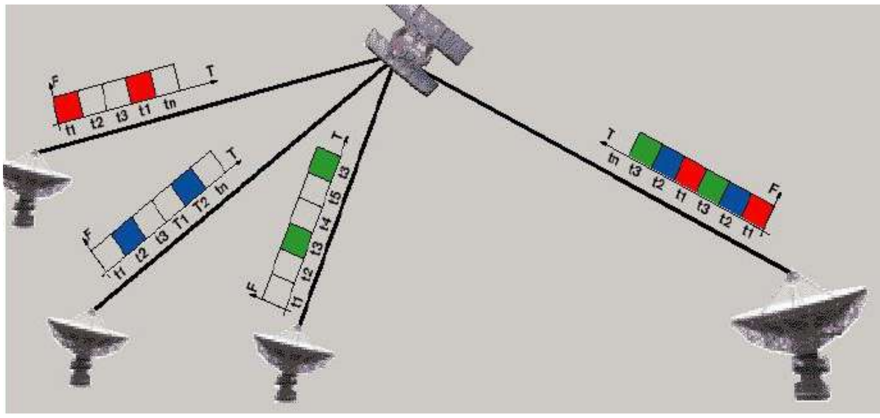


Fig.3.3.1.3 Time Division Multiplexing Access

TDMA systems still rely on switch to determine when to perform a handoff. Handoff occurs when a call is switched from one cell site to another while travelling. The TDMA handset constantly monitors the signals coming from other sites and reports it to the switch without caller's awareness. The switch then uses this information for making better choices for handoff at appropriate times. TDMA handset performs hard handoff, i.e., whenever the user moves from one site to another, it breaks the connection and then provides a new connection with the new site.

3.3.1.4 THE CONCEPT OF CELLULAR NETWORK

Mobile telephone networks are based on the concept of **cells**, circular zones that overlap to cover a geographical area.

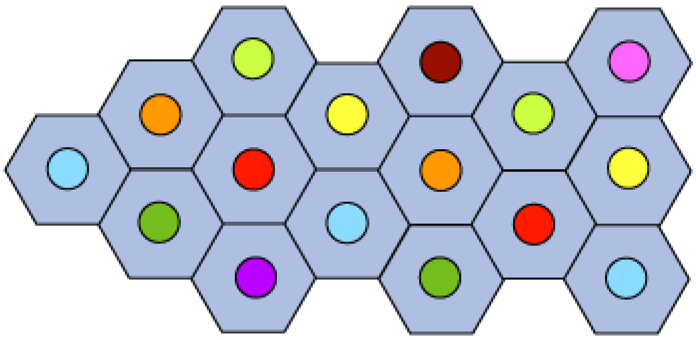


Fig.3.3.1.4 The Concept of cellular network

Cellular networks are based on the use of a central transmitter-receiver in each cell, called a "**base station**" (or *Base Transceiver Station*, written **BTS**). The smaller the radius of a cell, the higher is the available bandwidth. So, in highly populated urban areas, there are cells with a radius of a few hundred meters, while huge cells of up to 30 kilometers provide coverage in rural areas.

In a cellular network, each cell is surrounded by 6 neighboring cells (thus a cell is generally drawn as a hexagon). To avoid interference, adjacent cells cannot use the same frequency. In practice, two cells using the same frequency range must be separated by two to three times the diameter of the cell.

3.3.1.5 GSM MODEM

A GSM modem is a wireless modem that works with a GSM wireless network. A wireless modem behaves like a dial-up modem. The main difference between them is that a dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves.

Fig.3.3.1.5.1 GSM/GPRS Modem

A GSM modem can be an external device or a PC Card / PCMCIA Card. Typically, an external GSM modem is connected to a computer through a serial cable or a USB cable. A GSM modem in the form of a PC Card / PCMCIA Card is designed for use with a laptop computer. It should be inserted into one of the PC Card / PCMCIA Card slots of a laptop computer. Like a GSM mobile phone, a GSM modem requires a SIM card from a wireless carrier in order to operate.

A SIM card contains the following information:

- Subscriber telephone number (MSISDN)
- International subscriber number (**IMSI, International Mobile Subscriber Identity)**
- State of the SIM card
- Service code (operator)
- Authentication key
- PIN (***Personal Identification Code***)
- PUK (***Personal Unlock Code***)

Computers use AT commands to control modems. Both GSM modems and dial-up modems support a common set of standard AT commands. In addition to the standard AT commands, GSM modems support an extended set of AT commands. These extended AT commands are defined in the GSM standards. With the extended AT commands, the following operations can be performed:

- Reading, writing, and deleting SMS messages.
- Sending SMS messages.
- Monitoring the signal strength.
- Monitoring the charging status and charge level of the battery.
- Reading, writing, and searching phone book entries.

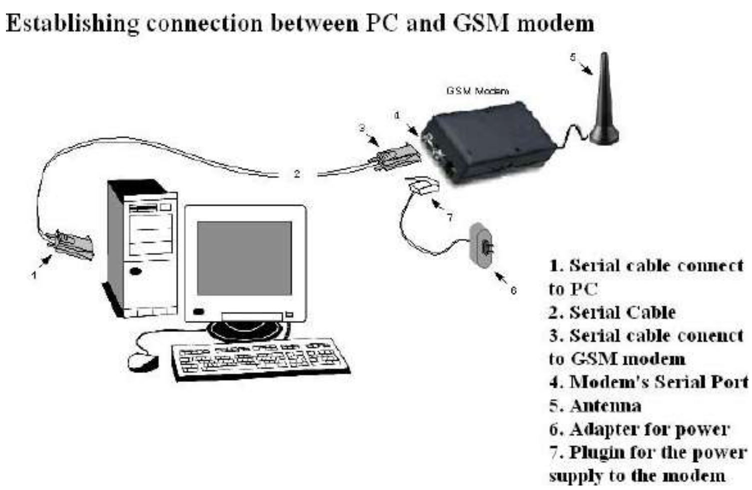


Fig.3.3.1.5.2 Establishing connection between PC and GSM modem

The number of SMS messages that can be processed by a GSM modem per minute is very low i.e., about 6 to 10 SMS messages per minute.

3.3.1.6 INRODUCTION TO AT COMMANDS

AT commands are instructions used to control a modem. AT is the abbreviation of Attention. Every command line starts with "AT" or "at". That is the reason, modem commands are called AT commands. Many of the commands that are used to control wired dial-up modems, such as ATD (Dial), ATA (Answer), ATH (Hook control) and ATO (Return to online data state) are also supported by GSM modems and mobile phones.

Besides this common AT command set, GSM modems and mobile phones support an AT command set that is specific to the GSM technology, which includes SMS-related commands like AT+CMGS (Send SMS message), AT+CMSS (Send SMS message from storage), AT+CMGL (List SMS messages) and AT+CMGR (Read SMS messages).

It should be noted that the starting "AT" is the prefix that informs the modem about the start of a command line. It is not part of the AT command name. For example, D is the actual AT command name in ATD and +CMGS is the actual AT command name in AT+CMGS.

Some of the tasks that can be done using AT commands with a GSM modem or mobile phone are listed below:

Get basic information about the mobile phone or GSM modem. For example, name of manufacturer (AT+CGMI), model number (AT+CGMM), IMEI number (International Mobile Equipment Identity) (AT+CGSN) and software version (AT+CGMR).

Get basic information about the subscriber. For example, MSISDN (AT+CNUM) and IMSI number (International Mobile Subscriber Identity) (AT+CIMI).

Get the status of the mobile phone or GSM/GPRS modem. For example, mobile phone activity status (AT+CPAS), mobile network registration status (AT+CREG), radio signal strength (AT+CSQ), battery charge level and battery charging status (AT+CBC).

Establish a data connection or voice connection to a remote modem (ATD, ATA, etc.).

Send and receive fax (ATD, ATA, AT+F*).

Send (AT+CMGS, AT+CMSS), read (AT+CMGR, AT+CMGL), write (AT+CMGW) or delete (AT+CMGD) SMS messages and obtain notifications of newly received SMS messages (AT+CNMI).

Read (AT+CPBR), write (AT+CPBW) or search (AT+CPBF) phonebook entries.

Perform security-related tasks, such as opening or closing facility locks (AT+CLCK), checking whether a facility is locked (AT+CLCK) and changing passwords (AT+CPWD). (Facility lock examples: SIM lock [a password must be given to the SIM card every time the mobile phone is switched on] and PH-SIM lock [a certain SIM card is associated with the mobile phone. To use other SIM cards with the mobile phone, a password must be entered.])

Control the presentation of result codes / error messages of AT commands. For example, the user can control whether to enable certain error messages (AT+CMEE) and whether error messages should be displayed in numeric format or verbose format (AT+CMEE=1 or AT+CMEE=2).

Get or change the configurations of the mobile phone or GSM/GPRS modem. For example, change the GSM network (AT+COPS), bearer service type (AT+CBST), radio link protocol parameters (AT+CRLP), SMS center address (AT+CSCA) and storage of SMS messages (AT+CPMS).

Save and restore configurations of the mobile phone or GSM/GPRS modem. For example, save (AT+CSAS) and restore (AT+CRES) settings related to SMS messaging such as the SMS center address.

It should be noted that the mobile phone manufacturers usually do not implement all AT commands, command parameters and parameter values in their mobile phones. Also, the behavior of the implemented AT commands may be different from that defined in the standard. In general, GSM modems, designed for wireless applications, have better support of AT commands than ordinary mobile phones.

3.3.2 GLOBAL POSITIONING SYSTEM

3.3.2.1 INTRODUCYION TO GPS

Fig 3.3.2.1 Interface ublox NEO-6M GPS Module with Arduino

Give your next Arduino project ability to sense locations with NEO-6M GPS Module that can track up to 22 satellites and identifies locations anywhere in the world. It may serve as a great launch pad for anyone looking to get into the world of GPS.

They are low power (suitable for battery powered devices), inexpensive, easy to interface with and are insanely popular among hobbyists.

3.3.2.2 HARDWARE OVERVIEW OF NEO-6M GPS MODULE

NEO-6M GPS Chip

At the heart of the module is a NEO-6M GPS chip from u-blox. The chip measures less than the size of a postage stamp but packs a surprising number of features into its little frame.

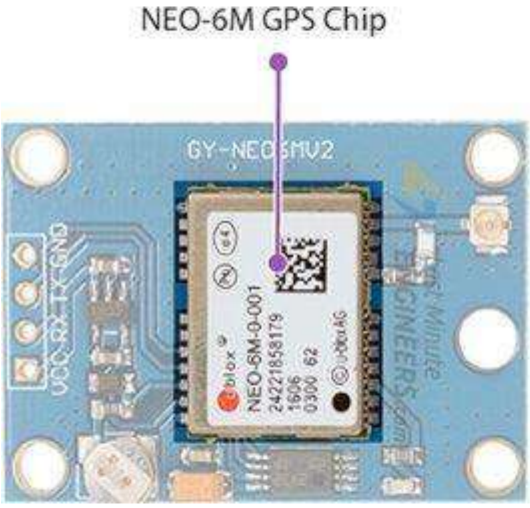


Fig.3.3.2.2 Overview of NEO-6M GPS Module

It can track up to 22 satellites on 50 channels and achieves the industry’s highest level of sensitivity i.e., -161 dB tracking, while consuming only 45mA supply current.

Unlike other GPS modules, it can do up to 5 location updates a second with 2.5m Horizontal position accuracy. The u-blox 6 positioning engine also boasts a Time-To-First Fix (TTFF) of under 1 second.

One of the best features the chip provides is Power Save Mode (PSM). It allows a reduction in system power consumption by selectively switching parts of the receiver ON and OFF. This dramatically reduces power consumption of the module to just 11mA making it suitable for power sensitive applications like GPS wristwatch.

3.3.2.3 POSITION FIX LED INDICATOR

The necessary data pins of NEO-6M GPS chip are broken out to 0.1″ pitch headers. This includes pins required for communication with a microcontroller over UART. The module supports baud rate from 4800bps to 230400bps with default baud of 9600.

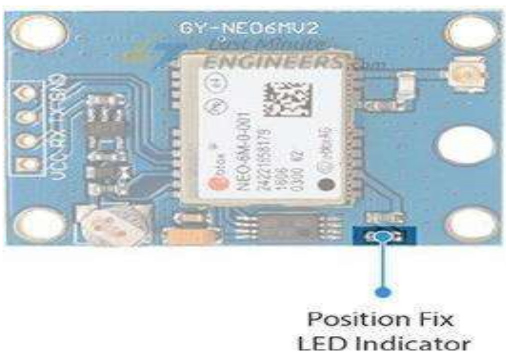


Fig.3.3.2.3 Position Fix LED Indicator

There is an LED on the NEO-6M GPS Module which indicates the status of Position Fix. It will blink at various rates depending on what state it’s in:

- No Blinking – It is searching for satellites.
- Blink every 1s – Position Fix is found (The module can see enough satellites).

3.3.2.4 3.3V LDO REGULATOR

Fig.3.3.2.4 3.3V LDO Regulator

The operating voltage of the NEO-6M chip is from 2.7 to 3.6V. But the good news is that the module comes with MIC5205 ultra-low dropout 3V3 regulator from MICREL.

The logic pins are also 5-volt tolerant, so we can easily connect it to an Arduino or any 5V logic microcontroller without using any logic level converter.

3.3.2.5 BATTERY & EEPROM

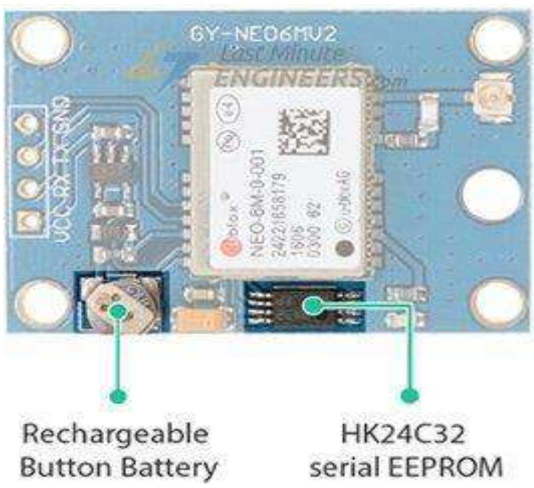


Fig. 3.3.2.5 Battery & EEPROM

The module is equipped with an HK24C32 two wire serial EEPROM. It is 4KB in size and connected to the NEO-6M chip via I2C.

The module also contains a rechargeable button battery which acts as a super-capacitor.

An EEPROM together with battery helps retain the battery backed RAM (BBR). The BBR contains clock data, latest position data (GNSS orbit data) and module configuration. But it's not meant for permanent data storage.

As the battery retains clock and last position, time to first fix (TTFF) significantly reduces to 1s. This allows much faster position locks.

Without the battery the GPS always cold start so the initial GPS lock takes more time.

The battery is automatically charged when power is applied and maintains data for up to two weeks without power.

3.3.2.6 ANTENNA

An antenna is required to use the module for any kind of communication. So, the module comes with a patch antenna having -161 dBm sensitivity.

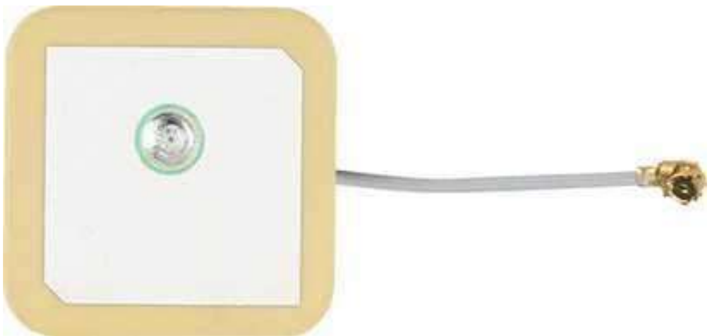


Fig. 3.3.2.6 Antenna

3.3.2.7 UFL CONNECTOR

You can snap-fit this antenna to small U.FL connector located on the module.

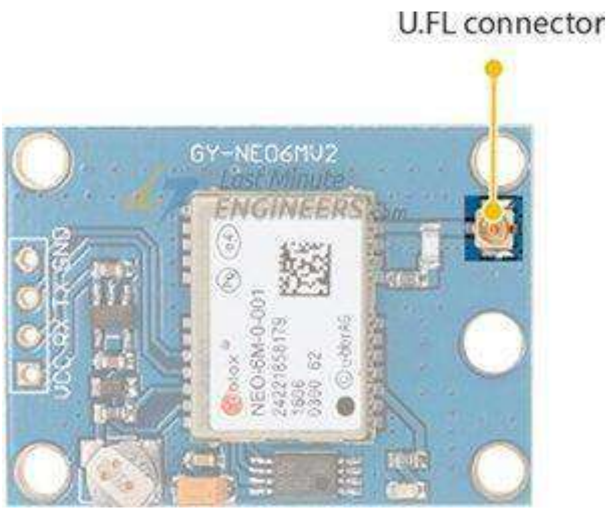


Fig.3.3.2.7 UFL Connector

Patch antenna is great for most projects. But if you want to achieve more sensitivity or put your module inside a metal case, you can also snap on any 3V active GPS antenna via the U.FL connector.

3.3.2.8 NEO-6M GPS MODULE PINOUT

The NEO-6M GPS module has total 4 pins that interface it to the outside world. The connections are as follows:

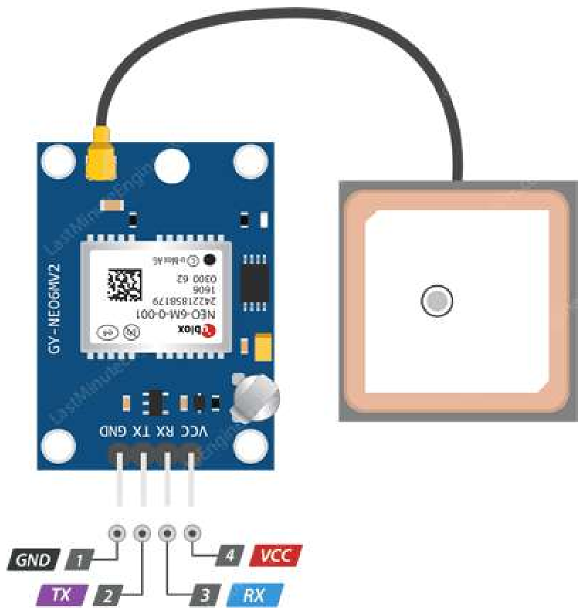


Fig.3.3.2.8 NEO-6M GPS MODULE PINOUT

GND is the Ground Pin and needs to be connected to GND pin on the Arduino.

TxD (Transmitter) pin is used for serial communication.

RxD (Receiver) pin is used for serial communication.

VCC supplies power for the module. You can directly connect it to the 5V pin on the Aíduino.

Wiring NEO-6M GPS module with Arduino UNO

Now that we know everything about the module, we can begin hooking it up to our Arduino!

Start by connecting the patch antenna to the U.FL connector. Remember to thread the U.FL cable through one of the mounting holes for robust connection.

The module usually comes with header pins unsoldered. So, you'll need to solder them.

Now, connect Tx and Rx pin on module to digital pin#2 and #3 respectively on Arduino; as we'll be using software serial to talk to the module.

Next, connect VCC pin to the 5V pin on the Arduino and GND to ground.

CHAPTER 4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM OF PROJECT

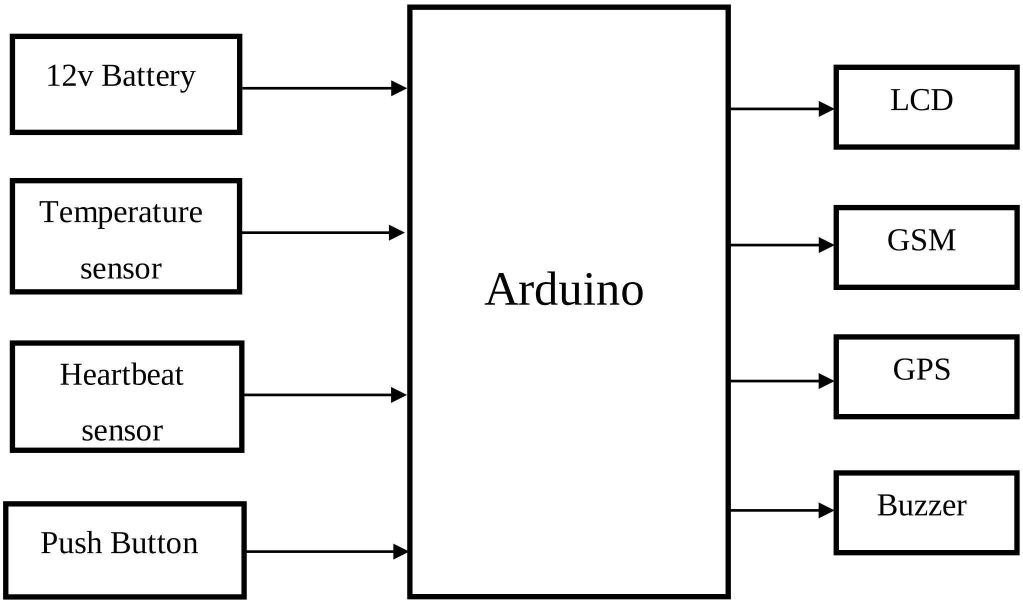


Fig.4.1 Block diagram of the Project.

4.2 FLOW CHART

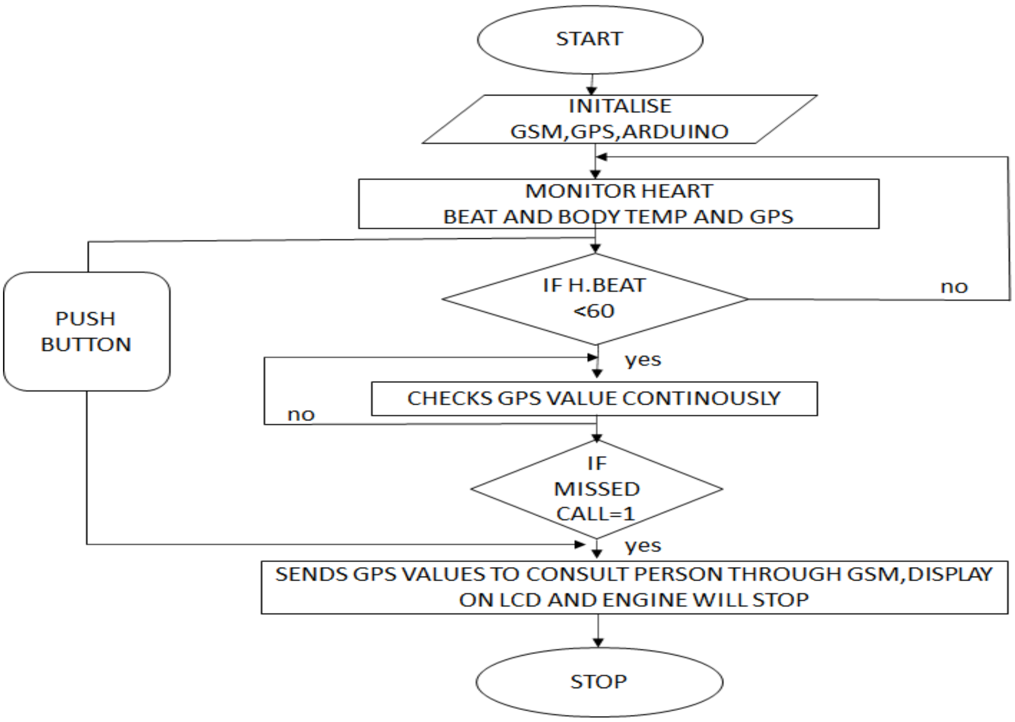


Fig.4.2 Flow Chart

4.3 IMPLEMENTATION STAGES

Stage-1: First on lcd we will get a message displaying “Welcome”.

Fig.4.3.1 Welcome Message

Step 2: It displaying biomedical sensors output and status of switch.

Fig.4.3.2 Biomedical sensors output and status of switch.

Step 3: Lcd displaying high temperature when threshold value is stricken.

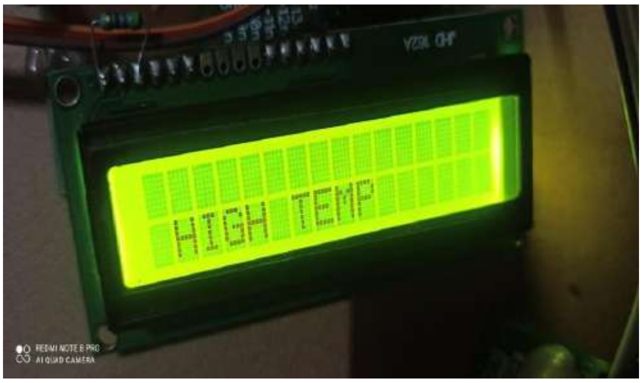


Fig.4.3.3 Lcd displaying high temperature

4.4 PROJECT CODE

```
#include <SoftwareSerial.h>
//SoftwareSerial mySerial(2,3);
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2,3);//rx , tx
#include <LiquidCrystal.h>
LiquidCrystal lcd (13,12,11,10,9,8);

unsigned int hbpin = 4;
int hcount = 0, hbval = 0;
int temp=A4;
int buz=7;
int sw=A5;
void serialFlush(){
while (Serial.available() > 0) {
char tt = Serial.read();
 }
}
void sendmsg(char *num,char * msg) //gsm
{
mySerial.print("AT+CMGS=\"");
mySerial.print(num);
mySerial.println("\""); delay (800);
mySerial.println(msg); delay (800);
mySerial.write(0x1a); delay (2000);
}

const char* number = "0123456789\0";/*Give the required mobile num to send the msg*/


char t;
```

```
 void setup ()
{
 char ret;
 pinMode(hbpin,  INPUT);
 pinMode(temp,   INPUT);
 pinMode(buz, OUTPUT);
 pinMode(sw, INPUT);

 mySerial.begin(9600);
 Serial.begin(9600);
 lcd.begin(16,2);
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("Welcome");
 Serial.println("WELCOME1");


 mySerial.println("AT"); delay (1000);
 mySerial.println("AT+CMGF=1"); delay (1000);
 mySerial.println("AT+CNMI=1,2,0,0"); delay (1000);
 mySerial.println("AT+CSMP=17,167,0,16"); delay (1000);
}

 char buff [200], buff2[100];
 char bf2[50];

  unsigned long duration = 0;
  int hbeat = 0;

 void loop ()
 {

 Serial.print("temperature = ");
 int td = analogRead(temp)/3;
 Serial.print(td);

 duration = pulseIn(hbpin, LOW, 5000000) / 1000;
 if (duration == 0)
 hbeat = 0;
 else
 hbeat = 64 + duration % 18;
 delay (1000);
 lcd.clear();
 lcd.setCursor(0, 1);
 hbval = hbeat;
 Serial.println(hbval);
```

```
 lcd.clear();
 //lcd.setCursor(0, 0);lcd.print(" T:");lcd.print(temp);
 lcd.setCursor(7, 0);lcd.print(" sw:");lcd.print(sw);
 lcd.setCursor(8, 1); lcd.print("HB:"); lcd.print(hbval); //lcd.print(" ");
delay (1000);



if (hbval > 78)
   {
    lcd.clear(); lcd.setCursor(0, 1); lcd.print("HIGH HEARTBEAT ");
    sendmsg(number,"HIGH HEARTBEAT ");
    digitalWrite(buz,LOW);
    delay (1000);
   }

if (hbval>10 &&hbval < 60)
   {
    lcd.clear(); lcd.setCursor(0, 1); lcd.print("LOW HEARTBEAT  ");
    sendmsg(number,"LOW HEARTBEAT ");
    digitalWrite(buz,LOW);
     delay (1000);
   }

   if(temp>35)
  {
   lcd.clear();lcd.setCursor(0, 0);lcd.print("temp high");
   sendmsg(number,"HIGH TEMPERATUR ");
   delay (1000);
   }

 if(sw==1)
  {
    lcd.clear();lcd.setCursor(0, 0);lcd.print("NEED HELP");
    sendmsg(number,"NEED HELP ");
    delay (1000);

   }

   }
```

4.5 RESULTS

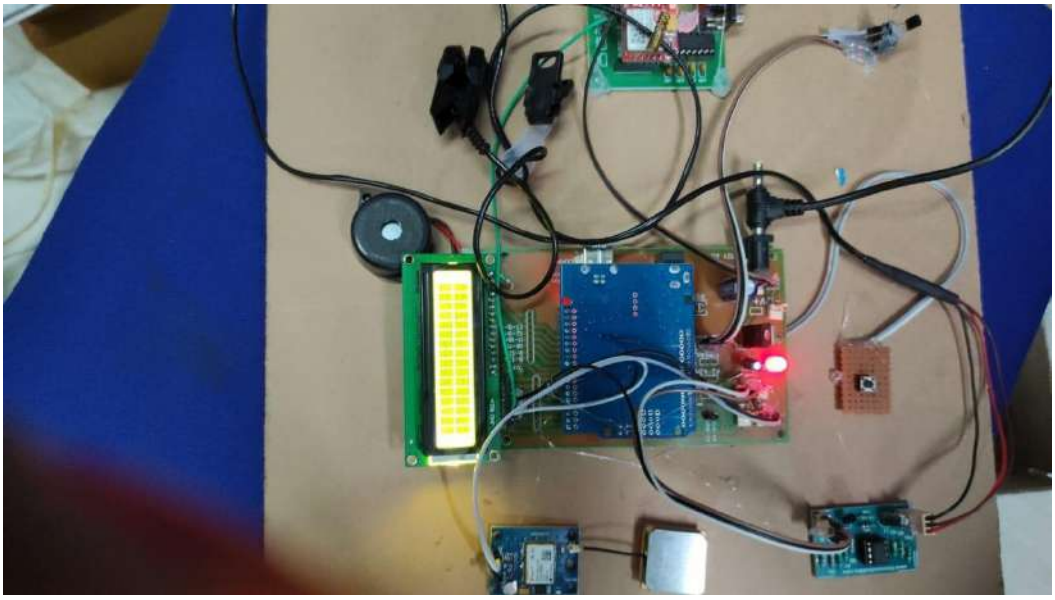


Fig 4.5 Final Output

CHAPTER 5

PROJECT TESTING

5.1 OVERVIEW OF TESTING METHODS

The soldier Health and Position Tracking System allows the military to track the current GPS position of a soldier and checks the health status including body temperature and heartbeats of the soldier. The System also consists of extra features with the help that a soldier can ask for help manually or send a distress signal to the military if he is in need. The GPS modem sends the latitude and longitude position with a link pattern with the help of that the military can track the current position of the soldier. The system is very helpful for getting health status information of soldiers and providing them instant help. Here are some techniques used to track current position and health condition. In today's era enemy warfare is a very important factor in any nation’s security. The national security mainly depends on the defense force. In this system we have come up with an idea of tracking soldiers as well as giving their status of their health condition and location during the war.

So, in implementation of the project, we encountered few problems such as the entire project depends on the wireless communication i.e., it depends on GSM module and signals. So, there are few problems we encountered and solved them.

- Signal problem in hill regions.
- Temperature differences in body that take place due to climatic changes.
- Rise of heartbeat in hill region when climbing.

1. Signal Problem:

   So, to overcome signal problem initially we had used helical antenna which perform well in open area and where there are no obstacles between antenna and signal towers which has many disadvantages, so it was replaced with another antenna which work works well in any environment and has proper bandwidth than helical antenna which is called as 3dBi.
   The 3dBi GSM antenna along with a U. FL to SMA adapter which can be obtained online for less than $3. You can snap-fit this antenna to small u.fl connector located on the top-left corner of the module. This type of antenna has a better performance and allows putting your module inside a metal case – as long the antenna is outside.

2. Temperature differences in body that take place due to climatic changes.

   Here we can change the value in program with respect to climate in outdoors which makes it flexible to any kind of climates can change the threshold values according to region of the soldier’s mission.

3. Rise of heartbeat in hill region when climbing.

   Here depending on region in which mission is given to soldier’s the threshold value can be varied.

CHAPTER-6

CONCLUSION AND FUTURESCOPE

6.1 CONCLUSION

- Above system, when completed, would make it easy to help in determining the health status of soldiers with measures of heart beats and body temperature.
- It would also help in tracking his position by using GPS modem and with GSM modem it can send all information to the base station so that further necessary action could be taken.

6.2 FUTURE SCOPE

- We can add a graphical display section to this project which helps to display a digital map which shows the position of all soldiers in the unit as they surround a block of buildings and launch their attacks.
- Welling to make live tracking possible.
- Helmet attached screens, accomplished by presenting information from maps and video using varieties of physiological sensors.

PUBLICATION

Submitted Paper in the Conference ICISECT - 21 with Paper ID (ICISECT21-0003) and got Acceptance for the Paper.

REFERENCES

[1]. P.chakravarth, S.Natarajan, M.Anto Bennete “GSM based soldier tracking system and monitoring using wireless communication” Department of Electronics and Communication, published in September 1st ,2017.

[2]. Akshita v.Armarkar, Deepika J, Punekar, Mrunali, p.Kapse, Shweta Kumari, Jayshree Shelka “Soldier health and position tracking system” Department of ETC engineering , volume 7 issue no..3

[3]. Patil Akshay , Shelake Balaji, Pinjari Raju, Mirajkar P.P “GPS based soldier tracking and health monitoring”, Department of ETC engineering, volume 40 issue:03 March 2017.

[4]. Shruthi Nikam, Supriya Patil, Prajkata Powar, V.S.Bendre “GPS based soldier tracking and health indication system”, Department of E&TC , PIMPRI CHINCHWA COLLEGE OF ENGINEERING, volume 2, issue 3, March 2013.

[5]. Govindiraj A, Dr.S.Sindhurja Banu “GPS based soldier tracking and health indication system with environmental analysis”, Department of Electronics and Communicaton Engineering volume 2, issue 12, Decemember 2013.

[6]. Zeeshan Raza,Kamaran Liaquat,Shahazad Ashraf “Monitoring of soldier health and transmition of secret codes”,NFC instutation of engineering volume 8 no 2,june 2017.

[7]. Shubhangi Gupta, Shivani Kulshrestha, Divya Singh, Ashish Kumar, Er.Hitendra Singh “GPS and GSM based soldier health monitoring and tracking system” issue 3 volume 6, March 2017.

[8]. JouniRantakoko,JoakimRydell and peter Sromback, “Accurate and Realiable soldier and first responder Poasitioning :Multisensor System and co-operative localization” April-2011

[9]. RavidraB.Sathe ,A.S.Bhide. “Gps-based soldier tracking and health monitoring system” conference on Advances in communication and computing April 2011-12.

[10]. VincentPereira,AudreyGiremus,andericgrivel “Modeling of multipath environment using copulas For particle filtering based GPS Navigation” June-2012

[11]. Shruti Nikam1, Supriya Patil2, Prajkta Powar3, V.S.Bendre-“ Gpsbased soldier tracking and health indication system”International Journal of Advanced

Research in Electrical, Electronics and Instrumentation Engineering.Vol. 2, Issue 3, March 2013.

[12]. Akshita v.Armarkar, Deepika J, Punekar, Mrunali, p.Kapse, Shweta Kumari, Jayshree Shelka “Soldier health and position tracking system” Department of ETC engineering , volume 7 issue no..3

[13]. Patil Akshay , Shelake Balaji, Pinjari Raju, Mirajkar P.P “GPS based soldier tracking and health monitoring”, Department of ETC engineering, volume 40 issue:03 March 2017.

BIBILOGRAPHY

[1]. JouniRantakoko,JoakimRydell and peter Sromback, “Accurate and Realiable soldier and first responder Poasitioning :Multisensor System and co-operative localization” April-2011

[2]. RavidraB.Sathe ,A.S.Bhide. “Gps-based soldier tracking and health monitoring system” conference on Advances in communication and computing April 2011-12.

[3]. VincentPereira,AudreyGiremus,andericgrivel “Modeling of multipath environment using copulas For particle filtering based GPS Navigation” June-2012

[4]. Shruti Nikam1, Supriya Patil2, Prajkta Powar3, V.S.Bendre-“ Gpsbased soldier tracking and health indication system”International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering.Vol. 2, Issue 3, March 2013.

[5]. Akshita v.Armarkar, Deepika J, Punekar, Mrunali, p.Kapse, Shweta Kumari, Jayshree Shelka “Soldier health and position tracking system” Department of ETC engineering , volume 7 issue no..3

[6]. Patil Akshay , Shelake Balaji, Pinjari Raju, Mirajkar P.P “GPS based soldier tracking and health monitoring”, Department of ETC engineering, volume 40 issue:03 March 2017.

APPENDICES

The project shows that the health monitoring and tracking system for soldiers. The proposed system can be mounted on the soldier's body to track their health status and current location using GPS. This information will be transmitted to the control room through IoT. The proposed system comprises of tiny wearable physiological equipment's, sensors, transmission modules. Hence, with the use of the proposed equipment, it is possible to implement a low-cost mechanism to protect the valuable human life on the battlefield.

A

PROJECT REPORT

On

STUDIES ON DIFFERENT CNN ALGORITHMS FOR FACE SKIN DISEASE CLASSIFICATION BASED ON CLINICAL IMAGE

Submitted by

1)G.S.Rohan(17K81A04K7) 2)D.Rammohan(17K81A04J9)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATIONS ENGINEERING

Under The Esteemed Guidance of

Mrs.K.DIVYA VANI

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATIONS ENGINEERING

St.MARTIN'S ENGINEERING COLLEGE

An Autonomous Institute

Dhulapally, Secunderabad – 500 100

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

AN UGC AUTONOMOUS INSTITUTE

Dhulapally, Secunderabad-500 100

NBA & NAAC A+ Accredited

Department Of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the project entitled **STUDIES ON DIFFERENT CNN ALGORITHMS FOR FACE SKIN DISEASE CLASSIFICATION BASED ON CLINICAL IMAGE**, is being submitted by **G.S.Rohan(17K81A04K7), D.Rammohan(17K81A04J9)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONCS AND COMMUNICATIONS ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Project guide	**Head of the Department**
MRS.K.DIVYA VANI	**Dr.B.HARI KRISHNA**
Assistant Professor	**Professor**
Department of ECE	**Department of ECE**
Internal Examiner	External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **G.S.ROHAN** WITH ROLL NO.**17K81A04K7, D.RAMMOHAN** WITH ROLL NO.**17K81A04J9**, OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED –**STUDIES ON DIFFERENT CNN ALGORITHMS FOR FACE SKIN DISEASE CLASSIFICATION BASED ON CLINICAL IMAGES**‖ AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS LASYA
IT SOLUTIONS PVT LTD.

DECLARATION

We, the student of **Bachelor of Technology** in Department of **ELECTRONICS AND COMMUNICATIONS ENGINEERING**‘, session: 2017 – 2021, St. Martin‘s Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled –**STUDIES ON DIFFERENT CNN ALGORITHMS FOR FACE SKIN DISEASE CLASSIFICATION BASED ON CLINICAL IMAGE**‖ is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

Mr.G.S.Rohan	**Mr.D.Rammohan**
(17K81A04K7)	**(17K81A04J9)**

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to our College Management for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, our beloved **Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna**, **Head &Professor**, the Dept. of Electronics and Communications Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our dept. Project Coordinator **Dr.A.Anand, Professor** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide, **Mrs.K.Divya Vani, Assistant Professor,** Department of ECE and our class Project Coordinator **Ms.P.Pushpa, Assistant Professor,** Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level Project Coordinators **Dr.Maalyadri**, **Professor in IT And Dr.R.Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

G.S.ROHAN	D.RAMMOHAN
(17K81A04K7)	(17K81A04J9)

ABSTRACT

Multiple input and multiple output orthogonal frequency division multiplexing (MIMO-OFDM) is used to improve the performance and capacity of wireless communication system. OFDM is an efficient and promising modulation technique or wireless transmission due to its high spectral efficiency and robustness to frequency selective adding channels. However, it has high peak to average ratio (PAPR) which is a main disadvantage. Partial transmit sequence (PTS) is one o the most widely used technique that gives a better performance in PAPR reduction or MIMO-OFDM system. However the computational complexity of traditional PTS method is tremendous. In this paper, A Low-Complexity Hybrid Subblock Segmentation PTS Scheme or PAPR Reduction in MIMO-OFDM System is proposed. For analytic purposes, we derive computational complexity expressions or the proposed segmentation method and analyze the computational complexity o the proposed segmentation method compared with that o the random segmentation method which has the best Peak to Average Power Ratio (PAPR) reduction perormance. The simulation results show that the PAPR reduction performance is degraded only slightly compared with random segmentation method.

TABLE OF CONTENTS

.

LIST OF TABLES

LIST OF FIGURES

GLOSSARY OF COMMONLY USED TERM:

JPEG/JFIF-JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format.

EXIF-The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras.

TIFF-The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension.

PNG-The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas.

GIF-GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images.

BMP-The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

Chapter 1

INTRODUCTION

Based on a survey in 2010, skin diseases had the fourth leading cause of nonfatal disease burden in the world, and three of the world‘s most common diseases were skin diseases . Skin diseases have caused enormous economic burdens both in high-income and low-income countries. For each individual, skin problems can have adverse effects on all aspects of life, including interpersonal relationships, work, social functioning, physical activity and mental health.

1.0 OVERVIEW OF THE PROJECT

Usually, skin diseases cause skin lesions, scales, plaques, pigmentation and other symptoms on the patient‘s skin . These symptoms result in long-term pain and disfigurement. Such damage not only injures physical health but also contribute to serious mental problems, especially when such damage occurs on face. Studies showed that patients with primary skin diseases (such as psoriasis, alopecia areata and vitiligo) have a higher potential for mental problems, such as anxiety and depression. In addition, some skin disease treatments also have the possibility of inducing mental illness (such as isotretinoin, an acne medication, may induce suicidal depression).

1.1 OBJECTIVES OF THE STUDY

Facial skin is exposed to the air almost all the time, so it has a higher risk of being damaged than other areas. Moreover, facial skin is the most important part of the body for people‘s appearance, so people are more concerned about their facial skin health than skin health anywhere else. Along with the availability of massive amounts data brought by the Internet [9] and the improvement of computing power brought by advanced hardware, deep learning algorithms have achieved human-level performance in many fields. For example, convolutional neural networks (CNNs) have made many breakthroughs in the field of medical image processing, especially for pathological, CT and MRI images, which have rigid features and high resolution. However, research on clinical images is relatively insufficient. For these reasons, clinical images always contain a very complex context, and it is hard to control the conditions of acquiring the image. These circumstances make image processing tasks difficult.

1.2 SCOPE OF THE STUDY

Furthermore, datasets of a certain part of the body, especially the face, re relatively scarce. At present, most of the available datasets are not clearly labeled with information on the body parts; for some datasets that provide this information, the proportion of facial images is always small . All of these conditions make research difficult. Therefore, this paper first constructed a skin image dataset based on 6 common facial skin diseases (seborrheic keratosis (SK), actinic keratosis (AK), rosacea (ROS), lupus erythematosus (LE), basal cell carcinoma (BCC), and squamous cell carcinoma (SCC)). It includes 2,656 facial images for a total of 4,394 images. We focus on these diseases for the following reasons: 1) LE, ROS, BCC and SCC frequently occur on the face; 2) AK and SK usually transition from benign to malignant without timely treatment.

Based on the dataset, experiments were carried out on 5 different CNN structures to verify whether these methods can effectively diagnose facial skin diseases using clinical images. In the test set consisting entirely of facial images, the structure named Inception-ResNet-v2 achieved the highest average precision (77.0%).

1.3 MATERIAL REQUIREMENT

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

1.4 PROCUMENT OF EQUIPMENT

1.4.1 THE MATLAB SYSTEM

The MATLAB system consists of five main parts

- **Development Environment**:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library**:

  This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language**:

  This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both ―programming in the small‖ to rapidly create quick and dirty throw-away programs, and ―programming in the large‖ to create large and complex application programs.

- **Graphics:**

  MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

  This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

GRAPHICAL USER INTERFACE (GUI):

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.

- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type

guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted,

GUIDE opens a new (i.e., blank) window.

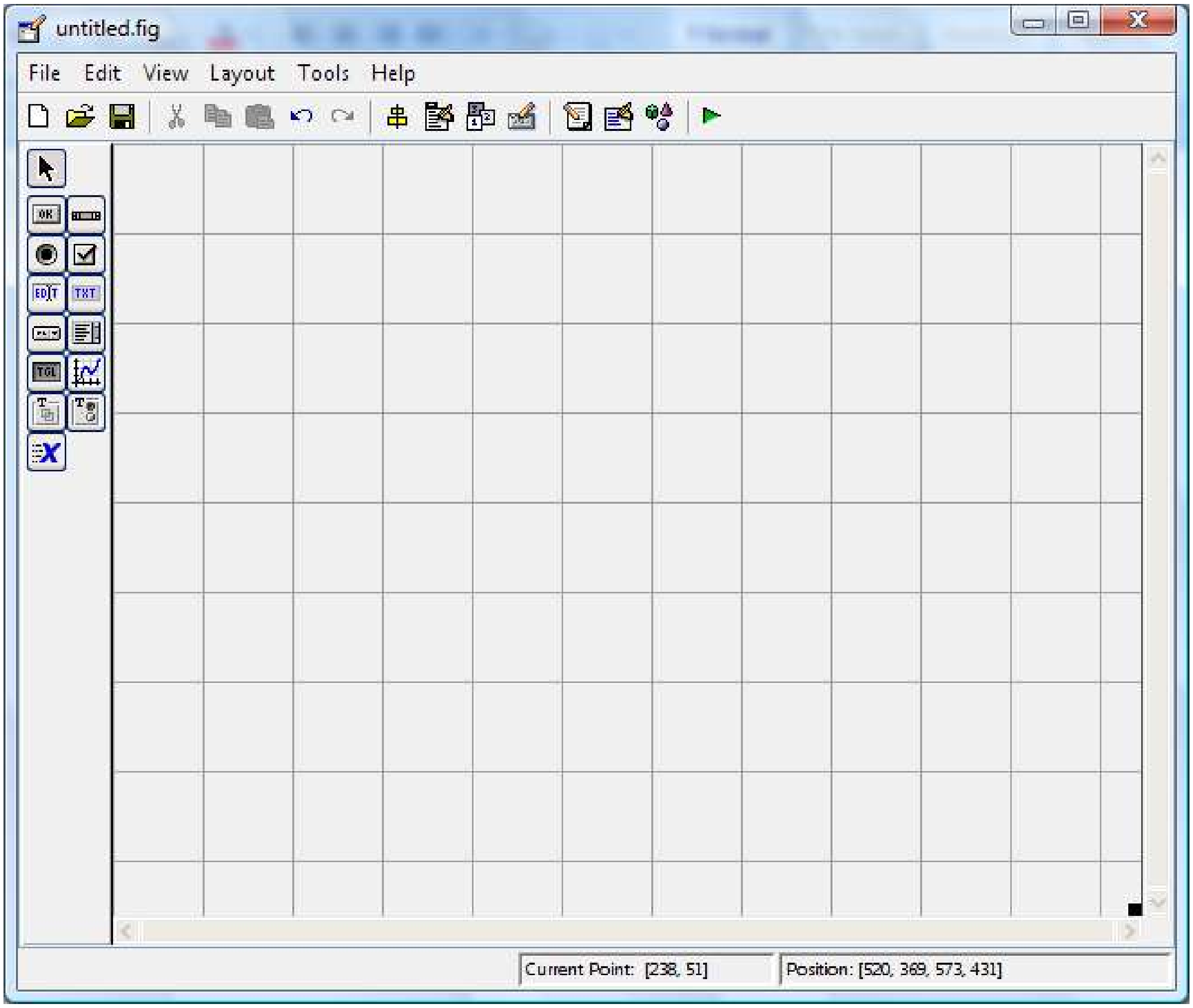


Fig:1.4.1 Matlab procedure

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

Chapter 2

LITERATURE REVIEW

2.0 LITERATURE REVIEW ON RESEARCH AREA:

Many studies have applied deep learning algorithms to skin diseases . For example, the performance in the task of classifying skin tumors using the Inception-v3 network has reached the level of professional dermatologists; for nine classes of tumors, a computer achieved an accuracy of 55.4%, and two dermatologists achieved accuracies of 53.3% and 55.0% . Using the same network structure, achieved an accuracy of 87.25 ± 2.24% on the dermoscopic images for four common skin diseases, including SK, BCC, psoriasis and melanocytic nevus. These studies show that current deep learning methods have the potential to be applied to dermatoses.

At the same time, the application of deep learning to facerelated diseases is also promising. Reference designed a deep learning algorithm called DeepGestalt and trained their model on more than 17,000 real facial images of genetic syndromes, and this model can identify more than 200 genetic syndromes using facial images with relatively high precision. Reference investigated using CNNs to classify acne into different severity grades ranging from clear to severe, and their results show that the accuracy of their method outperformed expert physicians. Initially, we investigated the proportion of facial images in the most commonly used public datasets for skin disease, which include AtlasDerm , DermlS , the ISIC Archive , Derm101 and Dermnet . Most of these datasets did not provide information about body parts. In [19], which does provide body parts information, there were only 195 facial images. It is difficult to perform further research on facial skin diseases using such limited data. As a result, building a specialized dataset for face images is extremely necessary for our research.

2.1 REVIEW ON RELATED LITERATURE

A neural network is a mathematical model inspired by the transfer process of biological neuron information, and its purpose is to learn a mapping from input to output. By using a loss function as a constraint and backpropagation to optimize the parameters, this method can automatically learn complex tasks for different fields. This method has reduced the need for human labor, such as manual feature extraction and data reconstruction for classification. A CNN is a type of neural network. It generally consists of an input layer, many hidden convolutional layers, and an output layer. Using this structure, the model can include a large number of parameters and obtain some usable properties, such as equivariance, for image-related tasks.

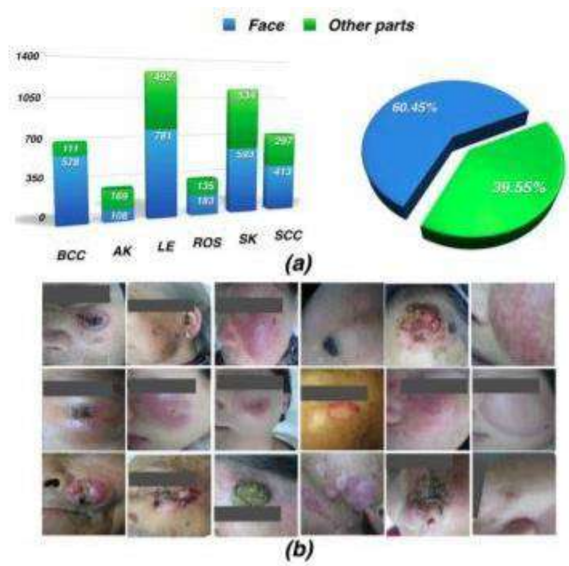


Fig:2.1 Diseases in the dataset

FIGURE 1. The stacked cylinder diagram ((a) left) and the pie chart ((a) right) show the distribution of the images of face and other body parts for different diseases in the dataset; (b) shows some examples of the dataset.

TABLE 1. Summary of dataset.

	IMAGE QTY	Face image	Train image	Test image
BCC	689	578	623	66
AK	277	108	219	58
LE	1273	781	1188	85
ROS	318	183	263	55
SK	1127	593	1075	52
SCC	710	413	638	72
Total	4394	2656	4006	388

Fig:2.1.1 Summary of dataset

we used five mainstream CNN algorithms that have been pretrained on ImageNet [9]. These five structures include ResNet-50, Inception-v3, DenseNet121, Xception and Inception-ResNet-v2. We used same pre-process for these images, including random reverse and crop. And to address the problem of data imbalance, we used different weights in the cost function for different diseases.

ResNet adds connections between the shallow and deep layers of the network. Such connections directly transmit the information of the shallow layer to the deep layer. On the other hand, the propagation of the gradient to the shallow layer during backpropagation greatly increases the number of network layers .

The basic module of the Inception structure is the inception block. There are different kernels in a block, and each type of kernel has a different shape; the output of the block is combines the output from different kernels. This improves the diversity of the network in terms of width and the diversity of the scale of the receptive field. Therefore, the model improved its recognition performance for objects with different sizes.

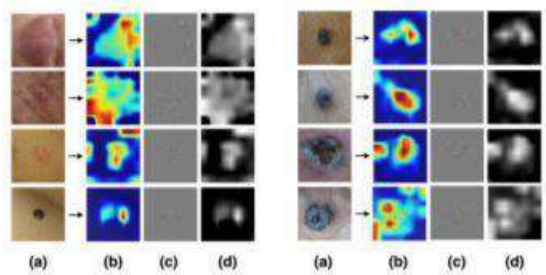


Fig:2.1.2 image heatmap

FIGURE 2. (a) is the input images, (b) is the heatmap for the combination of input images and the corresponding output of grad-CAM(d). (c) is the output of guided propagation.

DenseNet adds connections between each two layers; that is, the output feature maps of each layer will be used as the input for all subsequent layers. Using these dense connections, the network reuses features, thereby improving performance with fewer parameters, which makes the calculation more efficient .

Xception is an updated version of the Inception structure. Xception improves the Inception module with a depthwise separable convolution. This change decouples spatial correlations and cross-channel correlations. It can obtain a better performance than Inception-v3 with the same parameters . To some extent, Inception-ResNet is a combination of Inception and ResNet structures. By adding a residual connection to the Inception network, it can train deeper networks while maintaining the scale diversity of the network, thereby enhancing the performance .

we used the same 300∗300 input images for each network and did not change the basic structure from that in their origin paper. We replaced the first fully connected layer behind the last convolutional layer with global average pooling and a 1∗1 convolution to reduce the number of parameters and maintain spatial information. Finally, we used a 1024-d fully connected layer in each network and then used a softmax or logistic regression classifier to obtain 6 confidence outputs for six facial skin diseases. More details about the model structures are shown in Table 2, where inception block, dense block, transition layer,andinception resnet blockare modules that are the same as those from the origin papers.

2.2 CONCLUSION ON REVIEW:

This paper performed experiments using five mainstream CNN structures for the clinical image diagnosis of six common facial skin diseases and constructed a data set consisting mainly of facial skin disease images. The results demonstrate that CNNs have the ability to recognize facial skin diseases. Based on our experiments, we determined that different models to diagnose diseases on different body parts should be used. Furthermore, our experiments also showed that a more reasonable network structure could improve the performance of the model. The performance of the current network structure has been satisfactory in some diseases, but the overall performance has yet to be improved. As a result, if we want that people to actually use this technique to check their face skin health in their daily life, specialized improvements should be developed.

In our opinion, the application of artificial intelligence techniques in the medical field is not sufficient, and the datasets from this field should be improved both in quantity and quality. With the increasing amount of facial image data of various skin diseases and the continuous improvement of the network structure, CNN-based facial skin disease diagnosis algorithms will continue to improve in performance. We believe that, in the future, patients will use convenient CNN-based applications to keep their face skin healthy.

CHAPTER-3

PROJECT DESIGN

3.0 OVERVIEW OF THE PROJECT:

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An important characteristic underlying the design of image processing systems is the significant level of testing & experimentation that normally is required before arriving at an acceptable solution. This characteristic implies that the ability to formulate approaches &quickly prototype candidate solutions generally plays a major role in reducing the cost & time required to arrive at a viable system implementation.

3.1 EQUIPMENT ANALYSIS:

- Math AND Computation
- Algorithm Development
- Modeling, stimulation And Prototyping
- Data Analysis, Exploration And Visualization
- Application Development, including Graphical User Interface Building

3.2 DEFINE THE MODULES:

What is DIP

An image may be defined as a two-dimensional function f(x, y), where x & y are spatial coordinates, & the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital image by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensor, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis& computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting

& somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low-level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low- level process is characterized by the fact that both its inputs & outputs are images.

Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images. Finally higher- level processing involves –Making sense‖ of an ensemble of recognized objects, as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

What is an image?

An image is represented as a two dimensional function f(x, y) where x and y are spatial co-ordinates and the amplitude of _f' at any pair of coordinates (x, y) is called the intensity of the image at that point.

3.3 MODULE FUNCTIONALITIES:

Gray scale image:

A grayscale image is a function I (xylem) of the two spatial coordinates of the image plane.

I(x, y) is the intensity of the image at the point (x, y) on the image plane.

I (xylem) takes non-negative values assume the image is bounded by a rectangle [0, a] ×[0, b]I: [0, a] × [0, b] → [0, info)

Color image:

It can be represented by three functions, R (xylem) for red, G (xylem) for green *and* B (xylem) for blue.

An image may be continuous with respect to the x and y coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate's values is called sampling. Digitizing the amplitude values is called quantization.

Coordinate convention:

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image f(x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size M X N. The values of the coordinates (xylem) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates.

In many image processing books, the image origin is defined to be at (xylem)=(0,0).The next coordinate values along the first row of the image are (xylem)=(0,1).It is important to keep in mind that the notation (0,1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Following figure shows the coordinate convention. Note that x ranges from 0 to M-1 and y from 0 to N-1 in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (xylem) the toolbox uses the notation (race) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate topples, (alb), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentation refers to the coordinates. Less frequently the toolbox also employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refers to rows. This is the opposite of our use of variables x and y.

Image as Matrices:

The preceding discussion leads to the following representation for a digitized image function:

$$\text{f (xylem)=} \begin{matrix} f(0,0) & f(0,1) & \ldots\ldots\ldots.. & f(0,N-1) \\ f(1,0) & f(1,1) & \ldots\ldots\ldots\ldots & f(1,N-1) \\ . & . & & . \\ . & . & & . \end{matrix}$$

f (M-1,0)   f(M-1,1).................f(M-1,N-1)

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements. A digital image can be represented naturally as a MATLAB matrix:

f (1,1)   f(1,2)........... f(1,N)

f (2,1)   f(2,2)........... f (2,N)

.   .   .

f =   .   .   .

f (M,1)   f(M,2).......... f(M,N)

Where f (1,1) = f(0,0) (note the use of a monoscope font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation f(p ,q) denotes the element located in row p and the column q. For example f(6,2) is the element in the sixth row and second column of the matrix f. Typically we use the letters M and N respectively to denote the number of rows and columns in a matrix. A 1xN matrix is called a row vector whereas an Mx1 matrix is called a column vector. A 1x1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using mono-scope characters. We use conventional Roman, italic notation such as f(x ,y), for mathematical expressions

Reading Images:

Images are read into the MATLAB environment using function imread whose syntax is

Imread (_filename‘)

Format name	Description	recognized extension
TIFF	Tagged Image File Format	.tif, .tiff
JPEG	Joint Photograph Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	.xwd

Here filename is a spring containing the complete of the image file(including any applicable extension).For example the command line

```
>> f = imread (_8. jpg‘);
```

Reads the JPEG (above table) image chestxray into image array f. Note the use of single quotes (_) to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output If a semicolon is not included. MATLAB displays the results of the operation(s) specified in that line. The prompt symbol (>>) designates the beginning of a command line, as it appears in the MATLAB command window.

Data Classes:

Although we work with integers coordinates the values of pixels themselves are not restricted to be integers in MATLAB. Table above list various data classes supported by MATLAB and IPT are representing pixels values. The first eight entries in the table are refers to as numeric data classes. The ninth entry is the char class and, as shown, the last entry is referred to as logical data class.

All numeric computations in MATLAB are done in double quantities, so this is also a frequent data class encounter in image processing applications. Class unit 8 also is encountered frequently, especially when reading data from storages devices, as 8 bit images are most common representations found in practice. These two data classes, classes logical, and, to a lesser degree, class unit 16 constitute the primary data classes on which we focus. Many ipt functions however support all the data classes listed in table. Data class double requires 8 bytes to represent a number uint8 and int 8 require one byte each, uint16 and int16 requires 2bytes and unit 32.

Name **Description**

Double Double _ precision, floating_ point numbers the Approximate.

Uint8 unsigned 8_bit integers in the range [0,255] (1byte per Element).

Uint16 unsigned 16_bit integers in the range [0, 65535] (2byte per element).

Uint 32 unsigned 32_bit integers in the range [0, 4294967295](4 bytes per element). Int8 signed 8_bit integers in the range [-128,127] 1 byte per element)

Int 16 signed 16_byte integers in the range [32768, 32767] (2 bytes per element).

Int 32 Signed 32_byte integers in the range [-2147483648, 21474833647] (4 byte per element).

Single single _precision floating _point numbers with values

In the approximate range (4 bytes per elements)

Char characters (2 bytes per elements).

Logical values are 0 to 1 (1byte per element).

Int 32 and single required 4 bytes each. The char data class holds characters in Unicode representation. A character string is merely a 1*n array of characters logical array contains only the values 0 to 1,with each element being stored in memory using function logical or by using relational operators.

Image Types:

The toolbox supports four types of images:

1 .Intensity images;

2. Binary images;

3. Indexed images;

4. R G B images.

Most monochrome image processing operations are carried out using binary or intensity images, so our initial focus is on these two image types. Indexed and RGB colour images.

Intensity Images:

An intensity image is a data matrix whose values have been scaled to represent intentions. When the elements of an intensity image are of class unit8, or class unit 16, they have integer values in the range [0,255] and [0, 65535], respectively. If the image is of class double, the values are floating point numbers. Values of scaled, double intensity images are in the range [0, 1] by convention.

Binary Images:

Binary images have a very specific meaning in MATLAB.A binary image is a logical array 0s and1s.Thus, an array of 0s and 1s whose values are of data class, say unit8, is not considered as a binary image in MATLAB .A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, we create an array B using the statement.

B=logical (A)

If A contains elements other than 0s and 1s.Use of the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.

Using relational and logical operators also creates logical arrays.

To test if an array is logical we use the I logical function: islogical(c).

If c is a logical array, this function returns a 1.Otherwise returns a 0. Logical array can be converted to numeric arrays using the data class conversion functions.

Indexed Images:

An indexed image has two components:

A data matrix integer, x

A color map matrix, map

Matrix map is an m*3 arrays of class double containing floating point values in the range [0, 1].The length m of the map are equal to the number of colors it defines. Each row of map specifies the red, green and blue components of a single color. An indexed images uses ―direct mapping‖ of pixel intensity values color map values. The color of each pixel is determined by using the corresponding value the integer matrix x as a pointer in to map. If x is of class double ,then all of its components with values less than or equal to 1 point to the first row in map, all components with value 2 point to the second row and so on. If x is of class units or unit 16, then all components value 0 point to the first row in map, all components with value 1 point to the second and so on.

RGB Image:

An RGB color image is an M*N*3 array of color pixels where each color pixel is triplet corresponding to the red, green and blue components of an RGB image, at a specific spatial location. An RGB image may be viewed as –stack‖ of three gray scale images that when fed in to the red, green and blue inputs of a color monitor

Produce a color image on the screen. Convention the three images forming an RGB color image are referred to as the red, green and blue components images. The data class of the components images determines their range of values. If an RGB image is of class double the range of values is [0, 1].

Similarly the range of values is [0,255] or [0, 65535].For RGB images of class units or unit 16 respectively. The number of bits use to represents the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8bit image, the corresponding RGB image is said to be 24 bits deep.

Generally, the number of bits in all component images is the same. In this case the number of possible color in an RGB image is (2^b) ^3, where b is a number of bits in each component image. For the 8bit case the number is 16,777,216 colors

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

3.4 IMAGE

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Fig 3.4 General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

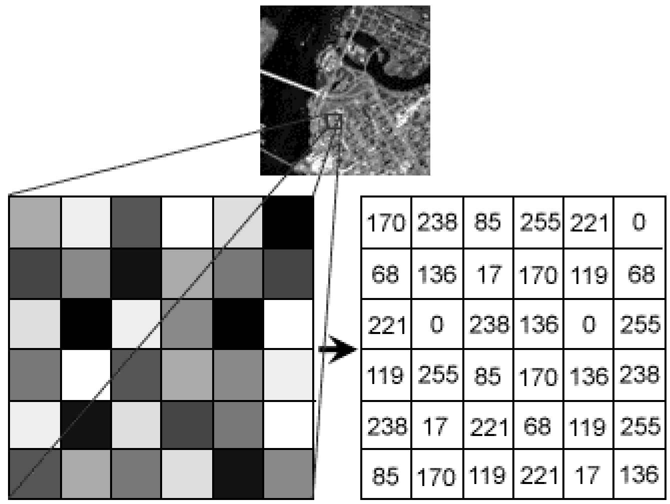


Fig 3.5 Image pixel

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

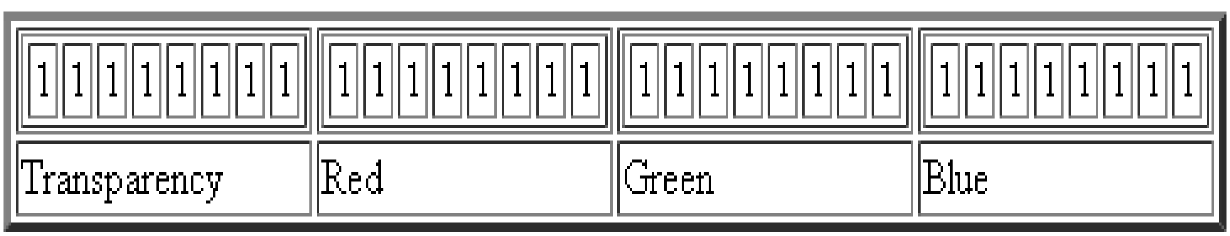


Fig 3.6 Transparency image

3.5 IMAGE FILE SIZES:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

3.6 IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

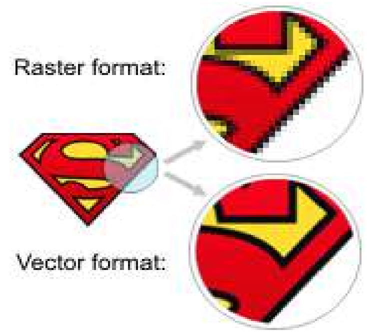


Fig3.7 Resolution image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

3.6.1 RASTER FORMATS:

These formats store images as bitmaps (also known as pixmaps)

- **JPEG/JFIF:**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

- **EXIF:**

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

- **TIFF:**

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

- **PNG:**

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

- **GIF:**

GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

- **BMP:**

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

3.6.2 VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM:**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

- **SVG:**

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

3.7 IMAGE PROCESSING:

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man‘s ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

3.8 FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:

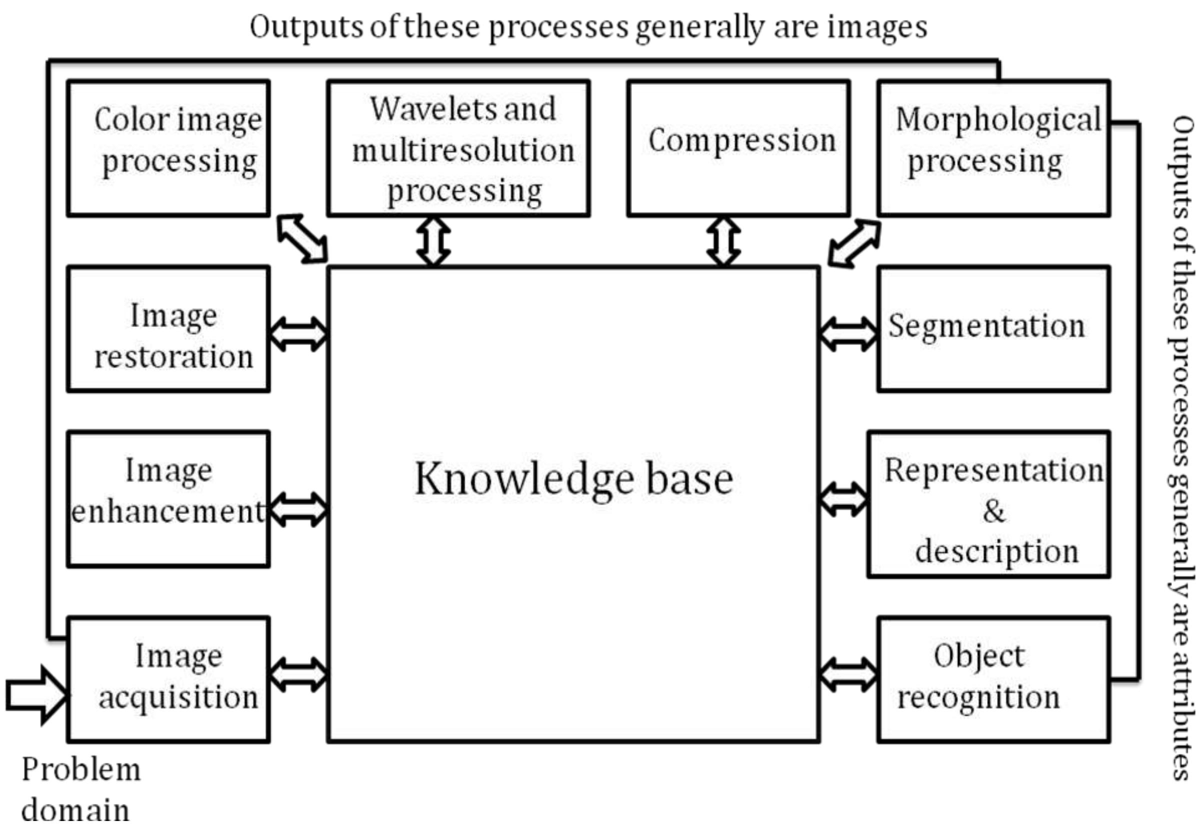


Fig 3.8 Image fundamental

3.8.1 Image Acquisition:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Fig 3.8.1 Digital camera image

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Fig 3.8.2 digital camera cell

3.8.2 Image Enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because –it looks better.‖ It is important to keep in mind that enhancement is a very subjective area of image processing.

Fig 3.8.3 Image enhancement

3.8.3 Image restoration:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Fig 3.8.4 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a –good‖ enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, where as removal of image blur by applying a deblurring function is considered a restoration technique.

3.8.4 Color image processing:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Fig 3.8.5 Color & Gray scale image

3.8.5 Wavelets and multiresolution processing:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform based image processing since the late1950's, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

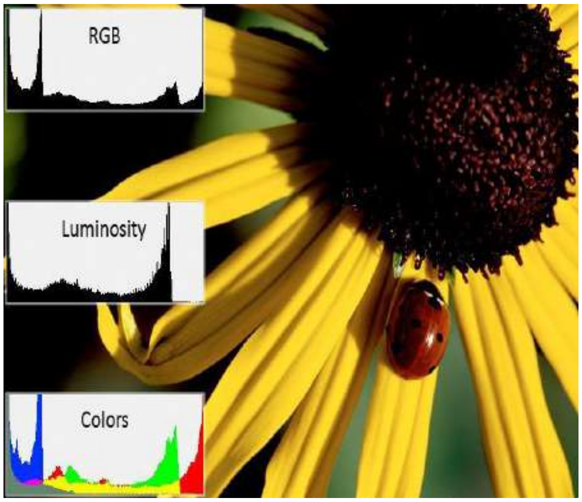


Fig 3.8.6 rgb histogram image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called **Multiresolution** theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

3.8.6 Compression:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

3.8.7 Morphological processing:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Fig 3.8.7 blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

3.8.8 Segmentation:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

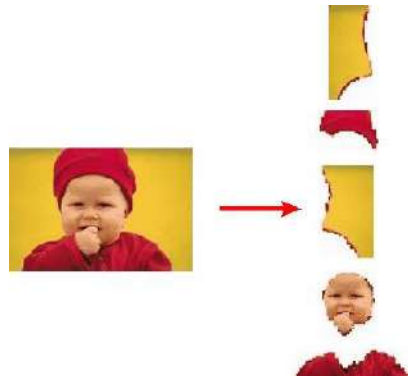


Fig 3.8.8 Image segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

3.8.9 Representation and description:

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

3.9 Object recognition:

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

3.9.1 Knowledgebase:

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an inter related to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

3.10 COMPONENTS OF AN IMAGE PROCESSING SYSTEM:

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

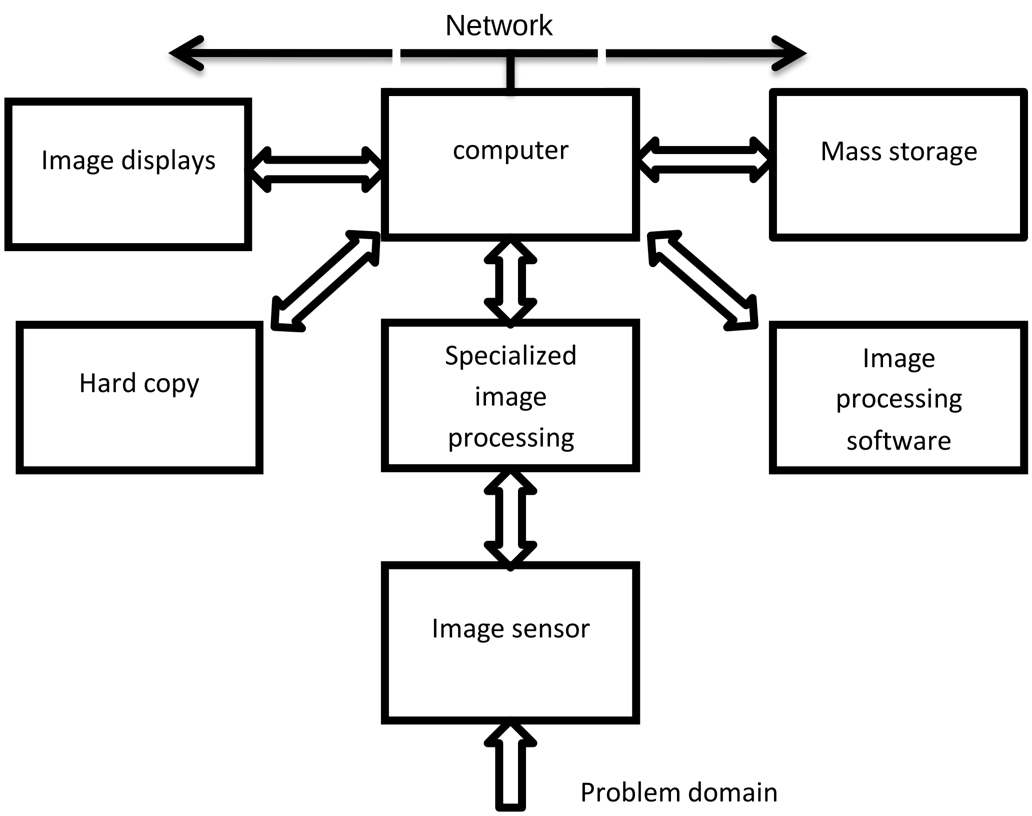


Fig 3.10 Component of image processing

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 1.24 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

- **Image sensors:**

With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

- **Specialized image processing hardware:**

  Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

- **Computer:**

  The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

- **Image processing software:**

  Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

- **Mass storage:**

  Mass storage capability is a must in image processing applications. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage forimage processing applications fall into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), Gbytes (meaning giga, or one billion, bytes), and Tbytes (meaning tera, or one trillion, bytes)

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig. 1.24. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in ―jukeboxes‖ are the usual media for archival applications.

- **Image displays:**

Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are there requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

- **Hardcopy:**

Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

- **Network:**

Networking is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of optical fiber and other broadband technologies.

Color and texture are two low-level features widely used for image classification, indexing and retrieval. Color is usually represented as a histogram, which is a first order statistical measure that captures global distribution of color in an image One of the main drawbacks of the histogram- based approaches is that the spatial distribution and local variations in color are ignored. Local spatial variation of pixel intensity is commonly used to capture texture information in an image. Grayscale Co-occurrence Matrix (GCM) is a well-known method for texture extraction in the spatial domain. A GCM stores the number of pixel neighborhoods in an image that have a particular grayscale combination. Let I be an image and let p and Np respectively denote any arbitrary pixel and its neighbor in a given direction. If GL denotes the total number of quantized gray levels and gl denotes the individual gray levels, where, gl $\in \{0, . . ., GL - 1\}$, then each component of GCM can be written as follows:

$$\mathrm{gcm}(i, j) = \Pr((\mathrm{gl}_{\mathrm{p}}, \mathrm{gl}_{N_{\mathrm{p}}}) = (i, j))$$

gcm(i, j) is the number of times the gray level of a pixel p denoted by glp equals i, and the gray level of its neighbor Np denoted by glNp equals j, as a fraction of the total number of pixels in the image. Thus, it estimates the probability that the gray level of an arbitrary pixel in an image is i, and that of its neighbor is j. One GCM matrix is generated for each possible neighborhood direction, namely, 0, 45, 90 and 135.Average and range of 14 features like Angular Second Moment, Contrast, Correlation, etc., are generated by combining all the four matrices to get a total of 28 features. In the GCM approach for texture extraction, color information is completely lost since only pixel gray levels are considered.

To incorporate spatial information along with the color of image pixels, a feature called color correlogram has recently been proposed. It is a three dimensional matrix that represents the probability of finding pixels of any two given colors at a distance ‗d' apart Auto correlogram is a variation of correlogram, which represents the probability of finding two pixels with the same color at a distance ‗d' apart. This atproach can effectively represent color distribution in an image. However, correlogram features do not capture intensity variation Many image databases often contain both color as well as gray scale images. The color correlogram method does not constitute a good descriptor in such databases.

Another method called Color Co-occurrence Matrix (CCM) has been proposed to capture color variation in an image. CCM is represented as a three-dimensional matrix, where color pair of the pixels p and Np are captured in the first two dimensions of the matrix and the spatial distance ‗d' between these two pixels is captured in the third dimension. This approach is a generalization of the color correlogram and reduces to the pure color correlogram for d = 1. CCM is generated using only the Hue plane of the HSV (Hue, Saturation and Intensity Value) color space. The Hue axis is quantized into HL number of levels. If individual hue values are denoted by hl, where $\mathrm{hl} \in \{0, \ldots, \mathrm{HL}-1\}$, then each component of CCM can be written as follows:

$$\mathrm{ccm}(i, j) = \Pr((\mathrm{hl}_{\mathrm{p}}, \mathrm{hl}_{N_{\mathrm{p}}}) = (i, j))$$

Four matrices representing neighbors at angles 0, 90, 180 and 270 are considered. This approach was further extended by separating the diagonal and the non-diagonal components of CCM to generate a Modified Color Co-occurrence Matrix (MCCM). MCCM, thus, may be written as follows: MCCM = (CCMD;CCMND)

Here, CCMD and CCMND correspond to the diagonal and off-diagonal components of CCM. The main drawback of this approach is that, like correlogram, it also captures only color information and intensity information is completely ignored.

An alternative approach is to capture intensity variation as a texture feature from an image and combine it with color features like histograms using suitable weights . One of the challenges of this approach is to determine suitable weights since these are highly application-dependent. In certain applications like Content-based Image Retrieval (CBIR), weights are often estimated from relevance feedback given by users.

While relevance feedback is sometimes effective, it makes the process of image retrieval user-dependent and iterative. There is also no guarantee on the convergence of the weight-learning algorithms. In order to overcome these problems, researchers have tried to combine color and texture features together during extraction.

proposed two approaches for capturing color and intensity variations from an image using the LUV color space. In the Single-channel Co-occurrence Matrix (SCM), variations for each color channel, namely, L, U and V are considered independently. In the Multi channel Co-occurrence Matrix (MCM), variations are captured taking two channels at a time – UV, LU and LV. Since the LUV color space separates out chrominance (L and U) from luminance (V), SCM in effect, generates one GCM and two CCMs from each image independently. As a result, correlation between the color channels is lost

However, in MCM, the count of pair wise occurrences of the values of different channels of the color space is captured. Thus, each component of MCM can be written as follows:

$$mcmUV(i; j) = Pr((up; vNp) = (i; j))$$

$$mcmLU(i; j) = Pr((lp; uNp) = (i; j))$$

$$mcmLV(i; j) = Pr((lp; vNp) = (i; j))$$

Here, mcmUV(i, j) is the number of times the U chromaticity value of a pixel p denoted by up equals i, and the V chromaticity value of its neighbor Np denoted by vNp equals j, as a fraction of the total number of pixels in the image. Similarly, mcmLU(i, j) and mcmLV(i, j) are defined. One MCM matrix is generated for each of the four neighborhood directions, namely, 0, 45, 90 and 135.

Deng and Manjunath (2001) proposed a two-stage method called JSEG, which combines color and texture after image segmentation. In the first stage, colors are quantized to the required levels for differentiating between various regions of an image. Pixel values of the regions are then replaced by their quantized color levels to form a color map. Spatial variation of color levels between different regions in the map is viewed as a type of texture composition of the image.

Yu et al. (2002) suggested the use of color texture moments to represent both color and texture of an image. This approach is based on the calculation of

Local Fourier Transformation (LFT) coefficients. Eight templates equivalent to LFT are operated over an image to generate a characteristic map of the image. Each template is a $3 \cdot 3$ filter that considers eight neighbors of the current pixel for LFT calculation. First and second order moments of the characteristic map are then used to generate a set of features.

In this paper, we propose an integrated approach for capturing spatial variation of both color and intensity levels in the neighborhood of each pixel using the HSV color space. In contrast to the other methods, for each pixel and its neighbor, the amount of color and intensity variation between them is estimated using a weight function. Suitable constraints are satisfied while choosing the weight function for effectively relating visual perception of color and the HSV color space properties. The color and intensity variations are represented in a single composite feature known as Integrated Color and Intensity Co-occurrence Matrix (ICICM). While the existing schemes generally treat color and intensity separately, the proposed method provides a composite view to both color and intensity variations in the same feature. The main advantage of using ICICM is that it avoids the use of weights to combine individual color and texture features. We use ICICM feature in an image retrieval application from large image databases.

Early result on this work was reported in (Vadivel et al., 2004a). In the next section, we describe the proposed feature extraction technique after introducing some of the properties of the HSV color space. Choice of quantization levels for color and intensity axes, selection of parameter values and a brief overview of the image retrieval application

Integrated color and intensity co-occurrence matrix:

We propose to capture color and intensity variation around each pixel in a two-dimensional matrix called Integrated Color and Intensity Co-occurrence Matrix (ICICM). This is a generalization of the Grayscale Co-occurrence Matrix and the Color Co-occurrence Matrix techniques. For each pair of neighboring pixels, we consider

their contribution to both color perception as well as gray level perception to the human eye. Some of the useful properties of the HSV color space and their relationship to human color perception are utilized for extracting this feature. In the next sub-section, we briefly explain relevant properties of the HSV color space. In the subsequent subsection, we describe how the properties can be effectively used for generating ICICM.

HSV color space:

HSV Color space: Basically there are three properties or three dimensions of color that being hue, saturation and value HSV means Hue, Saturation and Value. It is important to look at because it describes the color based on three properties. It can create the full spectrum of colors by editing the HSV values. The first dimension is the Hue. Hue is the other name for the color or the complicated variation in the color. The quality of color as determined by its dominant wavelength. This Hue is broadly classified into three categories. They are primary Hue, Secondary Hue and Teritiary Hue. The first and the foremost is the primary Hue it consists of three colors they are red, yellow and blue. The secondary Hue is formed by the combination of the equal amount of colors of the primary Hue and the colors of the secondary Hue which was formed by the primary Hue are Orange, Green and violet. The remaining one is the teritiary Hue is formed by the combination of the primary Hue and the secondary Hue. The limitless number of colors are produced by mixing the colors of the primary Hue in different amounts. Saturation is the degree or the purity of color. Then the second dimension is the saturation. Saturation just gives the intensity to the colors. The saturation and intensity drops just by mixing the colors or by adding black to the color. By adding the white to the color in spite of more intense the color becomes lighter. Then finally the third dimension is the Value. The value is the brightness of the color. When the value is zero the color space is totally black with the increase in the color there is also increase in the brightness and shows the various colors. The value describes the contrast of the color. That means it describes the lightness and darkness of the color. As similar to the saturation this value consists of the tints and shades. Tints are the colors with the added white and shades are the colors with the added black.

Properties of the HSV color space:

Sensing of light from an image in the layers of human retina is a complex process with rod cells contributing to scotopic or dim-light vision and cone cells to photopic or bright-light vision (Gonzalez and Woods, 2002). At low levels of illumination, only the rod cells are excited so that only gray shades are perceived. As the illumination level increases, more and more cone cells are excited, resulting in increased color perception. Various color spaces have been introduced to represent and specify colors in a way suitable for storage, processing or transmission of color information in images. Out of these, HSV is one of the models that separate out the luminance component (Intensity) of a pixel color from its chrominance components (Hue and Saturation). Hue represents pure color, which is perceived when incident light is of sufficient illumination and contains a single wavelength. Saturation gives a measure of the degree by which a pure color is diluted by white light. For light with low illumination, corresponding intensity value in the HSV color space is also low.

The HSV color space can be represented as a Hexa cone, with the central vertical axis denoting the luminance component, I (often denoted by V for Intensity Value). Hue, is a chrominance component defined as an angle in the range [0,2p] relative to the red axis with red at angle 0, green at 2p/3, blue at 4p/3 and red again at 2p. Saturation, S, is the other chrominance component, measured as a radial distance from the central axis of the hexacone with value between 0 at the center to 1 at the outer surface. For zero saturation, as the intensity is increased, we move from black to white through various shades of gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the color represented by its hue. When saturation is near 0, all the pixels in an image look alike even though their hue values are different.

As we increase saturation towards 1, the colors get separated out and are visually perceived as the true colors represented by their hues. Low saturation implies presence of a large number of spectral components in the incident light, causing loss of color information even though the illumination level is sufficiently high. Thus, for low values of saturation or intensity, we can approximate a pixel color by a gray level while for higher saturation and intensity, the pixel color can be approximated by its hue. For low intensities, even for a high saturation, a pixel color is close to its gray value. Similarly, for low saturation even for a high value of intensity, a pixel is perceived as gray. We use these properties to estimate the degree by which a pixel contributes to color perception and gray level perception.

One possible way of capturing color perception of a pixel is to choose suitable thresholds on the intensity and saturation. If the saturation and the intensity are above their respective thresholds, we may consider the pixel to have color dominance; else, it has gray level dominance. However, such a hard thresholding does not properly capture color perception near the threshold values. This is due to the fact that there is no fixed level of illumination above which the cone cells get excited. Instead, there is a gradual transition from scotopic to photopic vision. Similarly, there is no fixed threshold for the saturation of cone cells that leads to loss of chromatic information at higher levels of illumination caused by color dilution. We, therefore, use suitable weights that vary smoothly with saturation and intensity to represent both color and gray scale perception for each pixel.

NON INTERVAL QUANTIZATION:

Due to the large range for each component by directly calculating the characteristics for the retrieval then the computation will be very difficult to ensure rapid retrieval. It is essential to quantify HSV space component to reduce computation and improve efficiency. At the same time, because the human eye to distinguish colors is limited, do not need to calculate all segments. Unequal interval quantization according the human color perception has been applied on H , S ,V components.

Based on the color model of substantial analysis, we divide color into eight parts. Saturation and intensity is divided into three parts separately in accordance with the human eyes to distinguish. In accordance with the different colors and subjective color perception quantification, quantified hue(H), saturation(S) and value(V)

In accordance with the quantization level above, the H, S, V three-dimensional feature vector for different values of with different weights to form one dimensional feature vector and is given by the following equation:

$$G = Qs*Qv*H+Qv*s+V$$

Where Qs is the quantized series of S and Qv is the quantized series of V. And now by setting Qs = Qv = 3, Then G = 9H+3S+V

$$H = \begin{cases} 0 \ \ if \ \ h \in [316,20] \\ 1 \ \ if \ \ h \in [21,40] \\ 2 \ \ if \ \ h \in [41,75] \\ 3 \ \ if \ \ h \in [76,155] \\ 4 \ \ if \ \ h \in [156,190] \\ 5 \ \ if \ \ h \in [191,270] \\ 6 \ \ if \ \ h \in [271,295] \\ 7 \ \ if \ \ h \in [296,315] \end{cases} \qquad S = \begin{cases} 0 \ \ if \ \ s \in [0,0.2) \\ 1 \ \ if \ \ s \in [0.2,0.7) \\ 2 \ \ if \ \ s \in [0.7,1) \end{cases} \qquad V = \begin{cases} 0 \ \ if \ \ v \in [0,0.2) \\ 1 \ \ if \ \ v \in [0.2,0.7) \\ 2 \ \ if \ \ v \in [0.7,1) \end{cases}$$

In this way three component vector of the HSV from one dimensional vector, Which quantize the whole color space for the 72 kinds of the main colors. So we can handle 72 bins of one dimensional histogram. This qualification is effective in reducing the images by the effect of the light intensity, but also reducing the computational time and complexity.

IMAGE RETRIEVAL:

Image retrieval is nothing but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords or the descriptions to the images so that the retrieval can be performed. Manual image annotation is time consuming, expensive and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc..,

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others. one can today generate volumes of images with content as diverse as family get-togethers and national park visits. Low-cost storage and easy Web hosting has fueled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchable image data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real-world image search system designers.

As far as technological advances are concerned, growth in Content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real world implications, applications, and constraints of the technology. Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future.

An image retrieval system designed to serve a personal collection should focus on features such as personalization, flexibility of browsing, and display methodology. For example, Google‘s Picasa system [Picasa 2004] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods. The notable few applications include an FPGA implementation of a color-histogram-based image retrieval system [Kotoulas and Andreadis 2003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 2003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 2002].

Discussion. Regardless of the nature of the collection, as the expected user-base grows, factors such as concurrent query support, efficient caching, and parallel and distributed processing of requests become critical. For future real-world image retrieval systems, both software and hardware approaches to address these issues are essential. More realistically, dedicated specialized servers, optimized memory and storage support, and highly parallelizable image search algorithms to exploit cluster computing powers are where the future of large-scale image search hardware support lies.

OVERVIEW OF TEXTURE:

We all know about the term Texture but for defining it is a hard time. One can differentiate the two different Textures by recognizing the similarities and differences. Commonly there are three ways for the usage of the Textures:

Based on the Textures the images can be segmented To differentiate between already segmented regions or to classify them.We can reproduce Textures by producing the descriptions. The texture can be analyzed in three different ways. They are Spectral, Structural and Statistical:

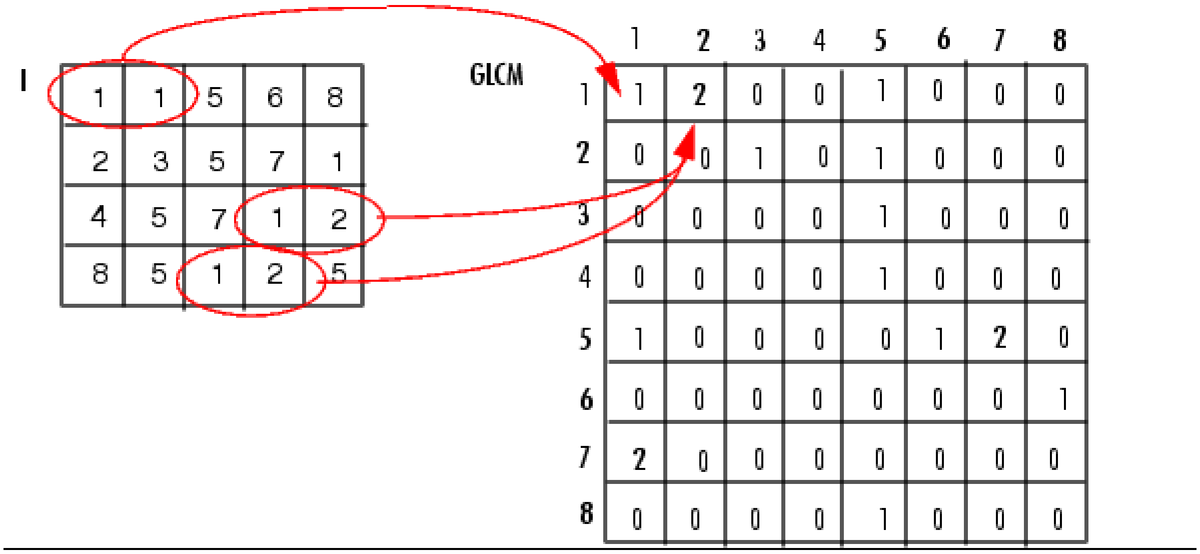

I
1 1 5 6 8
2 3 5 7 1
4 5 7 1 2
8 5 1 2 5
GLCM
1 2 3 4 5 6 7 8
1 1 2 0 0 1 0 0 0
2 0 0 1 0 1 0 0 0
3 0 0 0 0 1 0 0 0
4 0 0 0 0 1 0 0 0
5 1 0 0 0 0 1 2 0
6 0 0 0 0 0 0 0 1
7 2 0 0 0 0 0 0 0
8 0 0 0 0 1 0 0 0

Fig 3.11.1

CHAPTER 4

PROJECT IMPLEMENTATION:

4.0 IMPLEMNTATION STAGES:

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Dürer's matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

MATLAB displays the matrix you just entered.

```
A =

   16    3    2   13

    5   10   11    8
```

9	6	7	12
4	15	14	1

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

4.0.1 Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

4.0.2 Operators

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

Fig:4.0.2 Operators and precedence rules

4.0.3 Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.14159265...
I	Imaginary unit, $\sqrt{-1}$
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

Fig:4.0.3 Several special function

4.1 GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

4.1.1 Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

4.1.2 GUI Development Environment

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of its

children (uicontrols and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the

callbacks, which are defined as subfunctions. This M-file is referred to as the

application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

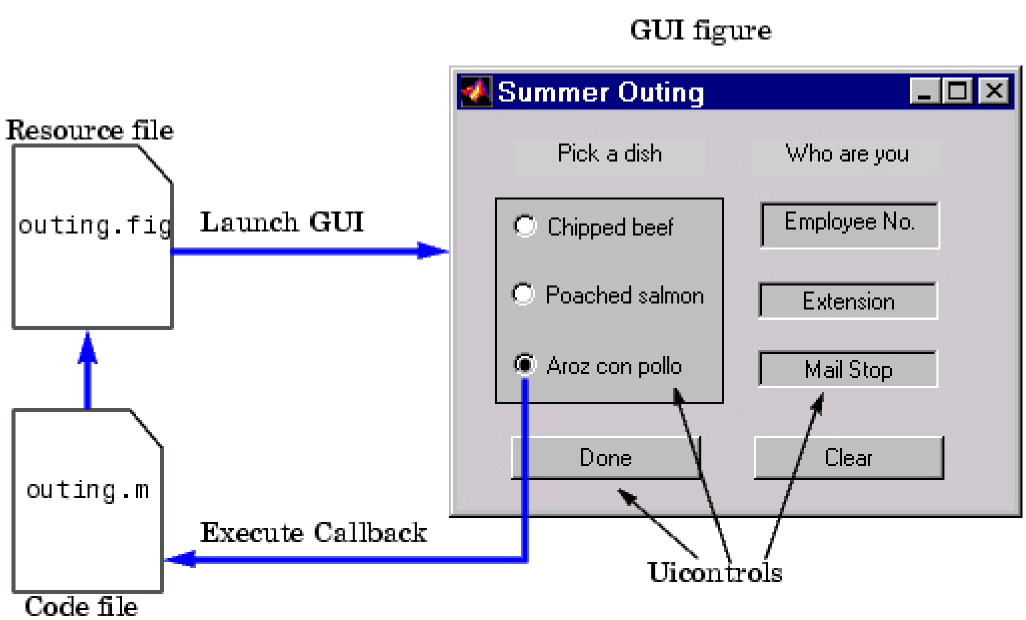


FIG 4.1.2 graphical user blocks

4.1.3 Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

4.1.4 Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques

used in the application M-file.

Application Examples - a collection of examples that illustrate techniques which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

4.1.5 User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- Listboxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)

button_state = get(h,'Value');
```

```
if button_state == get(h,'Max')
   % toggle button is pressed
elseif button_state == get(h,'Min')
   % toggle button is not pressed
end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);
a(:,:,2) = rand(16,128);
a(:,:,3) = rand(16,128);
set(h,'CData',a)
```

Radio Buttons

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)

set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)

off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];

mutual_exclude(off)

% Continue with callback

   .

   .

   .
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)

if (get(h,'Value') == get(h,'Max'))

   % then checkbox is checked-take approriate action

else

   % checkbox is not checked-take approriate action

end
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)

user_string = get(h,'string');

% proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)

user_entry = str2double(get(h,'string'));

if isnan(user_entry)

   errordlg('You must enter a numeric value','Bad Input','modal')

end

% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menubar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menubar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

Listboxes differentiate between single and double clicks on an item and set the figure SelectionType property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure SelectionType property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

switch val

case 1

% The user selected the first item

case 2

% The user selected the second item

% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

string_list = get(h,'String');

selected_string = string_list{val}; % convert from cell array to string

% etc.
```

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is

grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes ButtonDownFcn property to define the callback.

4.1.6 Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

```
axes(handles.axes1)
```

makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axes. See GUI with Multiple Axes for and example that uses two axes.

Figure

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control

4.2 PROJECT CODE

```
%Studies on Different CNN Algorithms for Face Skin Disease
Classification Based on
%Clinical Images
close all
clear all

input_image = imread('p.png');
% Customerlized Mask
m = zeros(size(input_image,1),size(input_image,2));
m(20:120,20:120) = 1;
seg = CNN_Algorithms(input_image,m,500,0.1,'chan'); %
CNN_Algorithms
% Built-in Mask
seg = CNN_Algorithms(input_image,'medium',400,0.02,'chan'); %
ability on gray image
%-- End

%P = imread('p.png');
% Imnoise the original input
% input_image = P;
%I(:,:,1) = imnoise(I(:,:,1),'speckle');
%I(:,:,2) = imnoise(I(:,:,2),'salt & pepper');
figure(),subplot(1,2,1),imshow(input_image),title('original
image');
subplot(1,2,2),imshow(input_image),title('original image with
two components adding noise')

% Normal Chan & Vese cannot work
seg = CNN_Algorithms(input_image,'large',300,0.02,'chan');

% Chan & Vese for vector image works here
seg = CNN_Algorithms(input_image,'large',300,0.02,'vector');
% Using built-in mask = 'whole' leads faster and better
segmentation
seg = CNN_Algorithms(input_image,'whole',800,0.02,'vector');
% input_image = imread('p.png');
seg = CNN_Algorithms(input_image,'whole',100,0.1,'multiphase');

seg = CNN_Algorithms(input_image,'whole',400,0.2,'multiphase');
%-- End
```

4.3 RESULT

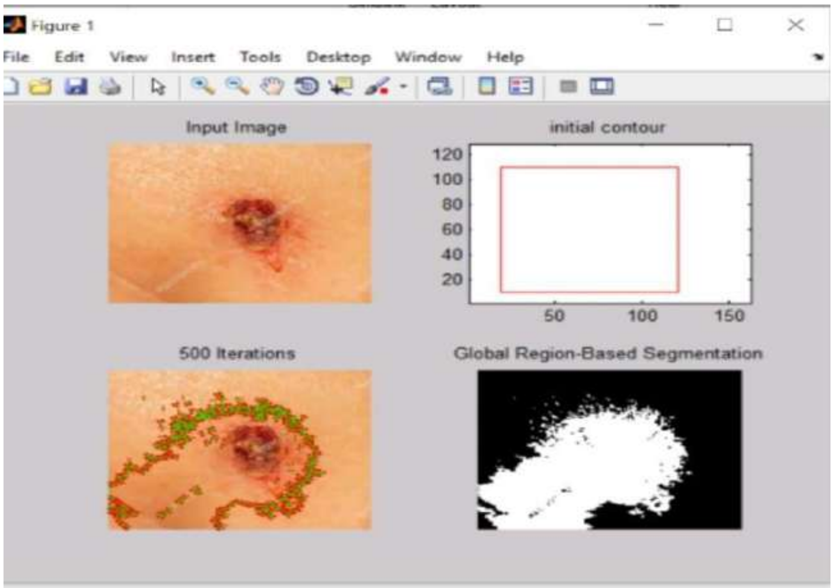


Fig:4.3.1 Input Image

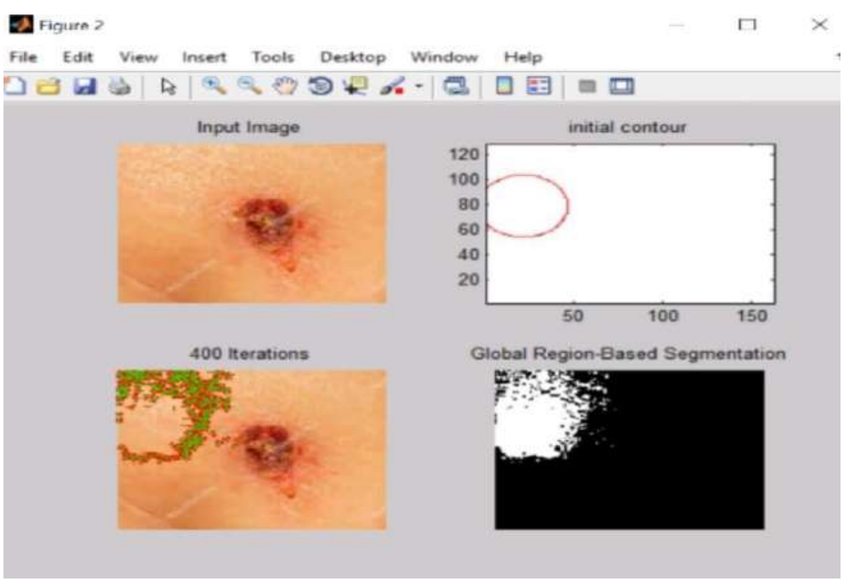


Fig:4.3.2 Random Detection

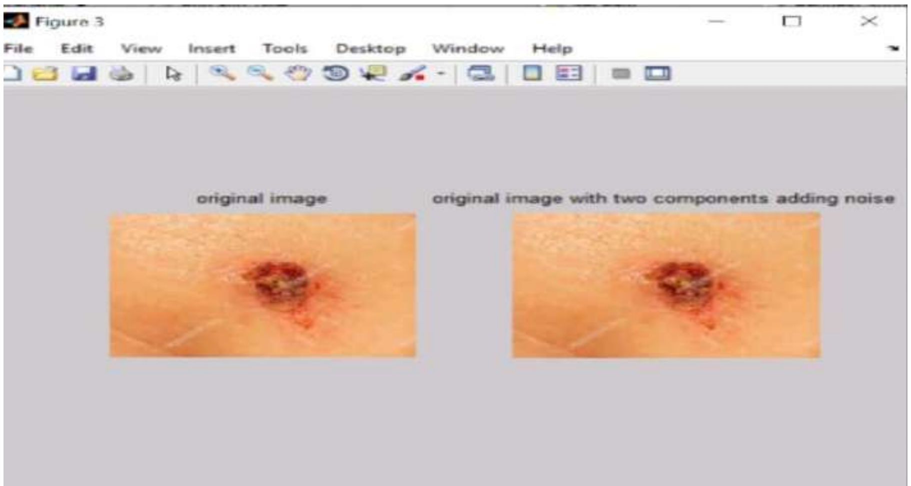


Fig:4.3.3 Original image adding noise

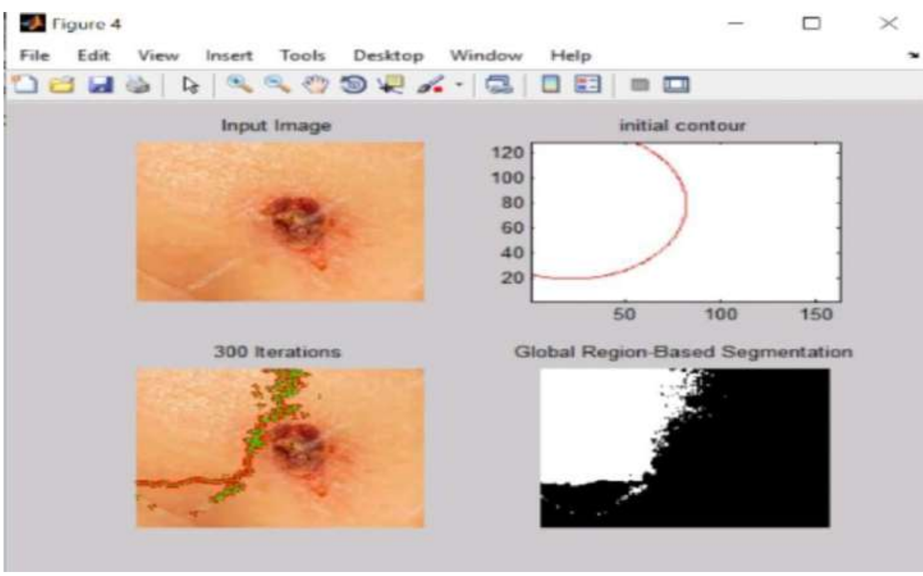


Fig:4.3.4 Detection processing

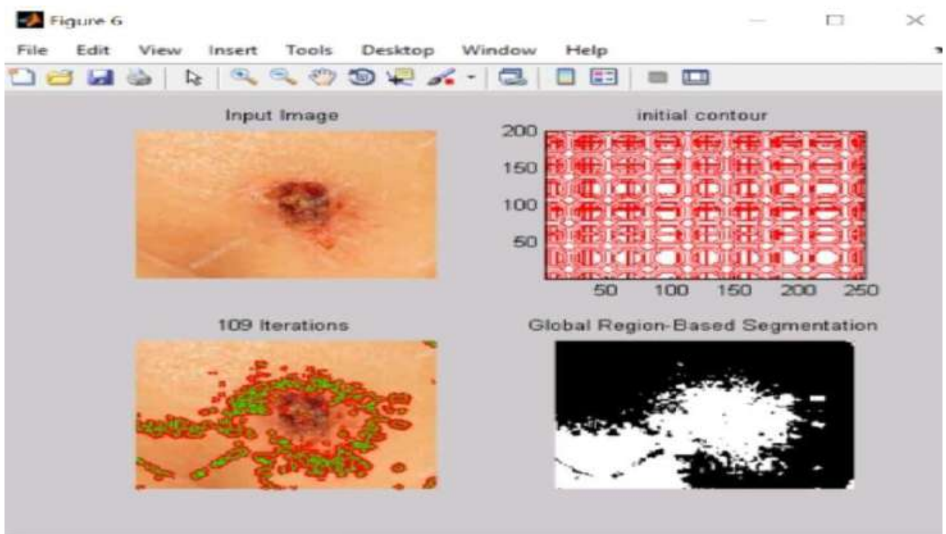


Fig:4.3.5 Final Detection

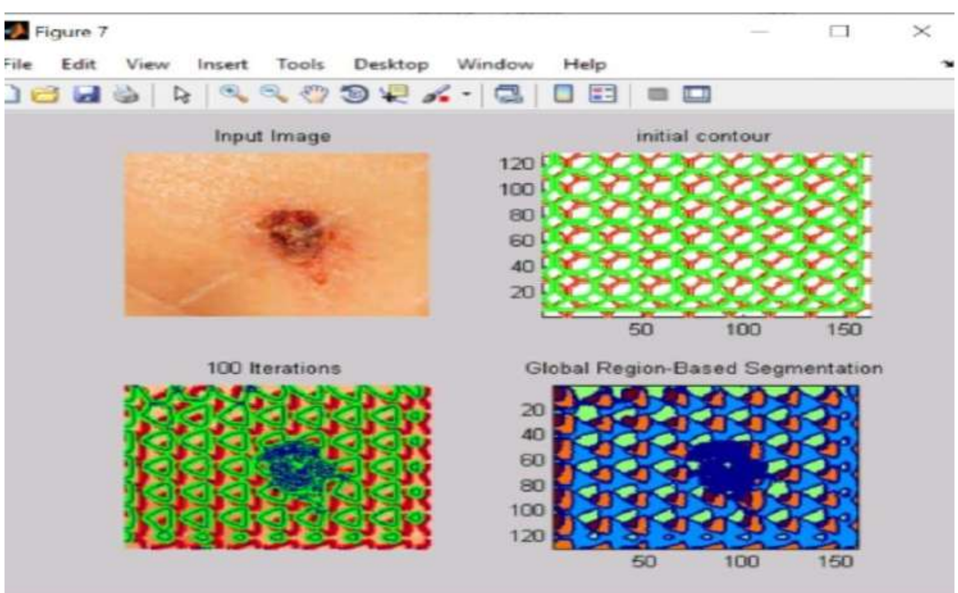


Fig:4.3.6 Final output

Chapter 5

PROJECT TESTING

5.0 OVERVIEW OF TESTING METHODS

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

5.0.1 Introduction - describes the components of the MATLAB system.

5.0.2 Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

5.0.3 Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

5.0.4 Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

5.0.5 Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

5.1 DEVELOPMENT

ENVIRONMENT5.1.1Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

5.1.2 Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

5.1.3 Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

5.1.4 MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

5.1.5 Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type helpbrowser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to runMATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

Chapter 6

CONCLUSION AND FUTURE ENCHANCEMENT

6.0 CONCLUSION:

- This paper performed experiments using five mainstream CNN structures for the clinical image diagnosis of six common facial skin diseases and constructed a data set consisting mainly of facial skin disease images.
- The results demonstrate that CNNs have the ability to recognize facial skin diseases. Based on our experiments, we determined that different models to diagnose diseases on different body parts should be used.
- Furthermore, our experiments also showed that a more reasonable network structure could improve the performance of the model. The performance of the current network structure has been satisfactory in some diseases, but the overall performance has yet to be improved.
- As a result, if we want that people to actually use this technique to check their face skin health in their daily life, specialized improvements should be developed.

6.1 FUTURE ENCHANCEMENT:

- In our opinion, the application of artificial intelligence techniques in the medical field is not sufficient, and the datasets from this field should be improved both in quantity and quality.
- With the increasing amount of facial image data of various skin diseases and the continuous improvement of the network structure, CNN-based facial skin disease diagnosis algorithms will continue to improve in performance.
- We believe that, in the future, patients will use convenient CNN-based applications to keep their face skin healthy.

PUBLICATION

Submitted Paper Conference ICISECT-21 with Paper ID (ICISECT21-0032) and got Acceptance for the Paper.

REFERENCES

[1] R. J. Hay, N. E. Johns, H. C. Williams, I. W. Bolliger, R. P. Dellavalle, and D. J. Margolis, ―The global burden of skin disease in 2010: An analysis of the prevalence and impact of skin conditions,‘‘ J. Investigative Dermatology, vol. 134, no. 6, pp. 1527–1534, 2014.

[2] X. Huang, J. Zhang, J. Li, S. Zhao, Y. Xiao, Y. Huang, D. Jing, L. Chen, X. Zhang, J. Su, Y. Kuang, W. Zhu, M. Chen, X. Chen, and M. Shen, ―Daily intake of soft drinks and moderate-to-severe acne vulgaris in Chinese Adolescents,‘‘ J. Pediatrics, vol. 204, pp. 256–262, Jan. 2018.

[3] Y. Deng, Q. Peng, S. Yang, D. Jian, B. Wang, Y. Huang, H. Xie, and J. Li, ―The rosacea-specific quality-of-life instrument (RosQol): Revision and validation among Chinese patients,‘‘ PLoS ONE, vol. 13, no. 2, Feb. 2018, Art. no. e0192487.

[4] C. Junchen, W. Zeng, W. Pan, C. Peng, J. Zhang, J. Su, W. Long, H. Zhao, X. Zuo, X. Xie, J. Wu, L. Nie, H.-Y. Zhao, H.-J. Wei, and X. Chen, ―Symptoms of systemic lupus erythematosus are diagnosed in leptin transgenic pigs,‘‘ PLoS Biol., vol. 16, no. 8, Aug. 2018, Art. no. e2005354.

[5] X. Xiaoyun, H. Chaofei, Z. Weiqi, C. Chen, L. Lixia, L. Queping, P. Cong, Z. Shuang, S. Juan, and C. Xiang, ―Possible involvement of F1F0-ATP synthase and intracellular ATP in Keratinocyte differentiation in normal skin and skin lesions,‘‘ Sci. Rep., vol. 7, Feb. 2017, Art. no. 42672.

[6] A. Bewley, ―The neglected psychological aspects of skin disease,‘‘ Brit. Med. J., vol. 358, p. 3208, Jul. 2017.

[7] W. Chen, X. Zhang, W. Zhang, C. Peng, W. Zhu, and X. Chen, ―Polymorphisms of SLCO1B1 rs4149056 and SLC22A1 rs2282143 are associated with responsiveness to acitretin in psoriasis patients,‘‘ Sci. Rep.. vol. 4, no. 1, 2018, Art. no. 13182. doi: 10.1038/s41598-018-31352-2.

[8] X. Zhou, W. Zhu, M. Shen, Y. He, C. Peng, Y. Kuang, J. Su, S. Zhao, X. Chen, and W. Chen, ‗‗Frizzled-related proteins 4 (SFRP4) rs1802073G allele predicts the elevated serum lipid levels during acitretin treatment in psoriatic patients from Hunan, China,'' PeerJ, vol. 13, no. 6, 2018, Art. no. e4637.

[9] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. F. Fei, ‗‗ImageNet: A large-scale hierarchical image database,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.

[10] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, ‗‗Dermatologist-level classification of skin cancer with deep neural networks,'' Nature, vol. 542, pp. 115–118, Feb. 2017.

[11] X. Zhang, S. Wang, J. Liu, and C. Tao, ‗‗Towards improving diagnosis of skin diseases by combining deep neural network and human knowledge,'' Med. Inform. Decis. Making, vol. 18, no. 2, p. 59, 2018.

[12] A. Masood and A. A. Al-Jumaily, ‗‗Computer aided diagnostic support system for skin cancer: A review of techniques and algorithms,'' Int. J. Biomed. Imag., vol. 2013, pp. 1–22, 2013.

[13] Y. Gurovich, Y. Hanani, O. Bar, G. Nadav, N. Fleischer, D. Gelbman, L. Basel-Salmon, P. M. Krawitz, S. B. Kamphausen, M. Zenker, L. M. Bird, and K. W. Gripp, ‗‗Identifying facial phenotypes of genetic disorders using deep learning,'' Nature Med., vol. 25, pp. 60–64, Jan. 2019.

[14] A. Melina, N. N. Dinh, B. Tafuri, G. Schipani, S. Nisticò, C. Cosentino, F. Amato, D. Thiboutot, and A. Cherubini, ‗‗Artificial intelligence for the objective evaluation of acne investigator global assessment,'' J. Drugs Dermatology, vol. 17, no. 9, pp. 1006–1009, 2018.

[15] A. Dermatology. Samuel Freire Da Silva, Delso Bringel Calheiros. [Online]. Available: http://www.atlasdermatologico.com.br

A

MAJOR PROJECT REPORT

On

IMAGE CONTRAST ENHANCEMENT IN AUTOMATIC MODE BY NON-LINEAR STRETCHING

Submitted by

1) Ms.B.Shireesha (17K81A04J6)   **2) Ms.K.Aruna (17K81A04L9)**

3) Mr.Satish Kumar Nirala (17K81A04P3)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Esteemed Guidance of

Dr.A.Anand

Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

BONAFIDE CERTIFICATE

This is to certify that the project entitled **"IMAGE CONTRAST ENHANCEMENT IN AUTOMATIC MODE BY NON-LINEAR STRETCHING",** is being submitted by 1.**Ms.B.Shiressha(17K81A04J6), 2.Ms.K.Aruna(17K81A04L9), 3.Mr.Satish Kumar Nirala(17K81A04P3)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Dr.A.Anand
Department of ECE

Head of the Department
Dr. B. HARI KRISHNA
Department of ECE

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT B SHIREESHA WITH ROLL NO.17K81A04J6, K ARUNA WITH ROLL NO.17K81A04L9, SATISH KUMAR NIRALA WITH ROLLNO.17K81A04P3, OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "IMAGE CONTRAST ENHANCEMENT IN AUTOMATIC MODE BY NONLINEAR STRETCHING" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of **Bachelor of Technology** in Department of ELECTRONICS AND COMMUNICATION ENGINEERING, session: <2017 – 2021>, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled **"IMAGE CONTRAST ENHANCEMENT IN AUTOMATIC MODE BY NONLINEAR STRETCHING"** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for awardof any degree.

1) B.Shireesha(17K81A04J6)

2) K.Aruna(17K81A04L9)

3) Satish Kumar Nirala(17K81A04P3)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr.B.HARI KRISHNA**, **Head &Professor**, Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in our memory for years to come.

We would like to express our gratitude to our dept. project coordinator **Dr.A.Anand, Professor** for his involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time to come.

We sincerely thank our project internal guide, **Dr.A.Anand**, **professor**, Department Of ECE and our class project coordinator **Ms. P. Pushpa, Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our project guide **DR.A.ANAND, Professor,** for his advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

1. B.Shireesha
2. K.Aruna
3. Satish Kumar Nirala

ABSTRACT

The problem of image enhancement in automatic mode is considered. A new method of contrast enhancement by adaptive nonlinear stretching using piecewise linear transformation of the brightness scale is proposed at which the average value of the brightness of transformed image is closest to the middle of the brightness range. The research of the effectiveness of the proposed and known methods of nonlinear stretching of monochrome images using the known no-reference metrics of integral contrast of image is carried out. Image enhancement is one of the most interesting and visually appealing areas of image processing. It involves operations such as enhancing contrast, reducing noise for improving the quality of the image. This paper presents an analysis of the mathematical morphological approach with comparison to various other state-of-art techniques for addressing the problems of low contrast in images. Histogram equalization (HE) is one of the common methods used for improving contrast in digital images.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT:

Image enhancement is a preprocessing step in many image processing applications. The aim of image enhancement is to improve the interpretability or perception of information in images for human viewers, or to provide better input for other automated image processing techniques. There are various reasons for poor quality of an image such as distortion being introduced by the imaging systems, lack of expertise of the operator or the adverse external conditions at the time ofimage acquisition.

Mainly, Image enhancement includes intensity and contrast manipulation, noise reduction, edges sharpening and filtering, etc. Contrast Enhancement is focused on the problem of improving the contrast in an image to make various features more easily perceived. Contrast of an image is determined by its dynamic range, which is defined as the difference between lowest and highest intensity level. Contrast enhancement techniques have various application areas for enhancing the visual quality of low contrast images. Many contrast enhancement algorithms have been developed over the years. Contrast enhancement algorithms can broadly be divided into two categories: spatial domain techniques and frequency domain techniques.

In spatial domain techniques, the image enhancement is basedon direct manipulation of the pixels in an image. Frequency domain processing techniques are based on modifying the Fourier transform of an image. In frequency domain methods, the image is first transferred in to frequency domain. It means that, the Fourier Transform of the image is computed first. Allthe enhancement operations are then performed on the Fouriertransform of the image and then the Inverse Fourier transform is performed to get the resultant image.

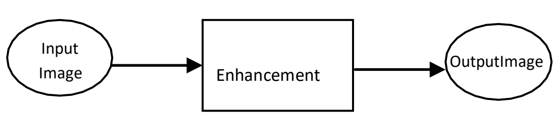


Fig 1.1: Image Enhancement

The problem of image enhancement in automatic mode is considered. A new method of contrast enhancement by adaptive nonlinear stretching using piecewise linear transformation of the brightness scale is proposed at which the average value of the brightness of transformed image is closest to the middle of the brightness range. The research of the effectiveness of the proposed and known methods of nonlinear stretching of monochrome images using the known no-reference metrics of integral contrast of image is carried out.

1.2 OBJECTIVES OF THE STUDY:

This project had been implemented by Adaptive non-linear stretching using piecewise transformation. It will enhance the image low to high contrast.

Contrast enhancement is one of the important research issues of image enhancement. There are many contrast enhancement methods which have been proposed in the literature. A very popular technique for image enhancement is histogramequalization (HE).But this technique is not much efficient to enhance small parts of an image so we are using adaptive non linear stretching. Some researchers havealso focused on improvement of histogram equalization based contrast enhancement such as adaptive non linear stretching which helps to enhance the contrast locally . Mathematical morphology is a relatively new approach to image processing and analysis.

(a) (b)

Fig 1.2 : Contrast Enhancement (a) A Low Contrast image (b) A High Contrast Image

Figure 2(b) shows the results of enhancing the contrast of the given image .

1.3 SCOPE OF THE STUDY:

The future of image processing will involve scanning the heavens for other intelligent life out in space. Also new intelligent, digital species created entirely by research scientists in various nations of the world will include advances in image processing applications.

Due to advances in image processing and related technologies there will be millions and millions of robots in the world in a few decades time, transforming the way the world is managed.

Advances in image processing and artificial intelligence6 will involve spoken commands, anticipating the information requirements of governments, translating languages, recognizing and tracking people and things, diagnosing medical conditions, performing surgery, reprogramming defects in human DNA, and automatic driving all forms of transport.

With increasing power and sophistication of modern computing, the concept of computation can go beyond the present limits and in future, image processing technology will advance and the visual system of man can be replicated.

The future trend in remote sensing will be towards improved sensors that record the same scene in many spectral channels. Graphics data is becoming increasingly important in image processing app1ications. The future image processing applications of satellite based imaging ranges from planetary exploration to surveillance applications.

Using large scale homogeneous cellular arrays of simple circuits to perform image processing tasks and to demonstrate pattern-forming phenomena is an emerging topic.

The cellular neural network is an implementable alternative to fully connected neural networks and has evolved into a paradigm for future imaging techniques. The usefulness of this technique has applications in the areas of silicon retina, pattern formation, etc.

1.4 MATERIAL REQUIREMENTS:

1.4.1 SOFTWARE REQUIREMENTS:

- Personal computer
- MATLAB Software

Matlab is a programming platform designed specifically for engineers and scientists to analyze and design systems and products that transform our world. The heart of MATLAB is the MATLAB language, a matrix-based language allowing the most natural expression of computational mathematics.

What Can I Do With MATLAB?

- Analyze data
- Develop algorithms
- Create models and applications

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH AREA:

Image Processing is among rapidly growing technologies today, with its applications in various aspects of a business. Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It forms core research area within engineering and computer science disciplines too. In addition to applications in medicineand the space program, digital image processing techniques now are used in a broad range of applications. Computerprocedures are used to enhance the contrast or other features of the image for easier interpretation of x-rays and other images used in industry, medicine and the biological sciences .

Image enhancement is one of the most interesting and visuallyappealing areas of image processing. The basic idea behind image processing techniques is to make details more obvious or to simply highlight certain features of interest in an image. There are many aspects of images that are ambiguous and uncertain. Image enhancement operation improves the qualities of an image in terms of contrast, brightness characteristics, reduction of noise contents etc. Sometimes an image may be too dark containing blurredness and therefore it is difficult to recognize the different objects or scenery contained in the image. This type of image requires enhancement. A large number of image enhancement techniques exist for reducing image noise, highlighting edges, or displaying digital images. It is difficult to judge the effectiveness of these techniques due to various reasons such as the outcome depends on the exact application. An enhancement technique performing well in enhancing biomedical images may not be identically efficient in enhancing satellite images. Thus, the objective of image enhancement is dependent on the application context.

2.2 REVIEW ON RELATED LITERATURE:

M. Shakeri proposed a contrast enhancement algorithm based on local histogram equalization which was used to calculate the total sub-histograms automatically and their division based on their density. The algorithm worked in three stages. Initially, the estimation of the number of clusters for image brightness levels is done using histogram equalization. In the next stage, the image bright-ness levels are clustered and finally include the contrast enhancement for each individual cluster separately. The algorithm is com-pared with other methods based on quality and quantity measurement.

Lalit Maurya proposed a social spider optimization algorithm which produces two quality images one with better contrast, in-creased entropy and the second image with increased peak signal to noise ratio. Both the images are combined to get an effective image later. Comparisons were done with HE, Linear contrast stretching.

Standard Particle Swarm Optimization. Results show that the pro-posed method achieves better Peak signal to noise ratio, preserves brightness, enhances the contrast of any given image resulting a quality visual. Se EunKim worked upon wavelet domain using entropy for contrast enhancement. Initially it uses a local entropy scaling in the wavelet domain to obtain the desired contrast .mathematical works were used and then a color enhancing method in the HSI color space was developed. The algorithm worked in two steps: Modification of the low frequencies in the wavelet domain and scaling HIS color space by enhancing the intensity component such that low light images will get detailed color information without any post processing. Anil Singh Parihar proposes an entropy-based dynamic sub-histogram equalization algorithm for contrast enhancement with better intensity levels over the entire dynamic range. Parameters were not used. Results were compared with conventional contrast enhancement algorithms. Jeyong Shin proposed HBLPCE,an optimization problem to preserve the localities of the histogram for performing contrast enhancement. By this method the shape of the enhanced image re-mains the same as the original image. HBLPCE was quite successfull on all images with different statistical properties. Huang Lidong proposed an image enhancement method CLAHE-DWT which combines both CLAHE and DWT. The algorithm works in three stage.

2.3 CONCLUSION ON REVIEWS:

Image enhancement techniques change images to provide a better representation of the information encapsulated in the image. In this paper is presented a review for various fields of image enhancement. For every purpose like underwater imaging or medical images, there are different algorithms and techniques suitable for image enhancement. As we can see from this review paper, and also from other review papers, there is no universal image enhancement technique, the most important reason for that is the fact that there are a lot of different factors, for example, fogg in images and videos, or the fact that medical images are mostly grayscale images.

CHAPTER-3

PROJECT DESIGN

3.1 OVERVIEW OF THE DESIGN:

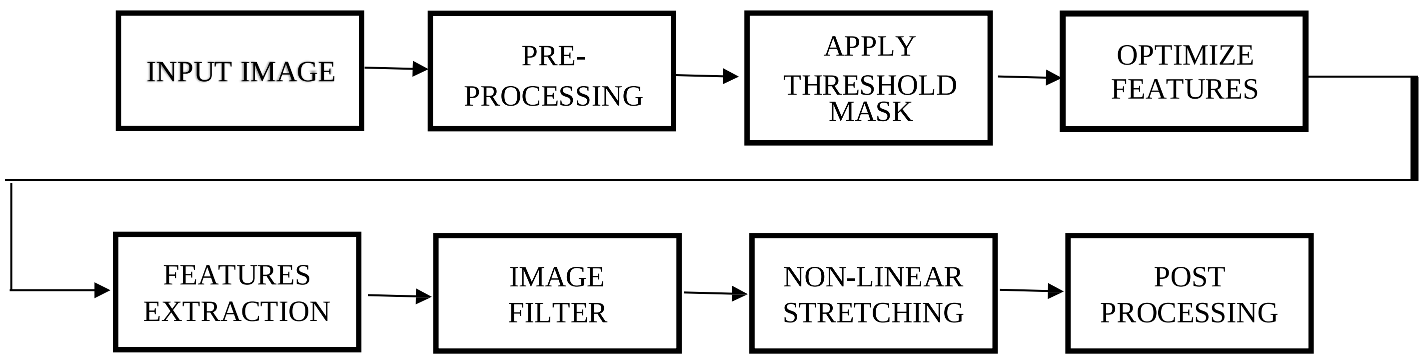


Fig.3.1 Overview of the design

3.2 EQUIPMENT ANALYSIS:

3.2.1 INTRODUCTION TO MATLAB:

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping

- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

3.3 THE MATLAB SYSTEM:

The MATLAB system consists of five main parts

- **Development Environment**:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library**:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language**:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both “programming in the small” to rapidly create quick and dirty throw-away programs, and “programming in the large” to create large and complex application programs.

- **Graphics:**

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using image processing toolbox.

3.3.1 GRAPHICAL USER INTERFACE (GUI):

MATLAB’s Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

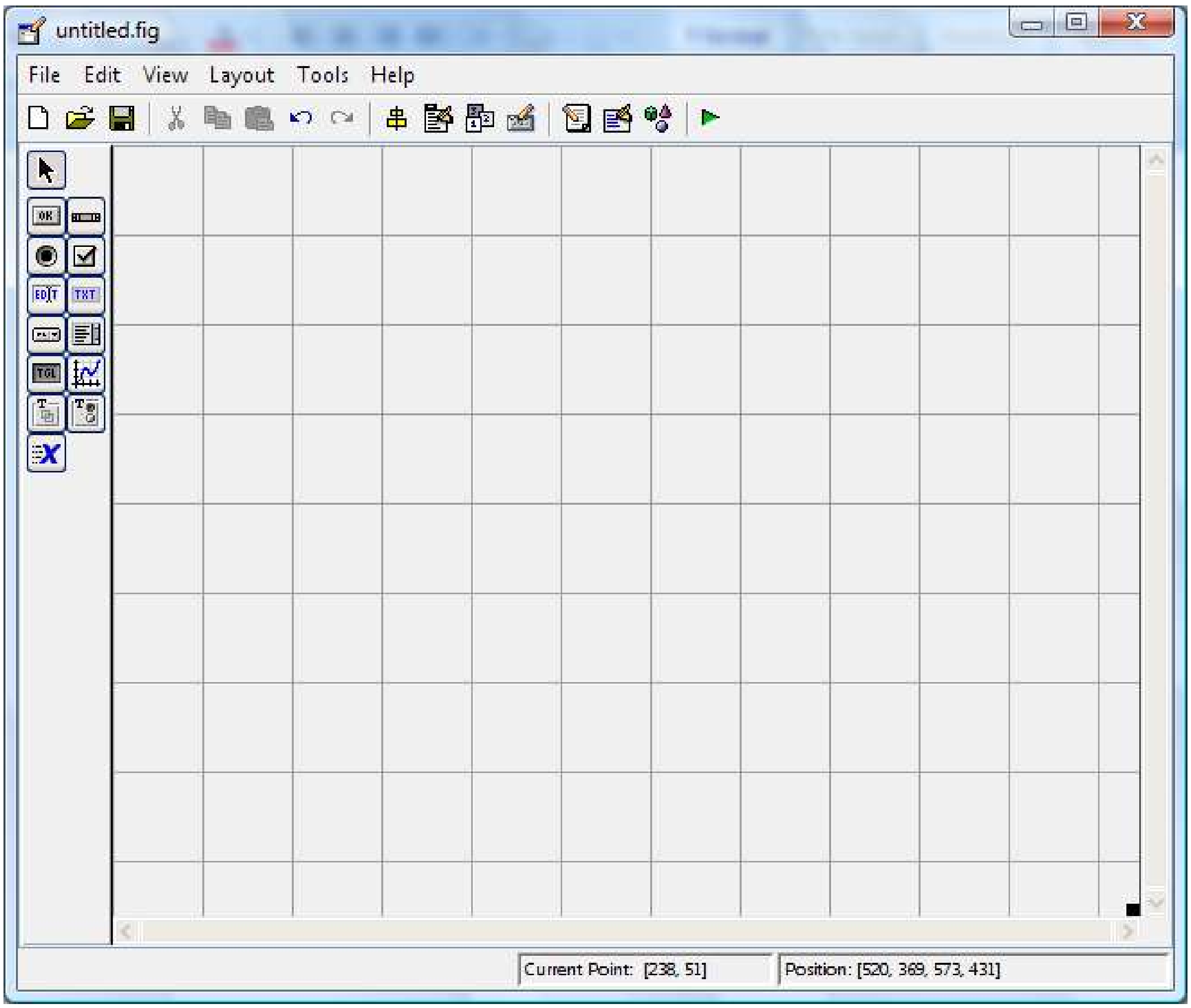
- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function’s GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.

- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type

guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted, GUIDE opens a new (i.e., blank) window.



A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

3.4 GETTING STARTED

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

3.4.1 Introduction - describes the components of the MATLAB system.

3.4.2 Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

3.4.3 Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

3.4.4 Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

3.4.5 Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

3.5 DEVELOPMENT ENVIRONMENT

3.5.1 Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

3.5.2 Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

3.5.3 Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

3.5.4 MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

3.5.5 Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command

History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type helpbrowser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator- Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to runMATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

3.6 MANIPULATING MATRICES

3.6.1 Entering Matrices

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Dürer's matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]

MATLAB displays the matrix you just entered.

```
A =
   16    3    2   13
    5   10   11    8
    9    6    7   12
    4   15   14    1
```

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

3.6.2 Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard.

Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

3.6.3 Operators

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

3.6.4 Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files.

Pi	3.14159265...
I	Imaginary unit, $\sqrt{-1}$
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}

Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

3.7 GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

3.7.1 Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

3.7.2 GUI Development Environment

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of its children (uicontrols and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the callbacks, which are defined as subfunctions. This M-file is referred to as the application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

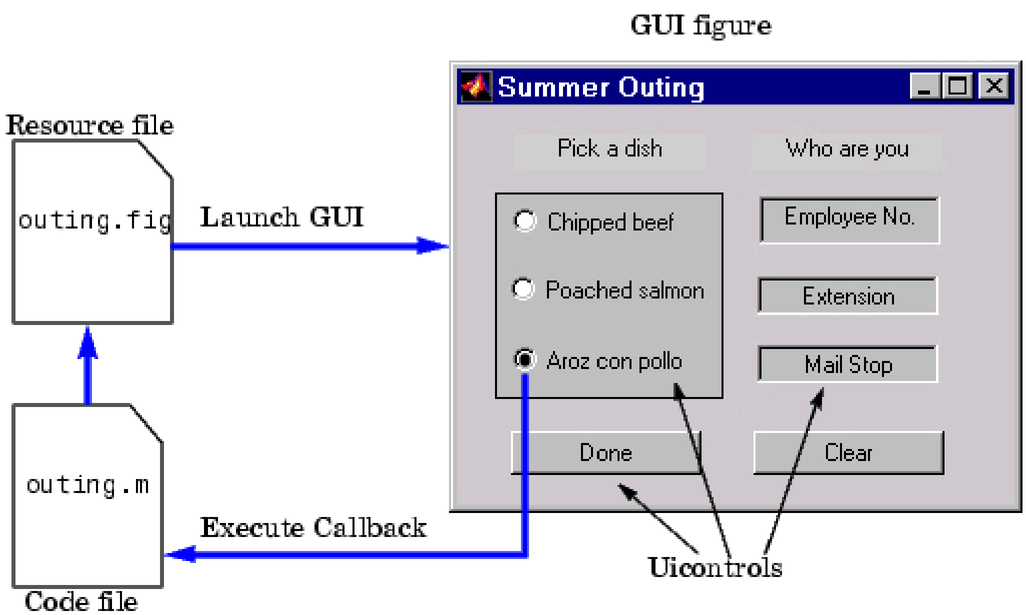


FIG 3.7.2 graphical user blocks

3.7.3 Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

3.7.4 Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques used in the application M-file.

Application Examples - a collection of examples that illustrate techniques which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

3.7.5 User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- Listboxes

- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)
button_state = get(h,'Value');
if button_state == get(h,'Max')
    % toggle button is pressed
```

```
elseif button_state == get(h,'Min')
   % toggle button is not pressed
end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);
a(:,:,2) = rand(16,128);
a(:,:,3) = rand(16,128);
set(h,'CData',a)
```

Radio Buttons

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)
set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)
off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];
mutual_exclude(off)
% Continue with callback
   .
   .
   .
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)
if (get(h,'Value') == get(h,'Max'))
   % then checkbox is checked-take approriate action
else
   % checkbox is not checked-take approriate action
end
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)
user_string = get(h,'string');
```

% proceed with callback...

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)
user_entry = str2double(get(h,'string'));
if isnan(user_entry)
   errordlg('You must enter a numeric value','Bad Input','modal')
end
% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menubar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menubar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

Listboxes differentiate between single and double clicks on an item and set the figure SelectionType property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure SelectionType property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)
val = get(h,'Value');
switch val
case 1
% The user selected the first item
case 2
% The user selected the second item
% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)
val = get(h,'Value');
string_list = get(h,'String');
selected_string = string_list{val}; % convert from cell array to string
```

% etc.

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes ButtonDownFcn property to define the callback.

3.7.6 Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure.

CHAPTER-4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM OF PROJECT:

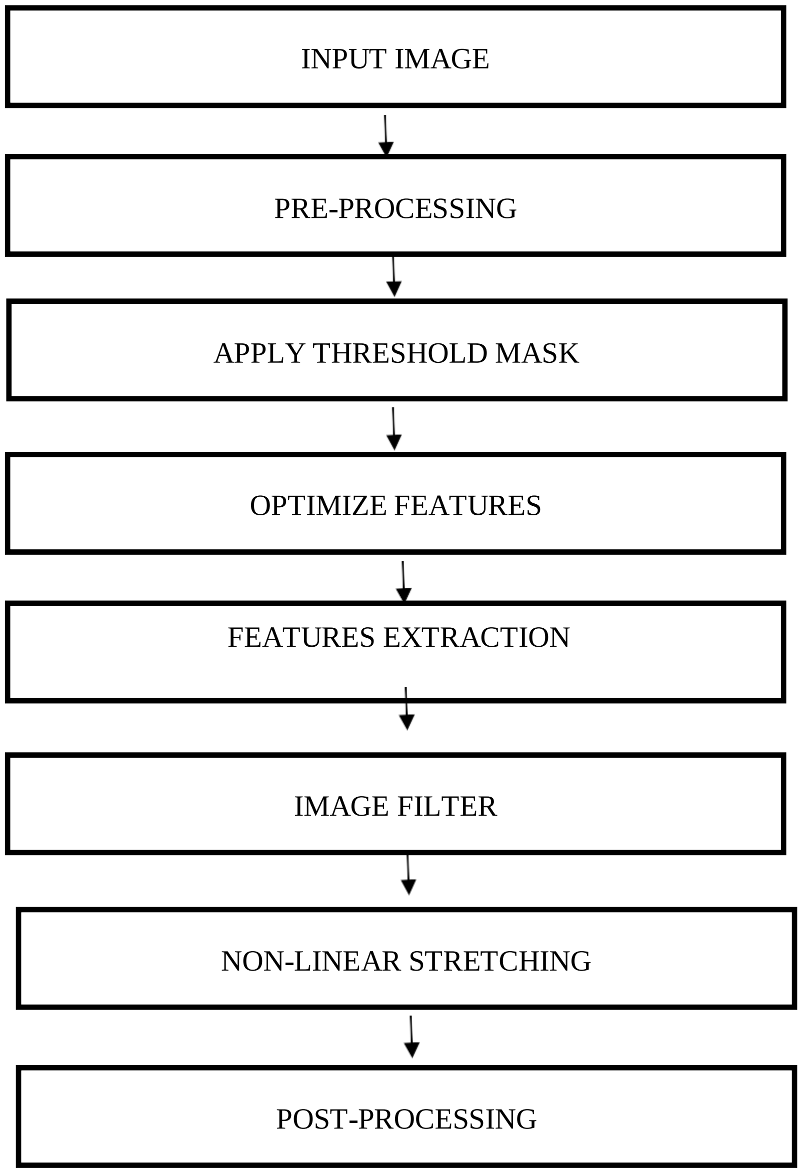


Fig.4.1: Block diagram of the project

4.1.1 INPUT IMAGE :

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

4.1.2 PRE-PROCESSING:

Image pre-processing is the name for operations on images at the lowest level of abstraction whose aim is an improvement of the image data that suppress undesired distortions or enhances some image features important for further processing. Image pre- processing use the redundancy in images.

4.1.3 APPLY THRESHOLD MASK:

Thresholds. Image thresholding is a simple form of image segmentation. It is a way to create a binary image from a single band or multi-band image. The process is typically done in order to separate "object" or foreground pixels from background pixels to aid in image processing.

4.1.4 OPTIMIZE FEATURES:

Optimizing web images is a process of delivering high-quality images in the right format, dimension, size, and resolution while keeping the smallest possible size. Image optimization can be done in different ways, be it by resizing the images, caching, or by compressing the size.

4.1.5 FEATURES EXTRACTION:

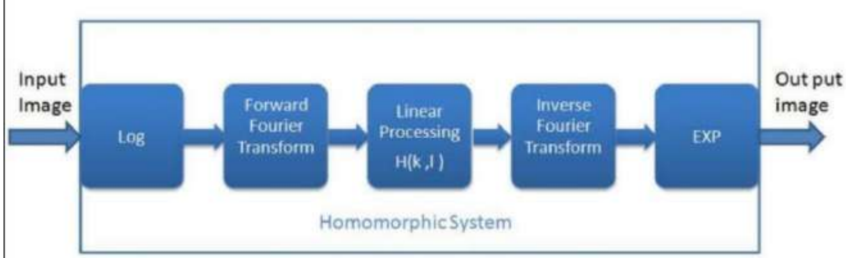
Feature extraction is a part of the dimensionality reduction process, in which, an initial set of the raw data is divided and reduced to more manageable groups. So when you want to process it will be easier. The most important characteristic of these large data sets is that they have a large number of variables. These variables require a lot of computing resources to process them. So Feature extraction helps to get the best feature from those big data sets by select and combine variables into features, thus, effectively reducing the amount of data. These features are easy to process, but still able to describe the actual data set with the accuracy and originality.

4.1.6 IMAGE FILTER:

An image filter is a technique through which size, colors, shading and other characteristics of an image are altered. An image filter is used to transform the image using different graphical editing techniques. Image filters are usually done through graphic design and editing software

4.1.6.1 HOMOMORPHIC FILTER

Homomorphic filtering is a generalized technique for signal and image processing, involving a nonlinear mapping to a different domain in which linear filter techniques are applied, followed by mapping back to the original domain.



4.1.7 NON-LINEAR STRETCHING:

Non linear stretching is the work horse for producing "pretty pictures" with astrophotography. This process completely destroys the data for any scientific use and is frowned upon by some purists. However, this is the only way to simultaneously show the faint and bright detail. Here's how to do it.

4.1.8 POST PROCESSING:

Image post processing enhances the quality of a finished image to prepare it for publication and distribution. It includes techniques to clean up images to make them visually clearer as well as the application of filters and other treatments to change the look and feel of a picture.

4.2 IMAGE:

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Fig 4.1.1 General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

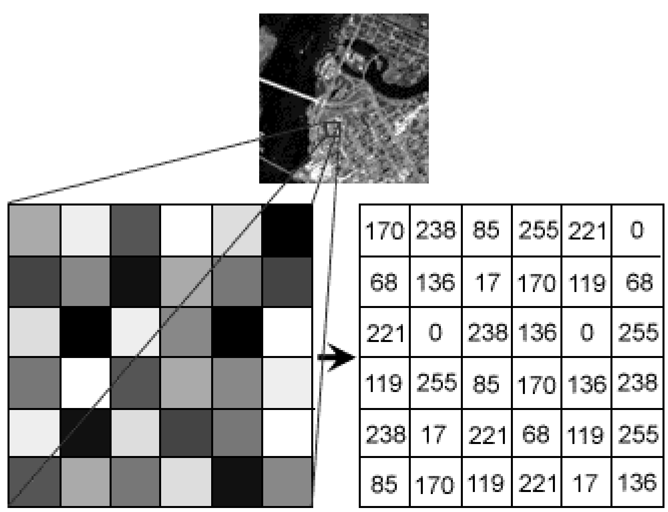


Fig 4.1.2 Image pixel

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

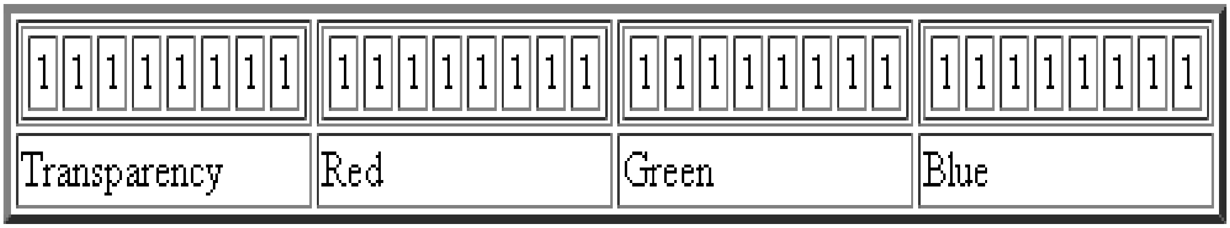


Fig 4.1.3. Transparency image

4.2.1 IMAGE FILE SIZES:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

4.2.2 IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

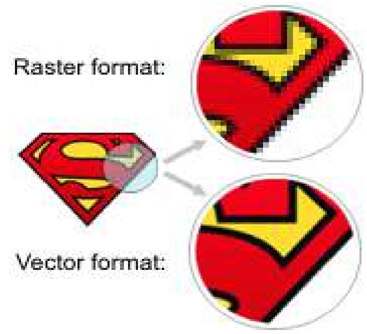


Fig4.1.4 Resolution image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

RASTER FORMATS:

These formats store images as bitmaps (also known as pixmaps)

- **JPEG/JFIF:**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

- **EXIF:**

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

- **TIFF:**

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

- **PNG:**

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

- **GIF:**

GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

- **BMP:**

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

4.2.3 VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM:**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

- **SVG:**

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

4.2.4 IMAGE PROCESSING:

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man’s ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

4.3 FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:

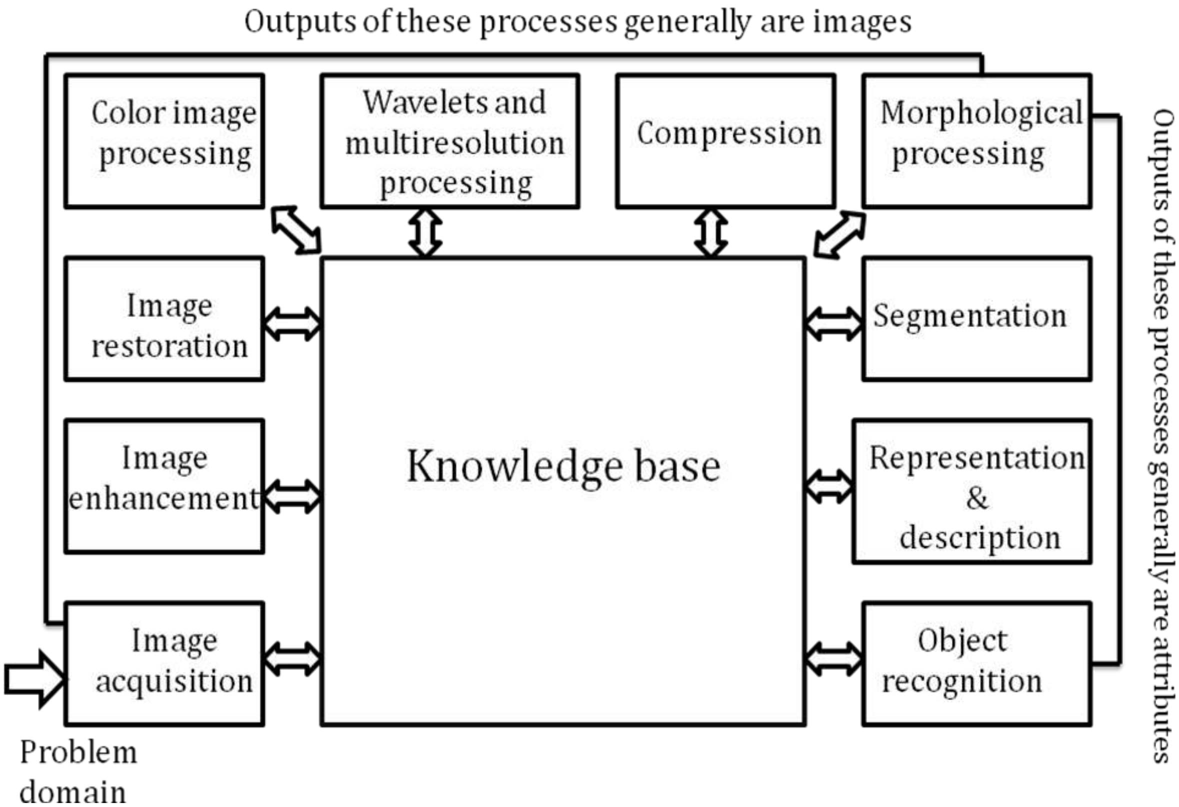


Fig 4.1.5 Image fundamental

4.3.1 Image Acquisition:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Fig 4.1.6 Digital camera image

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Fig 4.1.7 digital camera cell

4.3.2 Image Enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because "it looks better." It is important to keep in mind that enhancement is a very subjective area of image processing.

Fig 4.1.8 Image enhancement

4.3.3 Image restoration:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Fig 4.1.9 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, where as removal of image blur by applying a deblurring function is considered a restoration technique.

4.3.4 Color image processing:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Fig 4.1.10 Color & Gray scale image

4.3.5 Wavelets and multiresolution processing:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform based image processing since the late1950’s, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

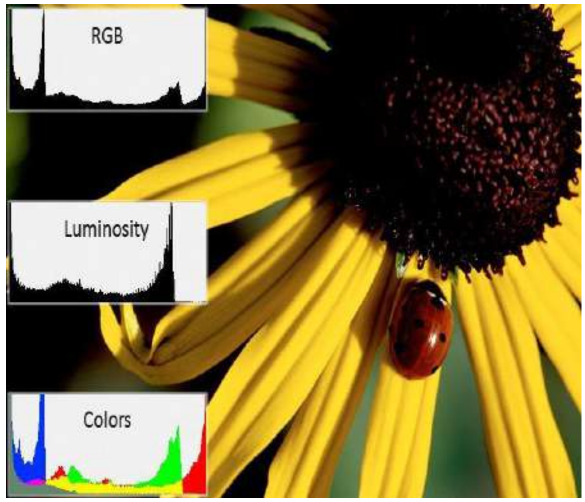


Fig 4.1.11 rgb histogram image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called **Multiresolution** theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

Compression:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

4.3.6 Morphological processing:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Fig 4.1.12 blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

4.3.7 Segmentation:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

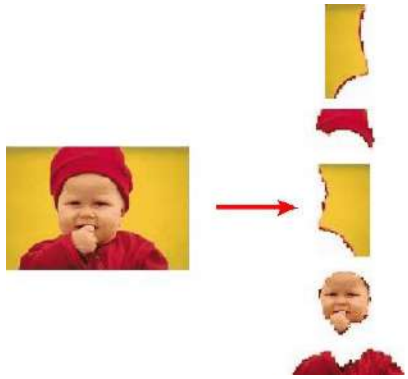


Fig 4.1.13 Image segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

4.3.8 Representation and description:

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

4.3.9 Object recognition:

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

4.3.10 Knowledgebase:

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an inter related to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

4.4 COMPONENTS OF AN IMAGE PROCESSING SYSTEM:

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

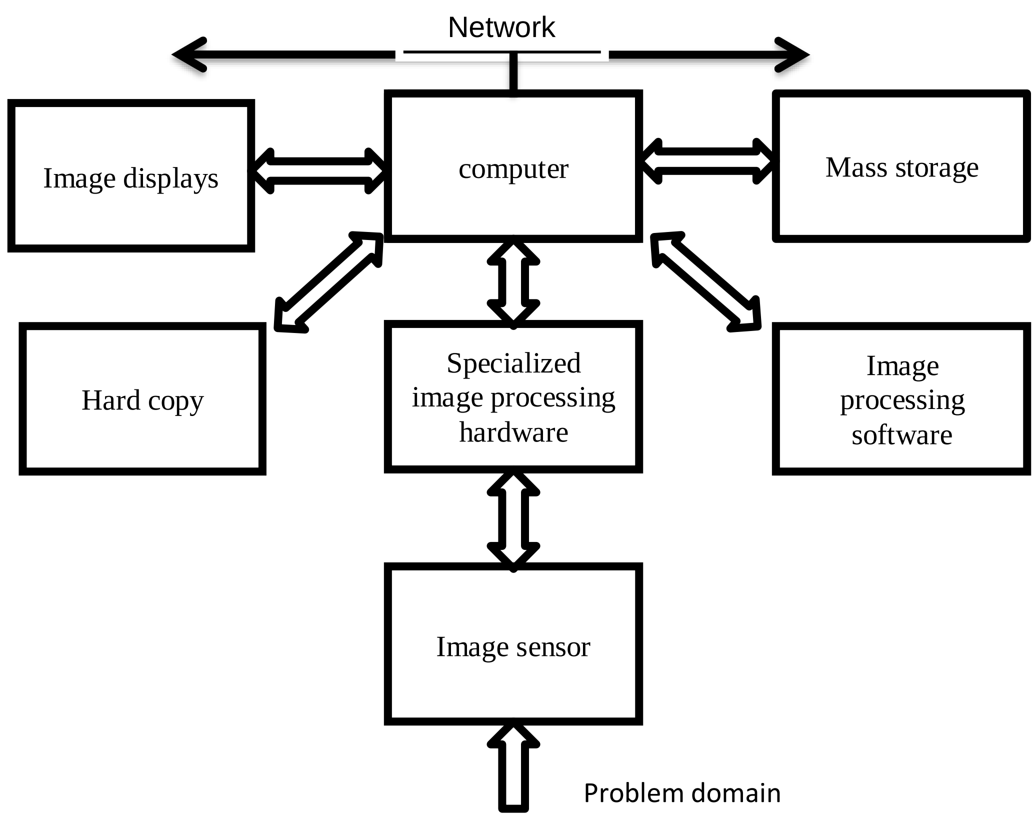


Fig 4.1.14 Component of image processing

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 1.24 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

- **Image sensors:**

  With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

- **Specialized image processing hardware:**

  Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

- **Computer:**

  The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

- **Image processing software:**

  Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

- **Mass storage:**

  Mass storage capability is a must in image processing applications. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage forimage processing applications fall into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), Gbytes (meaning giga, or one billion, bytes), and Tbytes (meaning tera, or one trillion, bytes)

  One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig. 1.24. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in “jukeboxes” are the usual media for archival applications.

- **Image displays:**

  Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are there requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

- **Hardcopy:**

  Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

- **Network:**

  Networking is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of optical fiber and other broadband technologies.

Color and texture are two low-level features widely used for image classification, indexing and retrieval. Color is usually represented as a histogram, which is a first order statistical measure that captures global distribution of color in an image One of the main drawbacks of the histogram- based approaches is that the spatial distribution and local variations in color are ignored. Local spatial variation of pixel intensity is commonly used to capture texture information in an image. Grayscale Co-occurrence Matrix (GCM) is a well-known method for texture extraction in the spatial domain. A GCM stores the number of pixel neighborhoods in an image that have a particular grayscale combination. Let I be an image and let p and Np respectively denote any arbitrary pixel and its neighbor in a given direction. If GL denotes the total number of quantized gray levels and gl denotes the individual gray levels, where, gl∈{0, . . .,GL _ 1}, then each component of GCM can be written as follows:

$$\mathrm{gcm}(i,j) = \Pr((\mathrm{gl}_{\mathrm{p}}, \mathrm{gl}_{N_{\mathrm{p}}}) = (i,j))$$

gcm(i, j) is the number of times the gray level of a pixel p denoted by glp equals i, and the gray level of its neighbor Np denoted by glNp equals j, as a fraction of the total number of pixels in the image. Thus, it estimates the probability that the gray level of an arbitrary pixel in an image is i, and that of its neighbor is j. One GCM matrix is generated for each possible neighborhood direction, namely, 0, 45, 90 and 135.Average and range of 14 features like Angular Second Moment, Contrast, Correlation, etc., are generated by combining all the four matrices to get a total of 28 features. In the GCM approach for texture extraction, color information is completely lost since only pixel gray levels are considered.

To incorporate spatial information along with the color of image pixels, a feature called color correlogram has recently been proposed. It is a three dimensional matrix that represents the probability of finding pixels of any two given colors at a distance 'd' apart Auto correlogram is a variation of correlogram, which represents the probability of finding two pixels with the same color at a distance 'd' apart. This atproach can effectively represent color distribution in an image. However, correlogram features do not capture intensity variation Many image databases often contain both color as well as gray scale images. The color correlogram method does not constitute a good descriptor in such databases.

Another method called Color Co-occurrence Matrix (CCM) has been proposed to capture color variation in an image. CCM is represented as a three-dimensional matrix, where color pair of the pixels p and Np are captured in the first two dimensions of the matrix and the spatial distance 'd' between these two pixels is captured in the third dimension. This approach is a generalization of the color correlogram and reduces to the pure color correlogram for d = 1. CCM is generated using only the Hue plane of the HSV (Hue, Saturation and Intensity Value) color space. The Hue axis is quantized into HL number of levels. If individual hue values are denoted by hl, where $\mathrm{hl} \in \{0, \ldots, \mathrm{HL}-1\}$, then each component of CCM can be written as follows:

$$\mathrm{ccm}(i,j) = \Pr((\mathrm{hl}_\mathrm{p}, \mathrm{hl}_{N_\mathrm{p}}) = (i,j))$$

Four matrices representing neighbors at angles 0, 90, 180 and 270 are considered. This approach was further extended by separating the diagonal and the non-diagonal components of CCM to generate a Modified Color Co-occurrence Matrix (MCCM). MCCM, thus, may be written as follows: MCCM = (CCMD;CCMND)

Here, CCMD and CCMND correspond to the diagonal and off-diagonal components of CCM. The main drawback of this approach is that, like correlogram, it also captures only color information and intensity information is completely ignored.

An alternative approach is to capture intensity variation as a texture feature from an image and combine it with color features like histograms using suitable weights . One of the challenges of this approach is to determine suitable weights since these are highly application-dependent. In certain applications like Content-based Image Retrieval (CBIR), weights are often estimated from relevance feedback given by users.

While relevance feedback is sometimes effective, it makes the process of image retrieval user-dependent and iterative. There is also no guarantee on the convergence of the weight-learning algorithms. In order to overcome these problems, researchers have tried to combine color and texture features together during extraction.

proposed two approaches for capturing color and intensity variations from an image using the LUV color space. In the Single-channel Co-occurrence Matrix (SCM), variations for each color channel, namely, L, U and V are considered independently. In the Multi channel Co-occurrence Matrix (MCM), variations are captured taking two channels at a time – UV, LU and LV. Since the LUV color space separates out chrominance (L and U) from luminance (V), SCM in effect, generates one GCM and two CCMs from each image independently. As a result, correlation between the color channels is lost

However, in MCM, the count of pair wise occurrences of the values of different channels of the color space is captured. Thus, each component of MCM can be written as follows:

$$mcmUV(i; j) = Pr((up; vNp) = (i; j))$$

$$mcmLU(i; j) = Pr((lp; uNp) = (i; j))$$

$$mcmLV(i; j) = Pr((lp; vNp) = (i; j))$$

Here, mcmUV(i, j) is the number of times the U chromaticity value of a pixel p denoted by up equals i, and the V chromaticity value of its neighbor Np denoted by vNp equals j, as a fraction of the total number of pixels in the image. Similarly, mcmLU(i, j) and mcmLV(i, j) are defined. One MCM matrix is generated for each of the four neighborhood directions, namely, 0, 45, 90 and 135.

Deng and Manjunath (2001) proposed a two-stage method called JSEG, which combines color and texture after image segmentation. In the first stage, colors are quantized to the required levels for differentiating between various regions of an image. Pixel values of the regions are then replaced by their quantized color levels to form a color map. Spatial variation of color levels between different regions in the map is viewed as a type of texture composition of the image.

Yu et al. (2002) suggested the use of color texture moments to represent both color and texture of an image. This approach is based on the calculation of

Local Fourier Transformation (LFT) coefficients. Eight templates equivalent to LFT are operated over an image to generate a characteristic map of the image. Each template is a 3 · 3 filter that considers eight neighbors of the current pixel for LFT calculation. First and second order moments of the characteristic map are then used to generate a set of features.

In this paper, we propose an integrated approach for capturing spatial variation of both color and intensity levels in the neighborhood of each pixel using the HSV color space. In contrast to the other methods, for each pixel and its neighbor, the amount of color and intensity variation between them is estimated using a weight function. Suitable constraints are satisfied while choosing the weight function for effectively relating visual perception of color and the HSV color space properties. The color and intensity variations are represented in a single composite feature known as Integrated Color and Intensity Co-occurrence Matrix (ICICM). While the existing schemes generally treat color and intensity separately, the proposed method provides a composite view to both color and intensity variations in the same feature. The main advantage of using ICICM is that it avoids the use of weights to combine individual color and texture features. We use ICICM feature in an image retrieval application from large image databases.

Early result on this work was reported in (Vadivel et al., 2004a). In the next section, we describe the proposed feature extraction technique after introducing some of the properties of the HSV color space. Choice of quantization levels for color and intensity axes, selection of parameter values and a brief overview of the image retrieval application

Integrated color and intensity co-occurrence matrix:

We propose to capture color and intensity variation around each pixel in a two-dimensional matrix called Integrated Color and Intensity Co-occurrence Matrix (ICICM). This is a generalization of the Grayscale Co-occurrence Matrix and the Color Co-occurrence Matrix techniques. For each pair of neighboring pixels, we consider

their contribution to both color perception as well as gray level perception to the human eye. Some of the useful properties of the HSV color space and their relationship to human color perception are utilized for extracting this feature. In the next sub-section, we briefly explain relevant properties of the HSV color space. In the subsequent subsection, we describe how the properties can be effectively used for generating ICICM.

HSV color space:

HSV Color space: Basically there are three properties or three dimensions of color that being hue, saturation and value HSV means Hue, Saturation and Value. It is important to look at because it describes the color based on three properties. It can create the full spectrum of colors by editing the HSV values. The first dimension is the Hue. Hue is the other name for the color or the complicated variation in the color. The quality of color as determined by its dominant wavelength. This Hue is broadly classified into three categories. They are primary Hue, Secondary Hue and Teritiary Hue. The first and the foremost is the primary Hue it consists of three colors they are red, yellow and blue. The secondary Hue is formed by the combination of the equal amount of colors of the primary Hue and the colors of the secondary Hue which was formed by the primary Hue are Orange, Green and violet. The remaining one is the teritiary Hue is formed by the combination of the primary Hue and the secondary Hue. The limitless number of colors are produced by mixing the colors of the primary Hue in different amounts. Saturation is the degree or the purity of color. Then the second dimension is the saturation. Saturation just gives the intensity to the colors. The saturation and intensity drops just by mixing the colors or by adding black to the color. By adding the white to the color in spite of more intense the color becomes lighter. Then finally the third dimension is the Value. The value is the brightness of the color. When the value is zero the color space is totally black with the increase in the color there is also increase in the brightness and shows the various colors. The value describes the contrast of the color. That means it describes the lightness and darkness of the color. As similar to the saturation this value consists of the tints and shades. Tints are the colors with the added white and shades are the colors with the added black.

Properties of the HSV color space:

Sensing of light from an image in the layers of human retina is a complex process with rod cells contributing to scotopic or dim-light vision and cone cells to photopic or bright-light vision (Gonzalez and Woods, 2002). At low levels of illumination, only the rod cells are excited so that only gray shades are perceived. As the illumination level increases, more and more cone cells are excited, resulting in increased color perception. Various color spaces have been introduced to represent and specify colors in a way suitable for storage, processing or transmission of color information in images. Out of these, HSV is one of the models that separate out the luminance component (Intensity) of a pixel color from its chrominance components (Hue and Saturation). Hue represents pure color, which is perceived when incident light is of sufficient illumination and contains a single wavelength. Saturation gives a measure of the degree by which a pure color is diluted by white light. For light with low illumination, corresponding intensity value in the HSV color space is also low.

The HSV color space can be represented as a Hexa cone, with the central vertical axis denoting the luminance component, I (often denoted by V for Intensity Value). Hue, is a chrominance component defined as an angle in the range [0,2p] relative to the red axis with red at angle 0, green at 2p/3, blue at 4p/3 and red again at 2p. Saturation, S, is the other chrominance component, measured as a radial distance from the central axis of the hexacone with value between 0 at the center to 1 at the outer surface. For zero saturation, as the intensity is increased, we move from black to white through various shades of gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the color represented by its hue. When saturation is near 0, all the pixels in an image look alike even though their hue values are different.

As we increase saturation towards 1, the colors get separated out and are visually perceived as the true colors represented by their hues. Low saturation implies presence of a large number of spectral components in the incident light, causing loss of color information even though the illumination level is sufficiently high. Thus, for low values of saturation or intensity, we can approximate a pixel color by a gray level while for higher saturation and intensity, the pixel color can be approximated by its hue. For low intensities, even for a high saturation, a pixel color is close to its gray value. Similarly, for low saturation even for a high value of intensity, a pixel is perceived as gray. We use these properties to estimate the degree by which a pixel contributes to color perception and gray level perception.

One possible way of capturing color perception of a pixel is to choose suitable thresholds on the intensity and saturation. If the saturation and the intensity are above their respective thresholds, we may consider the pixel to have color dominance; else, it has gray level dominance. However, such a hard thresholding does not properly capture color perception near the threshold values. This is due to the fact that there is no fixed level of illumination above which the cone cells get excited. Instead, there is a gradual transition from scotopic to photopic vision. Similarly, there is no fixed threshold for the saturation of cone cells that leads to loss of chromatic information at higher levels of illumination caused by color dilution. We, therefore, use suitable weights that vary smoothly with saturation and intensity to represent both color and gray scale perception for each pixel.

4.5 <u>NON INTERVAL QUANTIZATION:</u>

Due to the large range for each component by directly calculating the characteristics for the retrieval then the computation will be very difficult to ensure rapid retrieval. It is essential to quantify HSV space component to reduce computation and improve efficiency. At the same time, because the human eye to distinguish colors is limited, do not need to calculate all segments. Unequal interval quantization according the human color perception has been applied on H , S ,V components.

Based on the color model of substantial analysis, we divide color into eight parts. Saturation and intensity is divided into three parts separately in accordance with the human eyes to distinguish. In accordance with the different colors and subjective color perception quantification, quantified hue(H), saturation(S) and value(V) In accordance with the quantization level above, the H, S, V three-dimensional feature vector for different values of with different weights to form one dimensional feature vector and is given by the following equation:

$$G = Qs*Qv*H+Qv*s+V$$

Where Qs is the quantized series of S and Qv is the quantized series of V. And now by setting Qs = Qv = 3, Then G = 9H+3S+V

$$H = \begin{cases} 0 \ \textit{if} \ \ h \in [316,20] \\ 1 \ \textit{if} \ \ h \in [21,40] \\ 2 \ \textit{if} \ \ h \in [41,75] \\ 3 \ \textit{if} \ \ h \in [76,155] \\ 4 \ \textit{if} \ \ h \in [156,190] \\ 5 \ \textit{if} \ \ h \in [191,270] \\ 6 \ \textit{if} \ \ h \in [271,295] \\ 7 \ \textit{if} \ \ h \in [296,315] \end{cases} \qquad S = \begin{cases} 0 \ \textit{if} \ \ s \in [0,0.2) \\ 1 \ \textit{if} \ \ s \in [0.2,0.7) \\ 2 \ \textit{if} \ \ s \in [0.7,1) \end{cases} \qquad V = \begin{cases} 0 \ \textit{if} \ \ v \in [0,0.2) \\ 1 \ \textit{if} \ \ v \in [0.2,0.7) \\ 2 \ \textit{if} \ \ v \in [0.7,1) \end{cases}$$

In this way three component vector of the HSV from one dimensional vector, Which quantize the whole color space for the 72 kinds of the main colors. So we can handle 72 bins of one dimensional histogram. This qualification is effective in reducing the images by the effect of the light intensity, but also reducing the computational time and complexity.

4.6 IMAGE RETRIEVAL:

Image retrieval is nothing but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords or the descriptions to the images so that the retrieval can be performed. Manual image annotation is time consuming, expensive and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc..,

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others. one can today generate volumes of images with content as diverse as family get-togethers and national park visits. Low-cost storage and easy Web hosting has fueled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchable image data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real-world image search system designers.

As far as technological advances are concerned, growth in Content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real world implications, applications, and constraints of the technology. Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future.

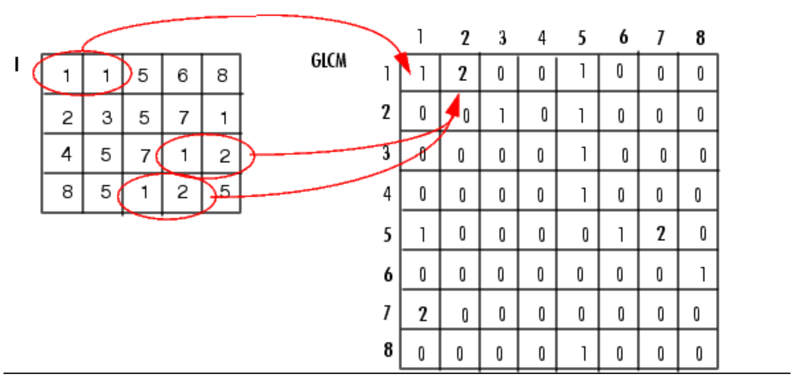
An image retrieval system designed to serve a personal collection should focus on features such as personalization, flexibility of browsing, and display methodology. For example, Google's Picasa system [Picasa 2004] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods. The notable few applications include an FPGA implementation of a color-histogram-based image retrieval system [Kotoulas and Andreadis 2003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 2003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 2002].

Discussion. Regardless of the nature of the collection, as the expected user-base grows, factors such as concurrent query support, efficient caching, and parallel and distributed processing of requests become critical. For future real-world image retrieval systems, both software and hardware approaches to address these issues are essential. More realistically, dedicated specialized servers, optimized memory and storage support, and highly parallelizable image search algorithms to exploit cluster computing powers are where the future of large-scale image search hardware support lies.

4.7 <u>OVERVIEW OF TEXTURE:</u>

We all know about the term Texture but for defining it is a hard time. One can differentiate the two different Textures by recognizing the similarities and differences. Commonly there are three ways for the usage of the Textures:

Based on the Textures the images can be segmented To differentiate between already segmented regions or to classify them.We can reproduce Textures by producing the descriptions. The texture can be analyzed in three different ways. They are Spectral, Structural and Statistical:



4.8 FLOW CHART

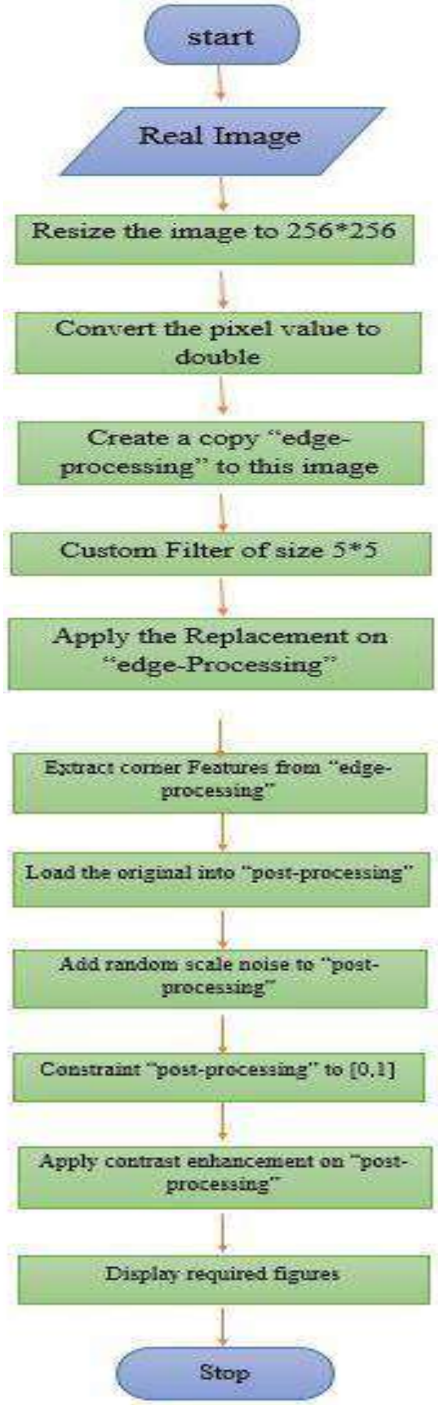


Fig.4.2 Flow Chart

4.8.1 START:

The 'start' parameter defines the starting centroids. The 'replicates' parameter indicates how many times to repeat the clustering. You only gave one set of starting values, so it doesn't know what to make of that.

4.8.2 REAL IMAGE:

This object is naturally suited to the representation of images, real-valued ordered sets of color or intensity data. MATLAB stores most images as two-dimensional matrices, in which each element of the matrix corresponds to a single discrete pixel in the displayed image.

4.8.3 RESIZE THE IMAGE 256*256:

Double is the default numeric data type (class) in MATLAB, providing sufficient precision for most computational tasks. Numeric variables are automatically stored as 64-bit (8-byte) double-precision floating-point values.

4.8.4 CONVERT THE PIXEL VALUE TO DOUBLE:

Double(s) converts the symbolic value s to double precision. Converting symbolic values to double precision is useful when a MATLAB function does not accept symbolic values. For differences between symbolic and double-precision numbers, see Choose Numeric or Symbolic Arithmetic.

4.8.5 CREATE A COPY ”EDGE PROCESSING” TO IMAGE:

Edges are one such feature. Edges are significant local changes in the image and are important features for analyzing images. An edge in an image is a significant local change in the image intensity, usually associated with a discontinuity in either the image intensity or the first derivative of the image intensity.

The edge function calculates the gradient using the derivative of a Gaussian filter. This method uses two thresholds to detect strong and weak edges, including weak edges in the output if they are connected to strong edges

4.8.6 CUSTOM FILTER OF SIZE 5*5:

Filters are data processing techniques that can smooth out high-frequency fluctuations in data or remove periodic trends of a specific frequency from data. In MATLAB, the filter function filters a vector of data x according to the following difference equation, which describes a tapped delay-line filter.

4.8.7 MEAN FILTER:

Median filtering is a nonlinear operation often used in image processing to reduce "salt and pepper" noise. A median filter is more effective than convolution when the goal is to simultaneously reduce noise and preserve edges.

4.8.8 VARIANCE FILTER:

V = var(A) returns the variance of the elements of A along the first array dimension whose size does not equal 1. If A is a vector of observations, the variance is a scalar. The variance is normalized by the number of observations -1 by default. If A is a scalar, var(A) returns 0.

4.8.9 APPLY THE REPLACEMENT ON ”EDGE-PROCESSING” :

newStr = replace(str , old , new) replaces all occurrences of the substring old with new . If old contains multiple substrings, then new either must be the same size as old , or must be a single substring.

After finding the mean and variance value we are replacing starting original pixel values with respect to the minimum value.

4.8.10 EXTRACT CORNER FEATURES FROM ”EDGE-PROCESSING”

Examples

1.Extract Corner Features from an Image. Open Live Script. Read the image. I = imread('cameraman. tif');

2.Extract SURF Features from an Image. Open Live Script. Read image. I = imread('cameraman. tif');

3.Extract MSER Features from an Image. Open Live Script. Read image. I = imread('cameraman. tif');

Feature extraction for image data represents the interesting parts of an image as a compact feature vector.

...

Feature extraction techniques provided by Computer Vision Toolbox™ and Image Processing Toolbox™ include:

1.Histogram of oriented gradients (HOG)

2.Speeded-up robust features (SURF)

3.Local binary pattern (LBP) features.

4.8.11 LOAD THE ORIGINAL INTO ”POST-PROCESSING” :

load (filename,'-mat',variables) loads the specified variables from filename. S = load (_) loads data into S, using any of the input arguments in the previous syntax group. If filename is a MAT-file, then S is a structure array. If filename is an ASCII file, then S is a double-precision array containing data from the file.

Load the original image into post-processing and normalize it.

4.8.12 ADD RANDOM SCALE NOSE TO ”POST-PROCESSING” :

J = imnoise(I ,'speckle') adds multiplicative noise using the equation J = I+n*I

where n is uniformly distributed random noise with mean 0 and variance 0.05.

Image noise is random variation of brightness or color information in images, and is usually an aspect of electronic noise. .... Image noise is an undesirable by-product of image capture that obscures the desired information.

4.8.13 CONSTRAINT TO ”POST-PROCESSING” TO [0,1] :

Constraints limit the set of x over which a solver searches for a minimum. You can have any number of constraints, which are inequalities or equalities. All Optimization Toolbox optimization functions minimize an objective function. To maximize a function f, apply an optimization routine to minimize –f.

In mathematical optimization, constrained optimization (in some contexts called constraint optimization) is the process of optimizing an objective function with respect to some variables in the presence of constraints on those variables.

4.8.14 APPLY CONTRAST ENHANCEMENT ON ”POST-PROCESSING” :

One of the important techniques in digital image processing is to enhance images. Contrast enhancement is a method that is used to enhance images for viewing process or for further analysis of images. Main idea behind contrast enhancement techniques is to increase contrast and to preserve original brightness of images. In this paper a contrast enhancement technique is proposed that first segments histogram of image recursively and then applies Adaptive Gamma Correction with Weighting Distribution (AGCWD) Technique. The proposed technique is basically an improvement over AGCWD technique and aims to get better contrast enhancement and brightness preservation than AGCWD technique. The image enhancement is one of the significant techniques in digital image processing. It has an important role in various fields where images are to be understood and analyzed. Image enhancement is done on an image to improve its visual effects and quality or to make it more appropriate for further processing by another application. An image can have low contrast or bad quality due to a number of reasons like poor quality of imaging device, adverse external conditions at the time of image acquisition and many more. The contrast enhancement is one of the commonly used image enhancement method .

Enhance the image using the three contrast adjustment techniques. pout = imread('pout. tif'); pout_imadjust = imadjust(pout); pout_histeq = histeq(pout); pout_adapthisteq = adapthisteq(pout); Display the original image and the three contrast adjusted images as a montage.

4.8.15 DISPLAY REQUIRED FIGURES:

To display image data, use the imshow function. The following example reads an image into the workspace and then displays the image in a figure window using the imshow function. moon = imread('moon. tif'); imshow(moon);

4.8.16 STOP:

To stop execution of a MATLAB command, press Ctrl+C or Ctrl+Break. On Apple Macintosh platforms, you also can use Command+

4.9 IMPLEMENTATION STAGES:

4.9.1 IMPLEMENTATION STAGES ON IMAGE 1:

Stage-1: original image with low contrast and high noise .

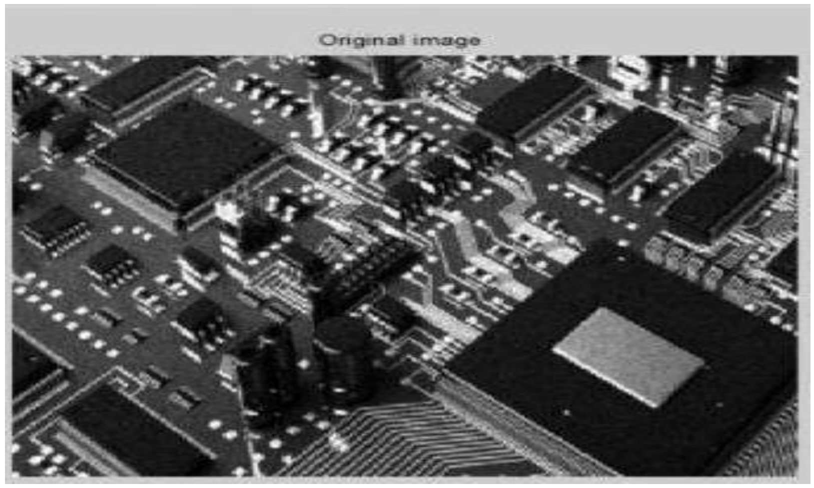


Fig 4.3.1 original image

Stage-2: processing image under goes many stages like contrast enhancement,so this contrast enhacement will produces an image that subjectively looks better than the original image by changing the pixel intensity.

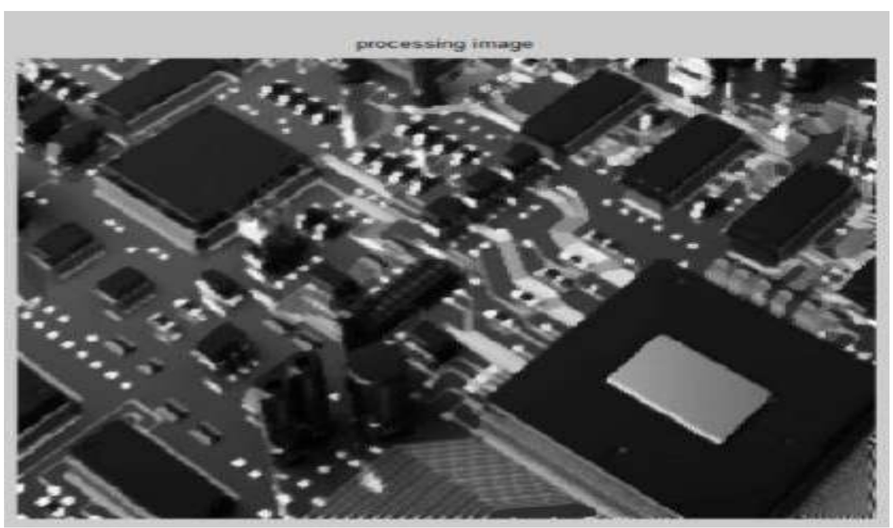


Fig.4.3.2 processing image

Stage-3: It is the output image by avoiding unwanted noise and low contrast and giving a high quality less noise and high contrast image.

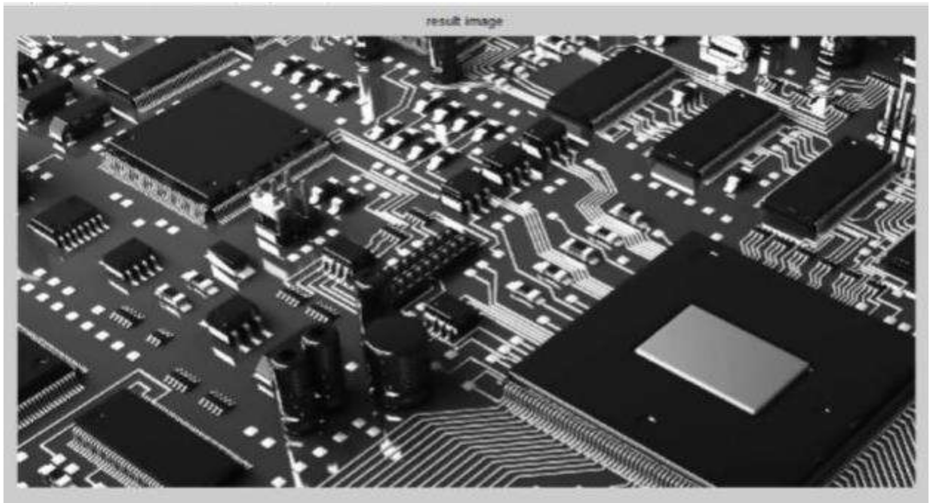


Fig.4.3.3 Result image

4.9.2 : IMPLEMENTATION STAGES ON IMAGE 2:

Stage-1: original image with low contrast and high noise .

Fig 4.3.4 original image

Step-2: Original image to gray scale image.

Fig 4.3.5 Original image to gray scale image

Stage-2: processing image under goes many stages like contrast enhancement,so this contrast enhacement will produces an image that subjectively looks better than the original image by changing the pixel intensity.

Fig 4.3.6 processing image

Stage-3: It is the output image by avoiding unwanted noise and low contrast and giving a high quality less noise and high contrast image.

Fig 4.3.7 Result image

CHAPTER-5
PROJECT TESTING

5.1 OVERVIEW OF TESTING METHODS

This project proposes to investigate and test a number of image processing techniques that enhance the identification of persons in consumer digital images. The final product of the research in an integrated software tool that allows for :

- Image marking : load an image from the database and mark important features on it (e.g. faces). Save the location within the picture and the type of marking into the database.
- Algorithm Testing : automatically apply an image processing algorithm (e.g. face detection and recognition) to a set of marked images (only to those that meet a specific criteria)

This masters research in carried out as part of the “Tools and Algorithms to Assist in Automatically Recognizing and Deducing Information about People in Consumer Digital Images” Enterprise Ireland funded project

The Research Team

Dr. Peter Corcoran Mentoring Supervisor NUIG Dr. Dana Vasiloaica Research Supervisor Department of Business & Humanities, IT, Sligo. Mr. Collin Callanan Research Student Department of Business & Humanities, IT Sligo.

Project Details

Aims and Objectives of the proposed programme of research.

– Research into the following topics

– User Interface Design

– Software Testing

– Distributed Databases

Prepare a series of test image databases for testing and verification of the algorithms developed as part of the “Tools and Algorithms to Assist in Automatically Recognizing and Deducing Information about People in Consumer Digital Images” Project.

Develop automatic and semi automatic tools to facilitate the marking and categorisation of the images mentioned in Objective 2.

Process image data-sets to generate test-sets for specific algorithm verification and comparative evaluation test.

Develop batch tests to support the testing of algorithms.

Prepare user manual and help files to assist users in working with the image marking and algorithm testing software application.

CHAPTER-6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

In this project the problem of operative (in real-time) image enhancement in automatic mode (with acceptable level of computational costs) for complex monochrome images was considered. The various known approaches to the image enhancement based on nonlinear statistical non-inertial transformations of image were considered. A new method of image enhancement by adaptive nonlinear stretching using piecewise linear transformation of the brightness scale was proposed at which the average value of the brightness of transformed image is closest to the middle of the brightness range. Research of the proposed and well-known histogram-based methods of image contrast enhancement in automatic mode was carried out using known no-reference contrast metrics. The proposed method of nonlinear stretching by adaptive piece-wise linear transformation of the dynamic range of image brightness provides an effective image enhancement for all test images without reducing the contrast of small-sized objects. The proposed method of adaptive nonlinear stretching provide effective contrast enhancement and can be recommended for image enhancement in automatic mode..

6.2 FUTURE ENHANCEMENT

The future of image processing will involve scanning the heavens for other intelligent life out in space. Also new intelligent, digital species created entirely by research scientists in various nations of the world will include advances in image processing applications.

PUBLICATION

Submitted Paper in the Conference ICISECT - 21 with Paper ID (ICISECT21-0061) and got Acceptance for the Paper

REFERENCES

[1] R. C. Gonzalez, R. E. Woods, Digital Image Processing, 2nd edn, Prentice Hall, New Jersey (2002).

[2] Z. Wang, A. C. Bovik, Modern Image Quality Assessment, in: Synthesis Lectures on Image, Video, and Multimedia Processing, Morgan and Claypool Publishers, New York (2006), Vol.2, No.1, pp.1-156.

[3] W. K. Pratt, Digital Image Processing: PIKS Inside. 3rd edn, John Wiley & Sons, New York (2001).

[4] S. Saruchi, "Adaptive Sigmoid Function to Enhance Low Contrast Images", International Journal of Computer Applications, 2012, Volume 55, No.4, pp. 45-49.

[5] Y. T. Kim, "Contrast Enhancement using Brightness Preserving BiHistogram equalization", IEEE Transactions on Consumer Electronics, Vol. 43, no. 1, pp. 1-8 (1997).

[6] N. Otsu, "A threshold selection method from gray-level histograms". IEEE Trans. Sys., Man., Cyber. 9: pp. 62-66 (1979).

[7] V. F. Nesteruk, V. A. Sokolova, "Questions of the theory of perception of subject images and a quantitative assessment of their contrast", Optikoelectronic industry. - 1980. – No. 5, pp. 11-13.

[8] E. Yelmanova, Y. Romanyshyn, "No-reference contrast assessment by image histogram", in Proceedings of 14th Int. Conf. CADSM 2017, pp. 148–152. IEEE, Lviv, Ukraine (2017).

[9] R.A. Vorobel, Loharyfmichna obrobka zobrazhen' [Logarithmic Image Processing], Naukova Dumka (2012), Kyiv, Ukraine.

[10] J. Cheng et al., A predicted compensation model of human vision system for low-light image, CISP'10 2010,605-609

[11] N. Zhou et al., Simulating human visual perception in nighttime illumination,Tsinghua Science and Technology 2009, 10:735-745

[12] G. Zhai et al., Efficient Deblocking With Coefficient Regularization, Shape-Adaptive Filtering, and QuantizationConstraint, IEEE Transactions on Multimedia 2008,14:133-1

[13] D. J. Jobson et al., A Multiscale Retinex for Bridging the Gap Between Color Images and the Human Observation of Scenes,IEEE TRANSACTIONS ON IMAGE PROCESSING 1997, 6:965 - 976

[14] D. A. Atchison et al. "Optics of the human eye." (2000):34-35.

[15] R.C. Gonzales and E. Woods,Digital ImageProcessing,Addison-Wesley, (1992)

[16] K. Zuiderveld, Contrast limited adaptive histogram equalization,Academic Press Professional (1994)

[17] C. Yang, Image enhancement by the modified highpass filtering approach, Opt. Int. J. Light Electron. Opt.

doi:10.1016/j.ijleo.2008. 03.016
[18] R. K. Jha, and R. Chouhan, Noise-induced contrastenhancement using stochastic resonance on singular values, Signal, Image and Video Processing, 2012.
[19] R.Chouhan, R. K. Jha, and P. K. Biswas, Wavelet-basedcontrast enhancement of dark images using dynamic stochastic resonance, Indian Conference on Computer Vision, Graphics and Image Processing, 2012, p. 73.
[20] Q. Ye, H. Haining, and C. Zhang, Image enhancement using stochastic resonance, ICIP'04, 2004, Vol. 1.
[21] V. P. Rallabandi, and P. K. Roy, Magnetic resonance image enhancement using stochastic resonance in Fourier domain, Magnetic Resonance Imaging, 2010, 28.9:1361-1373.
[22] R. K. Jha, R.Chouhan and P. K. Biswas, Noise-inducedcontrast enhancement of dark images using non-dynamicstochastic resonance, NCC'12, 2012, p. 1-5.
[23] J. Mukherjee and S. K. Mitra, Enhancement of color images by scaling the DCT coefficients. IEEE Transactions on Image Processing, 2008, 17(10):17831794.
[24] J. Tang, E. Peli and S. Acton. Image enhancement usinga contrast measure in the compressed domain. IEEE Signal Process. Lett., 2003, 10(10):289292.
[25] R. N. Strickland, C. S. Kim, and W. F. McDonnell,Digital color image enhancement based on the saturation component,1987 Opt. Eng., 26(7):609616.
[26] S. Wolf, R. Ginosar, and Y. Zeevi, Spatio-chromatic image enhancement based on a model of humal visual information system, 1998 J. Vis. Commun. Image Represent., 9(1):2537.
[27] S. Susstrunk,S. Winkler, Color image quality on theinternet, SPIE Electronic Imaging: Internet Imaging, 2004, p. 118131.
[28] Z. Wang, No-reference perceptual quality assessment ofjpeg compressed images, ICIP, 2002 vol. 1, p. 477480.
[29] D. Scharstein and R. Szeliski, High-accuracy stereo depth maps using structured light, CVPR'03, 2003, volume 1, pages 195-202.
[30] B. Funt et al, Retinex in MATLAB, 2004, Journal ofElectronic Imaging, 13(1), 48-57.
[31] Y. Yi-Bing et al.,Over exposed image informationrecovery via stochastic resonance, Chinese Physics B, 2012, 21(4), 044204.

BIBILOGRAPHY

[1] Bai X, Zhou F, “Multi Structuring Element Top-hat transform to detect linear features”, IEEE, ICSP, 2010, pp.877-880.

[2] Gonzalez R. C, Woods R, Digital Image Processing, 2nd Ed, Pearson Education,2004.

[3] Kaur M, Kaur J, Kaur J, “Survey of Contrast Enhancement Techniques based on Histogram Equalization”, IJASA, Vol. 2, No. 7, 2011, pp.137-141.

[4] Mukhopadhyay S, Chanda B, “A multiscale morphological approach. to local contrast enhancement”, Signal Processing, pp.685- 696,2000.

[5] Ruberto C. D, Dempster A. G, Khan S, Jarra B, “Segmentation of blood images using morphological operators”, International Conference on Pattern Recognition, Barcelona, Spain, IEEE, 2000,pp.397-400.

[6] Stojic T, Reljin I., “Local Contrast Enhancement of Digital Mammography by using Mathematical Morphology”, IEEE, 2005, pp.609-612.

[7] Sun K, Sang N, “Enhancement of Vascular Angiogram by Multiscale Morphology”, IEEE, 2007,pp.1311-1313.

[8] Wei Z, Hua Y., “X-ray Image Enhancement Based on Multiscale Morphology”, IEEE, 2007,pp.702-705.

[9] Wenzhong Y, “Mathematical Morphology Based Enhancemnet for Chromosome Images”, IEEE, 2009, pp.1-3.

[10] Wirth M, Fraschini M, Lyon J, “Contrast Enhancement of micro calcifications in mammograms using morphological enhancement and non-flat structuring elements”, Proceedings of the 17th IEEE Symposium on Computer-based Medical systems, 2004.

[11] Zadorozony A, Zhang H, “Contrast Enhancement using Morphological Scale Space”, Proceedings of the IEEE International Conference on Automation and Logistics, 2009,pp.804-807.

APPENDICES

Histogram equalization is a method in image processing of contrast adjustment using the image's histogram.

A histogram which is zero apart from a central area containing strong peaks is transformed by stretching the peaked area to fill the entire x-axis.

Histograms of an image before and after equalization.

Spread out the frequencies in an image(or equalising the image) is a simple way to improve dark or washed out images.

The formula for histogram equalisation is given below

$$\boxed{\begin{aligned} s_k &= T(r_k) \\ &= \sum_{j=1}^{k} p_r(r_j) \\ &= \sum_{j=1}^{k} \frac{n_j}{n} \end{aligned}}$$

A

MAJOR PROJECTREPORT

On

IOT CAR PARKING SYSTEM

Submitted by

1) Ms.S.HRITHIKA(17K81A04P2) 2) Ms.S.SUPRIYA(17K81A04P5)

3) Ms.Y.KRUTHI(17K81A04Q0)

In partial fulfillment for the award

of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Esteemed Guidance of

V. V RAMANA RAO, MTECH

Associate Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

AN UGC AUTONOMOUS INSTITUTE

Dhulapally, Secunderabad-500 100

NBA & NAAC A+ Accredited

Department Of Electronics & Communication Engineering

BONAFIDE CERTIFICATE

JUNE 2021

This is to certify that the project entitled "IOT CAR PARKING SYSTEM", is being submitted by **1.Ms.Sandhiri Hrithika(17K81A04P2), 2.Ms.Siddi Supriya(17K81A04P5), 3.Ms.Yerannagari Kruthi(17K81A04Q0)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

PROJECT GUIDE

Mr.V V RAMANA RAO
Associate Professor
Department of ECE

HEAD OF THE DEPARTMENT

Dr. B. HARI KRISHNA
Professor
Department of ECE

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

<u>INTERNSHIP CERTIFICATE</u>

THIS IS TO CERTIFY THAT S HRITHIKA WITH ROLL NO.17K81A04P2, S SUPRIYA WITH ROLL NO.17K81A04P5, Y KRUTHI WITH ROLL NO.17K81A04Q0, OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN’S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED “IOT CAR PARKING SYSTEM” AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email :contact@lasyainfotech.com,ov@lasyainfotech.com
Website :www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of **Bachelor of Technology** in Department of ELECTRONICS AND COMMUNICATION ENGINEERING, session: <2017 – 2021>, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled "IOT CAR PARKING SYSTEM" is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

1) Ms.Sandhiri.Hrithika(17K81A04P2)

2) Ms.Siddi.Supriya(17K81A04P5)

3) Ms.Yerannagari.Kruthi(17K81A04Q0)

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research andproviding us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr.B.HARI KRISHNA**, **Head &Professor**, Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in our memory for years to come.

We would like to express our gratitude to our dept. project coordinator **Dr.A.Anand, Professor** for his involvement with originality has triggered and nourished our intellectual maturity that will help us for along time to come.

Finally, we are highly grateful and obliged to our project guide **Mr.V.V RAMANA RAO, Associate Professor,**for his advice, supervision, and the vital contribution as and when required during this research work and for their support and co-operation that is difficult to express in words.

S HRITHIKA	S SUPRIYA	Y Kruthi
(17K81A04P2)	(17K81A04P5)	(17K81A04Q0)

ABSTRACT

Car parking is a major issues in modern congested cities of today. There simply are too many vehicles on the road and not enough parking space. This has led to the need for efficient parking management systems. Thus we demonstrate the use of IOT based parking management system that allows for efficient parking space utilization using IOT technology. To demonstrate the concept we use IR sensors for sensing parking slot occupancy along with a dc motors to simulate as gate opener motors. We now use a WIFI modem for internet connectivity and an AVR microcontroller for operating the system. We use IOT Gecko for online connectivity and IOT management GUI design. The system detects if parking slots are occupied using IR sensors. Also it uses IR technology to sense if a vehicle has arrived on gate for automated gate opening. The system reads the number of parking slots available and updates data with the cloud server to allow for checking parking slot availability online. This allows users to check for available parking spaces online from anywhere and avail hassle free parking. Thus the system solves the parking issue for cities and get users an efficient IOT based parking management system.

TABLEOFCONTENTS

1

LIST OF TABLES

LISTOF FIGURES

GLOSSARY OF COMMONLY USED TERMS

ARRIVAL EXPERIENCE:

The arrival experience is the first impression customers/guests get after they have arrived. The moment the customer parks is the first experience he/she has with your service level. That is why it is so important to create a pleasant arrival experience.

(TRAFFIC) CONGESTION:

Congestion is also described as overcrowding or blockage. Traffic congestion occurs when the space of the road is not enough anymore to handle the amount of vehicles in traffic.

FLOATING PARKING SPACES:

Floating parking spaces are parking spaces that aren't marked with lines on the ground, for example along the street. Cars can freely park in a line depending on the size of the car. As floating parking spaces cannot be detected by ground sensors, Cleverciti works with overhead sensors that view the object, not the space.

INTERNET OF THINGS (IOT):

IoT refers to a system of objects that are interconnected and together make it possible to collect and transfer data over a wireless network without human intervention.

MULTIMODAL TRANSPORT:

Multimodal transport is the transportation of goods under a single contract(or), but performed with at least two different modes of transport.

ON-STREET PARKING:

On-street parking is parking by taking up room on the actual street alongside the curb, as opposed to parking in a parking garage or a designated parking lot.

OVERHEAD PARKING SENSORS:

Overhead parking sensors, unlike ground parking sensors, are mounted above parking areas to more efficiently analyze the individual cars below. This type of sensor uses AI and IoT technology while being conveniently mounted on existing lampposts or buildings.

PARKING ASSETS:

Assets are items of property owned by a person or a company, regarded as having value and available to meet debts, commitments, or legacies. With parking assets are meant: any assets that can be used for parking, such as surface parking lots, parking garages, on -street parking, etc.

PARKING CAPACITY / OCCUPANCY:

The parking capacity or occupancy is the measurement of the number of taken parking spots, preferably measured in real-time, so that data on how many parking spots are still free is available at any given time.

PARKING DATA:

Parking data is any data related to parking, e.g. parking occupancy, length of stay, specific rush hours in a region, average parking price, parking revenue generated.

PARKING GUIDANCE:

Parking guidance is a system that guides the driver to the best available parking space. The most reliable way to do this is with local digital signage.

PARKING REQUEST FOR PROPOSAL (RFP):

A parking request for proposal (RFP) is a public business document that announces and provides details about a parking project, as well as solicits bids from suppliers who will help complete the project. RFPs are commonly used by governments. RFPs usually don’t go into as much detail as tenders.

PARKING REVENUE:

Parking revenue is the revenue that is derived from the rental of parking space against a certain price.

(SMART) PARKING TENDER:

A (smart) parking tender is a formal, structured procedure for generating competing offers from different potential smart parking solutions providers.

PAYMENT COMPLIANCE:

Payment compliance is the inclination of drivers to pay for their parking spot when they are requested to do so. Cities around the globe struggle to increase parking payment compl iance.

CHAPTER-1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT:

As quote says" why walk when drive " that is when journey on wheels started and which has created hard ongoing problem to park vehicles into parking slots.in urban cities as population is going on increasing the production of vehicles has also increased but parking space available has become congested to park N number of vehicles out of that people are finding it a great problem to search the available space during peak hours and festive times which consumes lot of energy and waste valuable time which is a major drawback, in order to replace the above issue.

Internet of things was first introduced in 1999 at auto ID center and first used by Kevin Aston. As evolving this latest burning technology, it promises to connect all our surrounding things to a network and communicating with each other with less human involvement. Still internet of things is in its beginning stage and there is no common architecture exists till today. There is lot of research and implementations are currently being going on in all the respective areas. Thus there is no guidelines or boundaries exists to define the definition of internet of things. So depending on the context, application the internet of things has different definitions.

Some smart applications which it has implementing currently such as on smart grids, smart lighting, smart energy, smart city, smart health etc. This is broadly classified into three categories such as sensing, processing and connectivity. Whereas sensing includes sensing the speed of vehicles and humans or any objects (accelerometer), sensing of temperature, pressure etc. [9]. And these can be processing by using some processors such as network processor, hybrid processor MCU/MPU etc. And the devices are connected by using some technologies called GPS, Wi-Fi, BT/BTLE, RFID etc.

1.2 OBJECTIVES OF THE STUDY:

The main objectives of this project Industrial Protection Systems are:

i. It is to reduces the risk of finding the parking slots in any parking area.

ii. These connected devises are called as smart devices. or smart objects.

iii. It consists of smart machines which communicating interacting with other machines, environment, objects etc.

iv. It incorporates to connect any two machines, machine to human and vice versa etc.

v. IOT technology grows in various fields of smart applications but we have not yet found boundary constraints of this technology.

vi. IoT, in general consists of inter-network of the devices and physical objects, number of objects can gather the data at remote locations and communicate to units managing, acquiring, organizing and analyzing the data in the processes and services.

vii. It provides a vision where things (wearable, watch, alarm clock, home devices, surrounding objects with) become smart and behave alive through sensing, computing and communicating by embedded small devices which interact with remote objects or persons through connectivity.

viii. It eliminates the unnecessary travelling of vehicles across the filled parking slots in a city.

1.2.1 Specific Objectives

i. The devices could be tracked, controlled or monitored using remote computers connected through Internet.

ii. IoT extends the use of Internet providing the communication, and thus inter-network of the devices and physical objects, or ‘Things’.

iii. The two prominent words in IoT are “internet” and “things”. Internet means a vast global network of connected servers, computers, tablets and mobiles using the internationally used protocols and connecting systems.

iv. Internet enables sending, receiving, or communicating of information. To interface LM35 temperature sensor, GSM SIM800l Modem, 16X2 LCD, and process control devices (through relays) with the Arduino UNO.

v. These factors gave rise to the amalgamation of both technologies thus leading to the formation of a new technology called Cloud of Things (CoT).

vi. Due to high scalability in cloud any number of nodes could be added or removed from the IoT system on a real time basis.

vii. In simple terms IoT can be explained in form of an equation stating.

1.3 SCOPE OF THE STUDY:

The scope for this project is to adopt this automatic Smart Parking System (SPS) so that availability of spaces could be displayed on a smart phone Application or even to satellite navigation device so that drivers will always aware of whether there are free spaces are not. And also enhance to send some notifications to users smart phone when vehicle enters to particular shopping places and some streets in a city etc.The smart car parking system is expected to be significantly influenced by the arrival of automated vehicles(AVs).several cities around the world are already beginning to trial self-parking vehicles,specialized AV parking iots, and robotic parking valets. Here, there is no need of man. The circuit itself checks the presence of vehicle and automatically closes the gate by rising an alarm. Once we switch on the circuit, it automatically performs all these actions without man handling. It is the most advantage of this project. For this reason, in future, this project may be used in railways and also in apartments, military, etc. Future car collision-avoidance systems may be smart phone based. You could stick your cell phone on the dashboard, and it would use to provide the feedback needed by the system.

1.4 Significance of the project

i. Time saved,43% reduction in time spent looking for a parking space and reduction in traffic volumes in build-up areas.

ii. Decreased searching for spaces can reduce accidents by ensuring drivers maintain their attention rather than browsing for spaces for making rash manoeuvres.

1.5 MATERIAL REQUIREMENT:

1.5.1 HARDWARE REQUIREMENT:

- Arduino UNO
- LCD
- Power Supply
- IR Sensor
- Servo motor

1.5.2 SOFTWARE REQUIREMENT:

- Proteus software
- Arduino software

1.6 PROCUMENT OF EQUIPMENT:

About hardware pursuing…………………

Assembling it…………….

Connecting to Arduino…………

Coding is done in ide…………

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH AREA:

Parking scheduling is converted into an off-line problem. The offline problem is described as a linear problem. The linear problem was solved using an algorithm. Finally, experimental simulations were done. However, this paper does not deal with the guiding of vehicles. Paper proposes a solution for parking lot based on wireless sensor network and radio-frequency identification (RFID). The paper however does not deal with a large-scale parking lot. Paper has proposed a parking system based on ZigBee network. Here, a web service is used to collect information about the parking space. Our approach is based on 8051 type microcontroller that is Arduino microcontroller, Arduino runs with Arduino IDE application that should be installed in system. We do simple embedded C code in microcontroller and directly put it into the Arduino system. We do simple embedded C code in microcontroller and directly put it into the Arduino microcontroller. Hence, it works according to code system keeps track of number of cars entered in parking building. The counting will be display using liquid crystal display board.

This project introduces the concept of using IoT and Cloud based technology in car parking services in cities. A high-level view of the proposed system is outlined. Our solution makes the ancient parking system smarter by leveraging the power of IoT and embedding it with the latest innovation of electronic sensors & computers. An IoT-based intelligent car parking system is described. A number of software solutions, including Python, PHP web gateway with MySQL database, Cloud based storage and mobile applications, are proposed to provide pleasant parking experience to mobile users. Also, Data generated by the sensors, Image detection cameras and mobile application will be used to gain insights by storing it in cloud foundry and applying Big Data analytics using Hadoop.

2.2 REVIEW ON RELATED LITERATURE:

Car parking issue is a major contributor and has been yet a noteworthy issue with limited parking spots in urban communities. In this paper we present an IoT based smart parking system for large parking lot that can be used to efficiently manage the parking system by providing information on the nearest parking slot available through the mobile application and thereby reducing the congestion of parking seekers. In order to efficiently manage the parking system, a successful cloud-based smart parking system solution using the Internet of Things technology is been developed to guide the user to the nearest parking spot available.

TRAFFIC a direct or an indirect outcome of many minor issues has been creating havoc for an individual especially in urban areas. This paper proposes an architecture that introduces an efficient and eminent; rather a smart way to resolve the small module that actually counts i.e. Parking Area. We design a system so as to eliminate the time wastage and irrelevant frustration faced by the drivers based on IoT for real time monitoring of the empty slots for car parking from anywhere using a webpage or a mobile app; IoT the emerging research domain is the heart of the proposed system. Traditional parking systems are very dependent on manual labors and possess multiple problems. A long time to find available parking slot, a big amount of cost to pay parking staffs, and unavailability of parking data are some of the most common problems in this parking system. In this paper, we propose an Internet of Things (IoT) based smart parking system that has three main components, which are an application for parking management, application for parking users, and IoT platform. The IoT platform consists of two different granularities, which are in each parking slot and in each parking lot. IoT hardware in parking slot will send information regarding the slot status, and IoT hardware in parking lot will send information regarding any car that goes in or out. To reduce the number of sensors, crowd sensing from parking users and parking staffs is used. This proposed design is implemented in the case study of X University. We conduct an evaluation for the implemented system and receive multiple feedbacks from parking system staffs and parking users. Using the System Usability scale, we receive a score of 91 from parking system staffs and 80.14 from parking users. This evaluation result shows that the proposed IoT-based parking system is helpful and easy to use for parking staffs and users.

In the current decade we are facing parking problem with the advancement in the technology. The population in urban cities is dense due to which lots of vehicles are running on road leads to parking problem, traffic problem. World is facing new challenge of Car parking. It is observed that one million of vehicles consumes oil on daily basis. In this paper, an automatic real-time system for automated car parking is proposed. This system is implemented with the help of internet of things (IOTs). IOT generally exchanges information or data between the two physical devices. Arduino Uno is a microcontroller used in proposed system. The main use of Arduino in the proposed system is to provide platform to communicate digital devices and interactive objects that can sense and control physical devices. The proposed system is implemented with the help of Arduino Uno board for car parking and Node MCU to connect parking area with web or internet. The proposed system incorporated with an infrared sensor in each slot for getting information about vacancy position of parking slot. The user book parking slot well in advance, all the necessary information is available on server. Every user has exclusive username and password. In case any misuse happened then the system will alert the responsible person.

Traditional parking systems are very dependent on manual labors and possess multiple problems. A long time to find available parking slot, a big amount of cost to pay parking staffs, and unavailability of parking data are some of the most common problems in this parking system. In this paper, we propose an Internet of Things (IoT) based smart parking system that has three main components, which are an application for parking management, application for parking users, and IoT platform. The IoT platform consist of two different granularities, which are in each parking slot and in each parking lot. IoT hardware in parking slot will send information regarding the slot status, and IoT hardware in parking lot will send information regarding any car that goes in or out. To reduce the number of sensors, crowdsensing from parking users and parking staffs is used. This proposed design is implemented in the case study of X University. We conduct an evaluation for the implemented system and receive multiple feedbacks from parking system staffs and parking users. Using the System Usability scale, we receive a score of 91 from parking system staffs and 80.14 from parking users. This evaluation result show that the proposed IoT-based parking system is helpful and easy to use for parking staffs and users.

Since all people used vehicles as a necessity, the growing number of vehicles is increasing day by day. With the increasing number of vehicles on the road, the requirement of parking lot will also be restricted. Thus, it become a problem for the people to find a free parking space in the busy building especially in shopping complex. Internet of Things (IoT) is an interesting technology that can be applied in this paper because this technology has a capability to transfer the data over a network without requiring human intervention in almost all applications in today's society. In this paper, a user-friendly mobile application, named as Android-based Car Parking Monitoring System (ACPMS) is built to aid in locating a particular parking place. ACPMS can provide a user with the ability to check vacant parking space and locate the nearest parking lot. ACPMS obtained the parking location from the current user's position with the sensor located in the shopping complex's parking lot. ACPMS is testing in a realistic environment for the purpose of movement detection and location service to notify user using the mobile application. By considering seven test case scenarios, the combination of the ACPMS mobile application with a parking prototype kit shows that the proposed work able to solve the parking problem.

Nowadays, technology has become more reliable that allowed human interaction to decrease while the machines have started to take over in many fields of interest. The Internet of things (IoT) represents the idea of machines ability to control entire systems without any human interactions. This paper proposes an IoT system that takes a picture of parked cars in a garage, extract the plate number and the location of the car, and places this information in a database. At the garage entrance, a terminal linked to the system allows the user to locate the parked car using the license plate no. The system is tested for reading license plates several times, and is shown to perform with a low probability of error. The proposed IoT parking system is an integrated system combining the RPi operating system Motion Eye, with MATLAB segmentation code for character recognition, database tools, as well as web services all in order to make the user interface as friendly as possible.

A simple and easy task such as parking is thought as a tedious and time-consuming process due to mismanagement of parking system. Current parking systems involve huge manpower for management and requires user to search for parking space floor by floor. Such conventional systems utilize more power, along with user's valuable time. This paper presents a Smart Parking Energy Management solution for a structured environment such as a multi-storied office parking area. The system proposes implementation of state-of-the-art Internet of Things (IoT) technology to mold with advanced Honeywell sensors and controllers to obtain a systematic parking system for users. Unoccupied vehicle parking spaces are indicated using lamps and users are guided to an empty parking space, thus eliminating need for searching for a parking space. The occupied parking spaces are virtually stored to the cloud to be accessed by central system and direct the upcoming cars to empty spaces. The automatically controlled light illuminance helps reduce energy usage, along with lighting up the parking space to the user whilst in the parking space. The entire system being fully automatic leads to reduced manpower involved and improves illuminance aesthetics of the parking area. This paper aims at improving user's time value and convenience in a parking system.

2.3 CONCLUSION ON REVIEWS:

In recent times the concept of smart cities have gained grate popularity. Thanks to the evolution of Internet of things the idea of smart city now seems to be achievable. Consistent efforts are being made in the field of IoT in order to maximize the productivity and reliability of urban infrastructure.

Problems such as, traffic congestion, limited car parking facilities and road safety are being addressed by IoT. In this paper, we present an IoT based cloud integrated smart parking system. The proposed Smart Parking system consists of an on-site deployment of an IoT module that is used to monitor and signalize the state of availability of each single parking space. A mobile application is also provided that allows an end user to check the availability of parking space and book a parking slot accordingly. The paper also describes a high-level view of the system architecture. Towards the end, the paper discusses the working of the system in form of a use case that proves the correctness of the proposed model.

One of the main problems in many big and crowded cities is finding parking spaces for vehicles. With IoT technology and mobile applications, in this paper, we propose a design and development of a real smart parking system that can provide more than just information about vacant spaces but also help user to locate the space where the vehicle can be parked in order to reduce traffics in the parking area. Moreover, we use computer vision to detect vehicle plate number in order to monitor the vehicles in the parking area for enhancing security and also to help user find his/her car when he/she forgets where the car is parked. In our system, we also design the payment process using mobile payment in order to reduce time and remove bottleneck of the payment process at the entry/exit gate of the parking area.

Internet of Things (IoT) plays an indispensable role in bridging the gap between all the day to day things to the networking system, and creates an ease to access all the un-internet things from any distant location. Adaption to the growth in the recent trends is inexorable for the people. With all the advancement in the technology, finding a particular place to park our automobile becomes an exasperating issue. In our work we have designed a Smart Car Parking System (SCPS) with the help of infrared sensor and a database based on application of Iot, which permits the driver to find the proximate parking slot, and gives the number of free places available in that respective parking zone. This ideology mainly focuses on diminishing the time involved in discovering the parking space and also it decreases the unwanted travelling, through filled parking slots in a parking arena. This will in turn reduce the consumption of fuel, which would reduce carbon footprints in our environment. Thus, this will pave way for an eco-friendly surrounding.

CHAPTER-3

PROJECT DESIGN

3.1 OVERVIEW OF THE DESIGN:

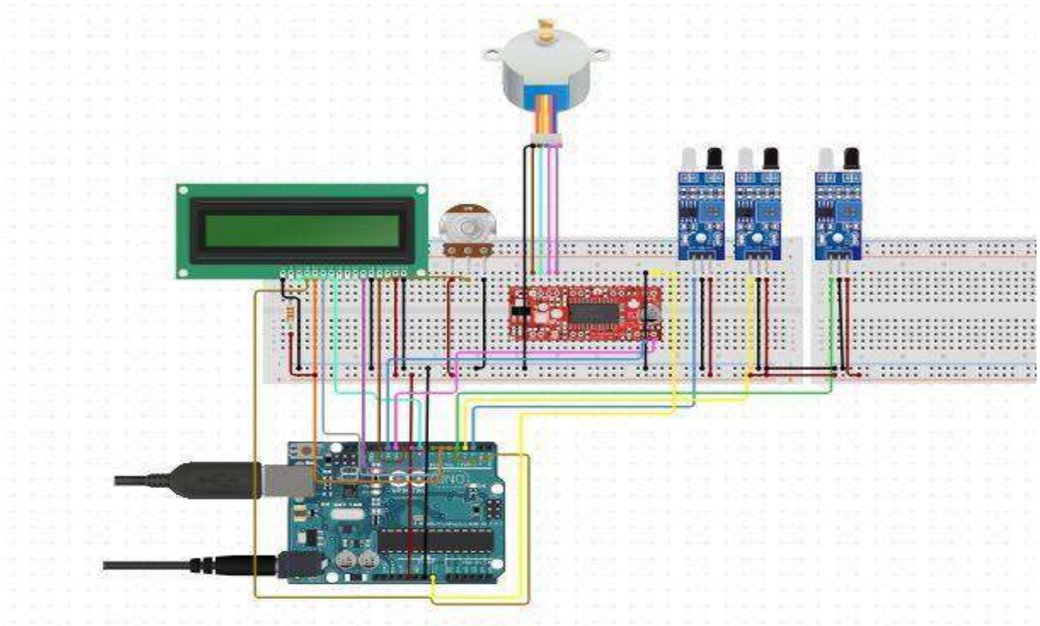


Fig.3.1 Overview of the design

3.2 EQUIPMENT ANALYSIS:

3.2.1 ARDUINO:

3.2.1.1 INTRODUCTION:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

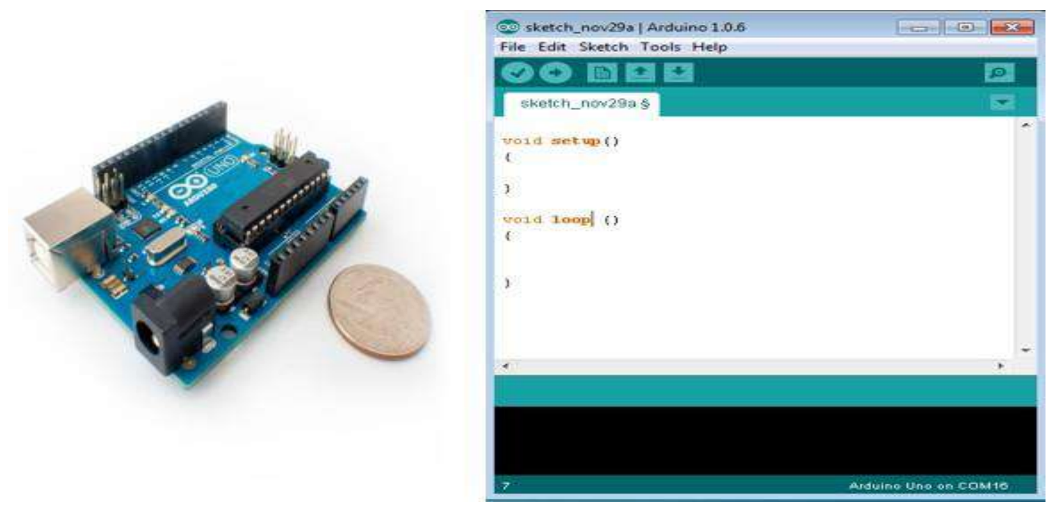


Fig 3.2.1.1 Arduino Uno

3.2.1.2 BOARD TYPES

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs,

and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.
Here is a list of different Arduino boards available.

Board Name	**Operating Volt**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header

Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.2.1.2.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	**Operating Volt**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB

Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 3.2.1.2.2 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 3.2.1.2.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table 3.2.1.2.4 Arduino boards based on AT91SAM3X8E microcontroller

3.2.1.3 BOARD DESCRIPTION:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

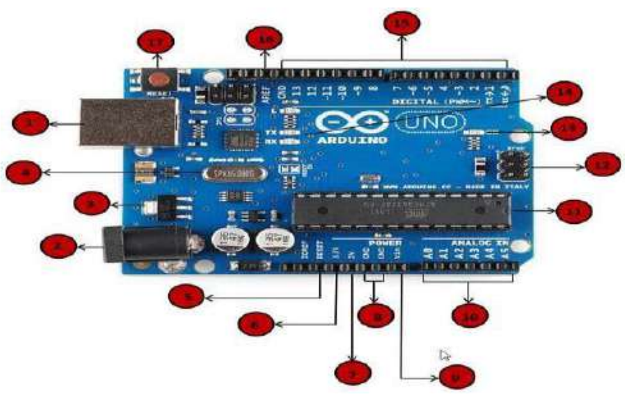


Fig.3.2.1.2 Board Description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).

3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC

	your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

3.2.1.4 ARDUINO FAMILY

Arduino makes several different boards, each with different capabilities. In addition, part of being open-source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

Fig.3.2.1.3: Arduino Family

3.2.1.5 SHIELDS:

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

Fig.3.2.1.4: Arduino Shields

3.2.1.6 PIN DESCRIPTION OF ATMEGA328:

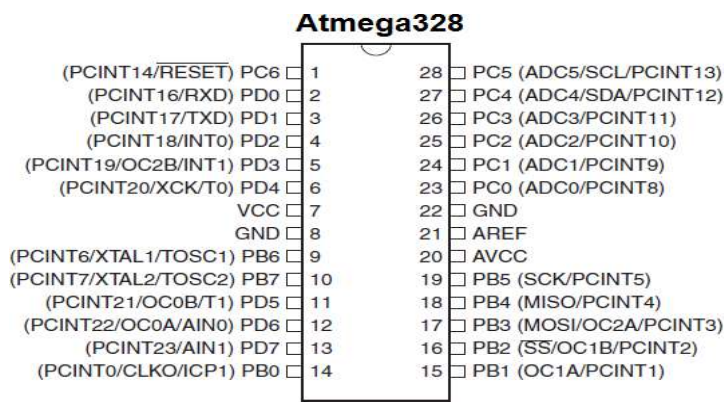


Fig.3.2.1.5 Pin description of ATMEGA328

3.2.1.7 APPLICATIONS:

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

3.2.2 16*2 LCD SCREEN DISPLAY

3.2.2.1 INTRODUCTION

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

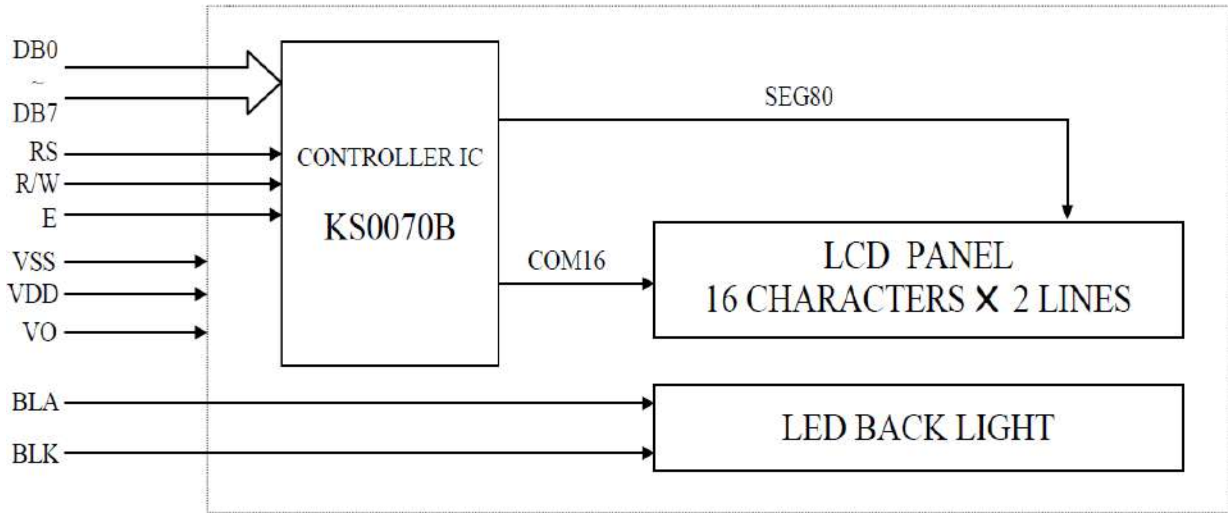


fig.3.2.2.1 Lcd block diagram

3.2.2.2 DISPLAY DATA RAM:

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.

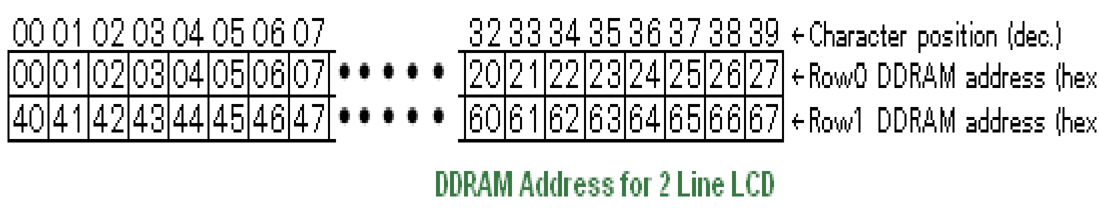


Fig.3.2.2.2 DDRAM addresses of 2 Line LCD

3.2.2.3 CHARACTER GENERATOR ROM:

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM.

3.2.2.4 BUS FLAG:

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e. BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

3.2.2.5 INSTRUCTION REGISTER (IR) AND DATA REGISTER (DR):

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g. LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD. Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

3.2.2.6 16 X 2 ALPHANUMERIC LCD MODULE FEATURES:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing

- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

3.2.2.7 CIRCUIT DESCRIPTION:

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

3.2.3 MOTOR

3.2.3.1 INTRODUCTION:

Motor is a device that creates motion, not an engine; it usually refers to either an electrical motor or an internal combustion engine.

It may also refer to:

- Electric motor, a machine that converts electricity into a mechanical motion
    - AC motor, an electric motor that is driven by alternating current
        - Synchronous motor, an alternating current motor distinguished by a rotor spinning with coils passing magnets at the same rate as the alternating current and resulting magnetic field which drives it
        - Induction motor, also called a squirrel-cage motor, a type of asynchronous alternating current motor where power is supplied to the rotating device by means of electromagnetic induction
    - DC motor, an electric motor that runs on direct current electricity
        - Brushed DC electric motor, an internally commutated electric motor designed to be run from a direct current power source
        - Brushless DC motor, a synchronous electric motor which is powered by direct current electricity and has an electronically controlled commutation system, instead of a mechanical commutation system based on brushes
    - Electrostatic motor, a type of electric motor based on the attraction and repulsion of electric charge
    - Servo motor, an electric motor that operates a servo, commonly used in robotics
    - Internal fan-cooled electric motor, an electric motor that is self-cooled by a fan, typically used for motors with a high energy density

Industrial motors come in a variety of basic types. These variations are suitable for many different applications. Naturally, some types of motors are more suited for certain applications than other motor types are. This document will hopefully give some guidance in selecting these motors.

3.2.3.2 SERVO MOTOR

A **servo motor** is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor which runs through a **servo mechanism**. If motor is powered by a DC power supply, then it is called DC servo motor, and if it is AC-powered motor then it is called AC servo motor. For this tutorial, we will be discussing only about the DC servo motor working**.** Apart from these major classifications, there are many other types of servo motors based on the type of gear arrangement and operating characteristics. A servo motor usually comes with a gear arrangement that allows us to get a very high torque servo motor in small and lightweight packages. Due to these features, they are being used in many applications like toy car, RC helicopters and planes, Robotics, etc.

Servo motors are rated in kg/cm (kilogram per centimetre) most hobby servo motors are rated at 3kg/cm or 6kg/cm or 12kg/cm. This kg/cm tells you how much weight your servo motor can lift at a particular distance. For example: A 6kg/cm Servo motor should be able to lift 6kg if the load is suspended 1cm away from the motors shaft, the greater the distance the lesser the weight carrying capacity. The position of a servo motor is decided by electrical pulse and its circuitry is placed beside the motor.

3.2.3.3 Servo Motor Working Mechanism

It consists of three parts:

1. Controlled device
2. Output sensor
3. Feedback system

It is a closed-loop system where it uses a positive feedback system to control motion and the final position of the shaft. Here the device is controlled by a feedback signal generated by comparing output signal and reference input signal.

Here reference input signal is compared to the reference output signal and the third signal is produced by the feedback system. And this third signal acts as an input signal to the control the device. This signal is present as long as the feedback signal is generated or there is a difference between the reference input signal and reference output signal. So the main task of servomechanism is to maintain the output of a system at the desired value at presence of noises.

3.2.3.4 Servo Motor Working Principle

A servo consists of a Motor (DC or AC), a potentiometer, gear assembly, and a controlling circuit. First of all, we use gear assembly to reduce RPM and to increase torque of the motor. Say at initial position of servo motor shaft, the position of the potentiometer knob is such that there is no electrical signal generated at the output port of the potentiometer. Now an electrical signal is given to another input terminal of the error detector amplifier. Now the difference between these two signals, one comes from the potentiometer and another comes from other sources, will be processed in a feedback mechanism and output will be provided in terms of error signal. This error signal acts as the input for motor and motor starts rotating.

Now motor shaft is connected with the potentiometer and as the motor rotates so the potentiometer and it will generate a signal. So as the potentiometer's angular position changes, its output feedback signal changes. After sometime the position of potentiometer reaches at a position that the output of potentiometer is same as external signal provided. At this condition, there will be no output signal from the amplifier to the motor input as there is no difference between external applied signal and the signal generated at potentiometer, and in this situation motor stops rotating.

3.2.3.5 Interfacing Servo Motors with Microcontrollers:

Interfacing hobby Servo motors like s90 servo motor with MCU is very easy. **Servos have three wires coming out of them**. Out of which two will be used for Supply (positive and negative) and one will be used for the signal that is to be sent from the MCU. An **MG995 Metal Gear Servo Motor** which is most commonly used for RC cars humanoid bots etc. The picture of MG995 is shown below:

The colour coding of your servo motor might differ hence check for your respective datasheet.

All servo motors work directly with your +5V supply rails but we have to be careful on the amount of current the motor would consume if you are planning to use more than two servo motors a proper servo shield should be designed.

3.2.3.6 Controlling Servo Motor:

All motors have three wires coming out of them. Out of which two will be used for Supply (positive and negative) and one will be used for the signal that is to be sent from the MCU.

Servo motor is controlled by PWM (Pulse with Modulation) which is provided by the control wires. There is a minimum pulse, a maximum pulse and a repetition rate. Servo motor can turn 90 degree from either direction form its neutral position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position, such as if pulse is shorter than 1.5ms shaft moves to 0° and if it is longer than 1.5ms than it will turn the servo to 180°.

Servo motor works on **PWM (Pulse width modulation)** principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically servo motor is made up of **DC motor which is controlled by a variable resistor (potentiometer) and some gears**. High speed force of DC motor is converted into torque by Gears. We know that WORK= FORCE X DISTANCE, in DC motor Force is less and distance (speed) is high and in Servo, force is High and distance is less. The potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on the required angle.

Servo motor can be rotated from 0 to 180 degrees, but it can go up to 210 degrees, depending on the manufacturing. This degree of rotation can be controlled by applying the **Electrical Pulse** of proper width, to its Control pin. Servo checks the pulse in every 20 milliseconds. The pulse of 1 ms (1 millisecond) width can rotate the servo to 0 degrees, 1.5ms can rotate to 90 degrees (neutral position) and 2 ms pulse can rotate it to 180 degree.

All servo motors work directly with your +5V supply rails but we have to be careful about the amount of current the motor would consume if you are planning to use more than two servo motors a proper servo shield should be designed.

3.2.3.7 Applications for servo motors

- Rotating stages for machine tools or inspection systems
- Control of presses
- Control of bonding machines
- Control of plastic moulding machines
- Control of amusement machines
- Radio control and other hobby applications
- Coating or vapor deposition machines
- Spin cleaning machines for LCDs and semiconductors

3.2.4 SOFTWARE EXPLANATION:

- Arduino software
- Proteus simulation

3.2.4.1 ARDUINO SOFTWARE:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments.

All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

3.2.4.2 WHY ARDUINO?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments.

Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step-by-step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phi gets, MIT's Handy board, and many others offer similar functionality.

All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.

- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows. Wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers c an make their own version.
Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.
- Getting Started with Arduino and Genuine products: -

Install the Arduino Software (IDE) on Windows PCs-

This document explains how to install the Arduino Software (IDE) on Windows machines.

✓ Download the Arduino Software (IDE)

✓ Proceed with board specific instructions.

3.2.4.3 HOW TO DOWNLOAD THE ARDUINO SOFTWARE (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

3.2.4.4 INSTALLATION:

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.3.2.4.4.1 Standard USB Cable (A plug to B plug)

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig.3.2.4.4.2 Standard USB Cable (an A to Mini-B plug)

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

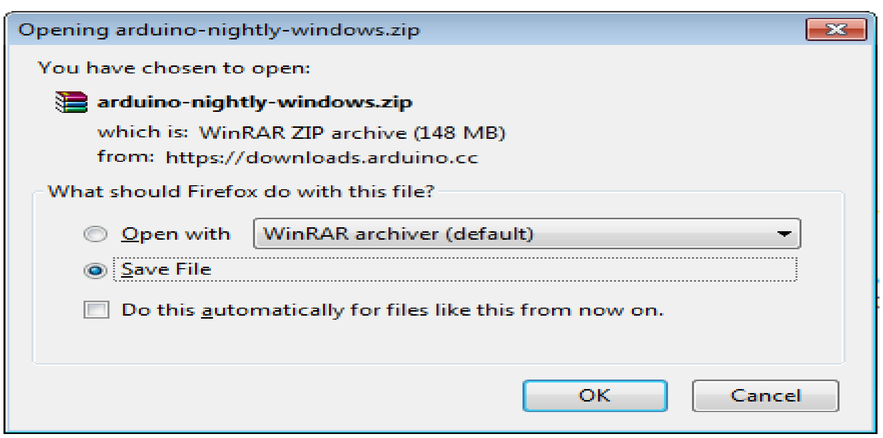


Fig.3.2.4.4.3 Download Arduino IDE Software

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

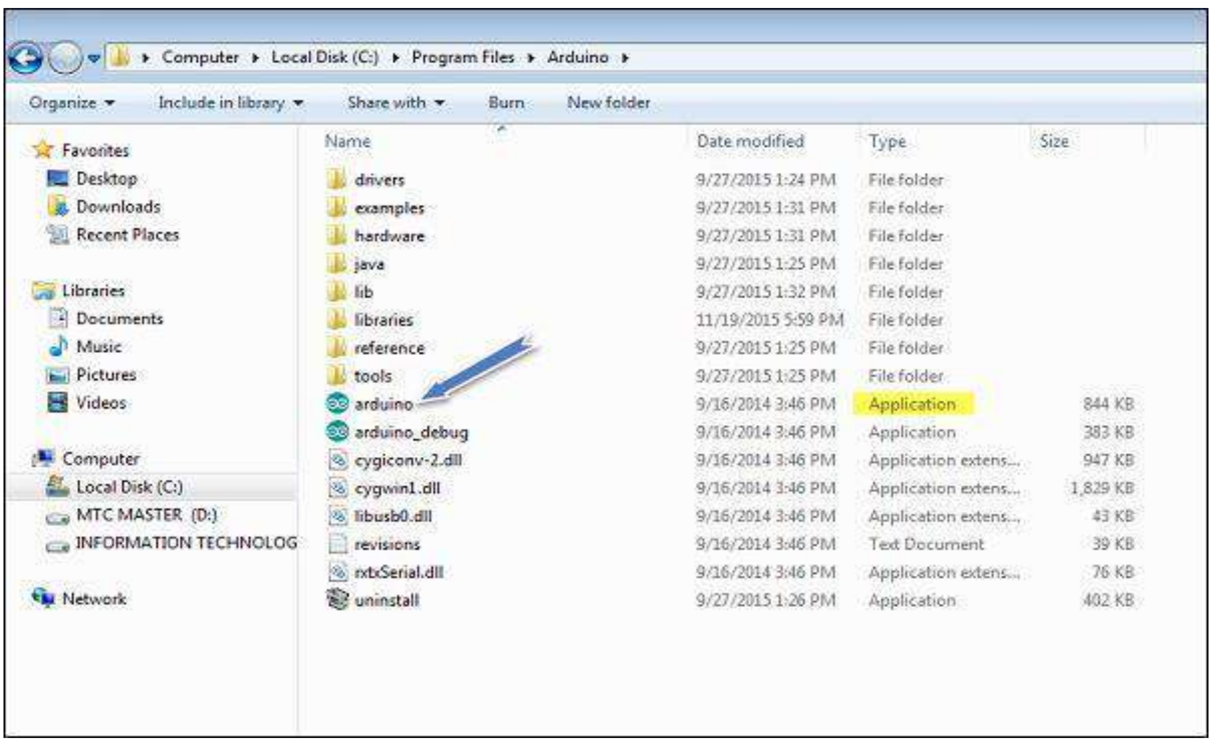


Fig.3.2.4.4.4 Launch Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**

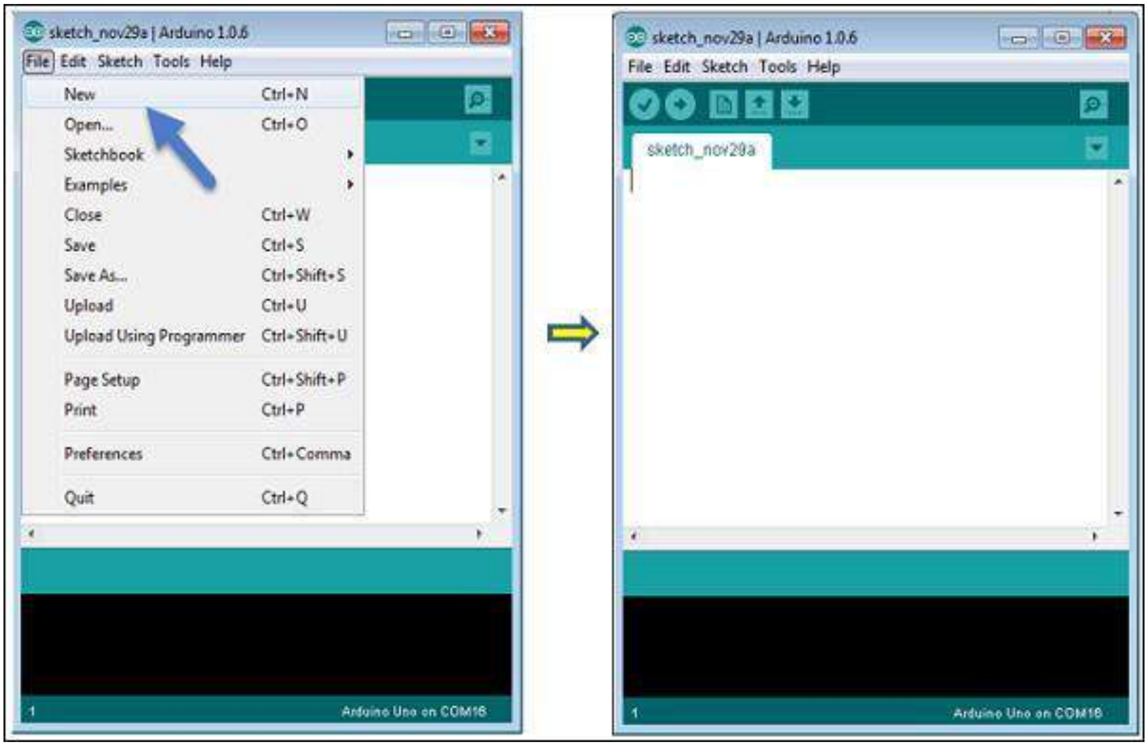


Fig.3.2.4.4.5 Open your first project.

To open an existing project example, select File → Example → Basics → Blink.

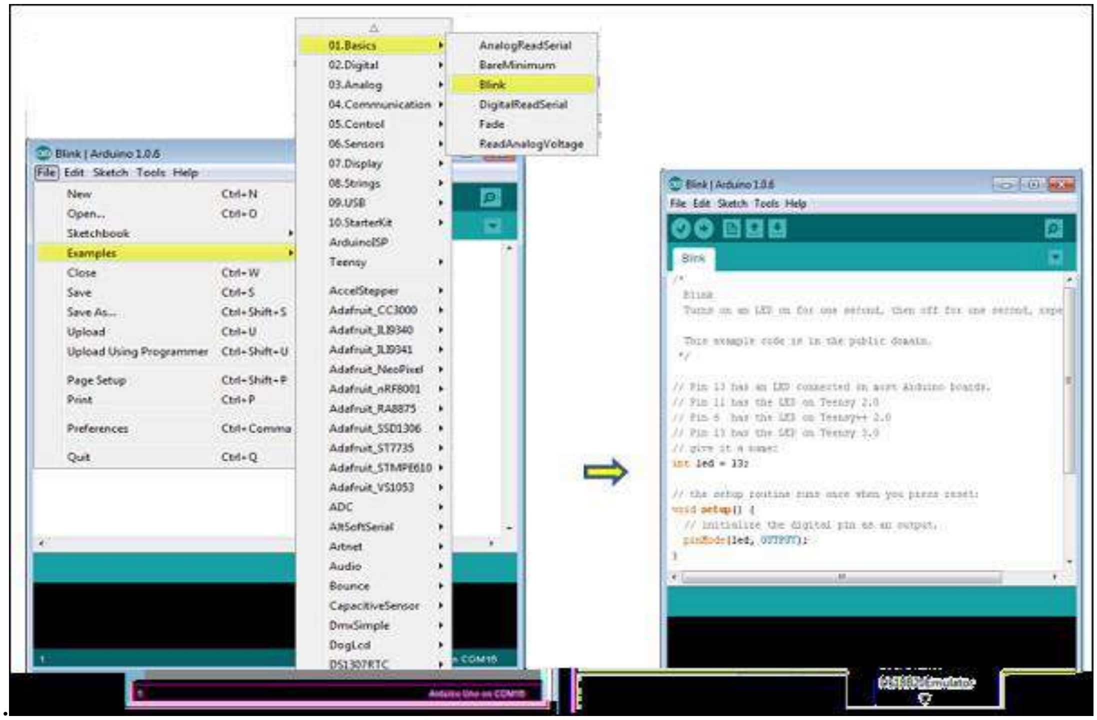


Fig.3.2.4.4.6 Open your first program.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

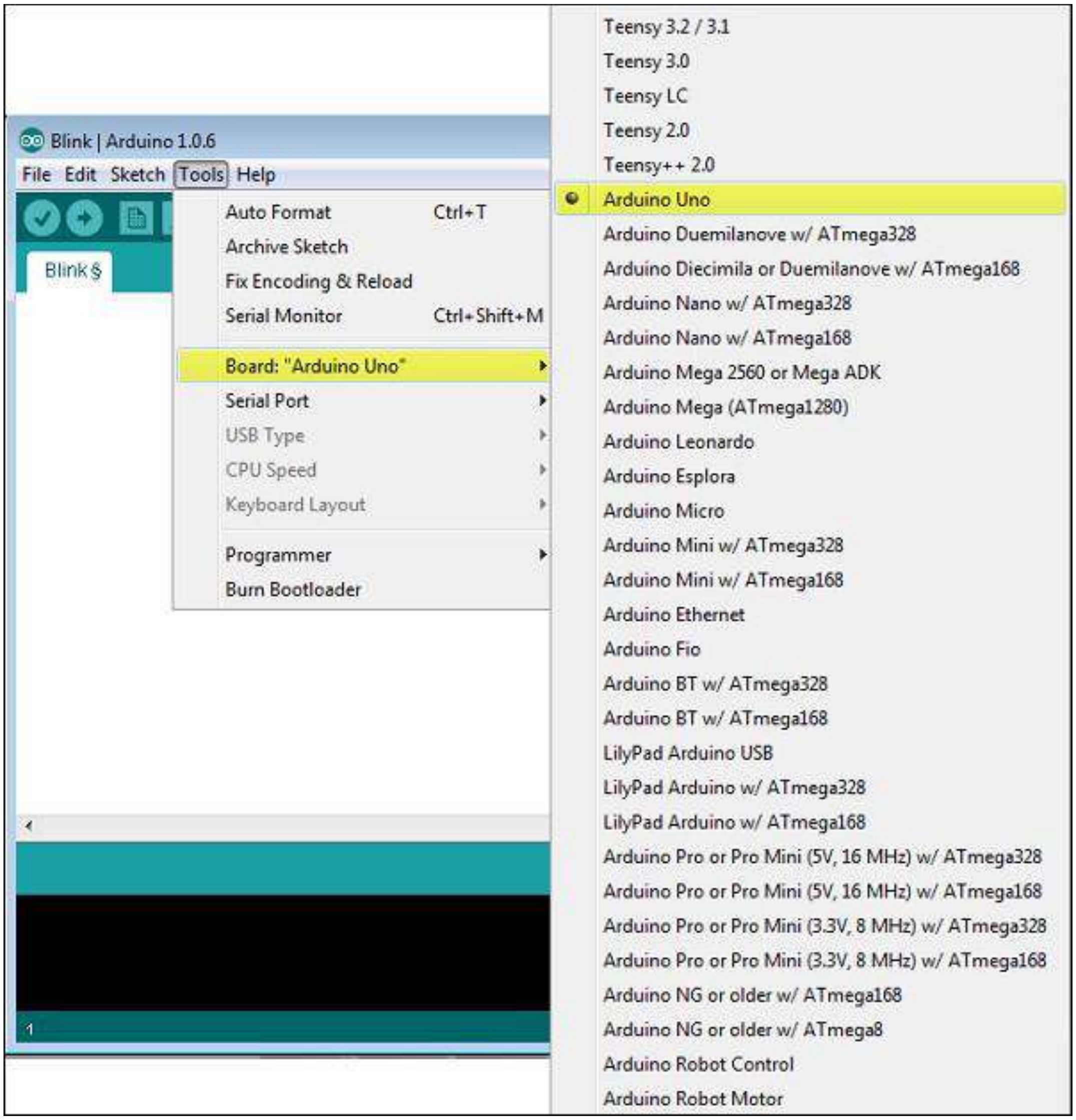


Fig.3.2.4.4.7 Select your Arduino board.

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 – Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

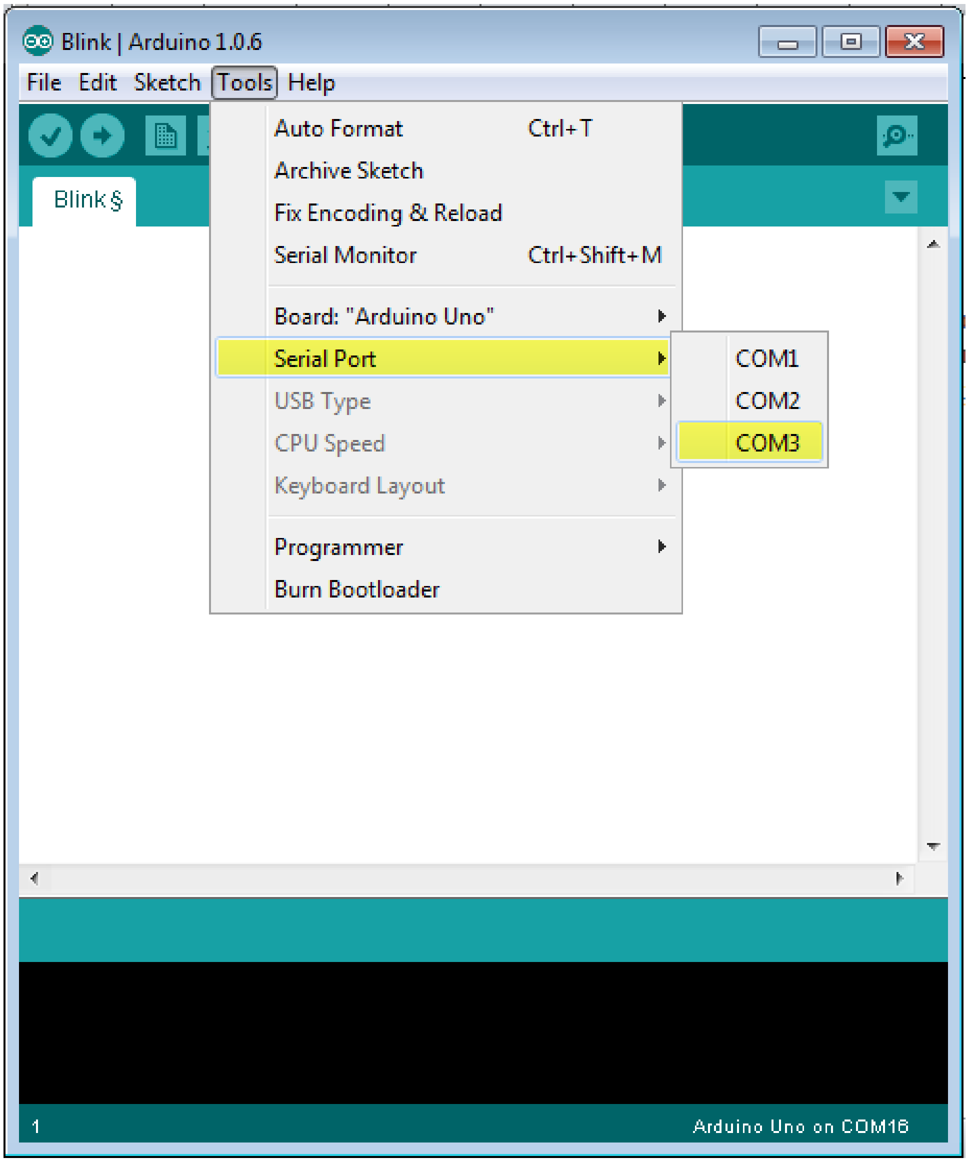


Fig.3.2.4.4.8 Select your serial port.

Step 8 – Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

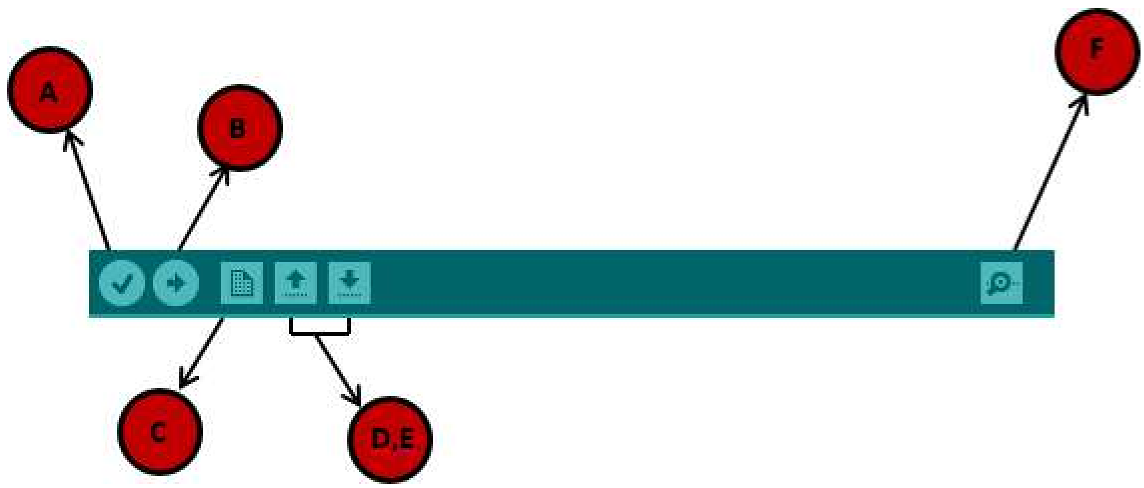


Fig.3.2.4.4.9 Upload the program to your board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from board & send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

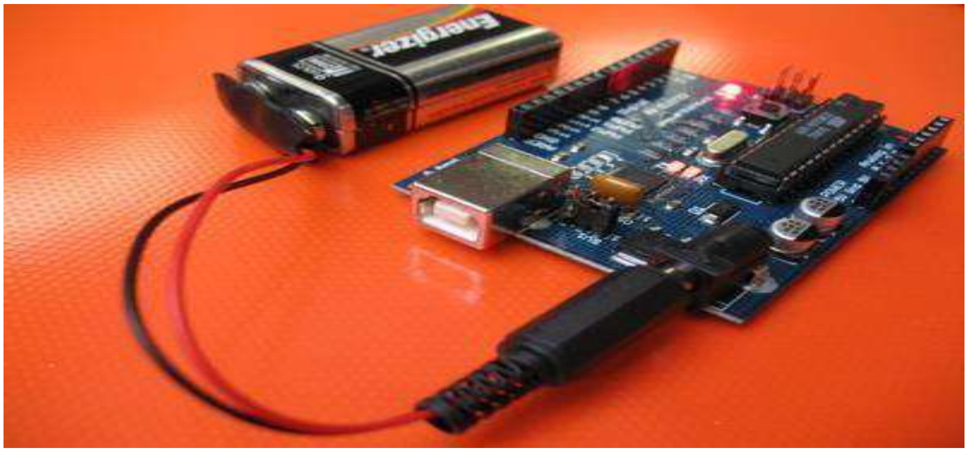


Fig.3.2.4.4.10 Connecting a Battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called “sketches”, but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

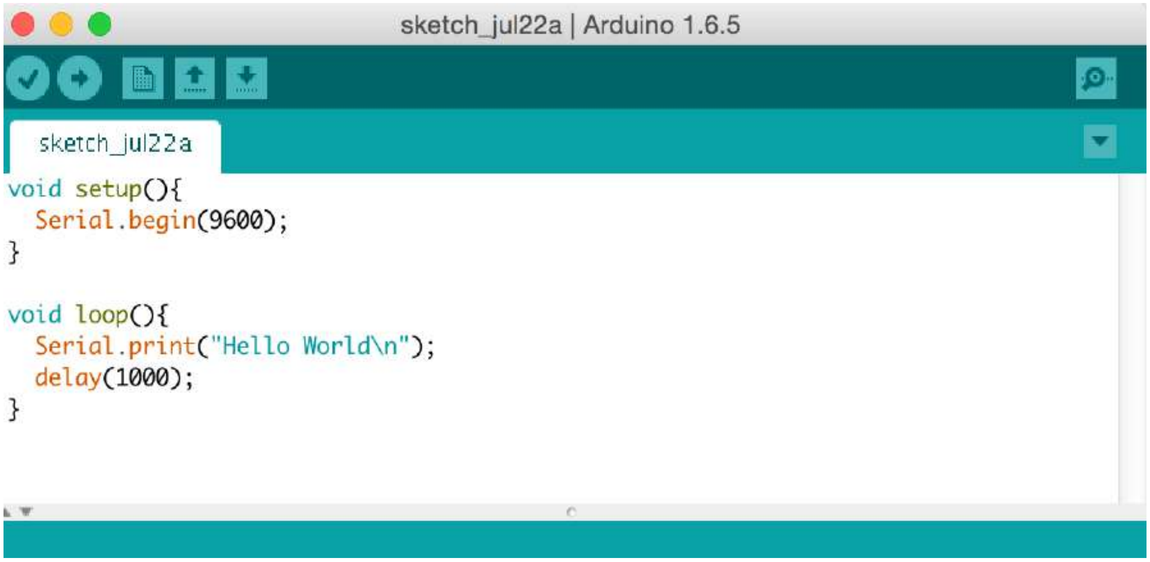


Fig.3.2.4.4.11 Example Program

3.2.5 PROTEUS:

PROTEUS:

Proteus is a simulation and design software tool developed by Lab center Electronics for Electrical and. It also possesses 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design:

Step 1: Open ISIS software and select new design in File menu

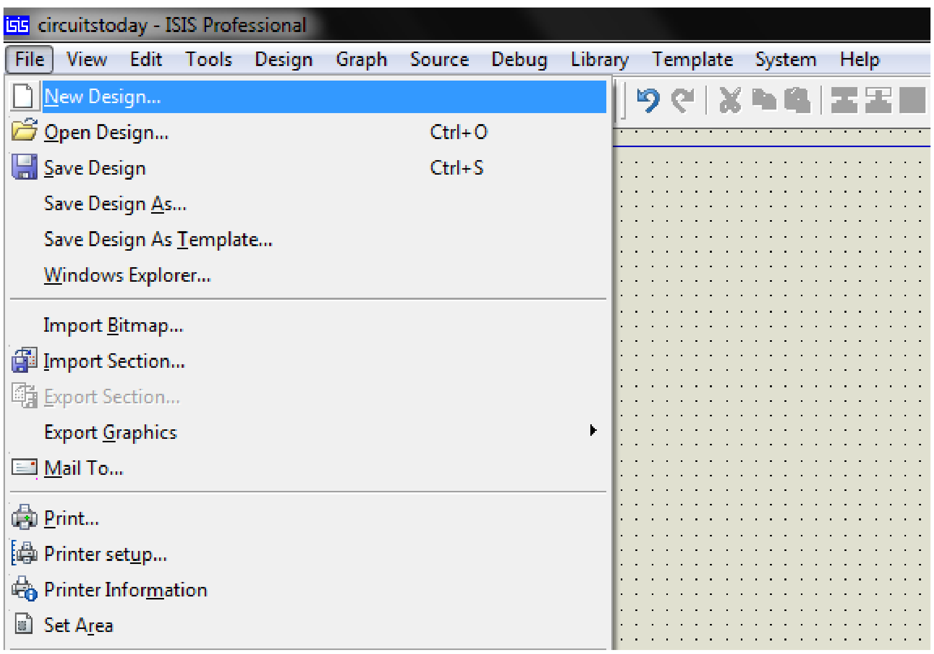


Fig.3.2.5.1Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

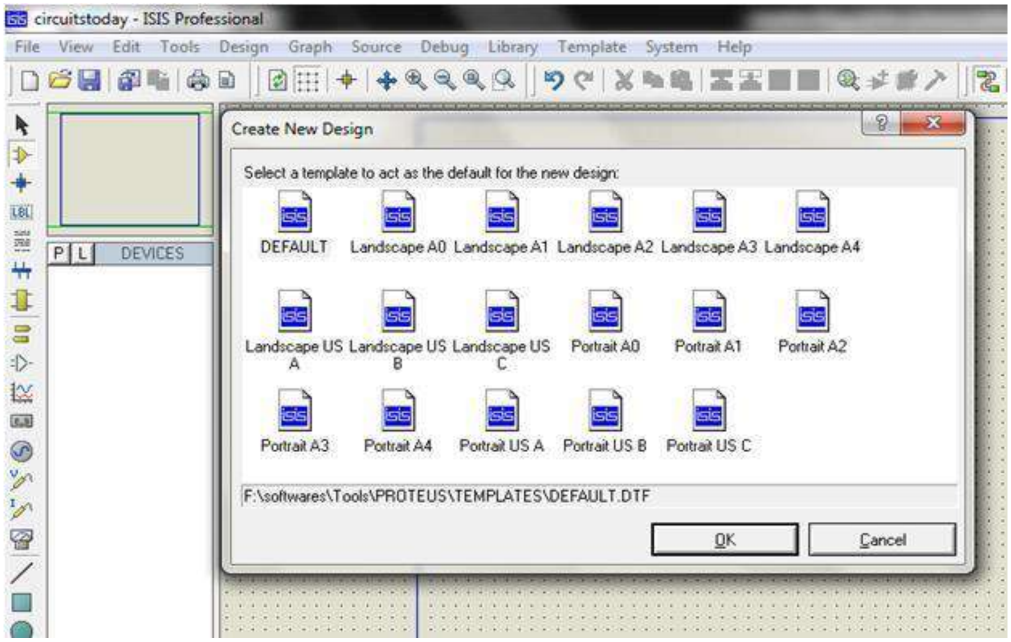


Fig. 3.2.5.2Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

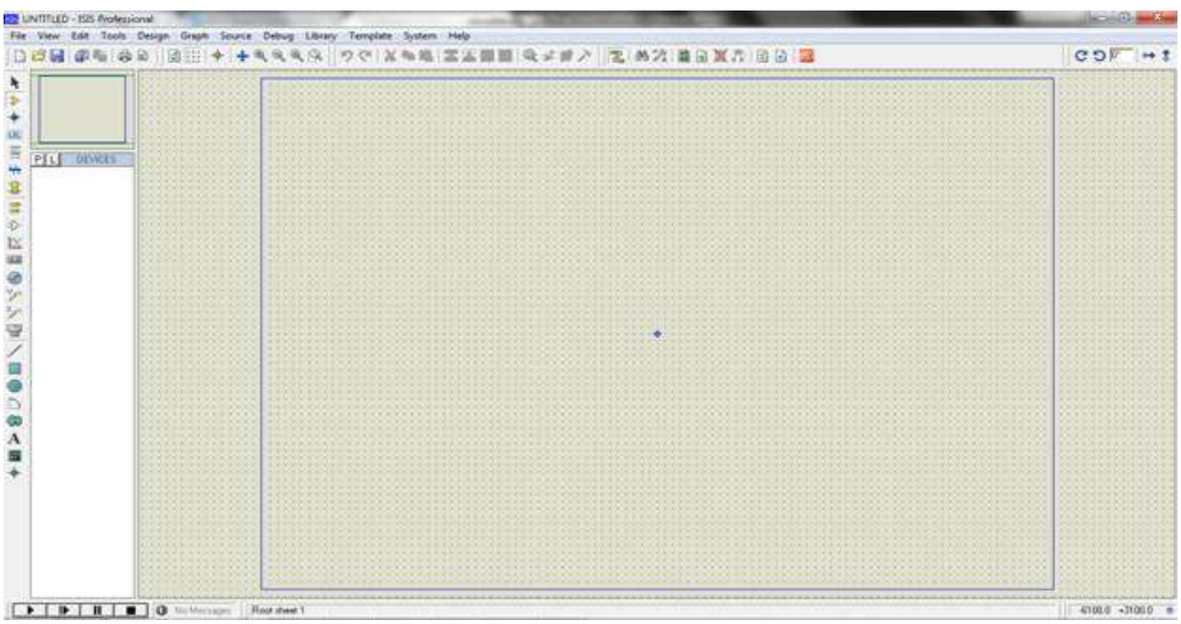


Fig. 3.2.5.3Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

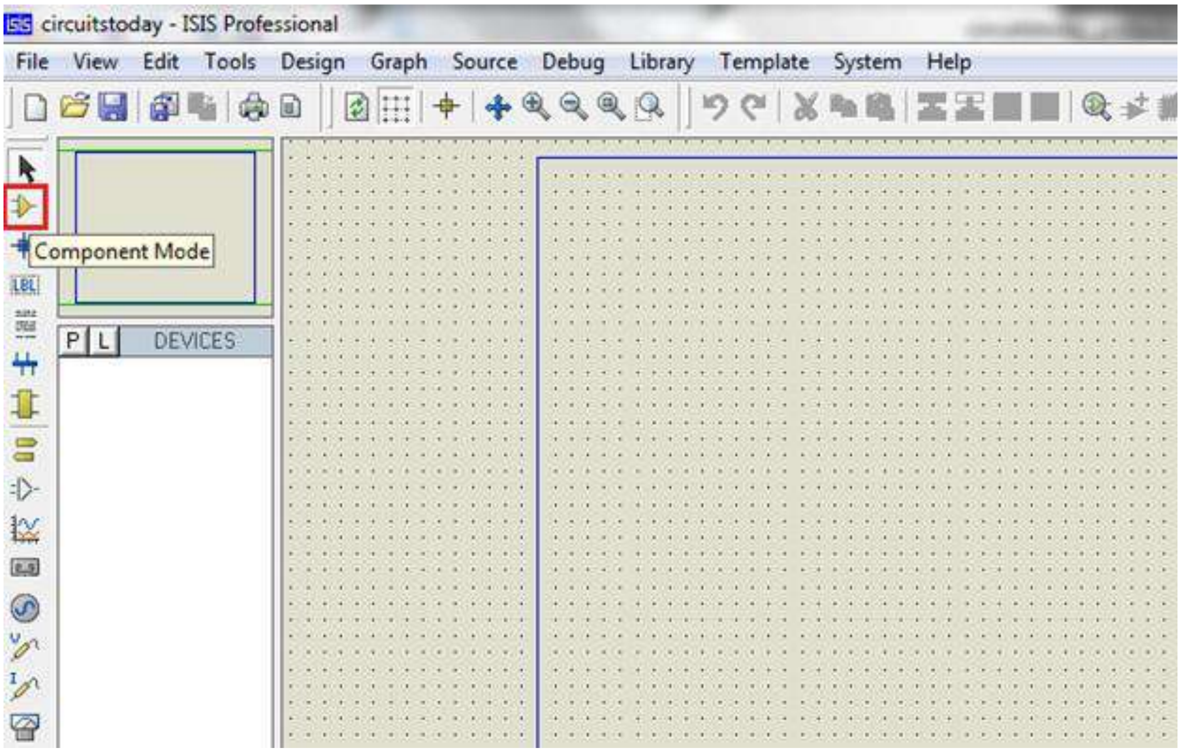


Fig. 3.2.5.4Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

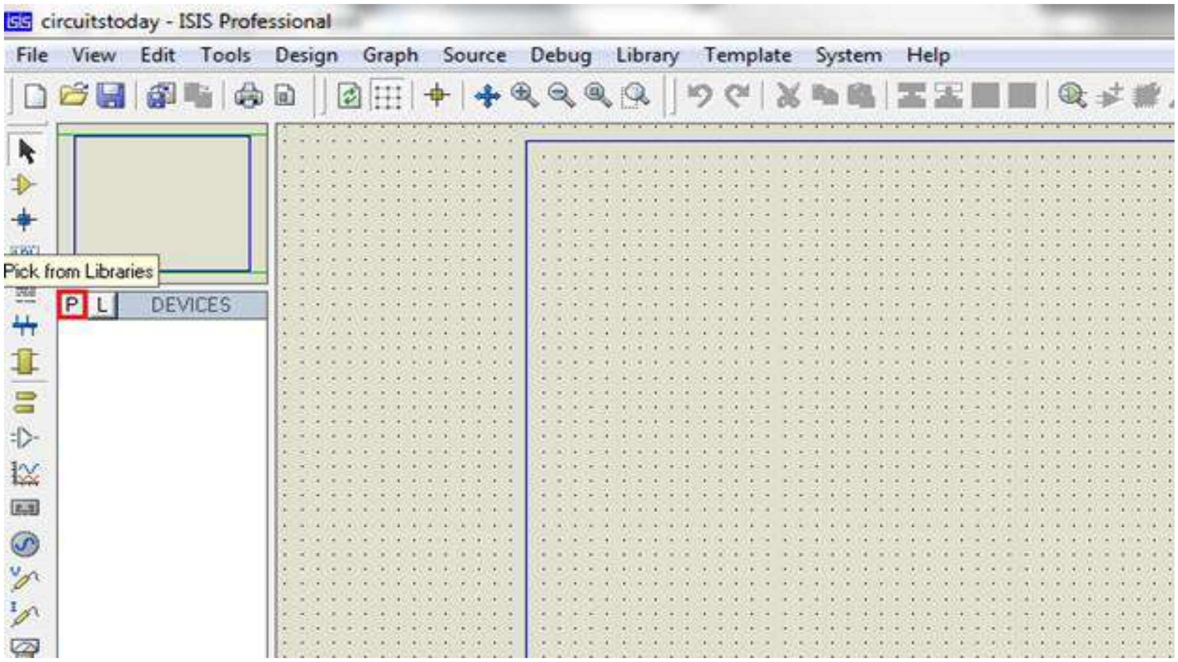


Fig. 3.2.5.5Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

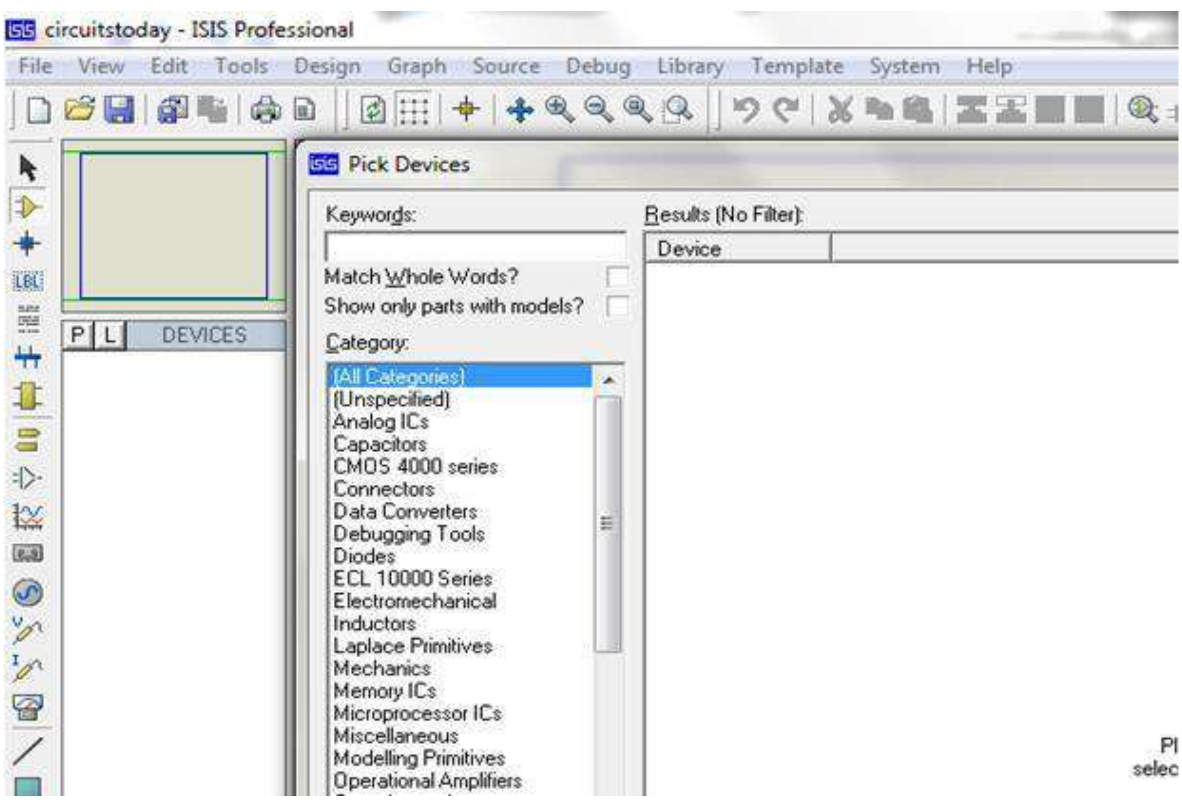


Fig. 3.2.5.6Keywords Textbox

Example shows selection of push button. Select the components accordingly.

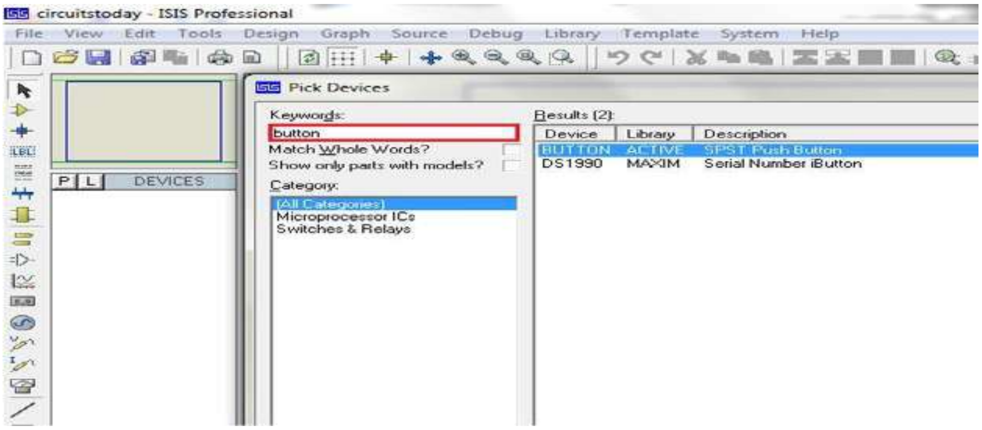


Fig.3.2.5.7Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

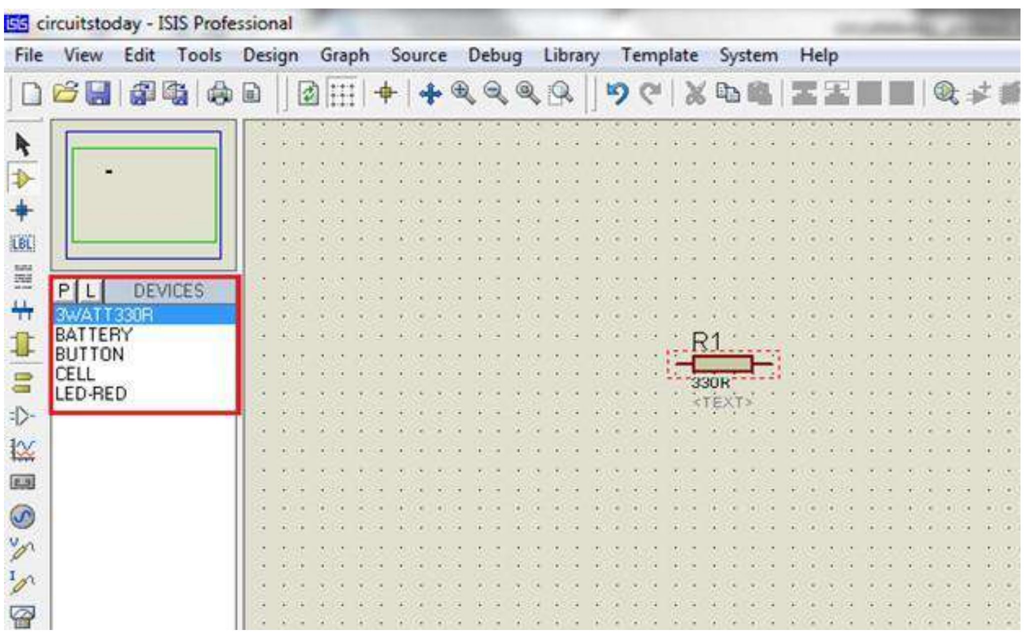


Fig.3.2.5.8Component Selection

Place all the required components and route the wires i.e., make connections. Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

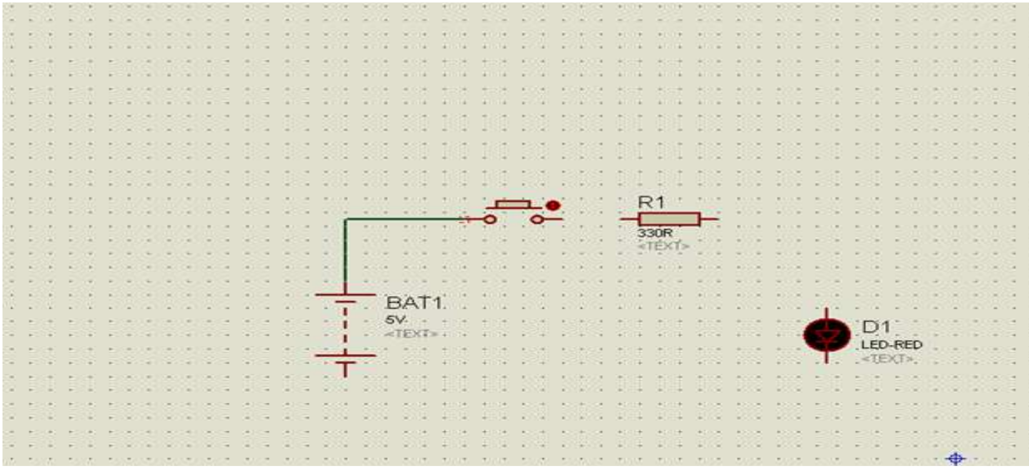


Fig.3.2.5.9Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

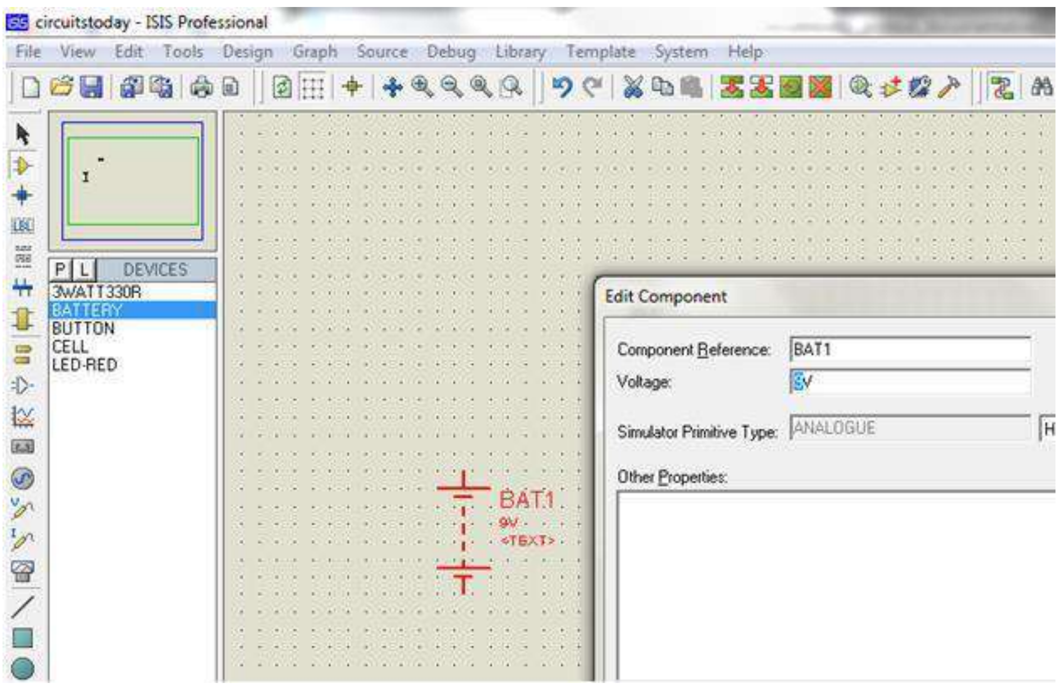


Fig. 3.2.5.10Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

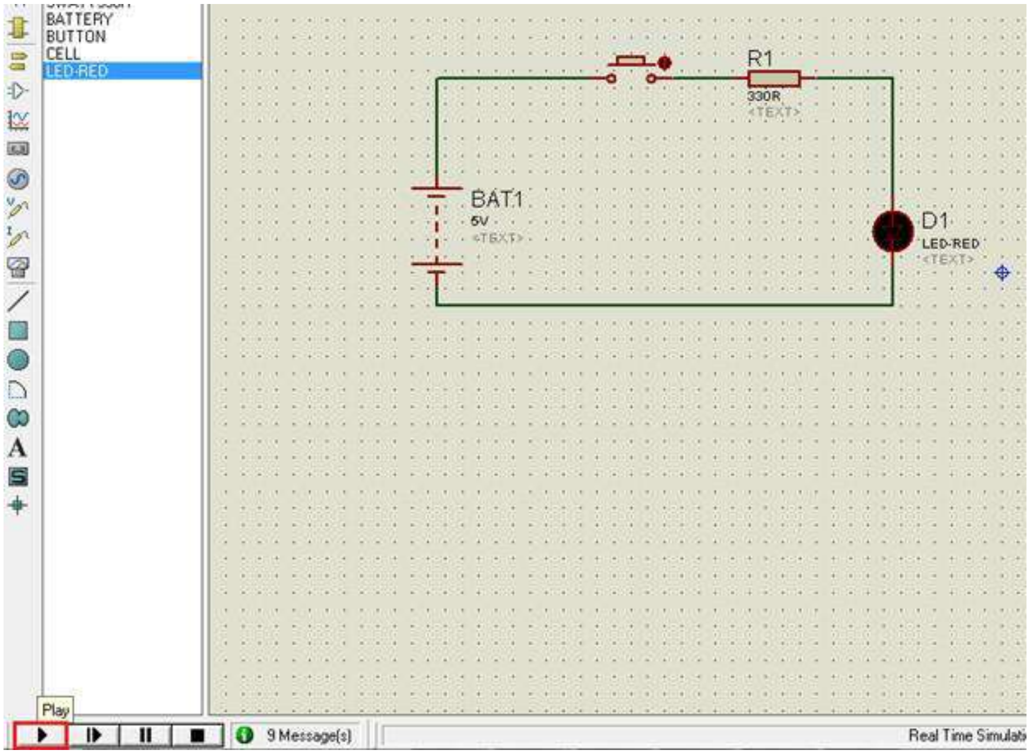


Fig. 3.2.5.11Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

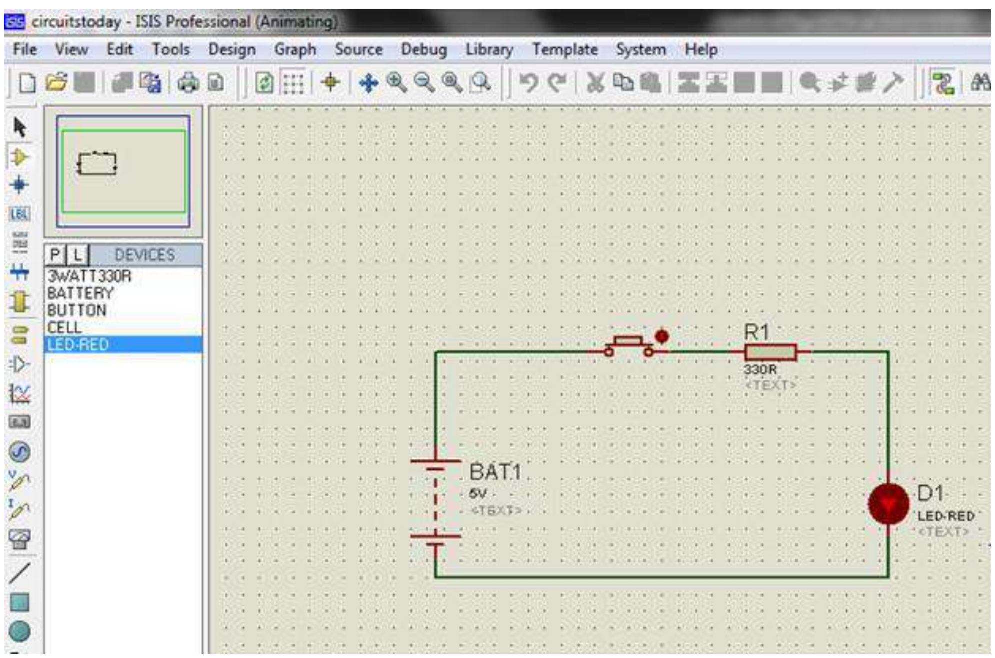


Fig. 3.2.5.12Simulation Animating

Simulation can be stepped, paused or stopped at any time.

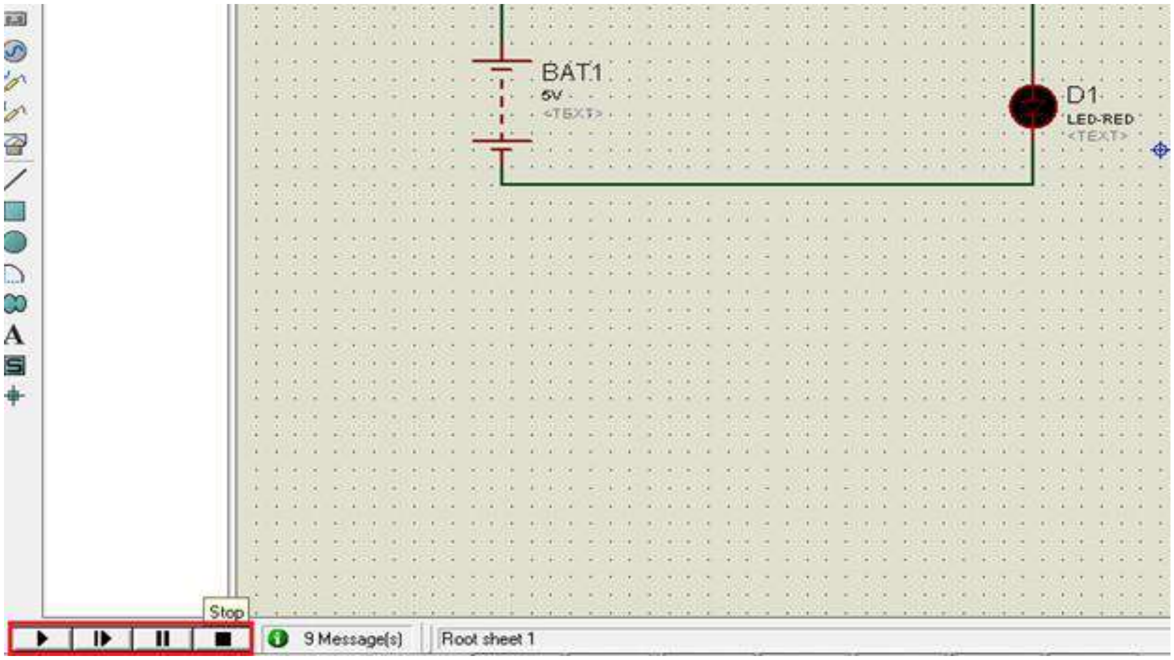


Fig. 3.2.5.13Simulation Step-Pause-Stop Buttons

According to its developers, Things Speak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. Things Speak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates.

Things Speak was originally launched by iot Bridge in 2010 as a service in support of IoT applications.

Things Speak has integrated support from the numerical computing software MATLAB from MathWorks, allowing Things Speak users to analyse and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license from Math works. [citation needed].

Things Speak has a close relationship with Math works, Inc. In fact, all of the Things Speak documentation is incorporated into the Math works' MATLAB documentation site and even enabling registered Math works user accounts as valid login credentials on the Things Speak website. The terms of service and privacy policy of ThingSpeak.com are between the agreeing user and Math works, Inc.

Setup Things Speak

Create a free MathWorks account or sign into Things Speak using an existing account.

Select the Things Speak channel you want your data to stream into. See Collect Data in a New Channel for help creating a new channel.

Record the following for the selected channel:

Channel ID, which is listed at the top of the channel view.

Write API key, which can be found on the API Keys tab of your channel view.

3.3 DEFINE

3.3.1 IR SENSOR

- IR stands infrared red spectrum, by using this region we can detect number of obstacles passing through a region.

- IR consists of two sections one is transmitter section and other is receiver section.

    - Transmitter section: it consists of a LED which continuously transmits a modulated signal of range (40 to 250khz).

    - Receiver section: Receiver section consists of a photo diode it will continuously track and receive the IR rays from LED.

- Functionality: The IR transmitter will continuously transmit IR rays and receiver (photo diode) will continuously receive the rays whenever some obstacle passes through the IR pair the transmission will be breaked by which we come to know that an intruder have been detected. In this way this communication will help us in safeguarding and many other applications.

3.3.2 LED

A light-emitting diode (LED) is a semiconductor light source. LEDs are used as indicator lamps in many devices, and are increasingly used for lighting. Introduced as a practical electronic component in 1962, early LEDs emitted low-intensity red light, but modern versions are available across the visible, ultraviolet and infrared wavelengths, with very high brightness.

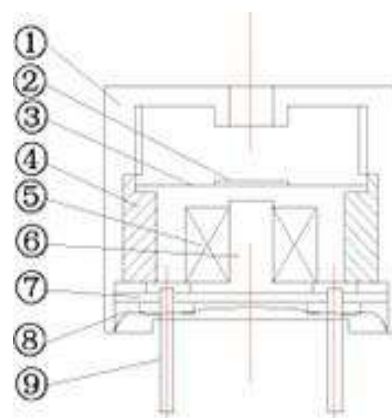
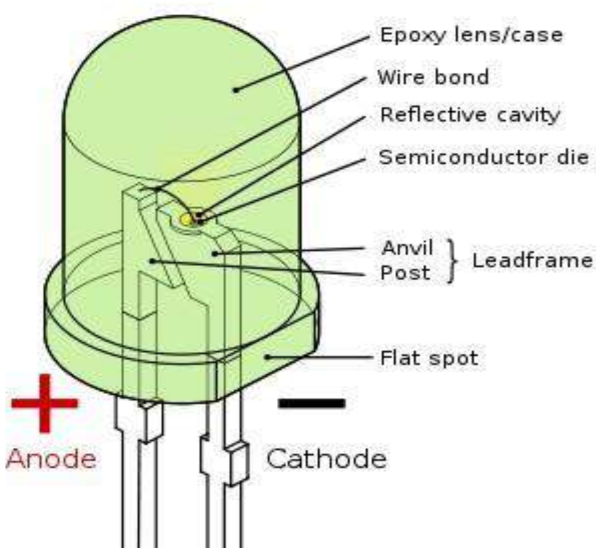
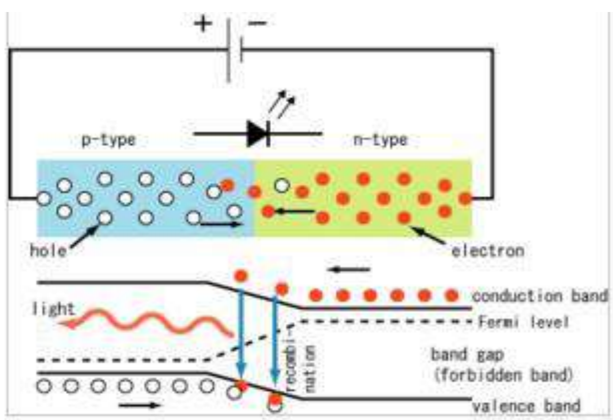


Fig.3.3.1 Magnetic Transducer

The LED is based on the semiconductor diode. When a diode is forward biased, electrons are able to recombine with holes within the device, releasing energy in the form of photons. This effect is called electroluminescence and the color of the light (corresponding to the energy of the photon) is determined by the energy gap of the semiconductor. An LED is usually small in area (less than 1 mm^2), and integrated optical components are used to shape its radiation pattern and assist in reflection. LEDs present many advantages over incandescent light sources including lower energy consumption, longer lifetime, improved robustness, smaller size, faster switching, and greater durability and reliability. However, they are relatively expensive and require more precise current and heat management than traditional light sources. Current LED products for general lighting are more expensive to buy than fluorescent lamp sources of comparable output.



Working: Charge-carriers electrons and holes—flow into the junction from electrodes with different voltages. When an electron meets a hole, it falls into a lower energy level, and releases energy in the form of a photon. The wavelength of the light emitted, and therefore its color, depends on the band gap energy of the materials forming the *p-n junction*. In silicon or germanium diodes, the electrons and holes recombine by a *non-radiative transition* which produces no optical emission, because these are indirect band gap materials. The materials used for the LED have a direct band gap with energies corresponding to near-infrared, visible or near-ultraviolet light.



The inner workings of an LED

Colors and materials :

	Color	Wavelength (nm)	Voltage (V)	Semiconductor Material
	Infrared	$\lambda > 760$	$\Delta V < 1.9$	Gallium arsenide (GaAs) Aluminum gallium arsenide (AlGaAs)
	Red	$610 < \lambda < 760$	$1.63 < \Delta V < 2.03$	Aluminum gallium arsenide (AlGaAs) Gallium arsenide phosphide (GaAsP) Aluminium gallium indium phosphide (AlGaInP) Gallium(III) phosphide (GaP)
	Orange	$590 < \lambda < 610$	$2.03 < \Delta V < 2.10$	Gallium arsenide phosphide (GaAsP) Aluminum gallium indium phosphide (AlGaInP) Gallium(III) phosphide (GaP)
	Yellow	$570 < \lambda < 590$	$2.10 < \Delta V < 2.18$	Gallium arsenide phosphide (GaAsP) Aluminium gallium indium phosphide (AlGaInP) Gallium(III) phosphide (GaP)
	Green	$500 < \lambda < 570$	1.9[42] < $\Delta V < 4.0$	Indium gallium nitride (InGaN) / Gallium(III) nitride (GaN) Gallium(III) phosphide (GaP) Aluminium gallium indium

			phosphide (AlGaInP) Aluminium gallium phosphide (AlGaP)
Blue	$450 < \lambda < 500$	$2.48 < \Delta V < 3.7$	Zinc selenide (ZnSe) Indium gallium nitride (InGaN) Silicon carbide (SiC) as substrate Silicon (Si) as substrate — (under development)
Violet	$400 < \lambda < 450$	$2.76 < \Delta V < 4.0$	Indium gallium nitride (InGaN)
Purple	multiple types	$2.48 < \Delta V < 3.7$	Dual blue/red LEDs, blue with red phosphor, or white with purple plastic
Ultraviolet	$\lambda < 400$	$3.1 < \Delta V < 4.4$	Diamond (235 nm) Boron nitride (215 nm) Aluminium nitride (AlN) (210 nm) Aluminium gallium nitride (AlGaN) Aluminium gallium indium nitride (AlGaInN) — (down to 210 nm)
White	Broad spectrum	$\Delta V = 3.5$	Blue/UV diode with yellow phosphor

3.4. POWER SUPPLY UNIT

3.4.1 INTRODUCTION:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as “Regulated D.C Power Supply”.

3.4.2 BLOCK DIAGRAM OF POWER SUPPLY:

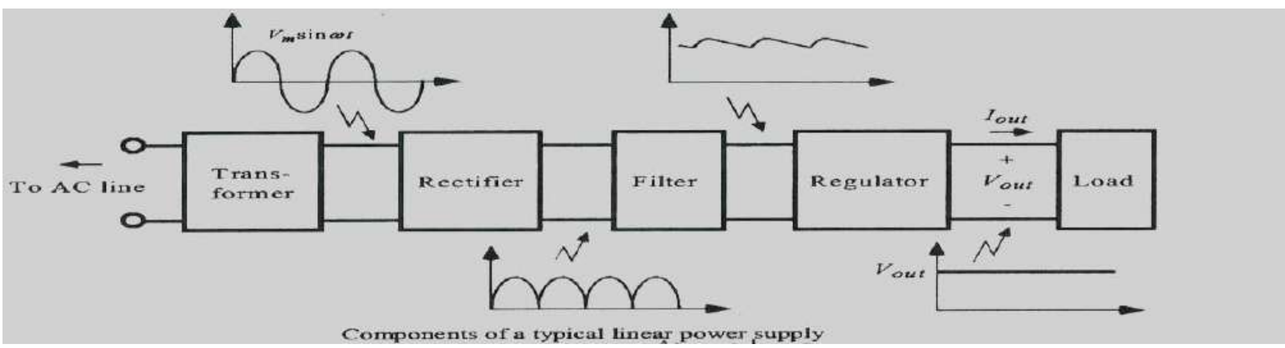


Fig.3.4.2: Block Diagram of Power Supply

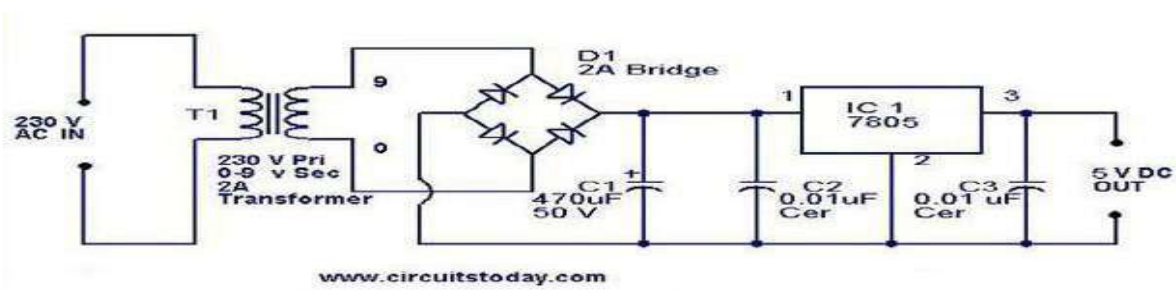


Fig.3.4.2(A): Schematic Diagram of Power Supply

3.4.3 DESCRIPTION OF POWER SUPPLY:

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a a different form of energy – such as solar, mechanical, or chemical - into electrical energy.

A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered. For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

3.4.4 TRANSFORMER:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

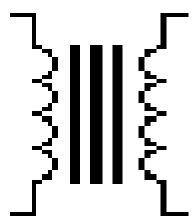


Fig.3.4.4: Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

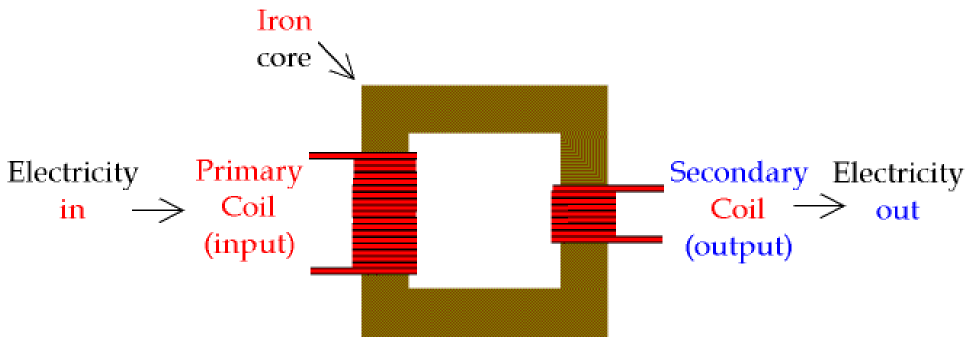


Fig.3.4.4.(A): Transformer

3.4.4.1 Basic Principle of Transformer:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

From Faraday's Law

$$\frac{V_S}{V_P} = \frac{N_S}{N_P}$$

For ideal transformer

The voltage ratio is equal to the turns ratio, and power in equals power out.

From conservation of energy

$$P_P = V_P I_P = V_S I_S = P_S$$

Working of Transformer:

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

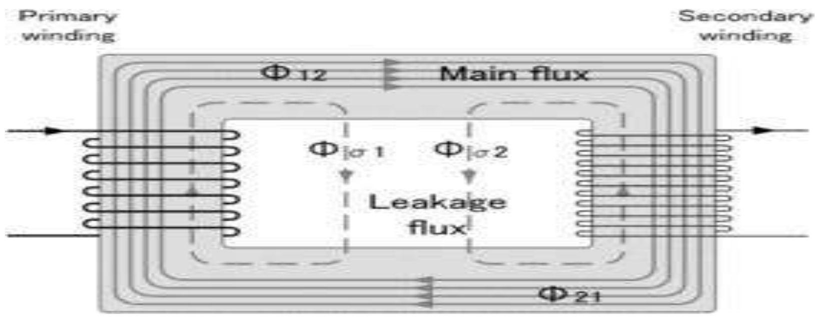


Fig.3.4.4.1: Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f. The iron core is laminated to prevent 'eddy currents' flowing in the core.

These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

3.4.4.2Classification of Transformer:

- Step-Up Transformer
- Step-Down Transformer

Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn’s ratio of the two sets of windings determines the amount of voltage transformation.

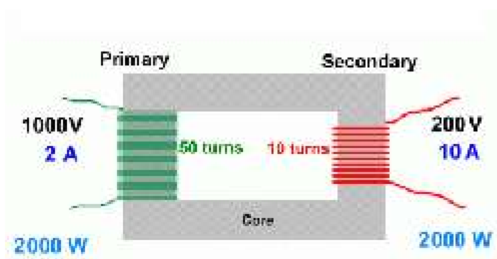


Fig.3.4.4.2: Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn’s ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn’s ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single phase steps down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase steps up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

Step-Up Transformer:

A step up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply. A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, Nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

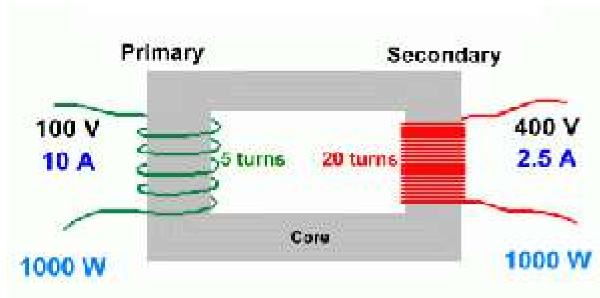


Fig.3.4.4.2(a): Step-Up Transformer

3.4.5 DIODES:

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

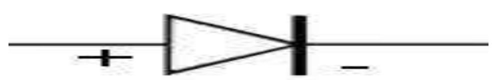


Fig.3.4.5: Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

3.4.6 RECTIFIER:

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

The Half-Wave Rectifier:

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure.

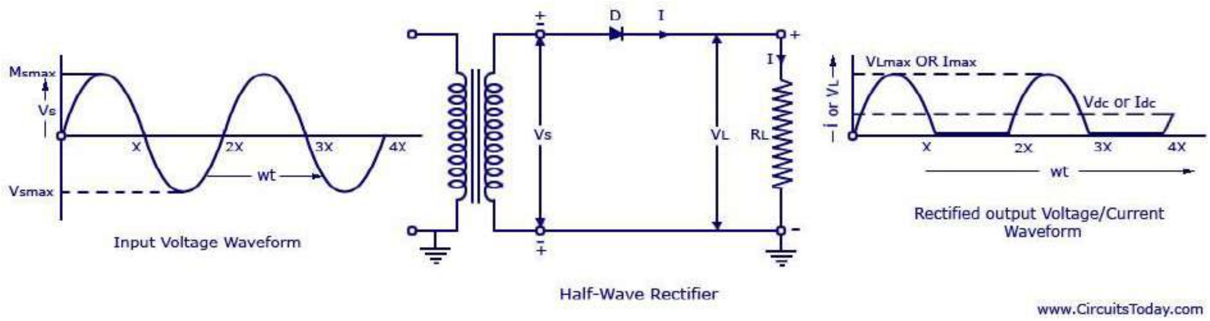


Fig.3.4.6(a): Half Wave Rectifier

Figure 2 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

The Full-wave Rectifier:

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC.

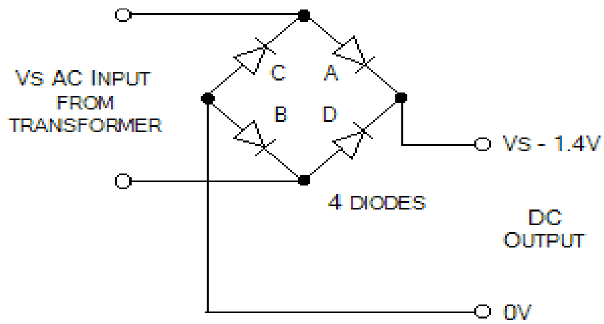


Fig.3.4.6(b): Full-Wave Rectifier

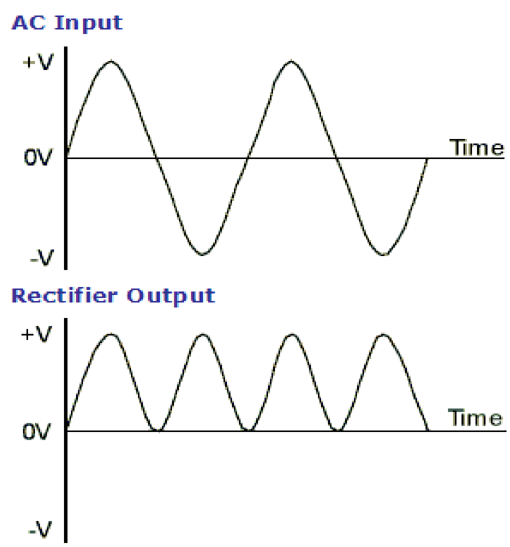


Fig.3.4.6(c): Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased. While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put ACin, you will 10.6V DC out.

3.4.7 CAPACITOR FILTER:

The capacitor-input filter, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

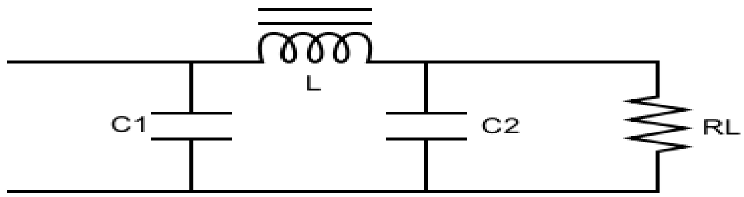


Fig.3.4.7: Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result, the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result, only the DC component appears across the load RL.

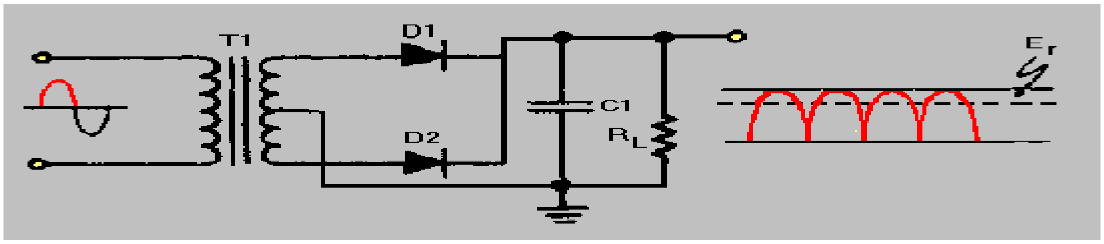


Fig.3.4.7(A): Cantered Tapped Full-Wave Rectifier with a Capacitor Filter

3.4.8 VOLTAGE REGULATOR:

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx 'indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx 'indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consist the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

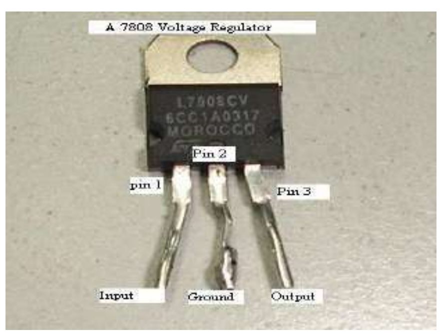


Fig.3.4.8 VOLTAGE REGULATOR

CHAPTER-4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM OF PROJECT:

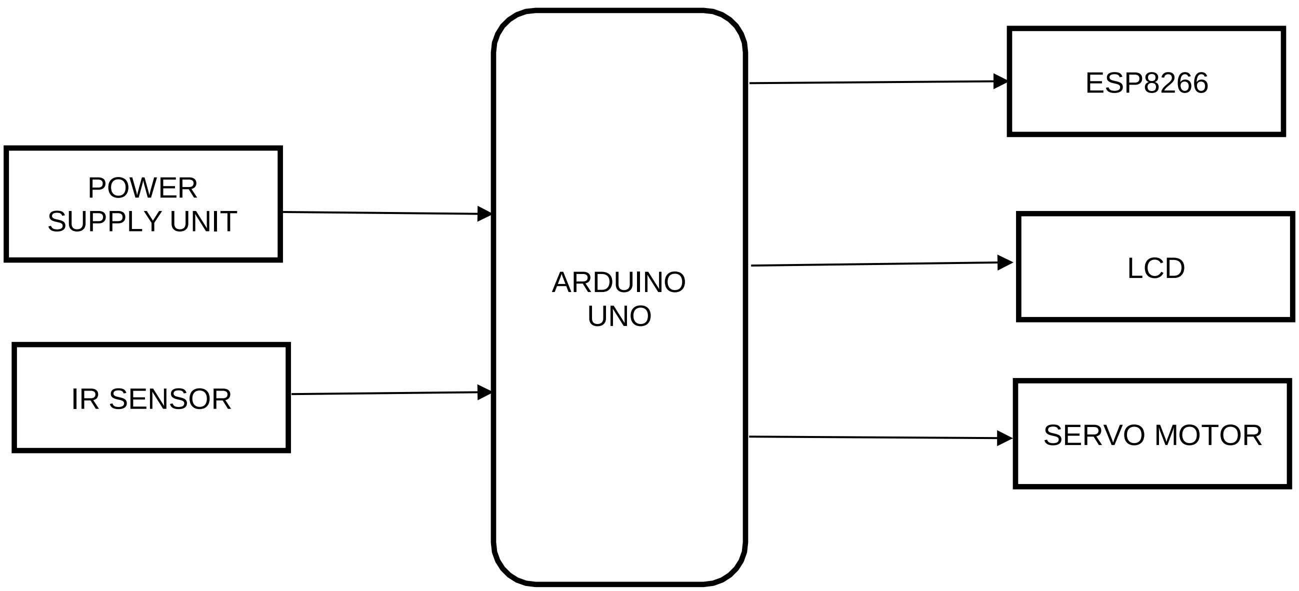


Fig.4.1: Block diagram of the project

4.2 FLOW CHART

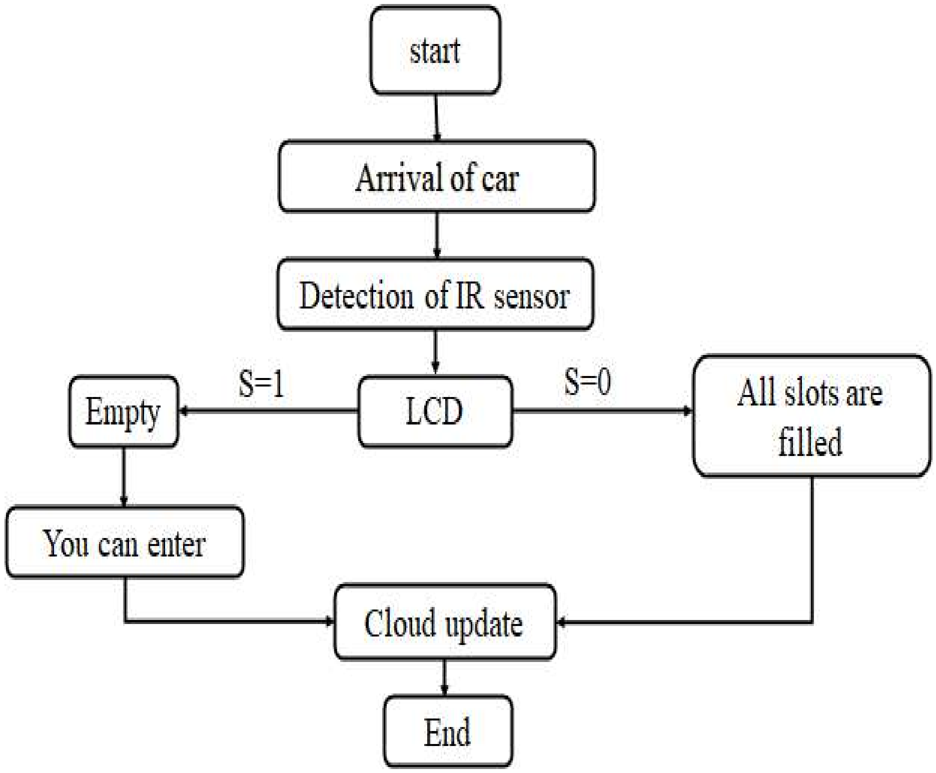


Fig.4.2 Flow Chart

4.3 IMPLEMENTATION STAGES

Stage-1: When the parking slot 1 is filled then the sensor detects and sends the data to the Arduino the LCD display as;

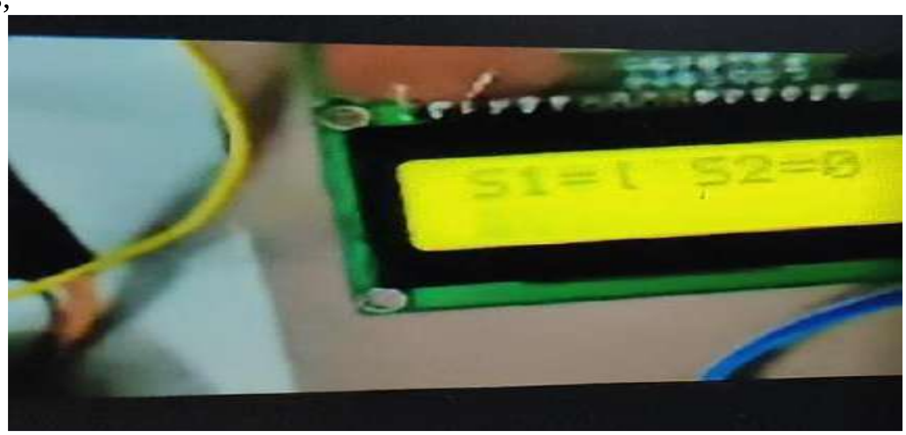


Fig.4.3 when the slot1 is filled.

Stage-2: When the parking slot 2 is filled then the sensor detects and sends the data to the Arduino and the LCD displays as:

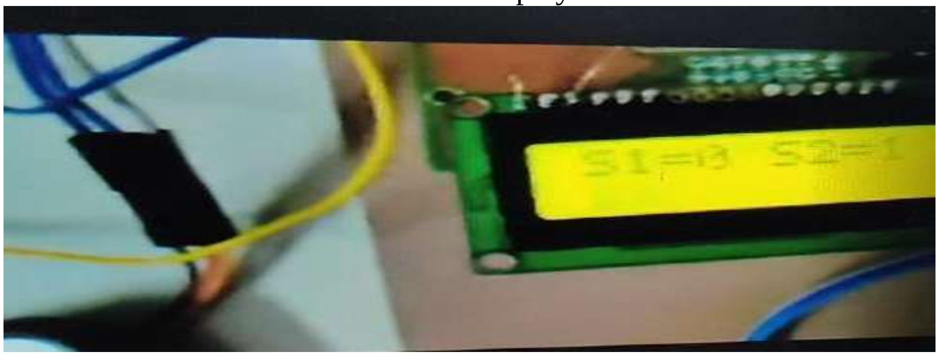


Fig.4.3.1 when the slot2 is filled.

Stage-3: When the both parking slots are filled then the sensor detects and sends the data to the Arduino and the LCD displays as:

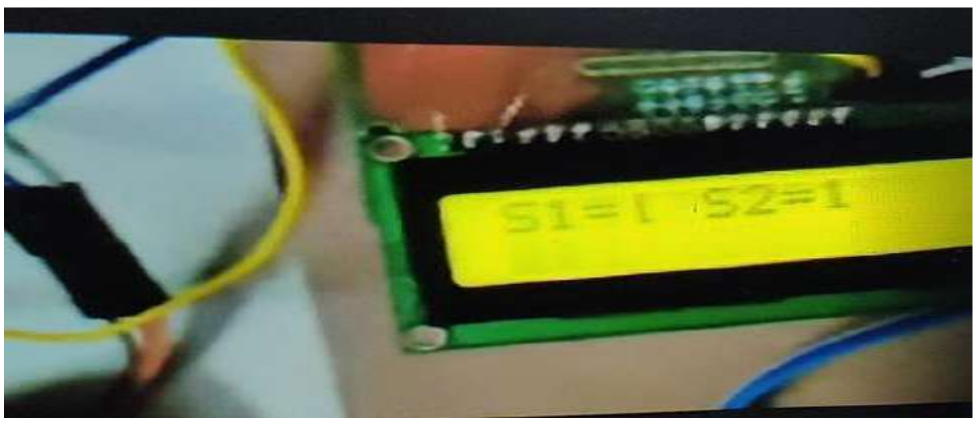


Fig.4.3.2 when both the slots are filled

Stage-4: When the parking slot are empty then the sensor detects and sends the data to the Arduino and the LCD displays as:

Fig.4.3.3 when the slots are empty

4.4 PROJECT CODE:

```
#include<LiquidCrystal.h>
#include<SoftwareSerial.h>
LiquidCrystallcd(13,12,11,10,9,8);
SoftwareSerialmySerial(2,3);//rx , tx //esp8266
#include <Servo.h>
int gate=5,s1=6,s2=7;
int servoPin = 4;
Servo Servo1;
void serialFlush()
{
while(Serial.available() > 0)
  {
    char tt = Serial.read();
  }
}
void sendwifi(String chr,unsigned int len)
{
mySerial.print("AT+CIPSEND=0,");
mySerial.println(len-1);
delay(2000);
mySerial.print(chr);
delay(2000);
}
char res[130];
char buff[130];
void setup()
{
```

```
pinMode(s1,INPUT);
pinMode(s2,INPUT);
pinMode(gate,INPUT);
Servo1.attach(servoPin);
Servo1.write(0);
lcd.begin(16,2);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("IOT Based");
lcd.setCursor(0,1);
lcd.print("Car Parking");
delay(2000);
Serial.begin(9600);
mySerial.begin(115200);//esp8266
mySerial.print("AT\r\n");
delay(1000);
mySerial.print("ATE0\r\n");
delay(1000);
mySerial.print("AT+CWMODE=3\r\n");
delay(1000);
  mySerial.print("AT+CWSAP=\"Project\",\"project1235\",5,0\r\n\r\n");
delay(1000);
mySerial.print("AT+CIPMUX=1\r\n");
delay(1000);
mySerial.print("AT+CIPSERVER=1,23\r\n");
delay(1000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("WATING FOR LINK");
while(1)
  {
if(mySerial.available())
    {
```

```
        //if(Esp.find("0,LINK"))
if(mySerial.find("0,CONNECT"))
        {
lcd.clear();lcd.setCursor(0, 0);lcd.print("LINK FOUND");
         break;
        }

      }
    }
delay(1000);
sendwifi("WELCOME \r\n",11);
delay(1000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("READY");
delay(2000);
}
void loop()
{
lcd.clear();lcd.setCursor(0,0);lcd.print("S1=");lcd.print(digitalRead(s1));lcd.print("
S2=");lcd.print(digitalRead(s2));
delay(1000);
  if(digitalRead(gate)==LOW)
  {
lcd.clear();lcd.setCursor(0,0);lcd.print("U can Enter");
    Servo1.write(90);
delay(3000);
    Servo1.write(0);
delay(500);
  }
if(digitalRead(s1)==LOW )
   {
    if(digitalRead(s2)==LOW )
```

```
    {
lcd.clear();lcd.setCursor(0,0);lcd.print(0,0);lcd.print("All Slots are Full");
lcd.setCursor(0,1);lcd.print("Please Wait");
delay(100);
sprintf(buff,"All Slots are Full \r\n",22);
sendwifi(buff,strlen(buff));
delay(3000);
      //Servo1.write(0);
    }
   }
   if(digitalRead(s1)==HIGH )
    {
sprintf(buff,"Slot 1 Empty \r\n",16);
sendwifi(buff,strlen(buff));
delay(3000);
    }
if(digitalRead(s2)==HIGH )
    {
sprintf(buff,"Slot 2 Empty \r\n",16);
sendwifi(buff,strlen(buff));
delay(3000);
    }
}
```

4.5 RESULTS

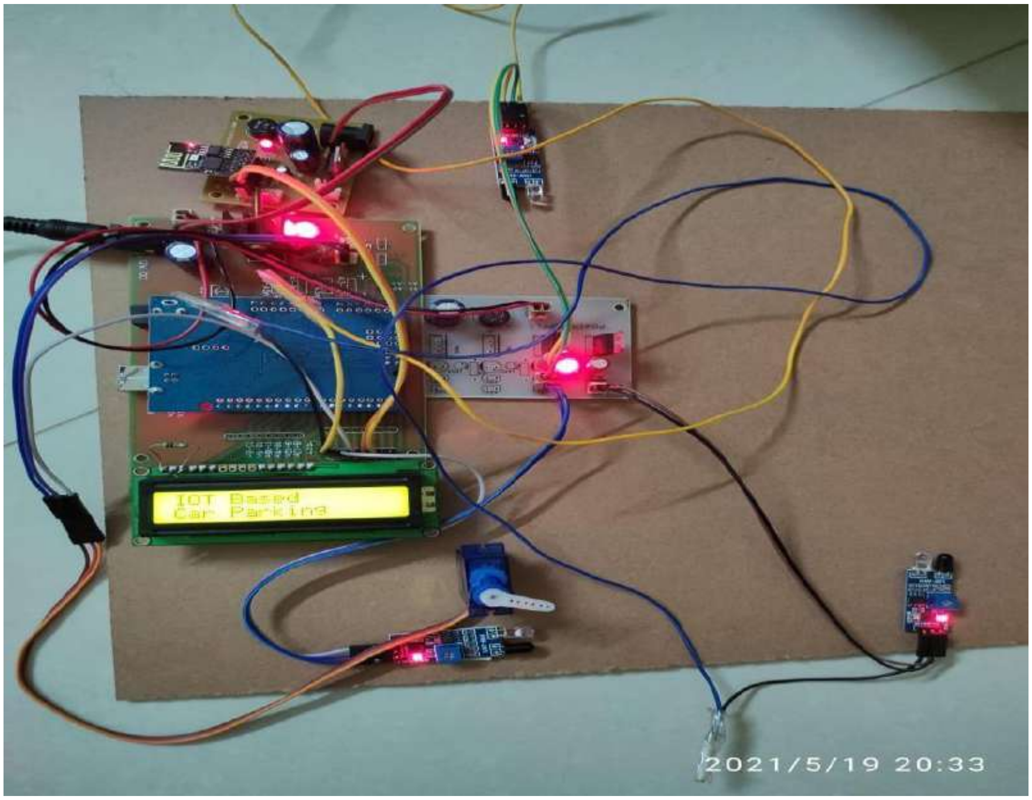


Fig.4.5 Final Output

CHAPTER-5

PROJECT TESTING

5.1 OVERVIEW OF TESTING METHODS

The test of the circuit data communication of the whole circuit to and from cloud is done by detecting the existence of the car 10 times with the assumption of 150 cm car height and floor to ceiling distance of 350 cm. The measuring point of this test is the success of the circuit in sending data, the length of time required, and the validity of the data. Calculation of respond time on the Arduino and server is done by the progr am by recording the difference betweenstart and end time of the program.

The calculation of respond time on cloud is done by calculating the difference between server sent time and Ubidots received time. Based on the testing result as shown in Table 1 below, The sensor circuit has 100 % success rate for sending data to cloud with an average total required time of 6.4004 s for each data. Data errors were not found during the test.

Testing the accuracy of sensor readings conducted by car detection test with the assumption that the car has a height of 150 cm. Testing is done as much as ten times for each circuit of sensors by shifting obstacles. If there is a parking car, then the obstacles will be placed between the sensor and the floor. If the condition of the parking slot is empty, then the barrier will be removed so that the sensor will immediately detect the floor.

- When a driver knows exactly where they need to go; it reduces idling and unnecessary driving – therefore optimises traffic flows in built-up areas.
- Smart Parking creates the possibility of new business models that are only made possible using technology.
- Decreased searching for spaces can reduce accidents by ensuring drivers maintain their attention rather than browsing for spaces or making rash maneuvers.
- One goal of Smart Parking is to reduce the time taken and the hassle factor of locating an available parking space detecting the existence of the car 10 times with the assumption of 150

cm car height and floor to ceiling distance of 350 cm. The measuring point of this test is the success of the circuit in sending data, the length of time required, and the validity of the data calculation of respond time on the Arduino and server is done by the pstart and end time of the program.

The calculation of respond time on cloud is done by calculating the difference between server sent time and Ubidots received time. Based on the testing result as shown in Table 1below, The sensor circuit has 100% success rate for sending data to cloud with an average total required time of 6.4004 s for each data.

The method used to test RFID circuit communications to cloud is by tapping RFID cards ten times. In the program, there are already methods of calculating the respond time such as testing the sensor circuit. The measured points are the success of the circuit sending the data, the time it takes to send from the circuit to cloud, and the validity of the data sent with the received.

Security: IoT is a combination of physical objects ("things"), sensors, embedded software and keeps a goal of effective connectivity and faithful data exchange. The current graphs imply that IoT market growing at remarkable rate. However, some of these devices suffer from limited memory, power consumption and processing power. These issues may cause IoT to become penetrable and hence security is at utmost priority to IoT domain.

As stated in a 2015 report by Hewlett-Packard on IoT research [3]:

1. Per 100, 70 devices still use un-encrypted network services.

2. Per 100, 90 devices collected at least one piece of personally identifiable information (via device, cloud or mobile app).

3. Per 100, 70 devices (with cloud and mobile app components) enabled an attacker to identify valid user accounts through enumeration.

4. 80% of devices (with cloud and mobile app components) failed to require passwords of sufficient complexity.

5.Behind every 10 device every 6 devices user interfaces (UI) were vulnerable

Following are the vulnerabilities of the IoT as per The Open Web Application Security Project (OWASP)

1) Insufficient protection compositions.

2) Dubiously secured Software/Firmware

3) Doubtful Network Services

4) Doubtful Web Interface

5) Worst Physical Security

6) Doubtful Cloud Interface

7) Doubtful Mobile Interface

8)Privacy Concerns

9) Doubtful Authentication/Authorization

10) Inadequate Transport Encryption Ongoing approaches to secure IoT have attempted to put a grip on communication protocol-based mechanisms, such as encryption for data-at-rest or in-transit. But this itself is doubtful if the respective endpoints themselves are capable of being modified either by local access or remote connections. Gartner claims that by year 2020 more than 25% of identified attacks in a particular company will be on IoT devices or systems, even though IoT will only contribute to less than 10% of IT security budgets.

CHAPTER-6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The smart parking system based on IoT concept has been implemented using various sensor circuitry and cloud (server). It is an efficient system for car parking which prevails traffic congestion. Smart parking systems gives rise to new solutions with the help of Internet of things . The system have always been at the core of constructing smart cities. In this project, we address the issue of parking. It presents an IoT based Cloud integrated smart parking system. With the help of this mobile application all the updated information can be checked. The efforts made in this project are indented to improve the parking facilities of a city and thereby aiming to enhance the quality. This designed automatic smart parking system which is simple, economic and provides effective solution to reduce carbon footprints in the atmosphere.

Thus it reduces the crisis of car parking across a remote city and also it eliminates unnecessary travelling of vehicles across the filled parking slots in a city. So it reduces time and it is cost effective.

6.2 FUTURE ENHANCEMENT

This work is further extended as smart car parking system with automatic billing system also fully automated system using multilayer parking method. Safety measures such as vehicle no. tracing, driver face recognition. Also care has taken so that there is no malfunction of wrong vehicle entering into the allocated slot by providing an unique OTP to each person and ensuring the same person parks in the given slot.

The future enhancement can be in such a way that by using GPS system we can find where the car is being parked and and also the number of vacancy available in the parking system.

PUBLICATION

Submitted Paper in the Conference ICISMEC-21 with Paper ID (ICISMEC21-0090) and got Acceptance for the Paper.

REFERENCES

[1] Thanh Nam Pham1, Ming-Fong Tsai1, Duc Bing Nguyen1, Chi-Ren Dow1 and Der-Jinn Deng2. “A Cloud Based Smart-Parking System Based on Internet-of-Things Technologies”. IEEE Access, volume 3, pp. 1581 – 1591, September 2015.

[2] Callum Rhodes, William Blewitt, Craig Sharp, Gary Upshaw and Graham Morgan. “Smart Routing: A Novel Application of Collaborative Path-finding to Smart Parking Systems”. Business Informatics (CBI), 2014 IEEE Conference on volume 1, pp. 119-126, 2014.

[3] Yamen Gang and Christos G. Cassandras. “A New Smart Parking System Based on Optimal Resource Allocation and Reservations”. IEEE Transaction on Intelligent Transportation Systems, volume 14, pp. 1129 -1139, April 2013.

[4] K.Ashokkumar a, Baron Sam , R.Arshadprabhu , Britto. "Cloud Based Intelligent Transport System". Procedia Computer Science, volume 50, pp. 58-63, 2015.

[5] Prof. D. J. Bonde , Rohit S. Shende, Ketan S. Gaikwad, Akshay S. Kusarigama U. Bhakra. “Automated Car Parking System Commanded by Android Application”, (IJCSIT) International Journal of Computer Science and Information Technologies, volume 5(3), pp. 1-4, 2014.

[6]. Zhou, F., & Li, Q. (2014, November). Parking Guidance System Basedon ZigBee and Geomagnetic Sensor Technology. In DistributedComputing and Applications to Business, Engineering and Science.(DCABES), 2014 13th International Symposium on (pp. 268-271). IEEE. [7]. V. P. Bilodeau, "Intelligent parking technology adoption," University of Southern Queensland, 2010.

[8]. ZhanlinJi; Ivan Ganchev; MáirtínO'Droma; Xueji Zhang, “A cloud-based intelligent car parking services for smart cities” 2014 XXXIth URSI General Assembly and Scientific Symposium (URSI GASS)2014IEEE Conference Publications.

[9]. Thanh Nam Pham; Ming-Fong Tsai; DucBinh Nguyen; Chyi-Ren Dow; Der-Jiunn Deng, “A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies” 2015, Volume: 3, IEEE Journals & Magazines.

BIBILOGRAPHY

[1] 2015- An approach to IOT based car parking and reservation system on cloud _Vaibhav Hans, Parminder Singh Sethi, JatinKinra.

[2] 2016- Novel vehicle booking system using IOT_S. Vidhya Sagar; B. Balakiruthiga; A. Sivanesh Kumar

[3] 2016- Performance notification in a reservation-basedparkingsystem _PujiantoYugopuspito; Ryant A. Herwansyah; Dion Krisnadi; SutrisnoCahya; Frans Panduwinata

[4] 2016- Toward service oriented design for reservation-based parking _PujiantoYugopuspito; Frans Panduwinata; Sutrisno; JulindaPangaribuan

[5] 2017- A navigation and reservation based smart parking platform using genetic optimization for smart cities-lhan Aydin; Mehmet Karakose; Ebru Karakose.

[6] Zheng, Y., Rajasegarar, S., & Leckie, C. (2015, April). Parking availability prediction for sensor-enabled car parks in smart cities.In Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2015 IEEE Tenth International Conference on (pp. 1-6).

[7] Han, D. M., & Lim, J. H. (2010). Smart home energy management system using IEEE 802.15. 4 and zigbee. Consumer Electronics, IEEE Transactions on, 56(3), 1403-1410.

[8] Parwekar, P. (2011, September). From Internet of Things towards cloud of things. In Computer and Communication Technology (ICCCT), 2011 2nd International Conference on (pp. 329-333). IEEE.

[9] Rao, B. B. P., Saluia, P., Sharma, N., Mittal, A., & Sharma, S. V. (2012, December). Cloud computing for Internet of Things & sensing based applications. In Sensing Technology (ICST), 2012 Sixth International Conference on (pp. 374-380).

APPENDICES

Now a days the concept of smart cities have gained grate popularity. Thanks to the evolution of Internet of things the idea of smart city now seems to be achievable. Major efforts are being made in the field of IOT in order to maximize the productivity and reliability of urban areas. Problems such as, traffic congestion, limited car parking facilities and road safety are being addressed by IOT. In this paper, we present an IOT based Server integrated smart parking system. This system consists of an IOT module which check the state of availability of parking lot. A mobile application is also provided that allows an end user to check the availability of parking space and book a parking slot accordingly. The paper also describes a high-level view of the system architecture. Towards the end, the paper discusses the working of the system in form of a use case that proves the correctness of the proposed model.

A

MAJOR PROJECT REPORT

On

Selection of a Stopping Criterion for Anisotropic Diffusion Filtering in Ultrasound Images

Submitted by

1)Mr. B.BHARATH KUMAR (17K81A04J4) 2)MR.R.ANILKUMAR (18K85A0437)

3)MR.B.NAGARAJU(18K85A0441)

in partial fulfillment for the award of the degreeof

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Mr.Dr.A.CHAITANYA KRISHNA

Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST.MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500010

St. MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

AN UGC AUTONOMOUS INSTITUTE

Dhulapally, Secunderabad-500 100

NBA & NAAC A+ Accredited

Department Of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the project entitled "SELECTION OF A STOPPING CRITERION FOR ANISOTROPIC DIFFUSION FILTERING IN ULTRA SOUND IMAGES" of a is being submitted by1.**Mr.B.BHARATH KUMAR(17K81A04J4),2.Mr.R.ANILKUMAR(18K85A0437)**,**3.Mr.B.NAGARJU(18K85A0441)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN** <department name> is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Project Guide
Dr.A.CHAITANYA KRISHNA
Department of ECE

Head of the Department
Dr. B. HARI KRISHNA
Department of ECE

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **BANDARAM NAGARAJU** WITH ROLL NO.**18K85A0441, BATHUL BHARATH KUMAR** WITH ROLL NO.**17K81A04J4**, **RAMAVATH ANIL KUMAR** WITH ROLL NO.**18K85A0437,** OF B.TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**SELECTION OF A STOPPING CRITERION FOR ANISOTROPIC DIFFUSION FILTERING IN ULTRASOUND IMAGES**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

ORUGANTI VENKAT

DIRECTOR

TRAININGS & PLACEMENTS LASYA IT SOLUTIONS
PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com Website : www.lasyainfotech.co

DECLARATION

We, the student of **Bachelor of Technology** in Department of ELECTRONICS AND COMMUNICATION ENGINEERING,', session: <2017 – 2021>, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled "SELECTION OF A STOPPING CRITERION FOR ANISOTROPIC DIFFUSION FILTERING IN ULTRA SOUND IMAGES'' is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

B.BHARTHA KUMAR (17K81A04J4)
R.ANILKUMAR(18K85A0437)
B.NAGARAJU(18K85A0441)

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would b incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude to **Dr.B.HARI KRISHNA Head &Professor**, the Dept. of Information Technology for his/her advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to our dept .project coordinator Dr.ANAND.A KUMAR,PROFESSOR,for his/her involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide,**Mr.Dr.A.Chaitanya Krishna professer**, department ECE and our class project coordinator **Ms.P.Pushpa Assistanat professer**,department of ECE for guidance and encoragment in carrying out this project work

Finally, we are highly grateful and obliged to our college level Project Coordinator **Project Coordinators Dr.Maalyadri**, **Professor in IT** and **Dr.R.Santhosh Kumar**, **Associate Professor in CSE** and for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. B.BHARATH KUMAR(17K81A04J4)
2. R.ANILKUMAR(18K85A0441)

3. B.NAGARAJU(18K85A0441)

ABSTRACT

Ultrasound imaging is a safe and cost-effective diagnostic tool, but the quality of the images is affected by speckle noise and artifacts. Anisotropic diffusion filters can be used to reduce noise and preserve the edges in the image. However,this technique is very sensitive to the number of iterationsselected. This paper proposes a stopping criterion for effective noise removal without blurring the edges, based on the relativevariance between the estimated denoised image and the original one. Different quality metrics were evaluated in 25 test images.The results suggest that the proposed stopping criterion can be implemented efficiently and aids in the process of automation of the filter.Ultrasound image enhancement, speckle reduction,anisotropic diffusion filtering

TABLE OF CONTENTS

.

LIST OF TABLES

LIST OF FIGURES

LIST OF GRAPHS

GLOSSARY OF COMMONLY USED TERMS:

Anisotropic diffusion filter: This filter as commonly used in medical image enhacement.They are effective to reduces noise in homegeous regiouns eith out bluring the edges by preventing intra region smoothing.

Images shading shapes: The image shading shapes caused by the object surface illumation in space is used to reconstruct the surface orinetion.

Magnetism: Magnetism is a property of matter that is a result of the orbiting electrons in atoms. The orbiting electrons cause the atoms to have a magnetic moment associated with an intrinsic angular momentum called 'spin'. Magnetic field strengths are measured in units of Tesla (T).

Ferromagnetism: Ferromagnetic materials generally contain iron, nickel, or cobalt. These materials have a large positive magnetic susceptibility, i.e., when placed in a magnet field, the field strength is much stronger inside the material than outside

Para magnetism: Paramagnetic materials include oxygen and ions of various metals like Fe, Mg, and Gd. These ions have unpaired electrons, resulting in a positive magnetic susceptibility. The magnitude of this susceptibility is less than one one-thousand of that of ferromagnetic materials

Superparamagnetism: Superparamagnetic materials consist of individual domains of elements that have ferromagnetic properties in bulk. Their magnetic susceptibility is between that of ferromagnetic and paramagnetic materials. Figure illustrates the effect of a superparamagnetic material (grey circle) on the magnetic field flux lines (blue). Examples of superparamagnetic materials include iron containing contrast agents for bowel, liver, and lymph node imaging

Diamagnetism: weak negative magnetic susceptibility contributes to the loss of signal seen in bowel on MRI after administration of ba Diamagnetic materials have no intrinsic atomic magnetic moment, but when placed in a magnetic field weakly repel the field, resulting in a small negative magnetic susceptibility. Materials like water, copper, nitrogen, barium sulfate, and most tissues are diamagnetic. Figure 2.12 illustrates the effect of a diamagnetic material (grey circle) on the magnetic field flux lines (blue). The rium sulfate suspensions.

RF Coils: RF coils are the "antenna" of the MRI system that broadcasts the RF signal to the patient and/or receives the return signal. RF coils can be receive-only, in which case the body coil is used as a transmitter; or transmit and receive (transceiver)

Surface Coils : Surface coils are the simplest design of coil. They are simply a loop of wire, either circular or rectangular, that is placed over the region of interest. The depth of the image of a surface coil is generally limited to about one radius. Surface coils are commonly used for spines, shoulders, and other relatively small body parts.

Paired Saddle Coils: Paired saddle coils are commonly used for imaging of the knee. These coils provide better homogeneity of the RF in the area of interest and are used as volume coils, unlike surface coils. By running current in opposite directions in the two halves of the gradient coil, the magnetic field is made stronger near one and weaker near the other

The Helmholtz Pair Coils: The Helmholtz pair coils consist of two circular coils parallel to each other. They are used as z gradient coils in MRI scanners. They are also used occasionally as RF coils for pelvis imaging and cervical spine imaging.

The Bird Cage Coil: The bird cage coil provides the best RF homogeneity of all the RF coils. It has the appearance of a bird cage; hence, its name. This coil is commonly used as a transceiver coil for imaging of the head. This type of coil is also used occasionally for imaging of the extremities, such as the knees

Gradient Coils: Gradient coils are used to produce deliberate variations in the main magnetic field (Bo). There are usually three sets of gradient coils, one for each direction. The variation in the magnetic field permits localization of image slices as well as phase encoding and frequency encoding.

Image Reconstruction: The raw data generated by most sequences is separated into phase and frequency components by a mathematical procedure called Fourier transformation. To generate an anatomic image, the raw data must be inverse Fourier transformed

Aliasing or "Wrap-around: Aliasing or wrap-around is a common artifact that occurs when the field of view (FOV) is smaller than the body part being imaged. The part of the body that lies beyond the edge of the FOV is projected on to the other side of the image. This can be corrected, if necessary, by oversampling the data

Radio Frequency : The radio frequency power that is capable of being produced matches that of many smradio stations (15-20 kW). As a result there is the presence of heating effects from the RF.

CHAPTER-1
INTRODUCTION

1.1 OVERVIEW OF THE PROJECT:

In this chapter, we will discuss the basic principles behind the physics of magnetic resonance imaging (MRI). Some of these principles are explained using Newtonian physics, and some using quantum mechanics, whichever can convey the message more clearly. Although this might be confusing at times, it seems to be unavoidable. In any case, we'll try to keep it straight-forward. MRI is a powerful imaging modality because of its flexibility and sensitivity to a broad range of tissue properties. One of the original reasons for the excitement about MRI was, and continues to be, its relative safety, where the 'noninvasive' nature of the magnetic fields employed makes it possible to diagnose conditions of people of almost any age. Today MRI also offers great promise in understanding much more about the human body, both its form and function.

Magnetic Resonance Imaging:

The Name MRI stems from the application of nuclear magnetic resonance (NMR) to radiological imaging. The adjective 'magnetic' refers to use of an assortment of magnetic fields and 'resonance' refers the need to match the (radio)frequency of an oscillating field to the 'professional' frequency of the spin of some nucleus (hence the 'nuclear') in a tissue molecule. It might be more accurate to refer to this field as NMRI rather than MRI, but there is widespread concern over any phrase containing the word 'nuclear.' Although the nuclear component simply refers to a benign role of the 'spin' of the nucleus in the process, the word has been suppressed and the public and the profession have embraced the MRI acronym.

The Origin of Magnetic Resonance Imaging:

To describe the history of any technological advance in a given field is a very difficult. Still, the beginning student may be aided and inspired by even a short historical discussion. The concept of nuclear magnetic resonance had its underpinnings with the discovery of the spin nature of the proton. Learning on the work of Stern and Gerlach from the early 1920's, Rabi and coworkers pursued the spin of the proton and its interaction with a magnetic field in the 1930's. With this foundation in hand in 1946, Bloch and Purcell extended these early quantum mechanical concepts to a measurement of an effect of the precession of the spins around a magnetic field. These gentlemen successfully measure a precessional signal from a water sample and a paraffin sample, respectively. It may be said that MRI had its beginnings in 1973 with the seminal papers by Lauterbur and Mansfield. It was already well known that the intrinsic angular momentum (or 'spin') of a hydrogen nucleus (the proton) in a magnetic field precesses about that field at the 'Larmor frequency' which in

turn, depends linearly on the magnitude of the field itself. Their idea was very simple. If a spatially varying magnetic field is introduced across the object, the Larmor frequencies are also spatially varying. They proposed and showed that the different frequency components of the signal could be separated to give spatial information about the object. This key point of spatially encoding the data opened the door to MR imaging. Others also recognized the importance of this area, with early attention brought to tumor detection by Damadian.

Historical Development of MRI:

A brief developmental history is given below which proven to be the milestone in the journey of MRI.

1946 Elucidation of NMR phenomena and technique—Bloch, Purcell

1951 Single dimension spatial localization—Gabillard

1952 Nobel Prize to Bloch and Purcell

1959 Blood flow by NMR—Singer

1971 In vitro cancer detection by NMR—Damadian

1972 In vivo cancer detection by NMR—Weisman

1972 NMR imaging—Damadian.

1.2 OBJECTIVES OF THE STUDY:

1.2.1 Electromagnetism of MRI

Atomic Structure:

The atom consists of a central nucleus and orbiting electrons. The nucleus contains nucleons which are subdivided into protons and neutrons; protons are positively charged, neutrons have no net charge, and electrons are negatively charged. The atom is electrically stable if the number of negatively charged electrons orbiting the nucleus equals the number of positively charged protons in the nucleus. Atoms that are electrically unstable due to a deficit, or an excess number of electrons, are called ions.

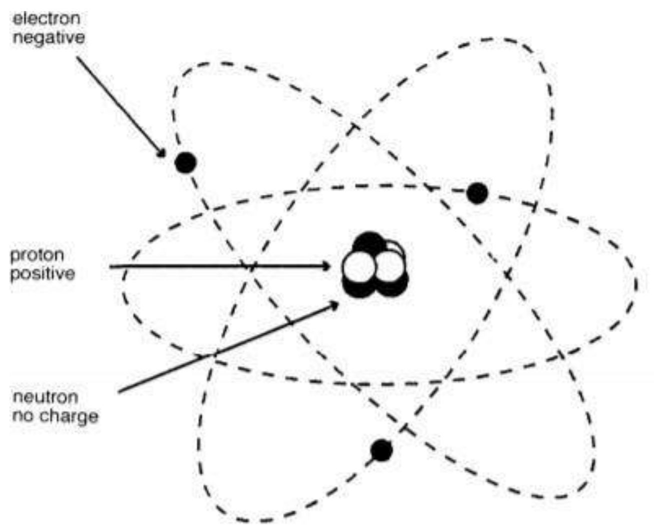


Figure 1.1: Atomic Structure.

Motion within the Atom:

Three types of motion are present within the atom. These are:

(1) Electrons spinning on their own axis,

(2) Electrons orbiting the nucleus

(3) The nucleus itself, spinning about its own axis.

The principles of MRI rely on the spinning motion of specific nuclei present in biological tissues. These are known as MR active nuclei.

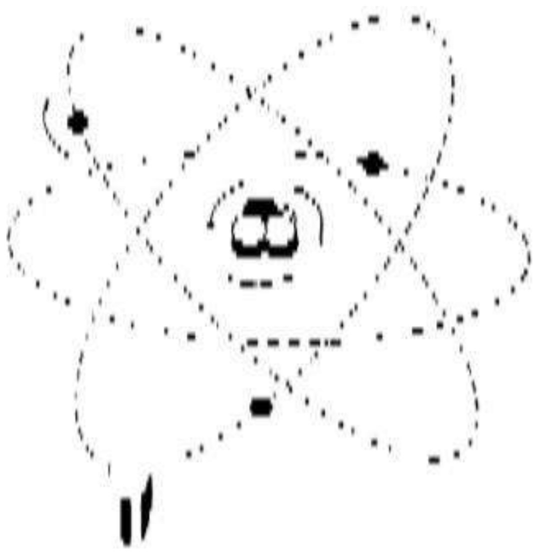


Figure 1.2: Motion of particles in the Atom.

MR Active Nuclei:

MR active nuclei are characterized by their tendency to align their axis of rotation to an applied magnetic field. Due to the laws of electromagnetic induction, nuclei that have a net charge and are spinning acquire a magnetic moment and are able to align with an external magnetic field. This occurs if the mass number is odd, i.e. there are even number of neutrons and odd number of protons or vice versa. The process of this interaction is angular momentum or spin.

Important examples of MR active nuclei are 1 H, 13C, 15N, 17O, 19F, 23Na and 31P.

Alignment can be measured as the total of the nuclear magnetic moments and is expressed as a vector sum. The strength of the total magnetic moment is specific to every nucleus and determines the sensitivity to magnetic resonance.

The Hydrogen Nucleus:

The hydrogen nucleus is the MR active nucleus used in clinical MRI. The hydrogen nucleus contains a single proton. It is used because it is very abundant in the human body, and because its solitary proton gives it a relatively large magnetic moment.

The Hydrogen Nucleus as a Magnet:

The laws of electromagnetism state that a magnetic field is created when a charge particle moves. The hydrogen nucleus has a magnetic field, because of a spinning positively charged proton that acts as a small magnet. The magnet of each hydrogen nucleus has in effect a north and a south pole of equal strength. The north/south axis of each nucleus is represented by a magnetic moment. The magnetic moment of each nucleus has vector properties, i.e. it has size and direction and is denoted by an arrow. The direction of the vector designates the direction of the magnetic moment, and the length of the vector designates the size of the magnetic moment.

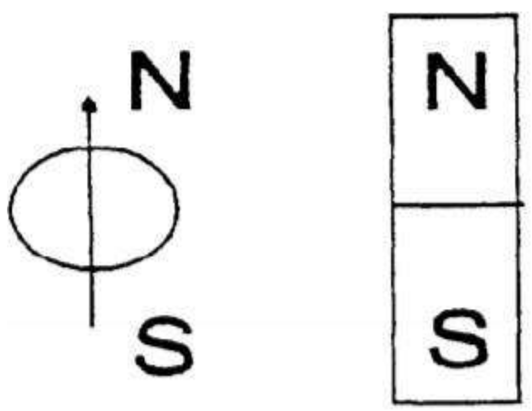


Figure 1.3: The magnetic moment of hydrogen nucleus.

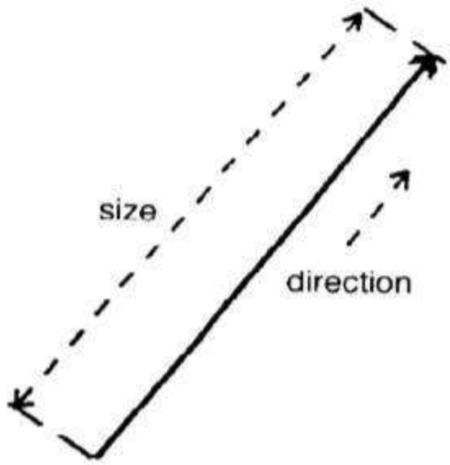


Figure 1.4: The magnetic vector.

Electromagnetic Waves:

To understand MRI, we first need to understand what electromagnetic waves are. All electromagnetic waves have certain fundamental properties in common:

(i) They all travel at the speed of light c = 3 × 108 m/sec.

(ii) They all have two components—an electric field E and a magnetic field B— that are perpendicular to each other and both are traveling at the speed of light (c).

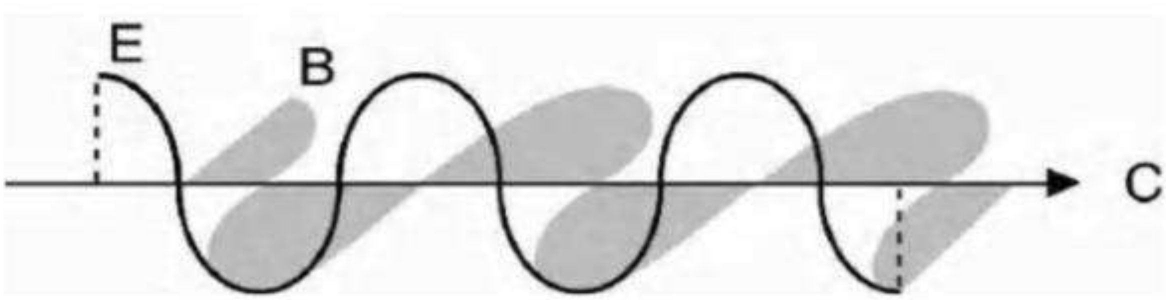


Figure 1.5: Two components of an electromagnetic wave, the electric component E and the magnetic component B. These two components are perpendicular to each other, are 90° out of phase, and travel at the speed of light (C).

(iii) We are interested in the magnetic field component—the electric field component is undesirable because it generates heat.

In MRI, we deal with much lower energies and lower frequencies than X-ray or even visible light. The wave lengths are much longer in the radio frequency (RF) window. Table 2.1 contains a few examples of frequency ranges in the electromagnetic spectrum.

AM radio frequency	0.54-1.6 MHz (540-1600 kHz)
TV (Channel 2)	Slightly over 64 MHz
FM radio frequency	88.8-108.8 MHz
RF used in MRI	3-100 MHz

Table 1.1: Frequency ranges in the EM spectrum.

This is why the electromagnetic pulse used in MRI to get a signal is called an RF (radio frequency) pulse—it is in the radio frequency range. It belongs to the radio frequency window of the electromagnetic spectrum.

Fundamental Interaction of a Proton Spin with the Magnetic Field:

MRI is based on the interaction of a nuclear spin with an external magnetic field, B0. The dominant nucleus in MRI is the proton in hydrogen and its interaction with the external field results in the precession of the proton spin about the field direction. Imaging of humans rests on the ability to manipulate, with a combination of magnetic fields, and then detect, the bulk precession of the hydrogen spins in water, fat and other organic molecules.

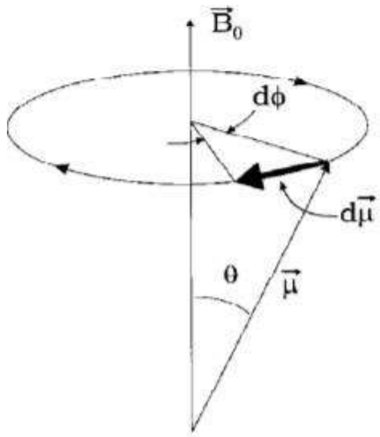


Figure 1.6: Precession of the proton spins about magnetic field direction.

By definition, precession is the circular motion of the axis of rotation of a spinning body about another fixed axis caused by the application of a torque in the direction of the precession. The interaction of the proton's spin with the magnetic field produces the torque, causing it to precess about B0 as the fixed axis. When looking down from above the vector B0, the precession of the magnetic moment vector μ, which is proportional to the spin vector, is clockwise. For the customary counterclockwise definition of polar angles, the differential dφ shown is negative.

It thus possesses an effective loop of electric current around the same axis about which it is spinning.

This effective current loop is capable of interacting with external magnetic fields as well as producing its own magnetic field. We describe the strength with which the loop produces its own field, in terms of the same 'magnetic dipole moment' vector µ. The direction of this vector is nothing other than the spin axis itself and, like a compass needle; the magnetic moment vector will tend to align itself along any external static magnetic field, B0. It can also be found that the precession angular frequency for the proton magnetic moment vector (and for the spin axis as well) is given by ω0 = B0

Where is a constant called the gyromagnetic ratio. In water, the hydrogen proton has a value of roughly 2.68 x 106 Gy/s/Tesla. For a 2 T field, for example, the spins precess at a radiofrequency of 85.2 MHz, just below the FM range for radio broadcasting. This precession frequency is referred to as the Larmor frequency and is referred to as the Larmor equation.

Relaxation

Relaxation is the process whereby nuclear magnetization returns to its resting state following a perturbation, such as by an RF pulse. Magnetic resonance is characterized by three forms of relaxation, T1 (longitudinal or spin-lattice) relaxation, T2 (transverse of spin-spin) relaxation, and T2* relaxation.

T1 Relaxation

The return of excited nuclei from the high energy state to the low energy or ground state is associated with loss of energy to the surrounding nuclei. Nuclear magnetic resonance was originally used to examine solids in the form of lattices, hence the name "spinlattice" relaxation. Macroscopically, T1 relaxation is characterized by the longitudinal return of the net magnetization to its ground state of maximum length in the direction of the main magnetic field. The rate of return is an exponential process as is shown in the following figure.

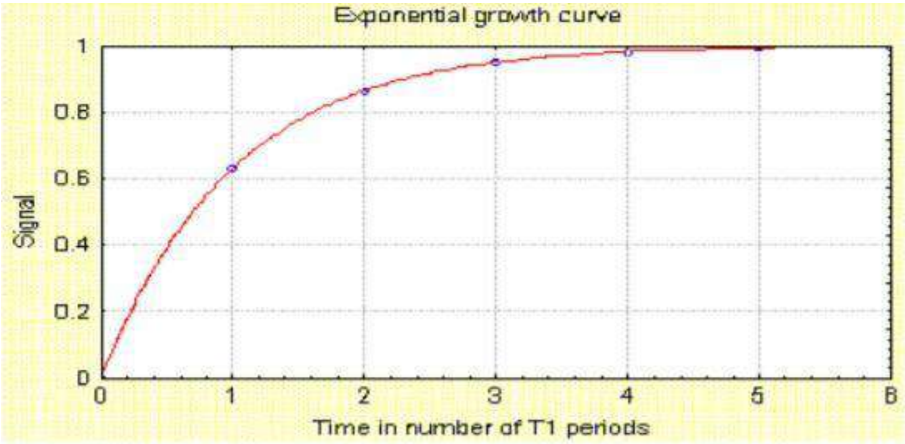


Figure 1.7: T1 relaxation graph.

The T1 relaxation time is the time for the magnetization to return to 63% of its original length. After two T1 times, the magnetization is at 86% of its original length. Three T1 times results in 95% recovery. Spins are considered completely relaxed after 3-5 T1 times. Another term that you may hear is the T1 relaxation rate. This is merely the reciprocal of the T1 time(1/T1). T1 relaxation is fastest when the motion of the nucleus (rotations and translations) matches that of the Larmor frequency. As a result, T1 relaxation is dependent on the main magnetic field strength which specifies the Larmor frequency. Higher magnetic fields are associated with longer T1 times.

T2 Relaxation

Microscopically, T2 relaxation or spin-spin relaxation occurs when spins in the high and low energy state exchange energy but do not loose energy to the surrounding lattice. 10 Macroscopically, this results in loss of the transverse magnetization. In pure water, The T2 and T1 times are approximately the same, 2-3 seconds. In biological materials, the T2 time is considerably shorter than the T1 time. T2 relaxation occurs exponentially like T1 relaxation with 63% of the transverse magnetization gone after one T2 period as shown in the graph.

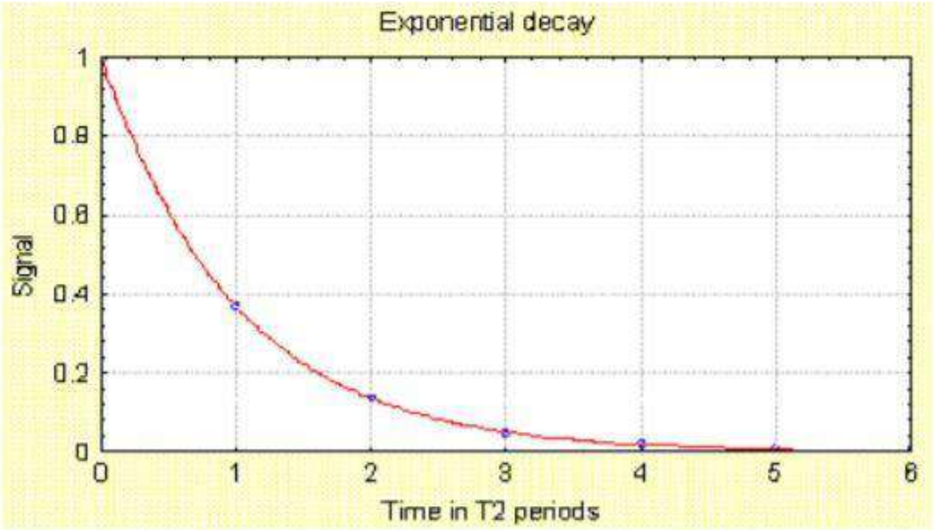


Figure 1.8: T2 relaxation graph.

T2* Relaxation

T2* relaxation is the loss of signal seen with dephasing of individual magnetizations. Macroscopically, it is characterized by a loss of transverse magnetization at a rate greater than T2. T2* relaxation is caused by the magnetic field inhomogeneity that is found in all magnets. Unlike spin echo sequences, gradient echo sequences do not refocus T2* decay. Therefore, gradient echo sequences are more susceptible to magnetic field inhomogeneity caused by ferromagnetic foreign bodies.

Magnetic Susceptibility

The intensity of magnetization, I, is related to the strength of the inducing magnetic field, H, through a constant of proportionality k, known as the magnetic susceptibility. $I = kH$

The magnetic susceptibility is a unit less constant that is determined by the physical properties of the magnetic material. It can take on either positive or negative values. Positive values imply that the induced magnetic field I is in the same direction as the inducing field, H. Negative values imply that the induced magnetic field is in the opposite direction as the inducing field.

Magnetic susceptibility represents a material's tendency to distort a magnetic field. Paramagnetic materials like gadolinium have some electrons that tend to line up with the magnetic field, producing an additive effect. Supermagnetic substances like hemosiderin more strongly attract magnetic lines of force. Ferromagnetic substances (e.g., iron alloys) remain permanently magnetized after being removed from a magnetic field.

Magnetism

Magnetism is a property of matter that is a result of the orbiting electrons in atoms. The orbiting electrons cause the atoms to have a magnetic moment associated with an intrinsic angular momentum called 'spin'. Magnetic field strengths are measured in units of Tesla (T). We will run across four terms describing the magnetic properties of materials, such as contrast agents, used in MRI. These terms are ferromagnetism, paramagnetism, superparamagnetism, and diamagnetism.

Ferromagnetism

Ferromagnetic materials generally contain iron, nickel, or cobalt. These materials have a large positive magnetic susceptibility, i.e., when placed in a magnet field, the field strength is much stronger inside the material than outside. Ferromagnetic materials are also characterized by being made up of clusters of 1017 to 1021 atoms called magnetic domains, that all have their magnetic moments pointing in the same direction. The moment of the domains is random in unmagnetized materials, and point in the same direction in magnetized materials. Figure 2.9 illustrates the effect of a ferromagnetic material (grey circle) on the magnetic field flux lines (blue). The ability to remain magnetized when an external magnetic field is removed is a distinguishing factor compared to paramagnetic, superparamagnetic, and diamagnetic materials. On MR images, these materials cause susceptibility artifacts characterized by loss of signal and spatial distortion.

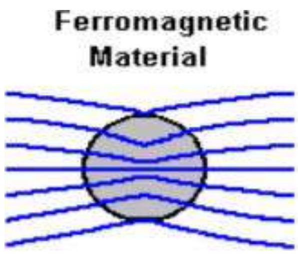


Figure 1.9: Ferromagnetism.

Para magnetism

Paramagnetic materials include oxygen and ions of various metals like Fe, Mg, and Gd. These ions have unpaired electrons, resulting in a positive magnetic susceptibility. The magnitude of this susceptibility is less than one one-thousand of that of ferromagnetic materials. The effect on MRI is increase in the T1 and T2 relaxation rates (decrease in the T1 and T2 times). Figure 2.10 illustrates the effect of a paramagnetic material (grey circle) on the magnetic field flux lines (blue). Gd is used as an MR contrast agent. At the proper concentration, Gd contrast agents cause preferential T1 relaxation enhancement, causing increase in signal on T1-weighted images. At high concentrations, as is sometimes seen in the urinary bladder, loss of signal is seen instead, a result of the T2 relaxation effects dominating.

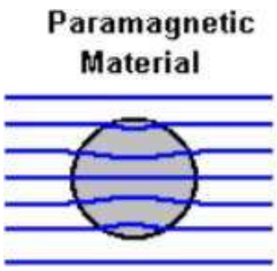


Figure 1.10: Paramagnetism.

Superparamagnetism

Superparamagnetic materials consist of individual domains of elements that have ferromagnetic properties in bulk. Their magnetic susceptibility is between that of ferromagnetic and paramagnetic materials. Figure illustrates the effect of a superparamagnetic material (grey circle) on the magnetic field flux lines (blue). Examples of superparamagnetic materials include iron containing contrast agents for bowel, liver, and lymph node imaging.

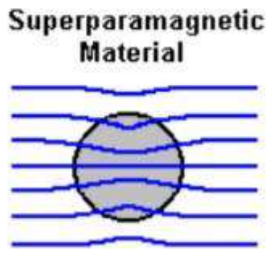
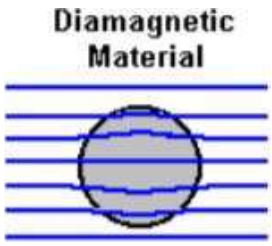


Figure 1.11: Superparamagnetism.

Diamagnetism

weak negative magnetic susceptibility contributes to the loss of signal seen in bowel on MRI after administration of ba Diamagnetic materials have no intrinsic atomic magnetic moment, but when placed in a magnetic field weakly repel the field, resulting in a small negative magnetic susceptibility. Materials like water, copper, nitrogen, barium sulfate, and most tissues are diamagnetic. Figure 2.12 illustrates the effect of a diamagnetic material (grey circle) on the magnetic field flux lines (blue). The rium sulfate suspensions.



1.12.DIAMAGNETIC MATERIAL

1.2.2 Hardware Considerations:

Resonance and RF Protons in a magnetic field have a microscopic magnetization and act like tiny toy tops that wobble as they spin. The rate of the wobbling or precession is the resonant or Larmor frequency. In the magnetic field of an MRI scanner at room temperature, there is approximately the same number of proton nuclei aligned with the main magnetic field Bo as counter-aligned. The aligned position is slightly favored, as the nucleus is at a lower energy in this position. For every one-million nuclei, there is about one extra nucleus aligned with the Bo field as opposed to the field. This results in a net or macroscopic magnetization pointing in the direction of the main magnetic field. Exposure of individual nuclei to RF radiation (B1 field) at the Larmor frequency causes nuclei in the lower energy state to jump into the higher energy state.

On a macroscopic level, exposure of an object or person to RF radiation at the Larmor frequency causes the net magnetization to spiral away from the Bo field. In the rotating frame of reference, the net magnetization vector rotates from a longitudinal position to an angle proportional to the length of time of the RF pulse. After a certain length of time, the net magnetization vector rotates 90 degrees and lies in the transverse or x-y plane. It is in this position that the net magnetization can be detected by an MRI receiver coil. The angle of rotation of the net magnetization vector is commonly called the "flip" or "tip" angle. At angles greater than or less than 90 degrees there will still be a component of the magnetization that will be in the x-y plane, and therefore be detectable.

RF and Gradient Coils

I have grouped these two topics together, although they serve different purposes.

RF Coils

RF coils are the "antenna" of the MRI system that broadcasts the RF signal to the patient and/or receives the return signal. RF coils can be receive-only, in which case the body coil is used as a transmitter; or transmit and receive (transceiver).

Surface Coils

Surface coils are the simplest design of coil. They are simply a loop of wire, either circular or rectangular, that is placed over the region of interest. The depth of the image of a surface coil is generally limited to about one radius. Surface coils are commonly used for spines, shoulders, and other relatively small body parts.

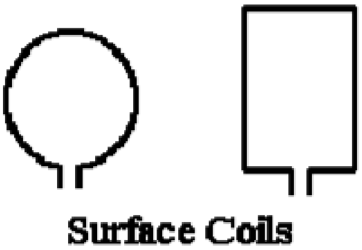


Figure 1.13: Surface Coils.

Paired Saddle Coils

Paired saddle coils are commonly used for imaging of the knee. These coils provide better homogeneity of the RF in the area of interest and are used as volume coils, unlike surface coils. By running current in opposite directions in the two halves of the gradient coil, the magnetic field is made stronger near one and weaker near the other.

Figure 1.14: Paired Saddle Coil.

The Helmholtz Pair Coils

The Helmholtz pair coils consist of two circular coils parallel to each other. They are used as z gradient coils in MRI scanners. They are also used occasionally as RF coils for pelvis imaging and cervical spine imaging.

Figure 1.15: Helmholtz Pair Coil.

The Bird Cage Coil

The bird cage coil provides the best RF homogeneity of all the RF coils. It has the appearance of a bird cage; hence, its name. This coil is commonly used as a transceiver coil for imaging of the head. This type of coil is also used occasionally for imaging of the extremities, such as the knees.

Figure 1.16: Bird Cage Coil.

Gradient Coils

Gradient coils are used to produce deliberate variations in the main magnetic field (Bo). There are usually three sets of gradient coils, one for each direction. The variation in the magnetic field permits localization of image slices as well as phase encoding and frequency encoding. The set of gradient coils for the z axis are Helmholtz pairs, and for the x and y axes, paired saddle coils.

1.2.3 Producing a Magnetic Resonance Signal

Spatial Encoding of the MR Signal

As the tissue protons are perturbed and then realign with the main magnetic field, they produce characteristic radiofrequency signals. This phenomenon is the basis for nuclear magnetic spectroscopy. To produce images however, this information must be capable of being localized. This is achieved by spatially encoding the magnetic spins. The most common way of achieving spatial encoding is to employ two-dimensional Fourier transform (2DFT) imaging. By applying incremental changes across the magnetic field, three orthogonal axes are imposed on the system as follows:

1. **Slice selection**: One gradient is applied at the same time as the RF pulse to selectively excite the spins in a specific slice.
2. **Frequency encoding:** A second gradient encodes location along one in plane dimension based on the frequency of the MR signal.
3. **Phase encoding:** A third gradient encodes along the other in plane dimension based on the phase differences of the MR signal.

These three gradients define three orthogonal planes and they may be aligned arbitrarily to produce any desired oblique projection.

Image Reconstruction

The raw data generated by most sequences is separated into phase and frequency components by a mathematical procedure called Fourier transformation. To generate an anatomic image, the raw data must be inverse Fourier transformed. The Fourier co domain of the image space is referred to as the k-space. k- and image spaces have an identical matrix size and contain equivalent imaging information in terms of spatial resolution and contrast. An exact one-to-one relationship does not exist, however, as the signal contains information from the entire image. The k-space data points are ordered according to the phase and frequency conferred on them by the spatial encoding process. The data points nearest the center of k-space contain the strongest signal elements because they are the least dephased by spatial encoding gradients. In contrast, data points at the periphery of k-space, though weaker, are responsible for reproducing the finer details in the image.

Pulse Sequences

There are many pulse sequences available for imaging (and spectroscopy). Listed below are two of the commonly used pulse sequences in MRI.

Spin Echo Sequence

The spin echo pulse sequence is the most commonly used pulse sequence. The pulse sequence timing can be adjusted to give T1-weighted, Proton or spin density, and T2- weighted images. Dual echo and multiecho sequences can be used to obtain both proton density and T2-weighted image simultaneously. The two variables of interest in spin echo sequences is the repetition time (TR) and the echo time (TE). All spin echo sequences include a slice selective 90 degree pulse followed by one or more 180 degree refocusing pulses as shown in Figure.

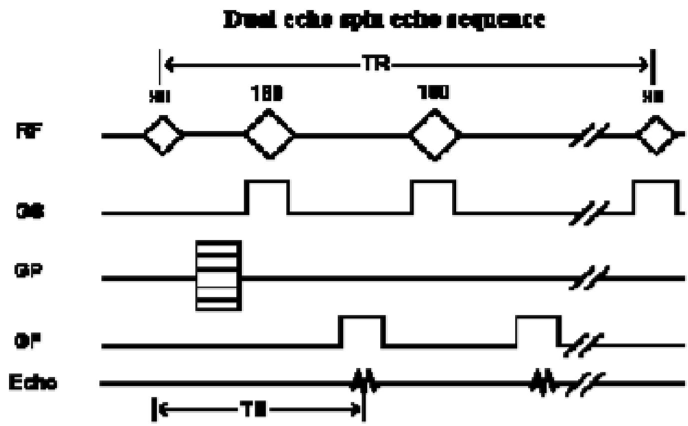


Figure 1.17: Dual echo spin echo sequence.

In this diagram, RF is the radio frequency pulse, a slice selective 90 degree pulse followed by two 180 degree refocusing pulses. GS, GP, and GF are the slice selective, phase encoding, and frequency encoding gradients, respectively. "Echo" represents the signal received from the slice of interest in the body. A short TR an short TE will give a T1-weighted image, a long TR and short TE (first echo) will give a proton density image, and a long TR and long TE (second echo) will give a T2-weighted image. An example of the changes in the net magnetization vector for a spin echo sequence with one echo is shown below. After 1/2 of the TE time, dephasing of individual spins in the volume occurs as represented by the black and red arrows. The 180 degree RF pulse flips these spins to the opposite side in the x-y plane so that after another 1/2 TE, they refocus and are detected.

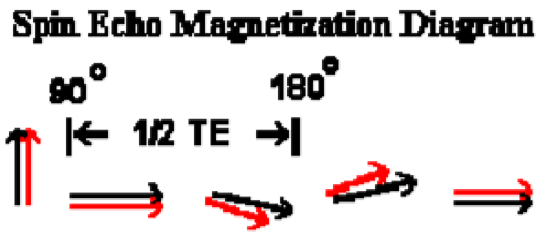


Figure 1.18: Spin Echo Magnetization Diagram.

Inversion Recovery Sequence

Inversion recovery pulse sequences are used to give heavy T1-weighting. The basic part of an inversion recovery sequence is a 180 degree RF pulse that inverts the magnetization followed by a 90 degree RF pulse that brings the residual longitudinal magnetization into the x-y or transverse plane where it can be detected by an RF coil. In imaging, the signal is usually refocused with a 180 degree pulse as in a spin echo sequence. The time between the initial 180 degree pulse and the 90 degree pulse is the inversion time (TI). A diagram of the sequence is shown below.

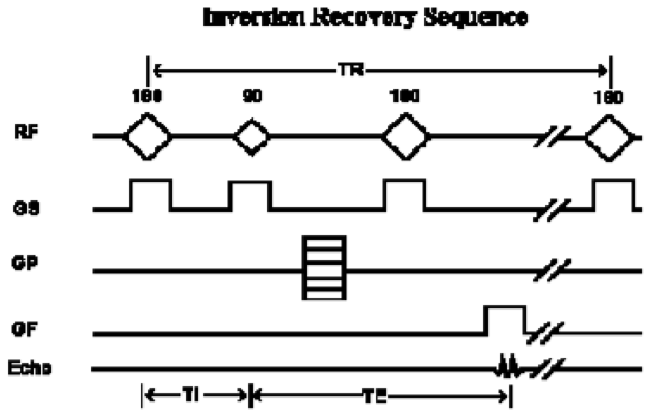


Figure 1.19: Inversion recovery sequence.

With a TI of about 140 ms on a 1.5 T MRI machine, the fat signal is nulled while the water proton signal is still present. This occurs because the T1 of fat is significantly smaller than the T1 of water. The diagram below shows the magnetization of water (black arrow) and fat (red arrow) during the STIR sequence used for fat suppression. One drawback of this sequence is the partial loss of proton signal during the TI time. Also the TR time must be longer than that of a spin echo sequence for recovery of longitudinal magnetization.

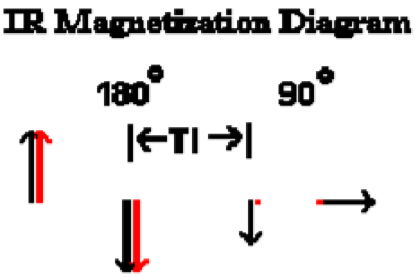


Figure 1.20: IR Magnetization Diagram.

1.2.4 - MRI Artifacts

There are many kinds of artifacts that can occur in MRI. Some affect the quality of the MRI exam while others do not affect the quality but may be confused with pathology.

Phase-Encoded Motion Artifacts

Phase-encoded motion artifacts appear as bright noise or repeating densities oriented in the phase direction, occurring as the results of motion during acquisition of a sequence. These artifacts may be seen from arterial pulsations, swallowing, breathing, peristalsis, and physical movement of a patient. Phase-encoded artifacts can be reduced by various techniques depending on their cause and location. Surface coil localization can reduce artifacts generated at a distance from the area of interest. The following axial image of the head shows a phase-encoded motion artifact running transversely across the back of the head as a result of venous flow in the transverse sinuses.

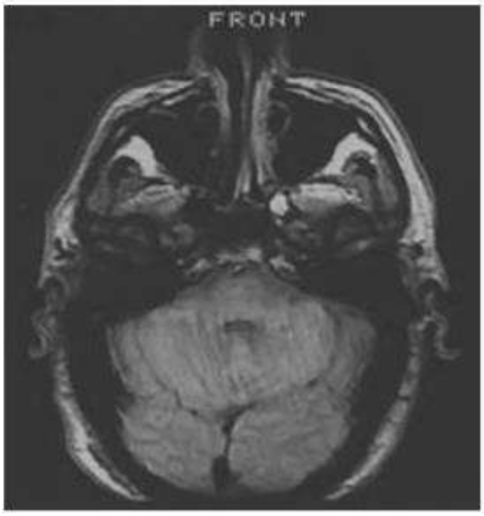


Figure 1.21: Axial image of the head showing a phase-encoded motion artifact running transversely across the back of the head.

Aliasing or "Wrap-around”

Aliasing or wrap-around is a common artifact that occurs when the field of view (FOV) is smaller than the body part being imaged. The part of the body that lies beyond the edge of the FOV is projected on to the other side of the image. This can be corrected, if necessary, by oversampling the data. The following axial images of the brain demonstrate this artifact. Figure shows wraparound of the back of the head on to the front of the head, where the phase-encoded direction is anterior-posterior. Figure has the phase and frequency directions reversed resulting in absence of the aliasing artifact. Oversampling was used in the frequency direction to eliminate the aliasing.

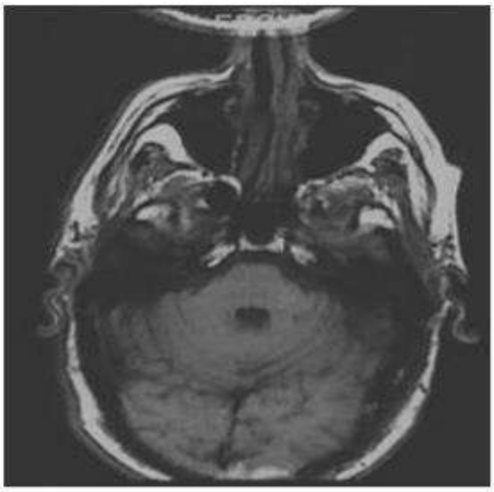


Figure 1.22: Axial images of the brain demonstrate Aliasing or wrap-around artifact.

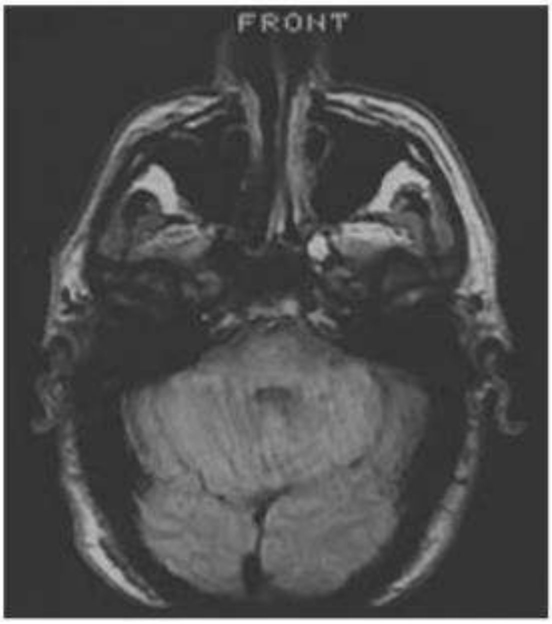


Figure 1.23: Oversampling used in the frequency direction to eliminate the aliasing.

Moirés Fringes

Moirés fringes are an interference pattern most commonly seen when doing gradient echo images with the body coil as shown in the figure 5.4A.

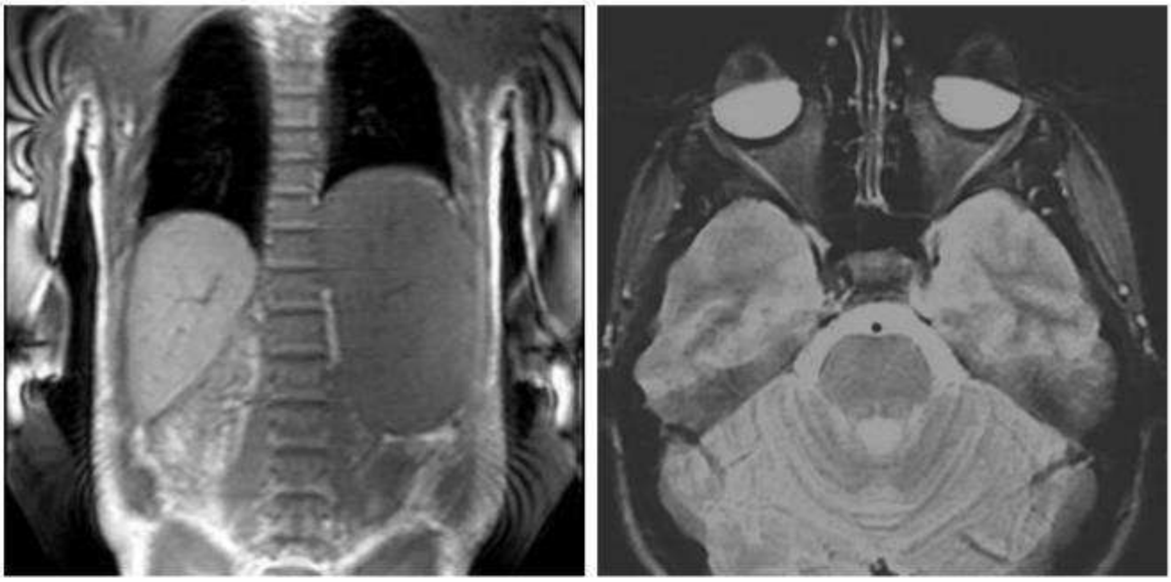


Figure 1.24: A: Moirés fringes in image of one side of the body. B: Susceptibility artifacts in axial MRI of the head in a patient with mascara on her eyelids.

Because of lack of perfect homogeneity of the main magnetic field from one side of the body to the other, aliasing of one side of the body to the other results in superimposition of signals of different phases that alternatively add and cancel. This causes the banding appearance and is similar to the effect of looking though two screen windows.

Susceptibility Artifacts

Susceptibility artifacts occur as the result of microscopic gradients or variations in the magnetic field strength that occurs near the interfaces of substance of different magnetic susceptibility. Large susceptibility artifacts are commonly seen surrounding ferromagnetic objects inside of diamagnetic materials (such as the human body). These gradients cause dephasing of spins and frequency shifts of the surrounding tissues. The net result is bright and dark areas with spatial distortion of surrounding anatomy. These artifacts are worst with long echo times and with gradient echo sequences. Figure 5.4B is showing an axial MRI of the head in a patient with mascara on her eyelids. Susceptibility artifacts from the mascara obscure the front half of the globes.

Practical Aspects of Magnetic Resonance Imaging

Biologic Effects

Three areas have received attention in terms of the biologic effects of MRI. The first area concerned the use of strong static magnetic fields (Bo). The primary theoretical concerns had to do with the effects of the magnetic field on the protons and electrons within molecules that could affect chemical bonding or biochemical reactions. Experiments have been conducted both in vitro as well as in vivo for cellular preparations, tissues, and organisms. The conclusion from these data is that no deleterious biologic effects have been demonstrated for magnetic field strengths commonly found in MRI. Currently, 8 T is accepted as the limit below which no deleterious biologic effects have been demonstrated.

The second area of concern centered on the use of time-varying magnetic fields (gradient fields). Changing magnetic fields can induce electric currents in biologic tissues. The two tissues of primary concern are the heart and the nervous system. In addition, it is possible to stimulate the retina with an appropriate gradient. Current induction is dependent upon the rate of magnetic field change (dB/dt). In the systems studied, the thresholds for these biological effects are sufficiently high that a biological stimulation does not occur for the gradient fields typically found in MRI, usually less than 400 T/s.

The third area of concern centered on the biological effects associated with the use of radio frequency radiation. This field of investigation has been the subject of scientific inquiry (apart from MRI) for quite a long time. The primary biologic effect of interest is that of heating due to the oscillation of the atoms and molecules stimulated by the radio frequency radiation. The amount of heating is dependent on several factors, including the RF frequency, the absorption of the RF energy, and the thermoregulatory mechanisms within humans.

Although a rise in body temperature is possible, it is usually not a problem in patients whose cardiovascular or thermoregulatory systems are not compromised. Hence, RF heating is not deemed a contraindication for the use of routine clinical MRI

Safety

Safety can be divided up into three areas:

Main Magnetic Field and Safety

The main magnetic field of a 1.5 T magnet is about 30,000 times the strength of the earth's magnetic field. It is strong enough to pull fork-lift tines off of machinery, pull heavy-duty floor buffers and mop buckets into the bore of the magnet, pull stretchers across the room and turn oxygen bottles into flying projectiles. Deaths have occurred from trauma as a result of these effects. Smaller objects such as pagers, bobby pins and pens have been known to be pulled off the person carrying them.

Some types of heart valves are torqued in a magnetic field: however, this torque is less than the stresses that occur normally as a result of blood flow. Therefore, heart valves are now considered not to be an absolute contraindication for MRI. More of an annoyance than a safety problem is the ability of the magnetic field of a MRI machine to erase the information contained on the magnetic strip on ATM and credit cards. This may occur a short distance inside of the scanner room of a MRI machine.

Varying Magnetic (Gradient) Fields

Varying magnetic fields are necessary in order to obtain images from MRI scanners. Changing magnetic field induce electrical currents in conductors. In patients with metal in their body, the potential exists for electrical currents being induced in the metal with subsequent heating. This may occur with metal foreign bodies or some surgical implants. It does not universally occur and patients may be scanned without harm. Very rapidly changing magnetic fields as may be achieved with echo planar imaging can cause nerve stimulation. This stimulation can affect motor nerves with resulting muscle contraction as well as the retina with resulting flickering lights.

Radio Frequency

The radio frequency power that is capable of being produced matches that of many small radio stations (15-20 kW). As a result there is the presence of heating effects from the RF. In most pulse sequences, the heating is insignificant and does not exceed the FDA guidelines. New pulse sequences such as for echo planar imaging and some spectroscopy localization techniques are capable of exceeding the FDA guidelines. Monitoring of the power deposition in patients is a requirement for FDA approval of clinical MRI scanners. Potential for electrical shock exists with RF coils so proper grounding and insulation of coils is necessary. Any damage to coils or their cables needs prompt attention. Also looping of the cable to a coil can result in burns to patients that come into contact with them. It is best to avoid all contact with the RF coil cables.

1.3 SCOPE OF THE STUDY:

- **EURASIP** journal on image and video processing is intended for researchers from both acasemia and industry, who are active in the multidisciplinary field of image and video processing

- The scope of the journal covers all therotical and practical aspect of the domain from, basic research to development of applicaton

- Contributed articles on image and video processing may be focused an specific techniques

- The classification of image processing comopression, recognition, restoration, and segmention. With in the context of various medical aaerial robitcs, security , communications arts are employing divers data format

- The image processing we have diifernt format black and white gary scale ,color, multi spectral , infra red ,video,stereo,3-d

1.4 MATERIAL REQUIREMENT

1.4.1 SOFTWARE REQUIREMENT:

MAT LAB SOFTWARE R2013B

PRESONAL COMPUTER

1.5 PROCUMENT OF EQUIPMENT:

Searchig mat lab programming ………………

Assembling it…………….

CHAPTER-2
SOFTWARE USED

2.1 Introduction to MATLAB:

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.

- Typical uses include
- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN. The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory

2.2 The MATLAB system:

The MATLAB system consists of five main parts

- **Development Environment:**

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library:**

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language:**

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both “programming in the small” to rapidly create quick and dirty throw-away programs, and “programming in the large” to create large and complex application programs.

Graphics:

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

The MATLAB Application Program Interface (API):

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files. Various toolboxes are there in MATLAB for computing recognition techniques, but we are using IMAGE PROCESSING toolbox.

2.3. GRAPHICAL USER INTERFACE (GUI):

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based Mfunction is executed.
- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed. To launch GUIDE from the MATLAB command window, type guide filename Where filename is the name of an existing FIG-file on the current path. If filename is omitted, GUIDE opens a new (i.e., blank) window.

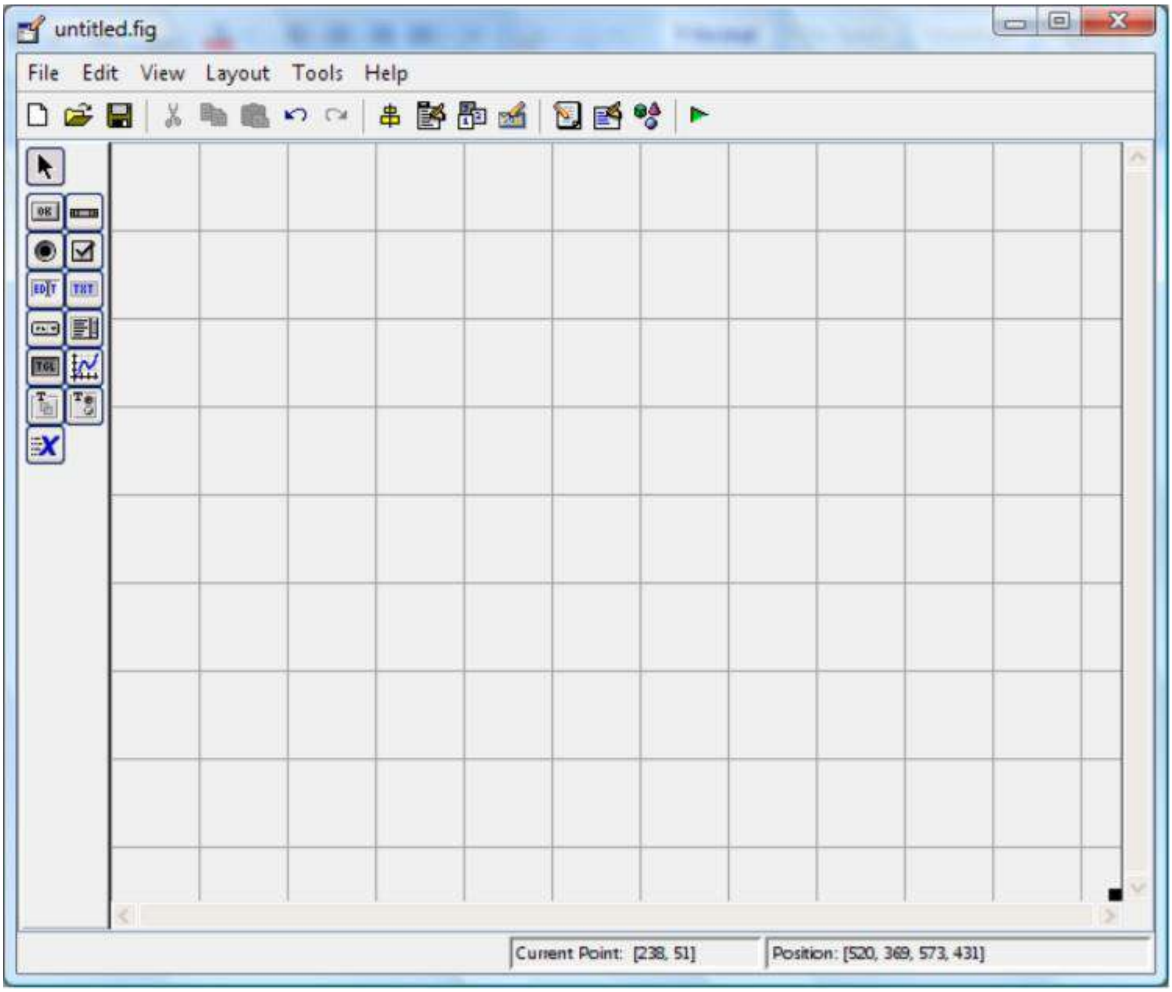


Figure 2.1 matlab procedure

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed. GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots

. **2.4 Getting Started:**

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos. At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

2.4.1 Introduction - describes the components of the MATLAB system.

2.4.2 Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

2.4.3 Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

2.4.4 Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

2.4.5 Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

2.5 DEVELOPMENT ENVIRONMENT

2.5.1Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

2.5.2 Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop. On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop. You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

2.5.3 Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

2.5.4 MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

2.5.5 Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools.

The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacsmagik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents. To open the Help browser, click the help button in the toolbar, or type help browser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go. Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find Mfiles and other MATLAB-related files, which are organized in directories on your file system.

Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos. To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function. The workspace is not maintained after you end the MATLAB session.

To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to runMATLAB functions. The Editor/Debugger provides a graphical user interface for basic textediting, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint

CHAPTER-3

PROJECT DESIGN

3.1 Anisotropic diffusion filters :

A large and growing body of literature has investigated about enhancement methods. Image enhancement deals with improving the quality of images, where the objective is to emphasize wanted features and make them less obscured. The area of image enhancement is very appealing, where many fundamental image enhancement techniques are developed based on a very simple concept. In the recent decades, image enhancement based on contrast enhancement has been focused

Many researchers argued that Histogram equalization (HE) is a simple and an easy method to enhance the contrast and improve the image quality. Since 1997, Yeong Kim raised several concerns about contrast problem and suggested Brightness preserving Bi-Histogram Equalization (BBHE) in order to enhance the contrast. The average intensity value was applied as a separating point to differentiate between a dark area and bright area. The above finding contradicts the study by Wang et al. The author presented that a median intensity value is more accurate as the separating point compared to the average intensity. These results were contradicted that suggested the minimum mean brightness between original and output image as the separating point is more specific and accurate compared to the BBHE and Dualistic Sub Image Histogram Equalization (DSIHE).

Research conducted by Ooi and Isa proposed a new improvement in histogram equalization known as Quadrant Dynamic Histogram Equalization (QDHE). The first step in this technique was to divide the histogram into four sub-quadrant histograms based on the median value of the original image. After normalizing each sub-histogram, finally, the image was equalized. A major advantage of QDHE is that it's enhanced the image without any intensity saturation, noise amplification, and over-enhancement. In 2010, Ooi et al. presented a new method based on Plateau level equation, namely Bi-Histogram Equalization with a Plateau Level (BHEPL). The main objective of this paper is to improve the BBHE technique in term of processing time.

The process of this method also involved mean brightness preserving histogram equalization method with a clipped histogram equalization method. However, interestingly, this is contrary to a study conducted by Sengee et al. They suggest an extension method of BBHE based on the Neighbourhood Metric. This method involved a few steps: First, a large histogram was divided into the sub-region using Neighbourhood Metric. Second, based on mean, the histogram of the original image was separated into two sub-regions and process independently. The results enhanced the local contrast and preserved the brightness of the original image. The comparison of result performance is illustrated in Figure 3.1.

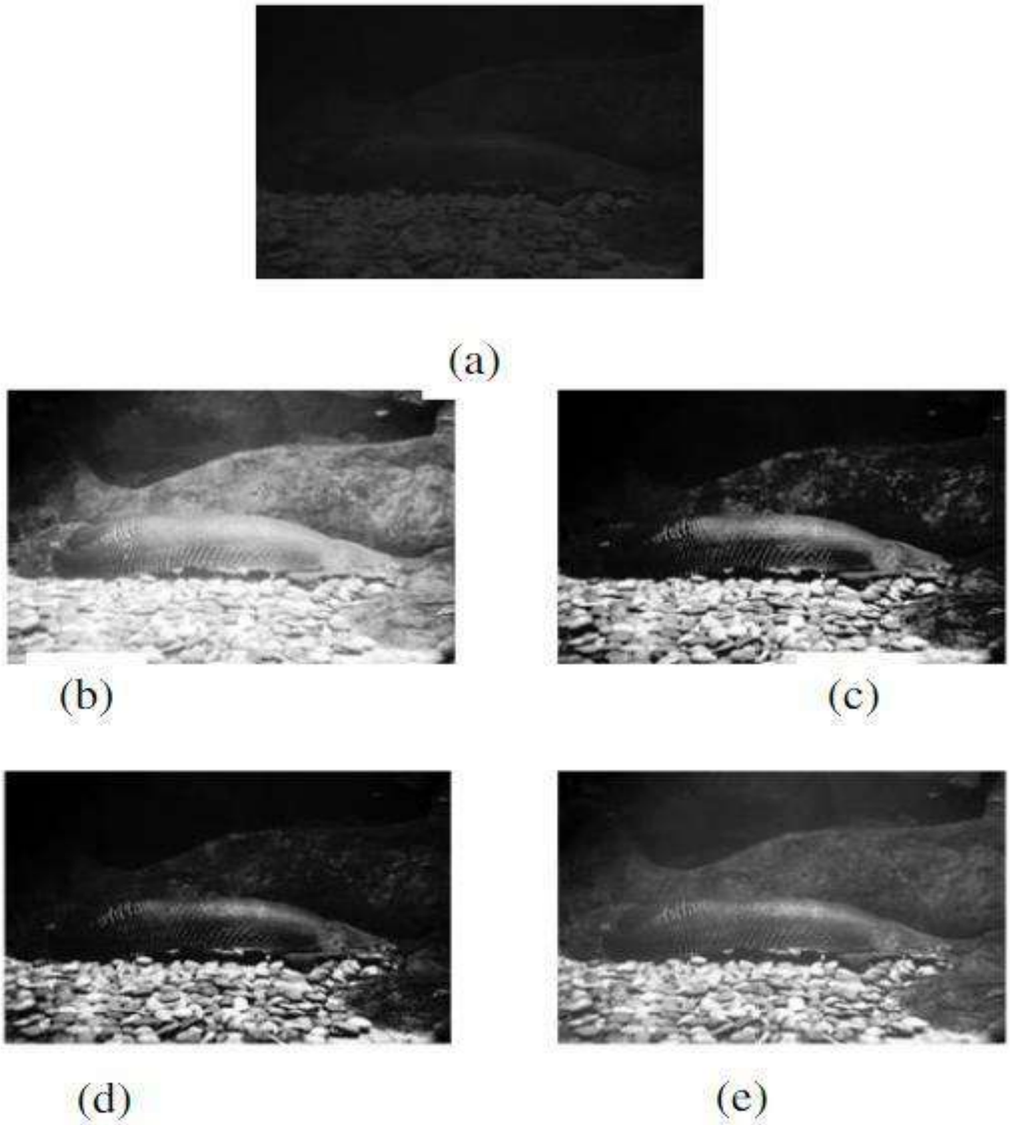


Figure 3.1. The comparison of enhancement result based on histogram modification: (a) Original image, (b)HE, (c) BBHE, (d) BHEPL, and (e) QDHE.

In a different study, Salah et al. explained a new approach to solving the illumination problem on the face images using Histogram Equalization (HE). The technique is based on the combination of gamma correction and the Retinal filter's compression function namely GAMMA-HM-COMP. The Retinal filter is a new enhancement method, and the result was effective compared to the three conventional enhancement methods which are histogram equalization, gamma correction and log transformation. In another study, Tan et al. proposed a Background Brightness Preserving Histogram Equalization (BBPHE) method based on non-linear Histogram equalization (HE). Based on the background and non-background level techniques, the original image was separated into three interval histograms:

(1) low grey level,
(2) medium grey level, and
(3) high grey level.

The objective of this method is to enhance the object contrast and maintaining the background brightness. Similarly, Moniruzzaman et al. proposed a modification of Brightness Preserving Bi-Histogram Equalization (BPBHE) using the edge pixels data. In order to prove the effectiveness, the Average Mean Brightness Error (AMBE) was calculated and the result was presented in table 3.1. The lowest of AMBE shows the high quality image and good performance technique.

Image	HE	BBHE	BPBHE
A	24.3350	29.7404	3.4737
B	42.0428	35.4111	2.5079
C	53.9887	13.6073	13.6073
D	11.386	8.5763	3.3188

Table 3.1. The AMBE result on different histogram technique.

Hashemi et al. proposed a novel enhancement method based on Genetic Algorithm using a simple chromosome structure and corresponding operator. The method was tested on the image that has a low dynamic range. In 2013, research finding by Chaudhary and Patil also suggested a simple method based on Genetic Algorithm. The advantages of both methods are fast processing time, efficient, and produce a high-quality image. Besides that, they also produced a comprehensive comparison between a BBHE, DSIHE, MMBBHE, MPHE, and RMSHE. The analyses were done based on PSNR and contrast ratio.

In another study, Shome et al. examined a method using Contrast Limited Adaptive Histogram Equalization (CLAHE) in order to normalize the contrast variation in the retinal image. CLAHE is an adaptive extension of Histogram Equalization followed by thresholding, which helps in the dynamic preservation of the local contrast features of an image. This proposed method used a non-mean based approach to improve the quality of the Diabetic Retinopathy (DR) image while preserving the sharpness and minutes of the details. The method also increased the local contrast pixels. The above view was supported studied by Sundaram et al.

where a CLAHE technique with a slight modification was suggested in order to enhance the mammogram images. The method was known Histogram ModifiedHashemi et al. proposed a novel enhancement method based on Genetic Algorithm using a simple chromosome structure and corresponding operator.

The method was tested on the image that has a low dynamic range. In 2013, research finding by Chaudhary and Patil also suggested a simple method based on Genetic Algorithm. The advantages of both methods are fast processing time, efficient, and produce a high-quality image. Besides that, they also produced a comprehensive comparison between a BBHE, DSIHE, MMBBHE, MPHE, and RMSHE. The analyses were done based on PSNR and contrast ratio.

In another study, Shome et al. examined a method using Contrast Limited Adaptive Histogram Equalization (CLAHE) in order to normalize the contrast variation in the retinal image. CLAHE is an adaptive extension of Histogram Equalization followed by thresholding, which helps in the dynamic preservation of the local contrast features of an image. This proposed method used a non-mean based approach to improve the quality of the Diabetic Retinopathy (DR) image while preserving the sharpness and minutes of the details. The method also increased the local contrast pixels. The above view was supported studied by Sundaram et al. where a CLAHE technique with a slight modification was suggested in order to enhance the mammogram images. The method was known Histogram ModifiedContrast Limited Adaptive Histogram Equalization (HM-CLAHE). The optimization technique was used to adjust the level of strong contrast and the local details for more relevant interpretation. Based on Enhancement Measure (EME) result, this method was better compared to the HE, Unsharp masking (USM), and CLAHE. Tan et al. introduced a modification of HE based on non-linear technique.

The aim is to enhance the image contrast, while preserving the background brightness for images with well-defined background brightness. The original image was divided into three sub-images by using the proposed algorithm. Finally, only the problem region was normalized. The correction process based on sub-images technique also was supported by Shanmugavadivu and Balasubramanian. They proposed a new method called as Thresholded and Optimized Histogram Equalization (TOHE). The main process has divided the histogram using the Otsu thresholding. Based on the result performance, this approaches is successful compared to the HE, BBHE, Range-Limited Bi-Histogram Equalization (RLBHE). Figure 3.2 shows the comparison result between TOHE and a few histogram methods. Circle in figure 3.2 represent the improvement and drawback of each method.

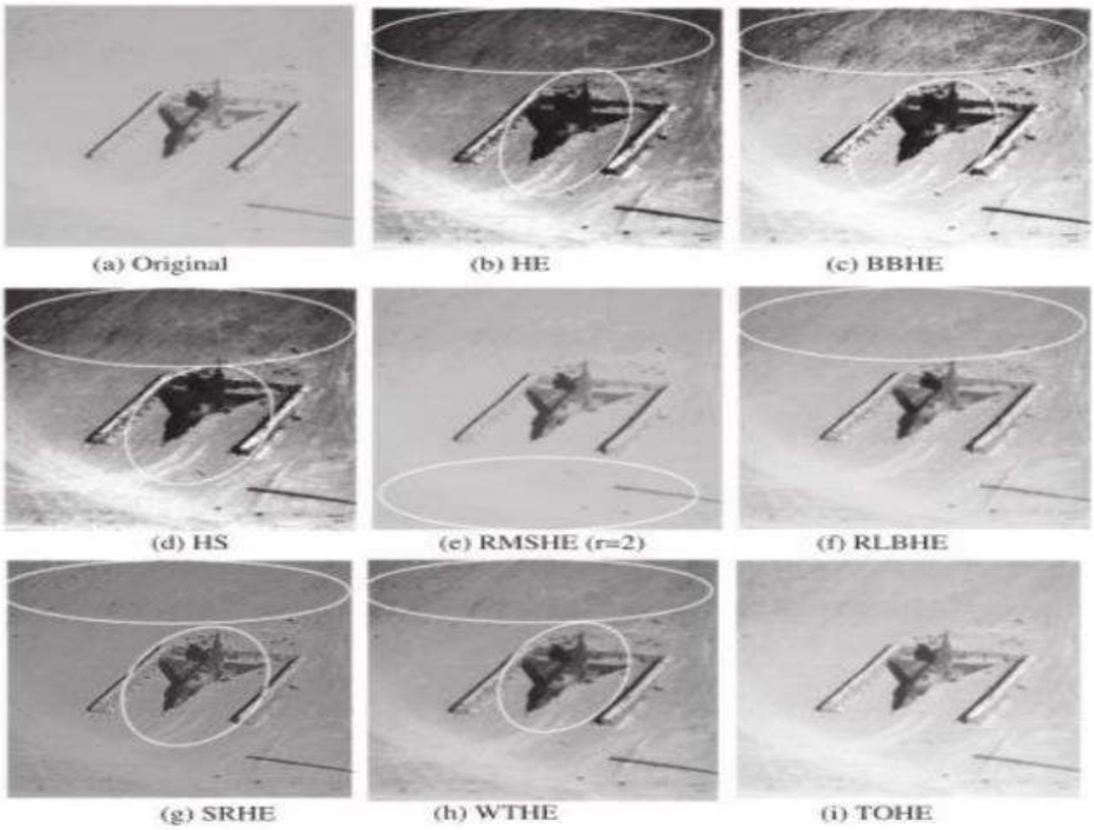


Figure 3.2. Comparison of the result performance after applying a different histogram process.

On the other side, Lin et al. [33] suggested a new HE method was applied to the colour image known as Averaging Histogram Equalization (AVHEQ). The technique also separates the original image into sub-images and equalized independently. A new mathematical algorithm in order to determine the optimal averaging threshold was proposed. The result is better compared to the conventional methods such as BBHE, DSIHE, and BHEPL.

Contrast-enhancement is an essential pre-processing step towards efficient clinical diagnosis of MR images. Various HE-based contrast-enhancement techniques are reported in literature. The main motivation of these variants HE is to preserve the best features of the classical HE technique in terms of brightness and contrast, while introducing modifications that will overcome the disadvantages associated to HE. In the next subsections we are going to study and describe some of these HE-based techniques.

3.2 Adaptive histogram equalization (AHE):

For images which contain local regions of low contrast bright or dark regions, global histogram equalization won't work effectively. A modification of histogram equalization called the Adaptive Histogram Equalization can be used on such images for better results. Adaptive histogram equalization works by considering only small regions and based on their local cdf, performs contrast enhancement of those regions.

Adaptive histogram equalization can be implemented by various methods and each of those methods have multiple variation. For the purpose of this project I have implemented AHE using a "tiled windows with interpolated mapping" method as described in the paper - "Adaptive Histogram Equalization and its Variations" by Pizer.

This function is implemented in the program ahe.cpp by the function tiledAHE. The main steps performed by this function are:

1. Calculate a grid size based on the maximum dimension of the image. The minimum grid size is 32 pixels square.
2. If a window size is not specified chose the grid size as the default window size.
3. Identify grid points on the image, starting from top-left corner. Each grid point is separated by grid size pixels.
4. For each grid point calculate the cdf of the region around it, having area equal to window size and centered at the grid point.
5. After calculating the mappings for each grid point, repeat steps 6 to 8 for each pixel in the input image.
6. For each pixel find the four closest neighboring grid points that surround that pixel.
7. Using the intensity value of the pixel as an index, find its mapping at the four grid points based on their cdfs.
8. Interpolate among these values to get the mapping at the current pixel location. Map this intensity to the range [min:max] and put it in the output image.

I have noticed that the interpolation equation given in the paper is incorrect. It is using the inverse interpolating factors for interpolating between the mapped value which would give incorrect results.

The command line parameters for this program are:

```
$ ./ahe inImage bins [w:windowsize] [c:clipLevel]
```

where in Image is the input image, bins specifies the number of bins to use for calculating the cdfs, windowsize specifies the window size to use and clipLevel specifies histogram clipping level for contrast limited ahe which will be discussed in the next section. The default and minimum window size is equal to the grid size, which in turn depends on the maximum dimension of the image. Having the window size the as the grid size results in dividing the image into non-overlapping regions. The user may specify a larger window size if he wants to.

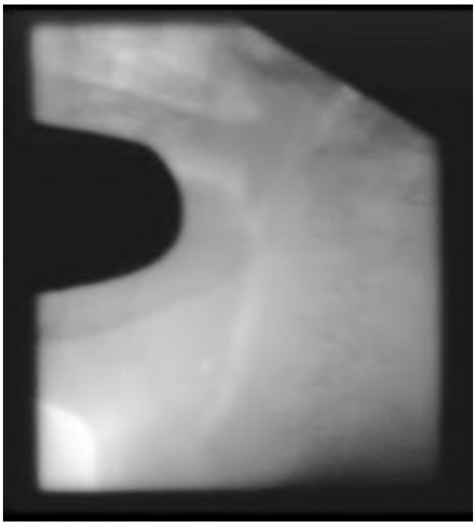


Figure 3.3: Input Image

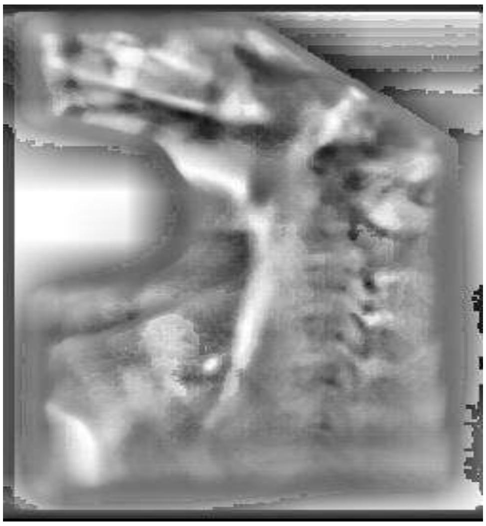


Figure 3.4: Output Image

Contrast-limited adaptive histogram equalization

While performing AHE if the region being processed has a relatively small intensity range then the noise in that region gets more enhanced. It can also cause some kind of artifacts to appear on those regions. To limit the appearance of such artifacts and noise, a modification of AHE called Contrast Limited AHE can be used. The amount of contrast enhancement for some intensity is directly proportional to the slope of the CDF function at that intensity level. Hence contrast enhancement can be limited by limiting the slope of the CDF. The slope of CDF at a bin location is determined by the height of the histogram for that bin. Therefore if we limit the height of the histogram to a certain level we can limit the slope of the CDF and hence the amount of contrast enhancement.

The only difference between regular AHE and CLAHE is that there is one extra step to clip the histogram before the computation of its CDF as the mapping function is performed. Hence CLAHE is implemented in the same function tiledAHE in ahe.cpp. The program "ahe" takes an additional optional parameter which specifies the level at which to clip the histogram. By default no clipping is performed. Valid values for clipping fall in the range from 1 to 1/bins.

Following is the overview of the algorithm for this function:

1. Calculate a grid size based on the maximum dimension of the image. The minimum grid size is 32 pixels square.
2. If a window size is not specified chose the grid size as the default window size.
3. Identify grid points on the image, starting from top-left corner. Each grid point is separated by grid size pixels.

4. For each grid point calculate the histogram of the region around it, having area equal to window size and centered at the grid point.
5. If a clipping level is specified clip the histogram computed above to that level and then use the new histogram to calculate the CDF.
6. After calculating the mappings for each grid point, repeat steps 6 to 8 for each pixel in the input image.
7. For each pixel find the four closest neighboring grid points that surround that pixel.
8. Using the intensity value of the pixel as an index, find its mapping at the four grid points based on their cdfs.
9. Interpolate among these values to get the mapping at the current pixel location. Map this intensity to the range [min:max) and put it in the output image.

Clipping the histogram itself is not quite straight forward because the excess after clipping has to be redistributed among the other bins, which might increase the level of the clipped histogram. Hence the clipping should be performed at a level lower than the specified clip level so that after redistribution the maximum histogram level is equal to the clip level.

To identify the point at which the clipping should be performed, I am using the binary search method as specified in the paper "Adaptive Histogram Equalization and its Variations". Following is an overview of the clipping algorithm.

1. Let the specified clip level be Top and 0 be the Bottom.
2. Until the difference between Top and Bottom become very small, perform the following steps:
    1. Calculate the Middle between Top and Bottom
    2. Find the sum of excess above Middle in each bin of the histogram
    3. If excess + Middle is greater than Clip level set Top = middle
    4. If excess + Middle is less than Clip level set Bottom = middle
    5. If excess + Middle is equal to Clip then Middle is the value at which clipping needs to be performed. Hence break out of the binary search loop.
3. Clip the histogram at the value of middle and redistribute the excess into each bin equally.

The function to clip the histogram and other histogram related functions are implemented in the C++ file - histogram.cpp.

The command line parameters for this program are:

$ ./ahe inImage bins [w:windowsize] [c:clipLevel]

where inImage is the input image, bins specifies the number of bins to use for calculating the cdfs, windowsize specifies the window size to use and clipLevel specifies histogram clipping level for contrast limited

CLAHE is only effective for images which contain relatively homogenous ere enhanced noise or artifacts may appear due to AHE.

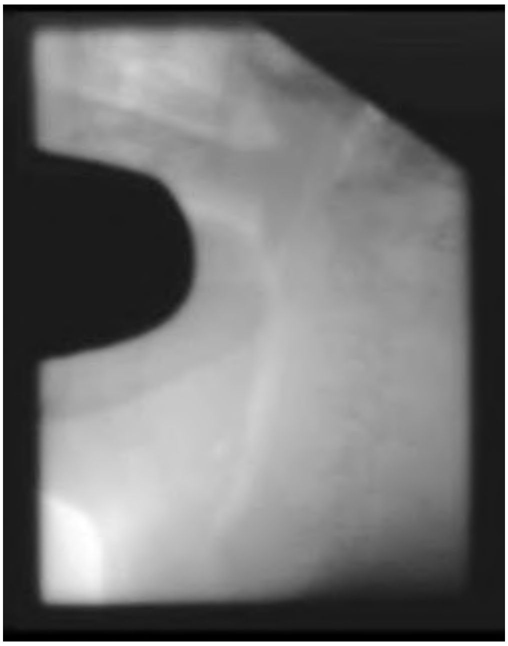


Figure 3.5: Input Image

Figure 3.6: Output Image

Brightness Preserving Dynamic Histogram Equalization (BPDHE)

Brightness preserving dynamic histogram equalization (BPDHE) which is proposed in this paper consists of five steps:

1. Smooth the histogram with Gaussian filter.
2. Detection of the location of local maximums from the smoothed histogram.
3. Map each partition into a new dynamic range.
4. Equalize each partition independently.
5. Normalize the image brightness.

Figure 3.7: Input Image

Figure 3.8: Output Image

Average Intensity Replacement based on Adaptive Histogram Equalization (AIR-AHE)

This study proposes an adaptive algorithm for brain MRI images called the Average Intensity Replacement based on Adaptive Histogram Equalization (AIR-AHE). AIR-AHE can automatically enhance the WMH region in FLAIR images. This work is served as the most important part of automatic segmentation of WMH on MRI (T2-WI and FLAIR) images in the brain at 1.5 T. The implementation of the proposed image enhancement algorithm for FLAIR sequences mode is chosen because it is capable of providing complimentary information of the WMH development as stated by Maillard et al. This algorithm combines several established enhancement algorithms (with some modification where needed), to improve the contrast of the image. The proposed algorithm consists of partial contrast stretching, contrast enhancement, window sliding neighborhood operation and new pixel centroid replacement. The sequences of MRI FLAIR images which have been used for segmentation have low contrast. Therefore, firstly, the contrast stretching is used to improve the quality of the image particularly the WMH region. After improving the quality of the image, the regions of high intensity are determined to represent the potential WMH area. Then, the Contrast-Limited Adaptive Histogram Equalization algorithm is used to enhance the contrast of each tile during the mapping of the WMH area to avoid amplifying any noise that might be present in the image, especially in the FLAIR image. A 3 by 3 window is used to scan the overall ROI to determine the average intensity among the local neighborhood pixels. Generally, the overall proposed algorithm procedure is as follows.

1. Load the image to be segmented.

2. Apply partial contrast stretching. Adjust the contrast in the image to setthe limits by saturating the upper and the lower contrasts by 1% respectively.

3. Apply the Contrast-Limited Adaptive Histogram Equalization algorithm with clipping limits in the range of 0–1.

4. Find the highest or maximum intensity regions of the WMH image.

5. Find the intensity of each pixel using 3 by 3 window sliding neighborhood operation. The new intensity for every pixel value of the center image is calculated based on the average value of the tile's window.

6. Use the new intensity pixel to update the center of the window while the other remaining pixels remain unchanged.

7. Repeat steps 5 to 7 for the sliding neighborhood operation to calculate and update all the pixels with a new calculated intensity value.

Figure 3.9: Input Image

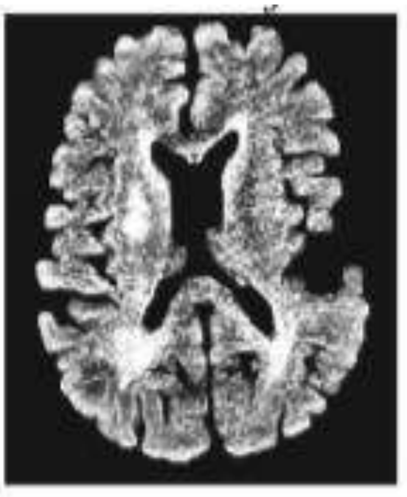


Figure 3.10: Output images

CHAPTER-4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM OF PROJECT:

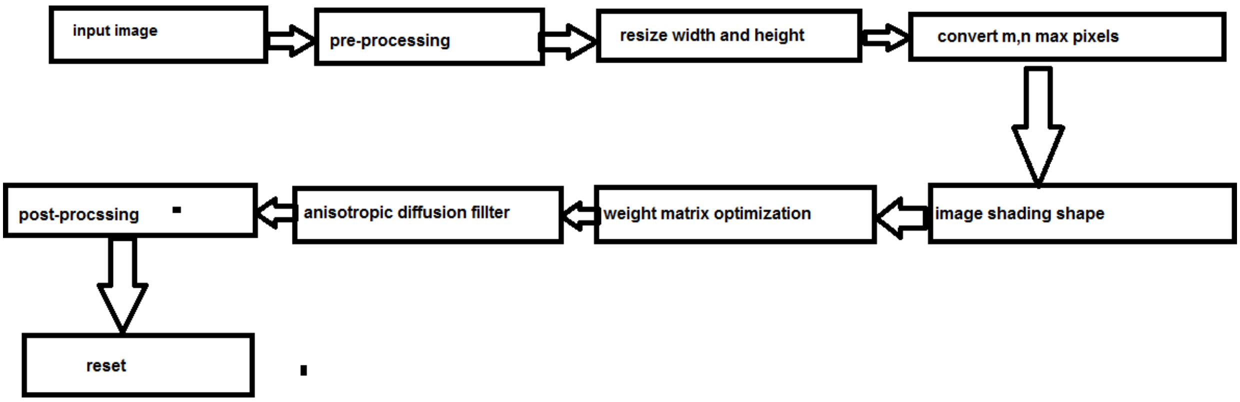


FIG:4.1 BLOCK DIAGRAM OF THE PROJECT

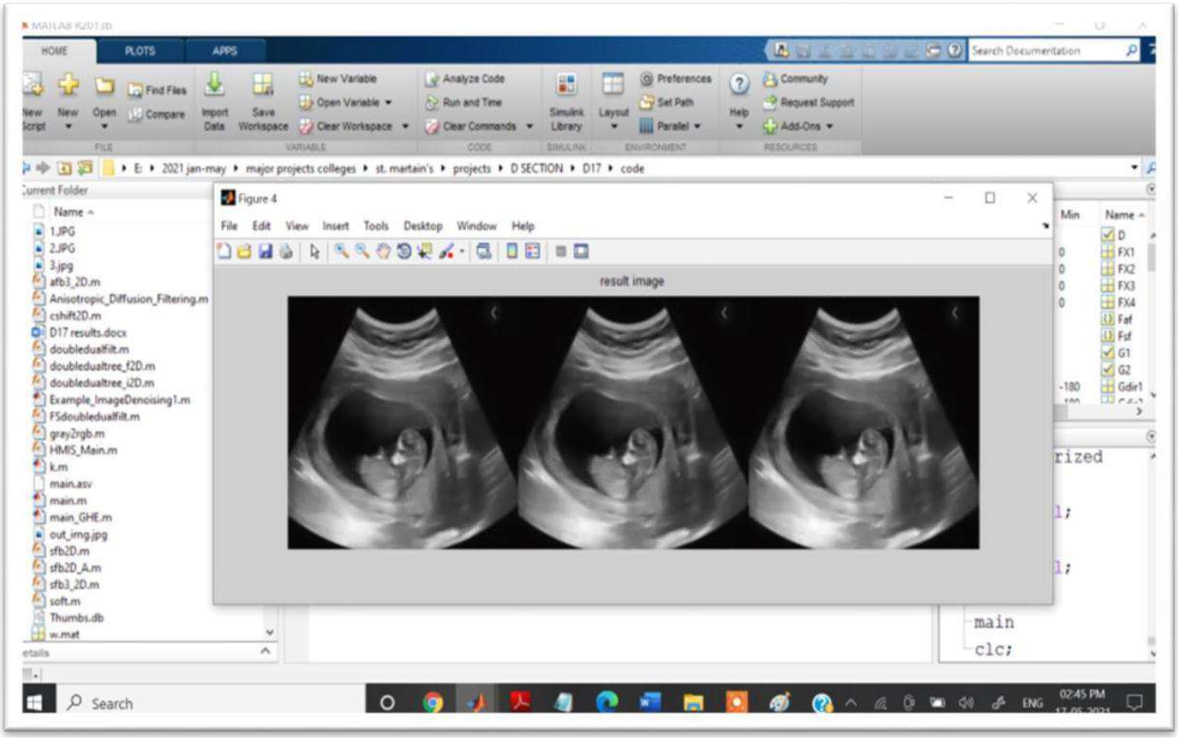
4.2 PROJECT CODE:

```
clc,
clear all,
close all,
warning off

[filename, pathname] = uigetfile({'*.png'},'pick an image');

if isequal(filename,0) || isequal(pathname,0)
   warndlg('File is not selected');
else
end
[pathstr,name,ext] = fileparts(filename);
filename11=char(filename);
I=imread(filename);
I=imresize(I,[256 256]);
nSig=imnoise(I,'poisson');
figure,imshow(nSig);title('Original image');
imwrite(nSig,'YO.jpg');

%%%%%%%%%%%%%%% Apply Discrete wavelet transform %%%%%%%%%%%%%%
[cA,cH,cV,cD]=dwt2(I ,'haar');
figure,subplot(2,2,1),imshow(mat2gray(cA));title('LL Image');
```

```
subplot(2,2,2),imshow(cH);title('LH Image');
subplot(2,2,3),imshow(cV);title('HL Image');
subplot(2,2,4),imshow(cD);title('HH Image');

%%%%%%%%%CALICULTE gradient vector for each DWT scale%%%%%%%%%%
if size(cA,3)==3
cA=rgb2gray(cA);
cH=rgb2gray(cH);
cV=rgb2gray(cV);
cD=rgb2gray(cD);
end
[FX1, Gdir1] = imgradient(cA,'prewitt');
figure,subplot(2,2,1),imshow(mat2gray(FX1));title('gradient LL comp');
[FX2, Gdir2] = imgradient(cH,'prewitt');
subplot(2,2,2),imshow(mat2gray(FX2));title('gradient LH comp');
[FX3, Gdir3] = imgradient(cV,'prewitt');
subplot(2,2,3),imshow(mat2gray(FX3));title('gradient HL comp');
[FX4, Gdir4] = imgradient(cD,'prewitt');
subplot(2,2,4),imshow(mat2gray(FX4));title('gradient HH comp');
a=imresize(I,[30 30]);
a = double(I);
[nr,nc]=size(I);
T = 128;
T0 =  .5;
G1 = I > T;
G2 = I <= T;
meanGL1 = mean(I(G1))
meanGL2 = mean(I(G2))
Tnew=(1/2) * (meanGL1 +meanGL2)
if (Tnew - T) < T0
end
k=0
I=nSig;
nSig=max(I);
k=6;
[Faf, Fsf] = FSdoubledualfilt;
[af, sf] = doubledualfilt;
im1 = double(imread('YO.jpg'));
im2 = double(imread('YO.jpg'));
% image decomposition
y = weight_matrix_optimization(im1,k);
w1=gray2rgb(y);
w1=num2cell(w1(:,1:7),2);
T = 6;                          % choose a threshold of 15
y = weight_matrix_optimization(im2,T);
w2=gray2rgb(y);
w2=num2cell(w2(:,1:7),2);
e_j=std(y);
% Image fusion process start here
for k1=1:k
   for  p=1:2
      for d1=1:2
         for d2=1:3
            k=6;
            x = w1{k};
```

```
          y = w2{k};
          x=sort(x);     %%%%%%%%%%% sorting or ordering rank coefficents
          D  = (abs(x)-abs(y)) >= 0;
          wf{k1}{p}{d1}{d2} = D.*x + (~D).*y; % image fusion
        end
      end
   end
end
load w
J=4;
y = Anisotropic_Diffusion_Filtering(w,J,Fsf,sf); % Anisotropic_Diffusion_Filtering
y=imadd(y,50);
y=double(y);
figure; imshow(mat2gray(y));title('result image');
imwrite(y,'out_img.jpg');
```

4.3 RESULTS:



4.3 FIG :FINAL OUTPUT

CHAPTER-5

PROJECT TESTING

5.1 OVERVIEW OF TESTING METHODS

Over the past several years, numerous MRI contrast enhancement and denoising approaches have been reported in the literature. This step of image preprocessing is very crucial for the efficiency of the subsequent image processing steps, for instance segmentation, classification and reconstruction. Among these approaches, one could report the linear filtering (spatial and temporal) and nonlinear filtering approaches, e.g. anisotropic diffusion filter (ADF), bilateral filtering. There are also the transform-based approaches, e.g. wavelet and curvelet transforms as well as statistical-based approaches e.g. maximum likelihood (ML), random forest and Markov Random Field (MRF), etc.

For MRI contrast enhancement, the histogram modificationsbased approaches have gained a great research interest by scientists due to their advantages: simplicity and ease of use, higher SNR as well as its low computational complexity. In fact, the most employed classical images contrast enhancement technique is the Histogram Equalization (HE), which provides good performance for ordinary images, such as human portraits or natural images. However, in MRI brain images, there are often more complex situations where the brain tumor White matter hyper intensities (WMH) signal is high but it may mistakenly be considered as other brain tissues such as Cerebrospinal fluid (CSF). In this paper, we will focus on the study of the major histogram modifications based approaches when applied to brain MRI images for contrast enhancement. Sazanita et al. proposed a new approach applied for FLAIR MRI images, known as the Average Intensity Replacement–Adaptive Histogram Equalization (AIR-AHE), which enhances the contrast without over amplifying the entire image. It improves the contrast of WMH relying on the intensity adjustment and contrast mapping techniques, more details of this method are explained in this paper.

The authors proposed a Hierarchical Correlation Histogram Analysis algorithm (HCHA) which provides an automatic contrast enhancement of images during the examination of atrophic cell areas of Parkinson Disease (PD) patients. This algorithm is based on the grayscale distribution degree of pixel intensity by constructing a correlation histogram. This method has as objective to provide the best segmentation accuracy result and facilitates subsequently CAD processes. SENTHILKUMARAN et al. performed a comparative study of different histogram-based techniques, mainly the histogram equalization (HE), BHE, modified BHE, AHE and CLAHE, in order to enhance the contrast of general MRI brain images. They used different evaluation metrics as Michelson contrast, RMS contrast, absolute mean brightness error (AMBE) and Pixel Distance. While the major comparative studies of histogram-based techniques are performed on general

MRI brain images, to the best our knowledge, there is no comparative study performed on MRI glioma brain tumors.

In fact, the main objective of the present comparative study is to find, over the existing recent contrast-enhancement methods, the most preferment and effective techniques for the MRI Glioma brain tumors images, specifically glioblastoma. Since glioblastomas are infiltrative tumors, their borders are often fuzzy and hard to distinguish from healthy tissues. Furthermore in some cases, the tumor region might share the same intensity profile with other normal regions within the image.These characteristics make the localization of tumor core edges a very difficult task for radiologists. Thus, the MRI contrast-enhancement is important on the one hand, it is a clinical necessity that facilitate, for radiologists and neurologists, the tumor detection and distinguish it from other tissues (edema and necrosis), and on the other hand, it makes easier for the CAD to have more efficiency and accuracy in different tasks such as automatic segmentation and even the classification task.

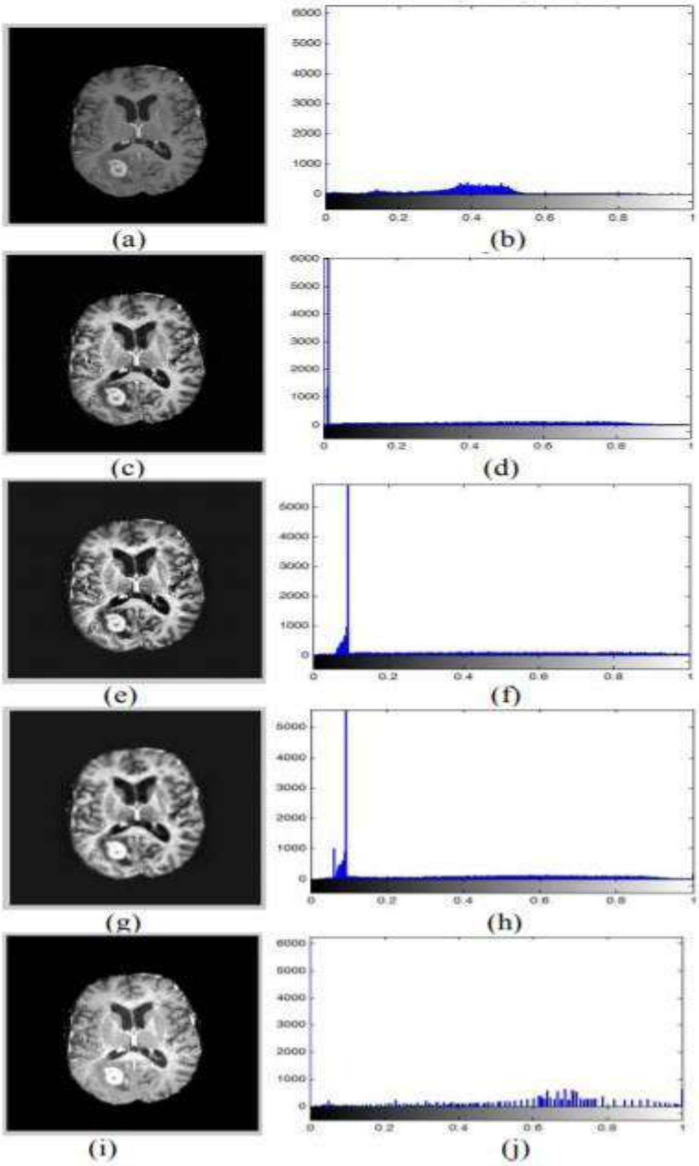


Fig. 5.1. Comparison of studied contrast enhancement techniques on Patient 2 of BRATS-2015 dataset: :(a) Original aquired image ; (b) histogram of original aquired image ; (c) AHE result ; (d) histogram of AHE (e) CLAHE result ; (f) histogram of CLAHE ; (g) AIR-AHE result ;(h) histogram of AIR-AHE (i) BPDHE result; (j) histogram of BPDHE

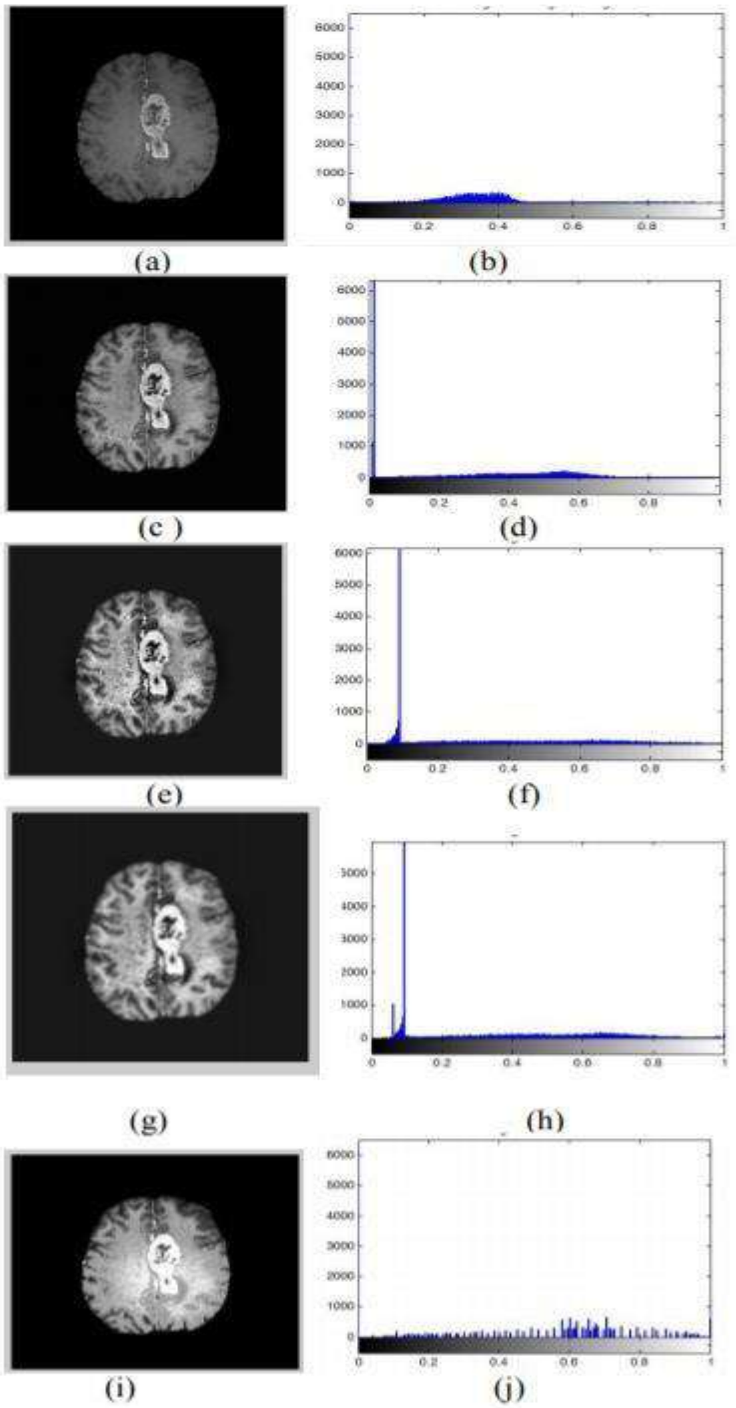


Fig. 5.2. Comparison of studied contrast enhancement techniques on Patient 07 from BRATS-2015 dataset: (a) Original aquired image ; (b) histogram of original aquired image ; (c) AHE result ; (d) histogram of AHE (e) CLAHE result ; (f) histogram of CLAHE ; (g) AIR-AHE result ;(h) histogram of AIR-AHE (i) BPDHE result; (j) histogram of BPDHE

CHAPTER-6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION:

- The main objective of this paper is to provide a comparative study of some existing techniques of contrast enhancement based on histogram equalization for MRI Glioblastoma brain tumor.

- we selected the most relevant slices where the tumor core appears clearly, then we computed the average value (with a standard deviation) of the quality evaluation metrics which makes the evaluation to be more precise.

- Through this study one could notice that, ADF technique provides efficient performances for MRI contrast enhancement compared to other studied techniques. For future works, we will focus on enhancing the AHE techniques by introducing filtering approaches that could improve the results in terms of accuracy and treatment efficiency

6.2 FUTURE ENHANCEMENT

- Future work includes the validation of the proposed method with a larger data base of the images.

- The use of the other mathods to evaluate the quality of the edges in the during the procee , this is the future enhancemrnt of the project.

PUBLICATION

Submitted Paper in the Conference ICISECT - 21 with Paper ID (ICISECT21-0030) and got

Acceptance for the Paper.

REFERENCES

[1] DANE, “Nacimiento y defunciones 2016-2017pr,” in Estadisticas vitales. DANE, mar 2018, p. 12.

[2] “Global health estimates 2016: Dalys by cause globally, 2016 and 2000.” World Health Organization, June 2018.

[3] Perona, P. and Malik, J., “Scale-Space and edge detection using anisotropic diffusion,” IEEE Transactions on pattern analysis and machine intelligence, vol. 12, no. 7, pp. 629–639, July 1990.

[4] Yu, Y. and Acton, S., “Speckle reducing Anisotropic diffusion,” IEEE Transactions on image processing, vol. 11, no. 1260-1270, pp. 629–639, November 2002.

[5] K.Z. Abd-Elmoniem, A.-B. Y. and Kadah, Y., “Real-time speckle reduction and coherence enhancement in ultrasound imaging via nonlinear anisotropic diffusion,” IEEE Transactions on Biomedical Engineering, vol. 49, no. 9, pp. 997–1014, Sept 2002.

[6] Mittal, D., Kumar, V., Saxena, S. C., Khandelwal, N., and Kalra, N., “Enhancement of the ultrasound images by modified anisotropic diffusion method,” Medical & biological engineering & computing, vol. 48, no. 12, pp. 1281–1291, 2010.

[7] Xu, J., Jia, Y., Shi, Z., and Pang, K., “An improved anisotropic diffusion filter with semi-adaptive threshold for edge preservation,” Signal Processing, vol. 119, pp. 80–91, 2016.

[8] Garg, A. and Khandelwal, V., “Combination of spatial domain filters for speckle noise reduction in ultrasound medical images,” Advances in Electrical and Electronic Engineering, vol. 15, no. 5, pp. 857–865, 2018.

[9] Weickert, J., “Coherence-enhancing diffusion of colour images1,” Image and Vision Computing, vol. 17, no. 3-4, pp. 201–212, 1999.

[10] Mrazek, P. and Navara, M., “Selection of optimal stopping time for ´ nonlinear diffusion filtering,” International Journal of Computer Vision, vol. 52, no. 2-3, pp. 189–203, 2003.

[11] Tsiotsios, C. and Petrou, M., “On the choice of the parameters for anisotropic diffusion in image processing,” Pattern recognition, vol. 46, no. 5, pp. 1369–1381, 2013.

BIBILOGRAPHY

Fernandez, J.-J. and Li, S., "An improved algorithm for anisotropic ´ nonlinear diffusion for denoising cryo-tomograms," Journal of structural biology, vol. 144, no. 1-2, pp. 152–161, 2003.

13 Giraldo-Guzman, J., Porto-Solano, O., Cadena-Bonfanti, A., and ´ Contreras-Ortiz, S. H., "Speckle reduction in echocardiography by temporal compounding and anisotropic diffusion filtering," in 10th International Symposium on Medical Information Processing and Analysis, vol. 9287. International Society for Optics and Photonics, 2015, p. 92871F.

14 Contreras-Ortiz, S. H. and Fox, M. D., "Hexagonal filters for ultrasound images," Journal of Electronic Imaging, vol. 23, no. 4, p. 043022, 2014.

15 Burckhardt, C. B., "Speckle in ultrasound b-mode scans," Sonics and Ultrasonics, IEEE Transactions on, vol. 25, no. 1, pp. 1–6, 1978.

16 Wang, Z. and Bovik, A. C., "A universal image quality index," IEEE signal processing letters, vol. 9, no. 3, pp. 81–84, 2002. 17

Jensen, J. A., "Field: A program for simulating ultrasound systems," Medical & Biological Engineering & Computing, vol. 34, no. sup. 1, pp. 351–353, 1997

A

MAJOR PROJECT REPORT

on

SYSTEM FOR MONITORING THE WATERASPECT AND LEAKS

Submitted by

1) Mr. K. Prem Nikit (17K81A04L5)
2) Mr. M. Shiva Vamshi (17K81A04N2)
3) Mr. P. Shashi Varma (17K81A04N6)

in partial fulfillment for the award of the degreeof

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Guidance of

Ms.K.Pritika

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500100

NBA &NAAC A+ Accredited

JUNE 2021

ST. MARTIN'S ENGINEERING COLLEGE
UGC AUTONOMOUS

Dhulapally, Secunderabad-500 100
NBA &NAAC A+ Accredited

BONAFIDE CERTIFICATE

This is to certify that the project entitled "System for Monitoring the Water Aspect and Leaks", is being submitted by **1. Mr. K. Prem Nikit 17K81A04L5, 2. Mr. M. Shiva Vamshi 17K81A04N2, 3. Mr. P. Shashi Varma 17K81A04N6** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Internal Guide

Ms. K. Pritika
Department of
Electronics and Communication
Engineering

Head of the Department

Dr. B. Hari Krishna
Department of
Electronics and Communication
Engineering

Internal Examiner

External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT KASI PREM NIKIT WITH ROLL NO.17K81A04L5, MUTYAMGARI SHIVA VAMSHI WITH ROLL NO.17K81A04N2, PETRAM SHASHI VARMA WITH ROLL NO.17K81A04N6, OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "SYSTEM FOR MONITORING THE WATER ASPECT AND LEAKS" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the students of '**Bachelor of Technology in Department of Electronics and Communication Engineering'**, session: 2017 – 2021, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled "**System For Monitoring The Water Aspect And Leaks**" is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in anyuniversity for award of any degree.

Mr.K.Prem Nikit	Mr.M.Shiva Vamshi	Mr.P.Shashi Varma
17K81A04L5	17K81A04N2	17K81A04N6

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

We would like to express our gratitude to **Dr.B.Hari Krishna**, **Head &Professor**, Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in our memory for years to come.

We would like to express our gratitude to our dept. project coordinator **Dr. A. Anand kumar, Professor** for his involvement with originality has triggered and nourished ourintellectual maturity that will help us for a long time to come.

We sincerely thank our **Class Project Coordinator, Ms. P. Pushpa,** Department of ECE and our project **Internal guide, Ms. K. Pritika, Assistant Professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level **Project Coordinators Dr. Maalyadri**, **Professor in IT** and **Dr. R. Santhosh Kumar**, **Associate Professor in CSE** for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. K. Prem Nikit (17K81A04L5)
2. M.Shiva Vamshi (17K81A04N2)
3. P.Shashi Varma (17K81A04N6)

ABSTRACT

For human life and the health of the environment, water is important. We have created an automated water quality management system for home, workplaces, etc. in order to achieve the good water quality needed by the citizens through IoT. Besides this, liquid level control systems are widely used for monitoring of liquid levels, reservoirs, silos, and dams etc.

To monitor the quality of water with the help of information sensed by the sensors immersed in water, so as to keep the water resource within a standard described for domestic usage and to be able to take necessary actions to restore the health of the degraded water body. Water pipelines leak detection, Pipeline systems are responsible for transporting vital materials such as water, oil and gas. We have used different sensors to design a device to calculate the water pH, the water pressure, flow, temperature etc. In this project, we have recommended the use of a smart interface sensor to track water, contamination and monitor leakages from water pipelines. A system with a higher spatial resolution would provide operators with a better understanding of their network. In buried pipeline monitoring, sensor nodes are deployed in soil. The underground environment imposes major limitations on sensor nodes, such as poor RF transmission and lack of maintainability

The water tank level, the water leak detector in the pipelines and the pH sensor have been used to monitor water quality and the water temperature sensor to track the water temperature. They use the ultrasonic sensor for tests.

TABLE OF CONTENTS

CHAPTER NO TITLE
PAGE NO

LIST OF TABLES

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT:

Water is a limited resource and is essential for agriculture, industry and for creatures existence on earth including human beings. Lots of people don’t realize the true importance of drinking enough water every day. More water are wasted by many uncontrolled way. This problem is quietly related to poor water allocation, inefficient use, and lack of adequate and integrated water management. Therefore, efficient use and water monitoring are potential constraint for home or office water management

Every living thing on earth needs water to survive. Human bodies are made up of more than 60 percent water. We use clean water to drink, grow crops for food, operate factories, and for swimming, surfing, fishing and sailing. Water is vitally important to every aspect of our lives. Monitoring the quality of surface water will help protect our waterways from pollution. Farmers can use the information to help better manage their land and crops. Our local, state and national governments use monitoring information to help control pollution levels By using water monitoring system, we avoid the water wastage, power consumption and easily prevent the water for our generation. Water monitoring day was established in 2003 by America’s clean water foundation as a global educational outreach program that aims to build public awareness and involvement in protecting water resources around the world. world water monitoring day is celebrated on September 18.

Tank Water Level Monitoring, is used to avoid overflowing and intimate level of water in the tank. Water controlling system implementation makes potential significance in home applications. The existing automated method of level detection is described and that can be used to make a device on/off.

Moreover, the common method of level control for home appliance is simply to start the feed pump at a low level and allow it to run until a higher water level is reached in the water tank. This is not properly supported for adequate controlling system. Besides this, liquid level control systems are widely used for monitoring of liquid levels, reservoirs, silos, and dams etc.

to monitor the quality of water with the help of information sensed by the sensors immersed in

1.2 OBJECTIVES OF THE STUDY:

The proposed method is to use an IoT system to automate most of the water leak detection system. In this method a system consisting of Arduino, flow sensor, solenoid valve and

GSM sim is connected to each end of the pipe which is saved in the database. The water flow through the pipes is recorded by the flow meter. This has an IoT sim card which sends the usage data to the server. The system checks the and compares the flow meter reading from the flow sensor at both ends of the same pipe. If the readings from both the sensors match the solenoid valve allows the water flow to continue, in case the readings d not match the solenoid valve shuts down the water flow at both ends of the pipe. Following closing the solenoid valve the sim card sends a message [2] to the concerned authority with the GPS coordinates of the shutdown valves. It is possible to remotely open the valve through the dashboard after the problem is fixed.

1.3 SCOPE OF THE STUDY:

To monitor the quality of water with the help of information sensed by the sensors immersed in water, so as to keep the water resource within a standard described for domestic usage and to be able to take necessary actions to restore the health of the degraded water body.

Monitoring the quality of surface water will help protect our waterways from pollution. Farmers can use the information to help better manage their land and crops. Our local, state and national governments use monitoring information to help control pollution levels By using water monitoring system Pipeline systems are responsible for transporting vital materials such as water, oil and gas. Any leakage in the pipe can cause major financial losses and possible environmental damages. Currently, buried pipelines are only monitored at key points, which can be spaced several kilometers apart.

1.4 MATERIAL REQUIREMENT:

1.4.1 HARDWARE REQUIREMENT:

- Arduino
- Wi-Fi module
- Ph sensor
- Temperature sensor
- Ultrasonic sensor
- Flow sensor

1.4.2 SOFTWARE REQUIREMENT:

- Proteus software
- Arduino software

1.5 PROCUMENT OF EQUIPMENT:

About hardware pursuing…………………

Assembling it…………….

Connecting to Arduino…………

Coding is done in ide…………

CHAPTER-2
LITERATURE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH:

We analyzed numerous existing systems built by researchers in order to construct a model of good quality [1]. During the study of parameters like temperature, pH and electrical conductivity, pressure different authors proposed differential model to test water quality and water leakage. We have developed a smart water control device that can perform all these monitoring functions by looking at all these details [2]. The [3] author indicated that the Internet of Things applications has been rising tremendously in smart homes recently. The wide variety of various IoT systems typically contributes to interoperability needs. Current IoT projects are implemented using physical platforms which lack decision-making intelligence [4]. In order to solve management of the heterogeneous IoTs in smart home, it is proposed an architecture that implements Event Condition-Action (ECA) process. Developed using a central repository for continuous data on IoT schedules, the constructive architecture has proven perfect for addressing interoperability in clever homes [5]. International Journal of Advanced Science and Technology Vol. 29, No. 9s, (2020), pp. 683-689 ISSN: 2005-4238 IJAST Copyright ⓒ 2020 SERSC 684 There must also be systems in place that actively test water quality and provide articulated sources to villages, towns and communities and the rivers, creeks and shores surrounding our towns and towns for drinking [6]. Better water quality is important to avoid waterborne diseases outbreaks as well as to improve the quality of life. Fiji islands are located in the vast Pacific which demands a frequent water quality monitoring network and the current measurements can be enhanced by IoT and RS. This paper presents a smart water quality monitoring system for Fiji, using IoT and remote sensing technology [7]. This problem affects various processes in water management, such as water consumption, distribution, system identification and equipment maintenance. OPC UA (Object Linking and Embedding for Process Control Unified Architecture) is a platform independent service-oriented architecture for the control of processes in the logistic and manufacturing sectors [8]. Based on this standard we propose a smart water management model combining Internet of Things technologies with business processes coordination and decision support systems. We provide an architecture for sub-system interaction and a detailed description of the physical scenario in which we will test our implementation, allowing specific vendor equipment to be manageable and interoperable in the specific context of processes water management [9]. In [10] the author shown how to monitor the water level of water systems such as water tanks, rivers, ground water table, and bore wells remotely. They also have shown that how to control the working of pump automatically and remotely. It can be used to remotely monitor the flood affected areas wirelessly and information can be sent to mobile wirelessly [11]. This system is designed to monitor the level of water with the help of water level sensors. This article includes an IoT tool for monitoring and preparing water use. This system is simple and long-lasting to install and maintain. The laser sensor is located on the tank that controls water level continuously in real time. This data is stored in the cloud and users can analyze water quantity [12]. The motor function is operated automatically, depending on the water level in the tank. The motor will automatically be turned on when the water level goes below the threshold level. In [13] paper, we present an IoT architecture for water monitoring and control that supports in real-time online data collection. The program addresses new problems in calculating the water flow rate and the need to research the water supply to minimize and encourage water pollution. By using pH and conductivity sensors, we also calculate the consistency of water distributed throughout every house. The conventional water measurement systems need frequent human maintenance intervention to make it uncomfortable and therefore less effective. In the absence of modern models, wired systems are used for smart monitoring and wireless communication of data The paper [14] explored various technologies and platforms required for an intelligent world. An architectural framework concept is proposed for intelligent water management and an implementation description is discussed of the intelligent water monitoring system.

2.2 REVIEW ON RELATED LITERATURE:

Nikhil Kedia entitled “Water Quality Monitoring for Rural Areas-A Sensor Cloud Based Economical Project.” Published in 2015 1st International Conference on Next Generation Computing Technologies (NGCT-2015) Dehradun, India. This paper highlights the entire water quality monitoring methods, sensors, embedded design, and information dissipation procedure, role of government, network operator and villagers in ensuring proper information dissipation. It also explores the Sensor Cloud domain. While automatically improving the water quality is not feasible at this point, efficient use of technology and economic practices can help improve water quality and awareness among people.[1] Jayti Bhatt,Jignesh Patoliya entitled “Real Time Water Quality Monitoring System”. This paper describes to ensure the safe supply of drinking water the quality should be monitored in real time for that purpose new approach IOT (Internet of Things) based water quality monitoring has been proposed. In this paper, we present the design of IOT based water quality monitoring system that monitor the quality of water in real time. This system consists some sensors which measure the water quality parameter such as pH, turbidity, conductivity, dissolved oxygen, temperature. The measured values from the sensors are processed by microcontroller and this processed values are transmitted remotely to the core controller that is raspberry pi using Zigbee protocol. Finally, sensors data can view on internet browser application using cloud computing.[2] Michal Lom, Ondrej Pribyl, Miroslav Svitek entitled “Industry 4.0 as a Part of Smart Cities”. This paper describes the conjunction of the Smart City Initiative and the concept of Industry 4.0. The term smart city has been a phenomenon of the last years, which is very inflected especially since 2008 when the world was hit by the financial crisis. The main reasons for the emergence of the Smart City Initiative are to create a sustainable model for cities and preserve quality of life of their citizens. The topic of the smart city Water Quality Monitoring System Based on IOT 1109 cannot be seen only as a technical discipline, but different economic, humanitarian or legal aspects must be involved as well. In the concept of Industry 4.0, the Internet of Things (IoT) shall be used for the development of so–called smart products. Subcomponents of the product are equipped with their own intelligence. Added intelligence is used both during the manufacturing of a product as well as during subsequent handling, up to continuous monitoring of the product lifecycle (smart processes). Other important aspects of the Industry 4.0 are Internet of Services (IoS), which includes especially intelligent transport and logistics (smart mobility, smart logistics), as well as Internet of Energy (IoE), which determines how the natural resources are used in proper way (electricity, water, oil, etc.). IoT, IoS, IoP and IoE can be considered as an element that can create a connection of the Smart City Initiative and Industry 4.0 – Industry 4.0 can be seen as a part of smart cities.[3] Zhanwei Sun,Chi Harold Li,Chatschik Bisdikian,Joel W.Branch and Bo Yang entitled “QOI-Aware Energy Management in Internet-of-Things Sensory Environments”. In this paper an efficient energy management frame work to provide satisfactory QOI experience in IOT sensory environments is studied. Contrary to past efforts, it is transparent and compatible to lower protocols in use, and preserving energy-efficiency in the long run without sacrificing any attained QOI levels. Specifically, the new concept of QOI-aware “sensor-to-task relevancy” to explicitly consider the sensing capabilities offered by an sensor to the IOT sensory environments, and QOI requirements required by a task. A novel concept of the “critical covering set” of any given task in selecting the sensors to service a task over time. Energy management decision is made dynamically at runtime, as the optimum for long-term traffic statistics under the constraint of the service delay. Finally, an extensive case study based on utilizing the sensor networks to perform water level monitoring is given to demonstrate the ideas and algorithms proposed in this paper, and a simulation is made to show the performance of the proposed algorithms.[4] Sokratis Kartakis, Weiren Yu, Reza Akhavan, and Julie A. McCann entitled “Adaptive Edge Analytics for Distributed Networked Control of Water Systems” This paper presents the burst detection and localization scheme that combines lightweight compression and anomaly detection with graph topology analytics for water distribution networks. We show that our approach not only significantly reduces the amount of communications between sensor devices and the back end servers, but also can effectively localize water burst events by using the difference in the arrival times of the vibration variations detected at sensor locations. Our results can save up to 90% communications compared with traditional periodical reporting situations.[5]

2.3 CONCLUSION ON REVIEWS:

Monitoring of Turbidity, PH & Temperature of Water makes use of water detection sensor with unique advantage and existing GSM network. The system can monitor water quality automatically, and it is low in cost and does not require people on duty. So the water quality testing is likely to be more economical, convenient and fast. The system has good flexibility. Only by replacing the corresponding sensors and changing the relevant software programs, this system can be used to monitor other water quality parameters. The operation is simple. The system can be expanded to monitor hydrologic, air pollution, industrial and agricultural production and so on. It has widespread application and extension value. By keeping the embedded devices in the environment for monitoring enables self protection (i.e., smart environment) to the environment. To implement this need to deploy the sensor devices in the environment for collecting the data and analysis. By deploying sensor devices in the environment, we can bring the environment into real life i.e. it can interact with other objects through the network. Then the collected data and analysis results will be available to the end user through the Wi-Fi.

CHAPTER-3

PROJECT DESIGN

3.0. OVERVIEW OF THE DESIGN

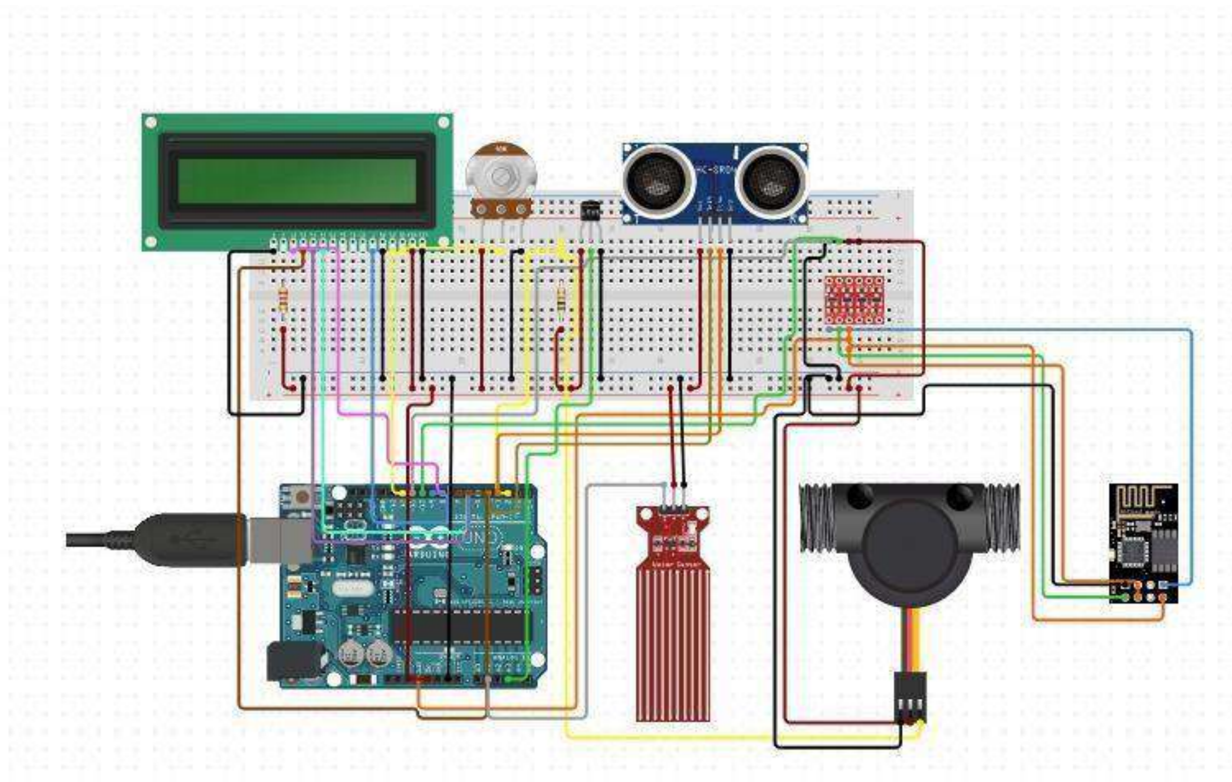


Fig.3.1 Overview of the design

3.2 EQUIPMENT ANALYSIS:

3.2.1 ARDUINO:

3.2.1.1 INTRODUCTION:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

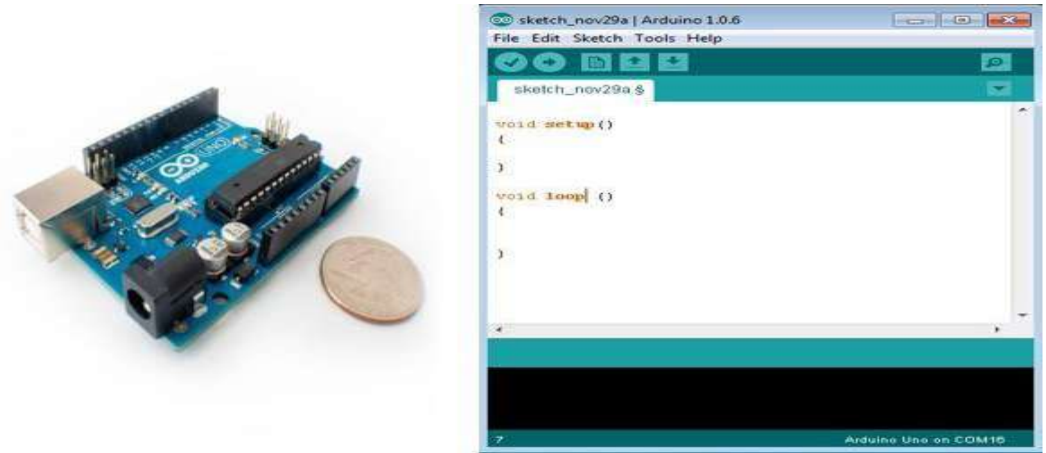


Fig 3.2.1.1 Arduino Uno

3.2.1.2 BOARD TYPES

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.
Here is a list of different Arduino boards available.

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface

Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header

Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.2.1.2.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB

Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 3.2.1.2.2 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 3.2.1.2.3 Arduino boards based on ATMEGA2560 microcontroller

Board	Operating	Clock	Digital	Analog	PWM	UART	Programming

Name	Volt	Speed	i/o	Inputs			Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table 3.2.1.2.4 Arduino boards based on AT91SAM3X8E microcontroller

3.2.1.3 BOARD DESCRIPTION:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

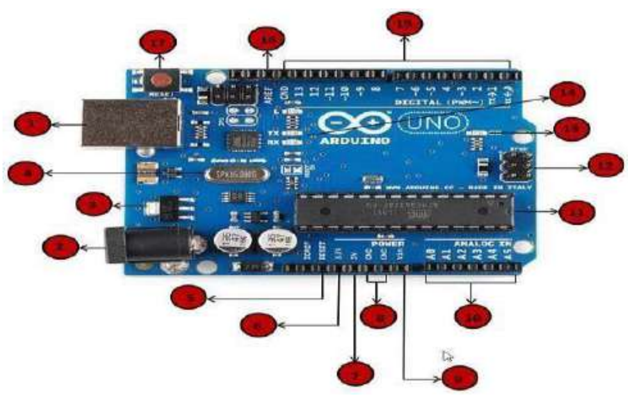


Fig.3.2.1.2 Board Description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).

3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC

	your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

3.2.1.4 ARDUINO FAMILY

Arduino makes several different boards, each with different capabilities. In addition, part of being open-source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

Fig.3.2.1.3: Arduino Family

3.2.1.5 SHIELDS:

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

Fig.3.2.1.4: Arduino Shields

3.2.1.6 PIN DESCRIPTION OF ATMEGA328:

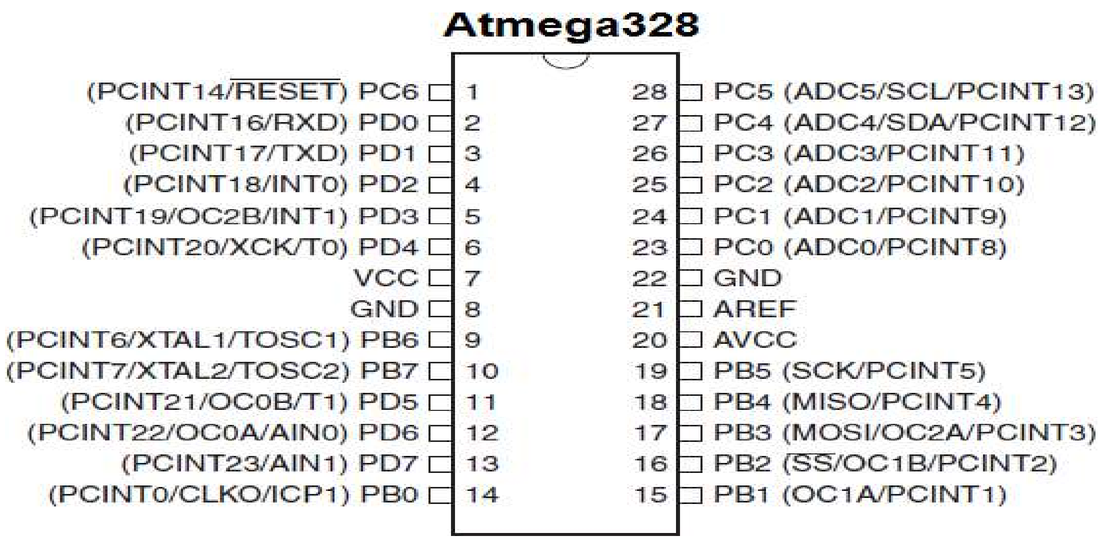


Fig.3.2.1.5 Pin description of ATMEGA328

3.2.1.7 APPLICATIONS:

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

3.2.2 16*2 LCD SCREEN DISPLAY

3.2.2.1 INTRODUCTION

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

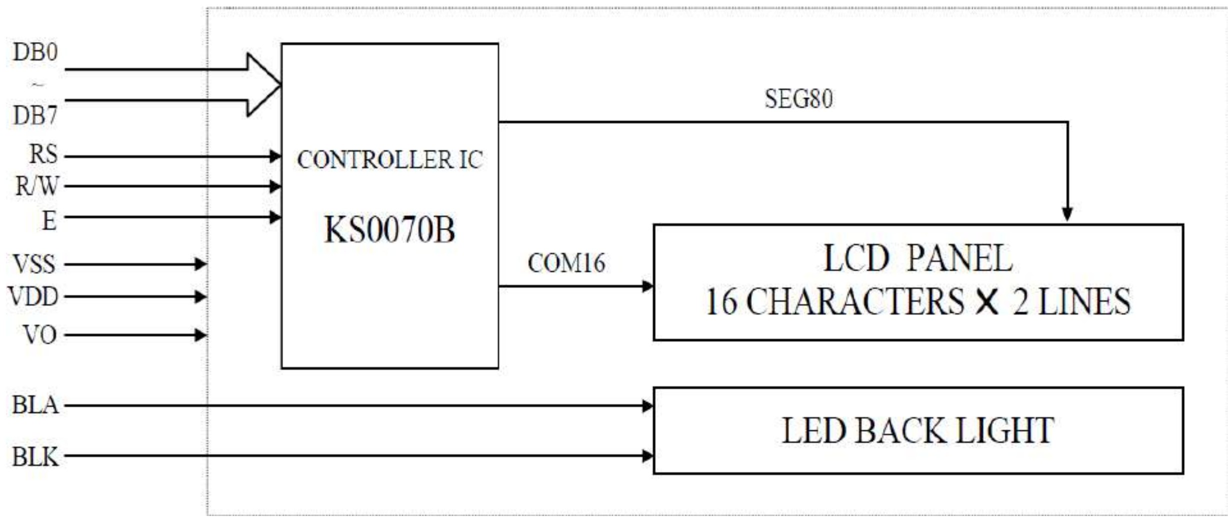
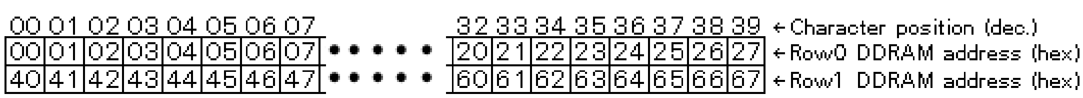


fig.3.2.2.1 Lcd block diagram

3.2.2.2 DISPLAY DATA RAM:

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.



DDRAM Address for 2 Line LCD

Fig.3.2.2.2 DDRAM addresses of 2 Line LCD

3.2.2.3 CHARACTER GENERATOR ROM:

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM.

3.2.2.4 BUS FLAG:

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e. BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

3.2.2.5 INSTRUCTION REGISTER (IR) AND DATA REGISTER (DR):

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g. LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD.
Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

3.2.2.6 16 X 2 ALPHANUMERIC LCD MODULE FEATURES:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

3.2.2.7 CIRCUIT DESCRIPTION:

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

3.2.3 POWER SUPPLY UNIT

3.2.3.1 INTRODUCTION:

The power supplies are designed to convert high voltage AC mains electricity to a suitable l voltage supply for electronic circuits and other devices. A power supply can by broken down into a se of blocks, each of which performs a particular function. A d.c power supply which maintains the out voltage constant irrespective of a.c mains fluctuations or load variations is known as “Regularted Power Supply”.

3.2.3.2 BLOCK DIAGRAM OF POWER SUPPLY:

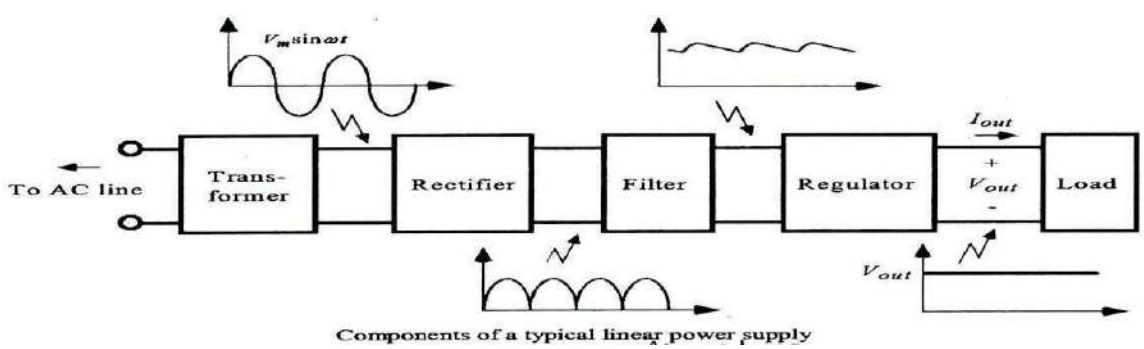


Fig.3.2.3.2: Block Diagram of Power Supply

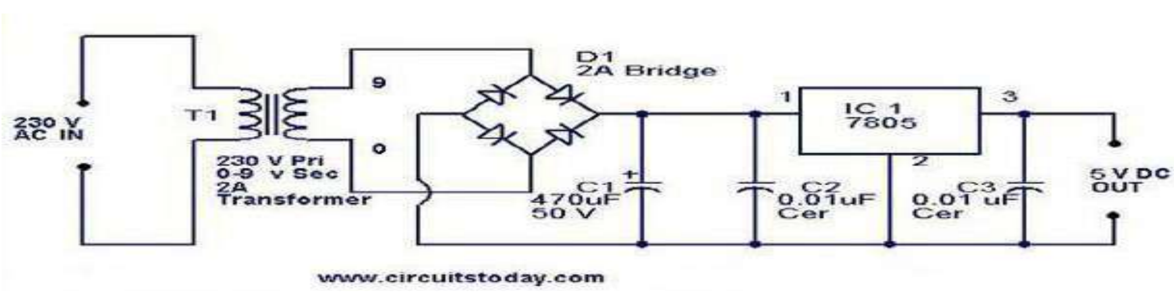


Fig.3.2.3.2(A): Schematic Diagram of Power Supply

3.2.3.3 DESCRIPTION OF POWER SUPPLY:

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a a different form of energy – such as solar, mechanical, or chemical - into electrical energy.

A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered. For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

3.2.3.4 TRANSFORMER:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

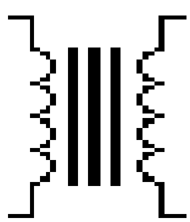


Fig.3.2.3.4: Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

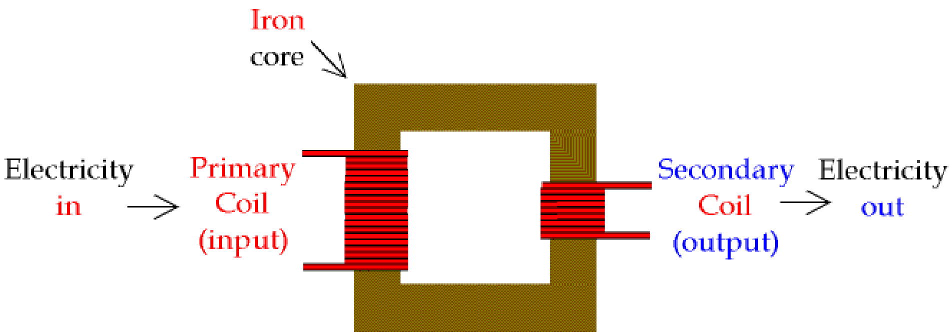
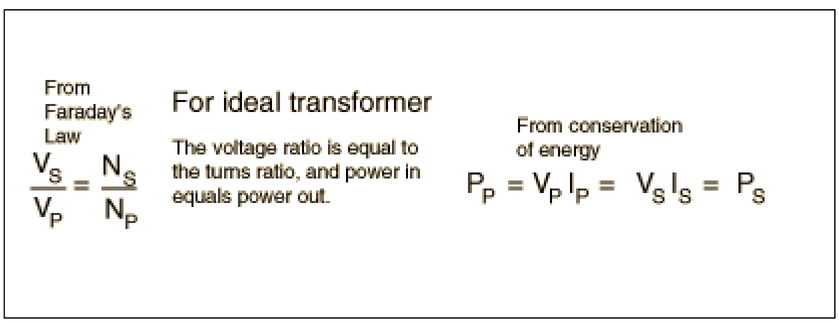


Fig.3.2.3.4(A):Transformer

3.2.3.5 Basic Principle of Transformer:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.



Working of Transformer:

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

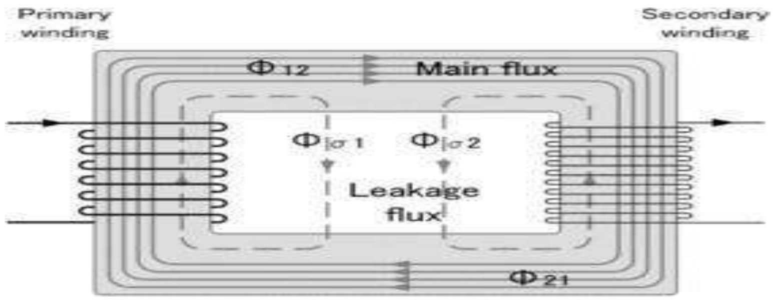


Fig.3.2.3.5: Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f. The iron core is laminated to prevent 'eddy currents' flowing in the core.

These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

3.2.3.6 Classification of Transformer:

- Step-Up Transformer
- Step-Down Transformer

Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn's ratio of the two sets of windings determines the amount of voltage transformation.

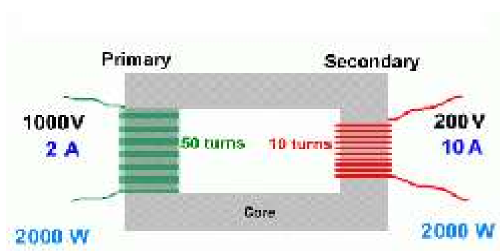


Fig.3.2.2.6: Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to

1.

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single phase steps down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase steps up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

Step-Up Transformer:

A step up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply. A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

Fig.3.2.3.6(a): Step-Up Transformer

3.2.3.7 DIODES:

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

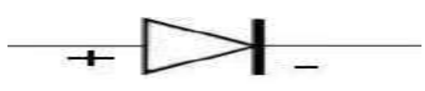


Fig.3.2.3.7: Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

3.2.3.8 Rectifier:

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

The Half-Wave Rectifier:

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure.

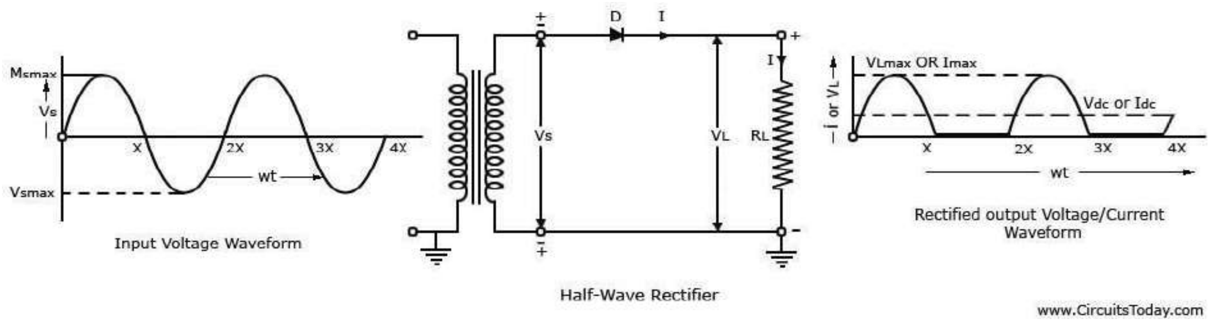


Fig.3.2.3.8(a): Half Wave Rectifier

Figure 2 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

The Full-wave Rectifier:

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC.

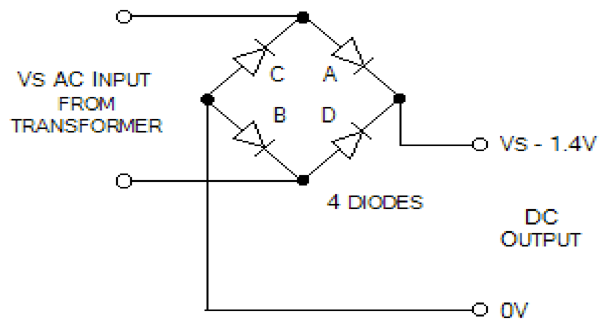


Fig.3.2.3.8(b): Full-Wave Rectifier

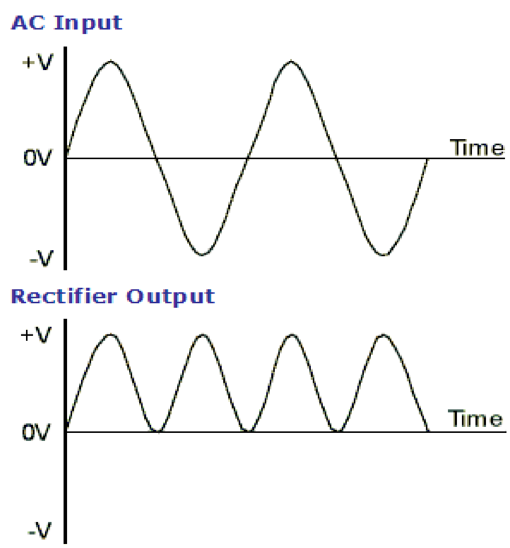


Fig.3.2.3.8(c): Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased. While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

3.2.4 CAPACITOR FILTER:

The capacitor-input filter, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

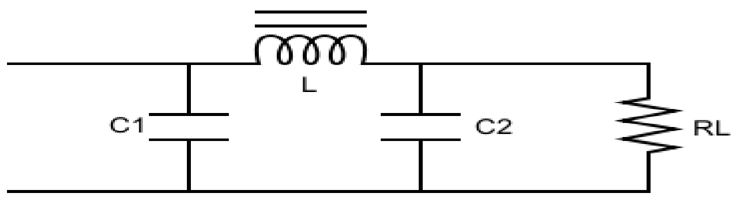


Fig.3.2.4: Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result, the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result, only the DC component appears across the load RL.

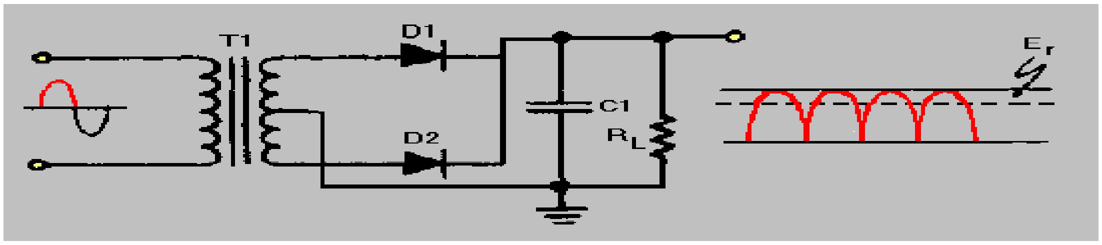


Fig.3.2.4.0(A): Centered Tapped Full-Wave Rectifier with a Capacitor Filter

3.2.5 VOLTAGE REGULATOR:

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx'indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx'indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

These regulators consist the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

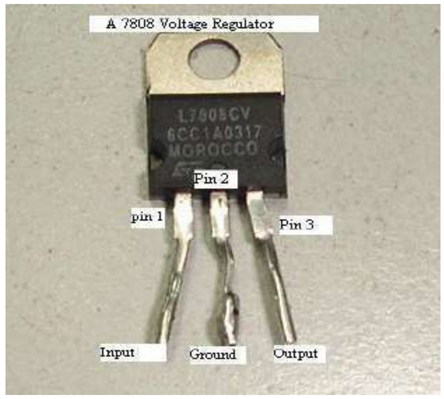


Fig.3.2.5: Regulator

3.2.6 SOFTWARE EXPLANATION:

- Arduino software.
- Proteus simulation.

3.2.6.1 ARDUINO SOFTWARE:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino

boards are
completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

3.2.6.2 WHY ARDUINO?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics
principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and
wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.Wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some

advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers c an make their own version Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money
- Getting Started with Arduino and Genuino products: -

Install the Arduino Software (IDE) on Windows PCs-
This document explains how to install the Arduino Software (IDE) on Windows machines.

✓ Download the Arduino Software (IDE)

✓ Proceed with board specific instructions.

3.2.6.3 HOW TO DOWNLOAD THE ARDUINO SOFTWARE (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the

Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

3.2.6.4 INSTALLATION:

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.3.2.6.4.1 Standard USB Cable (A plug to B plug)

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig. 3.2.6.4.2 Standard USB Cable (an A to Mini-B plug)

Step 2 – Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

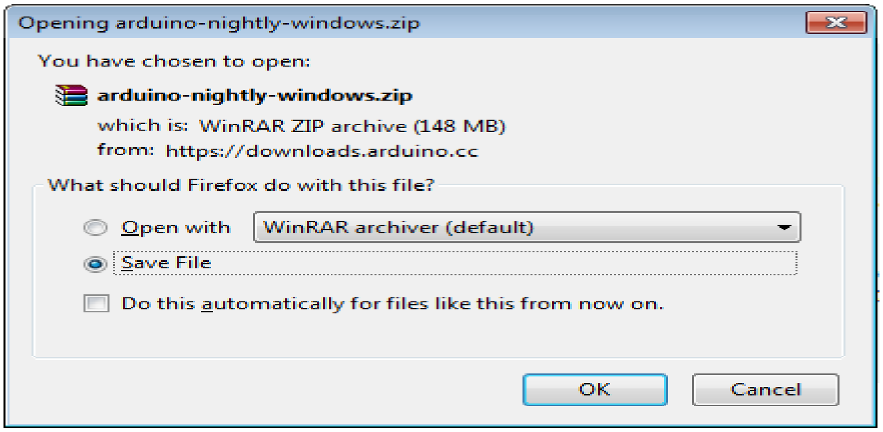


Fig. 3.2.6.4.3 Download Arduino IDE Software

Step 3 – Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 – Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

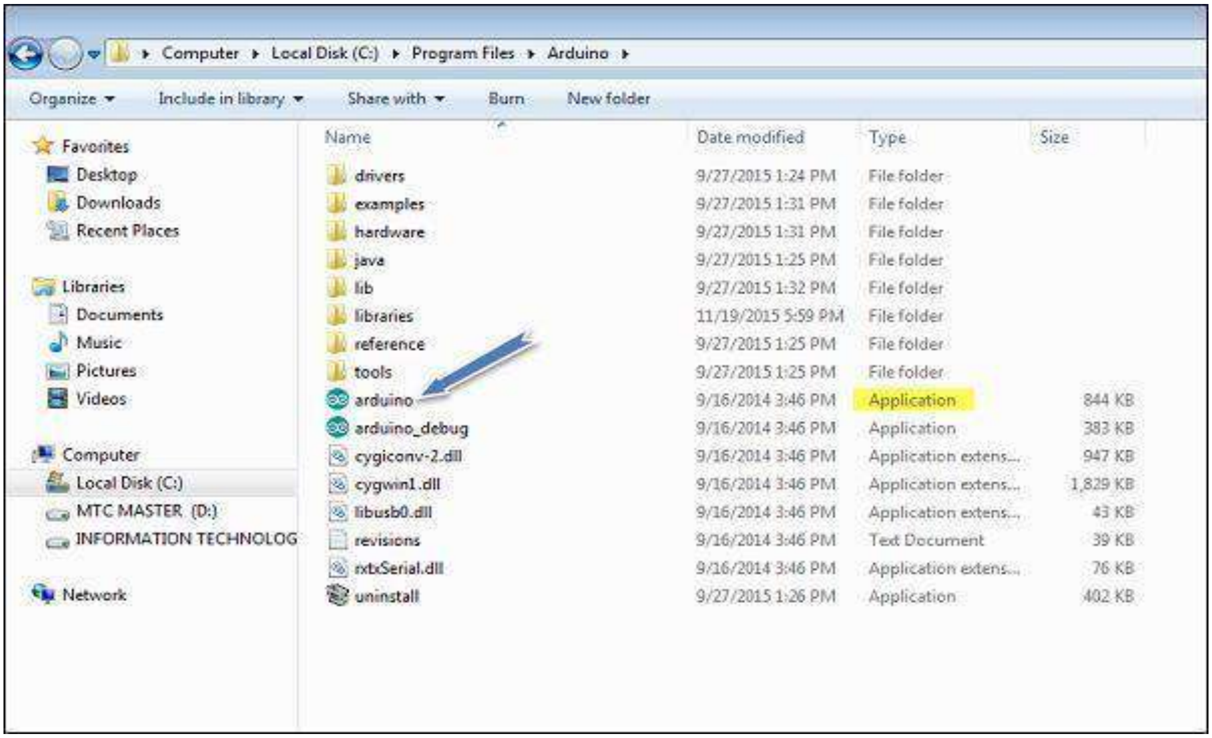


Fig. 3.2.6.4.4 Launch Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**

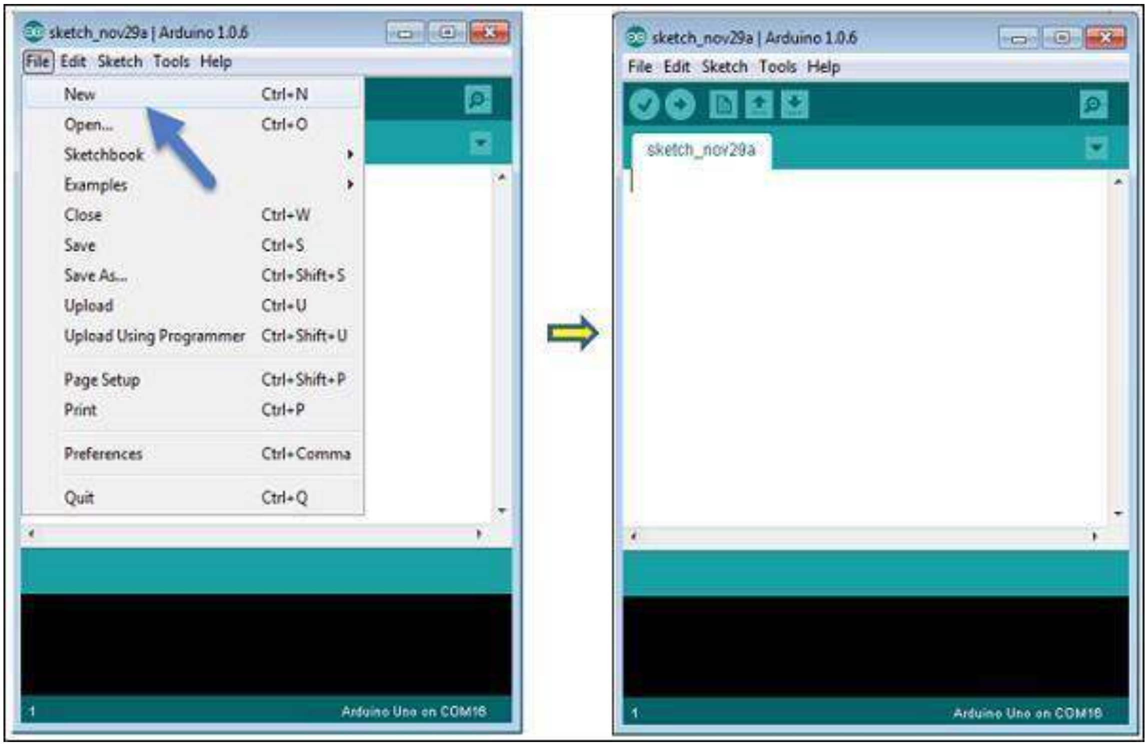


Fig. 3.2.6.4.5 Open your first project.

To open an existing project example, select File → Example → Basics → Blink.

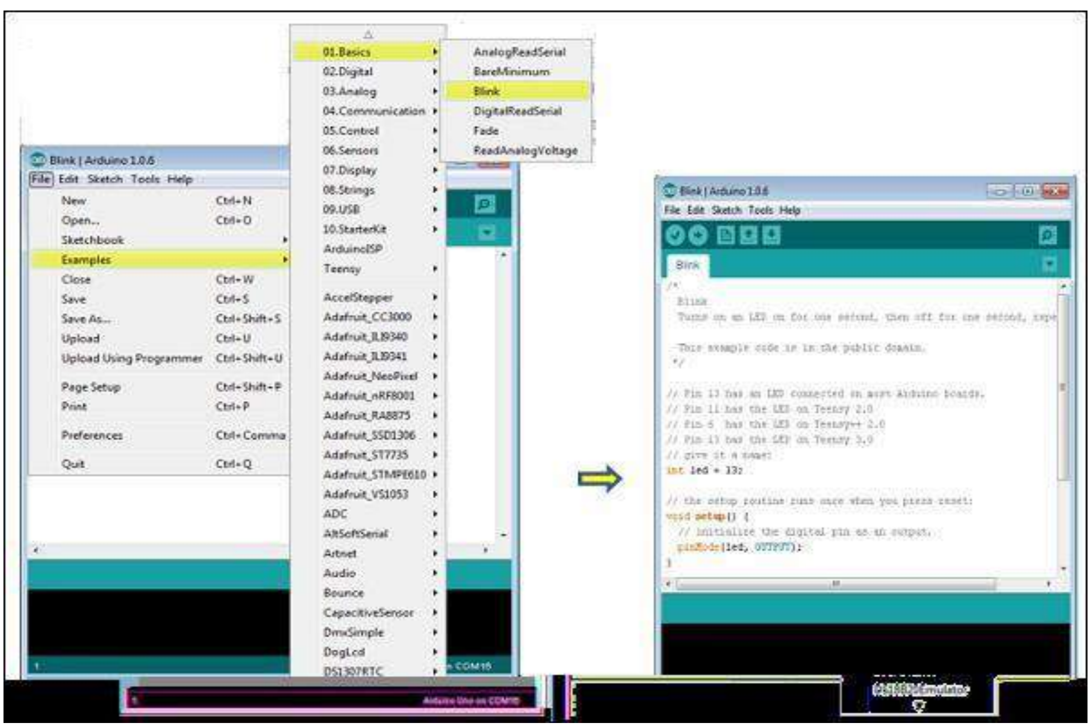


Fig. 3.2.6.4.6 Open your first program.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

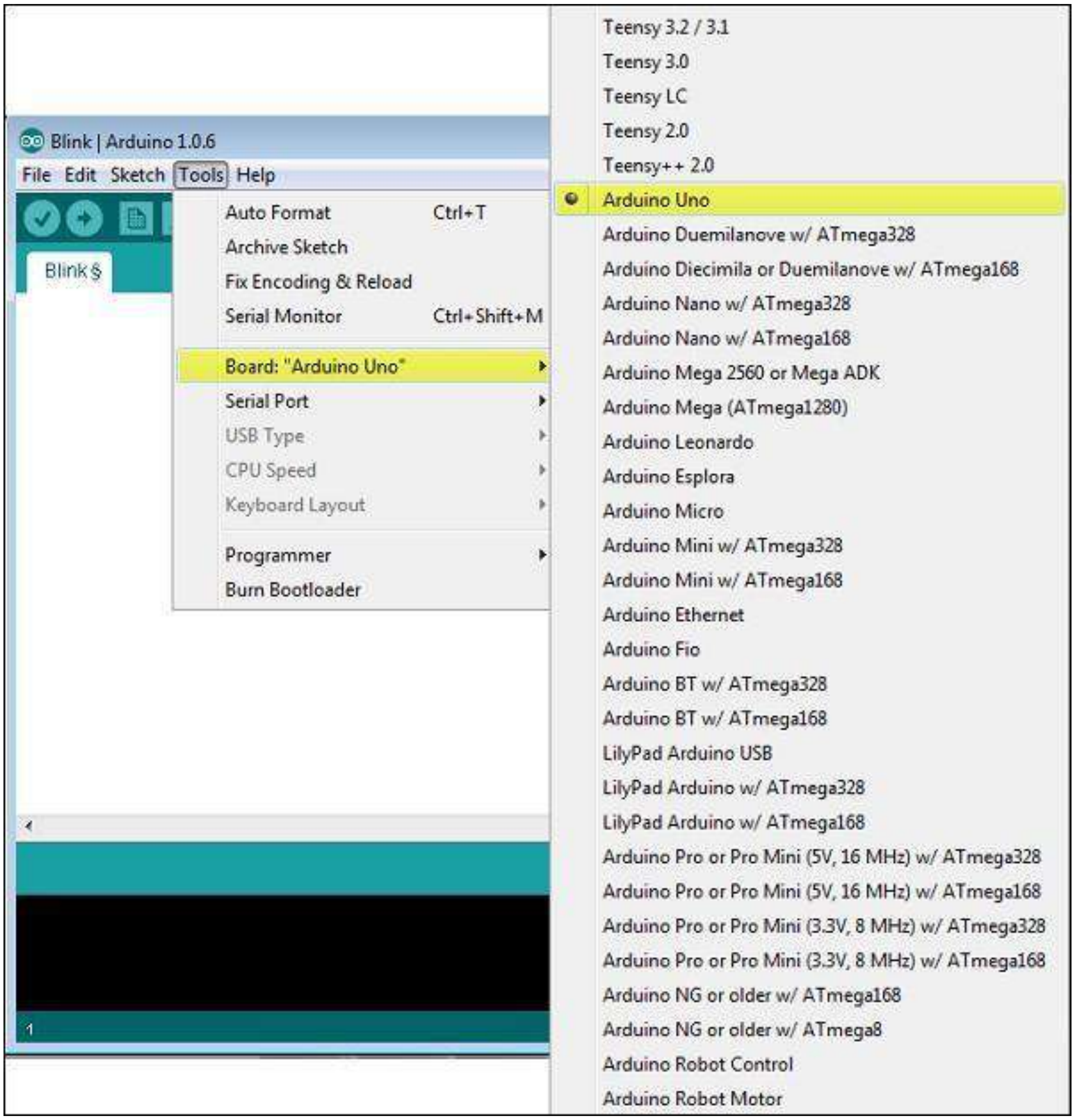


Fig. 3.2.6.4.7 Select your Arduino board.

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 – Select your serial port.

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

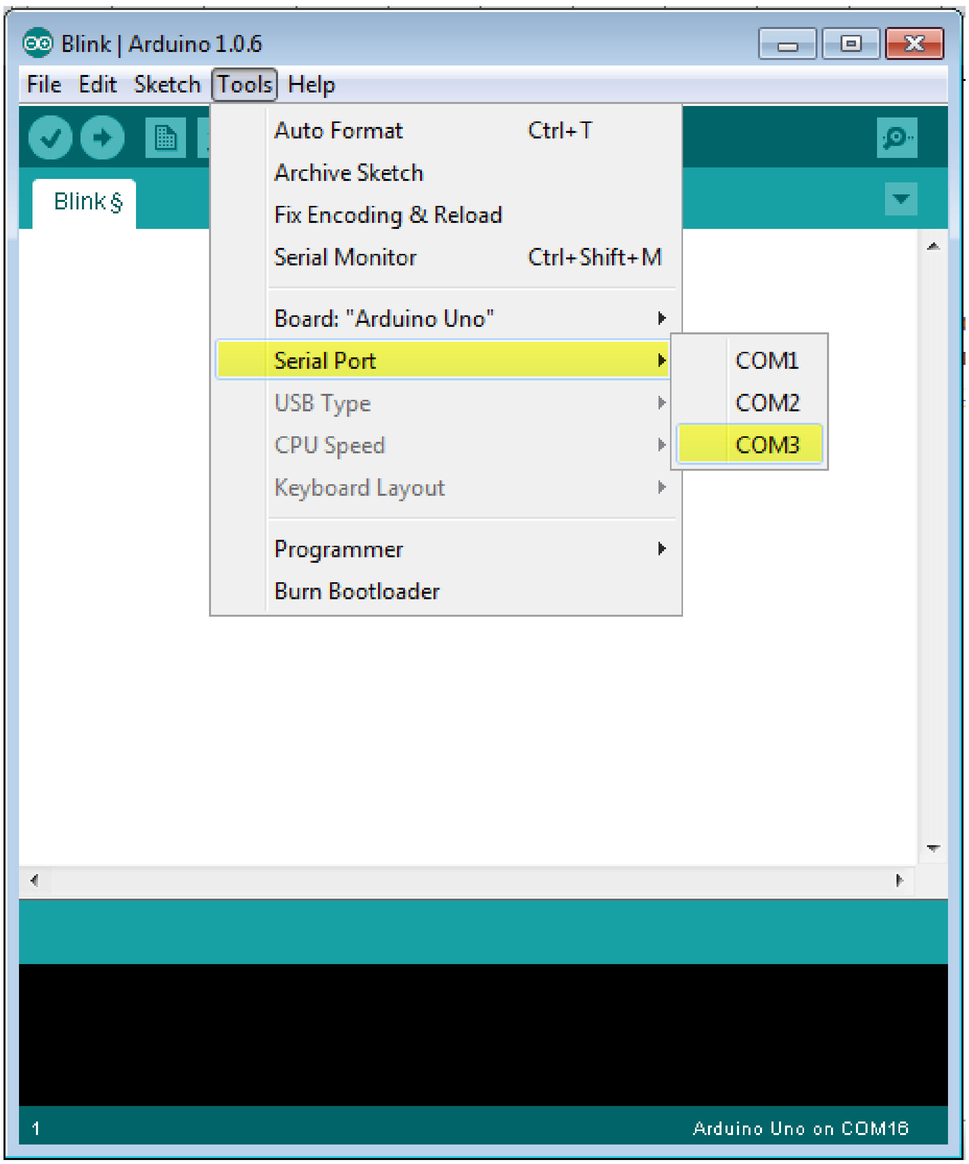


Fig. 3.2.6.4.8 Select your serial port.

Step 8 – Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

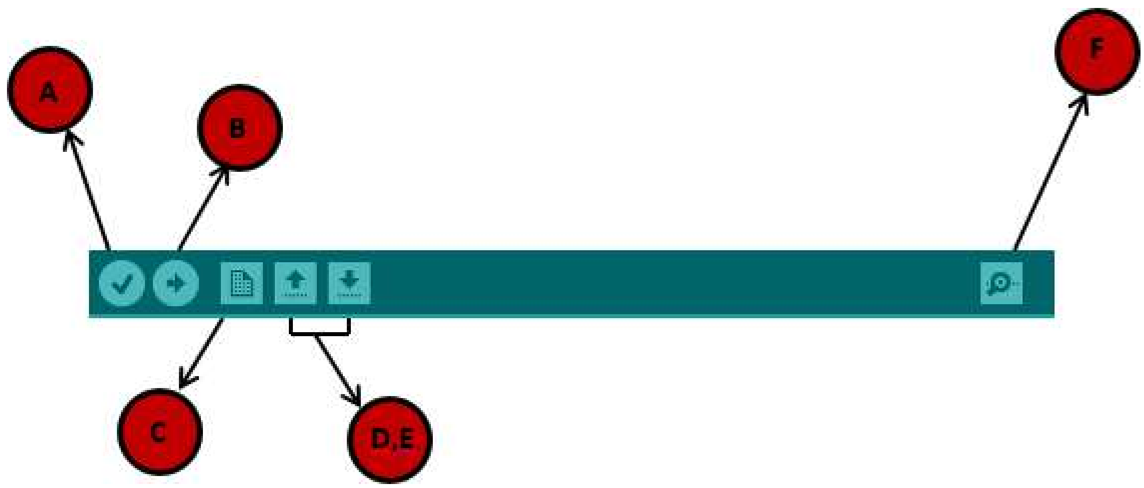


Fig. 3.2.6.4.9 Upload the program to your board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from board & send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

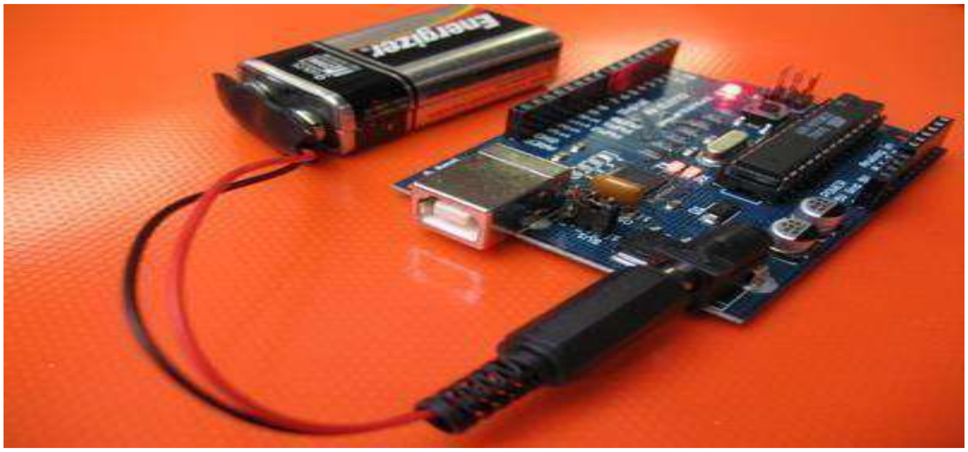


Fig. 3.2.6.4.10 Connecting a Battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called “sketches”, but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

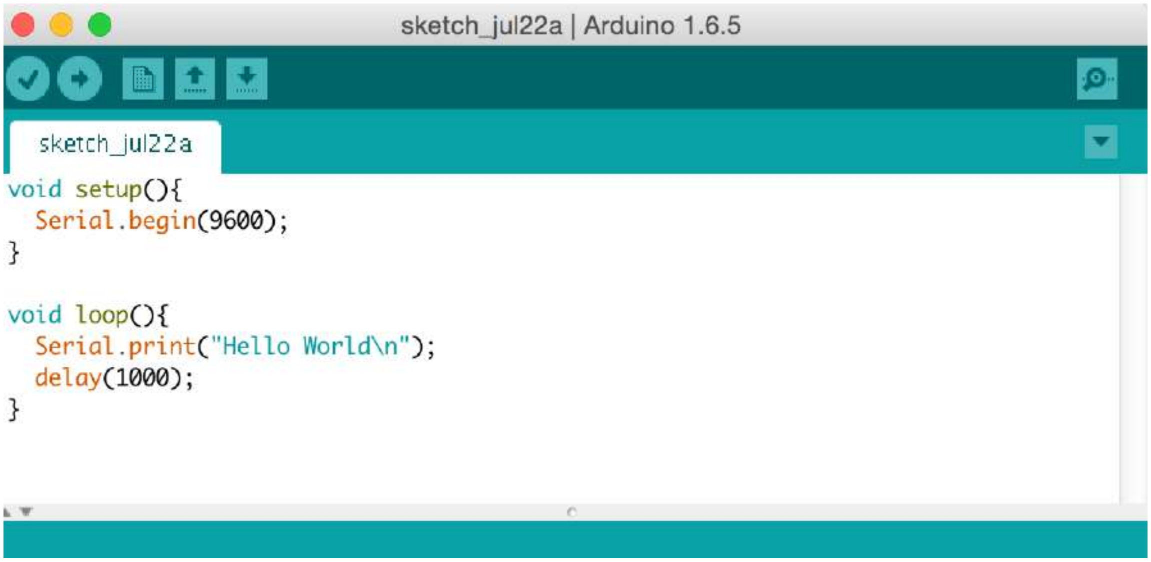


Fig. 3.2.6.4.11 Example Program

3.2.7 PROTEUS:

PROTEUS:

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline "From concept to completion".

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design:

Step 1: Open ISIS software and select New design in File menu

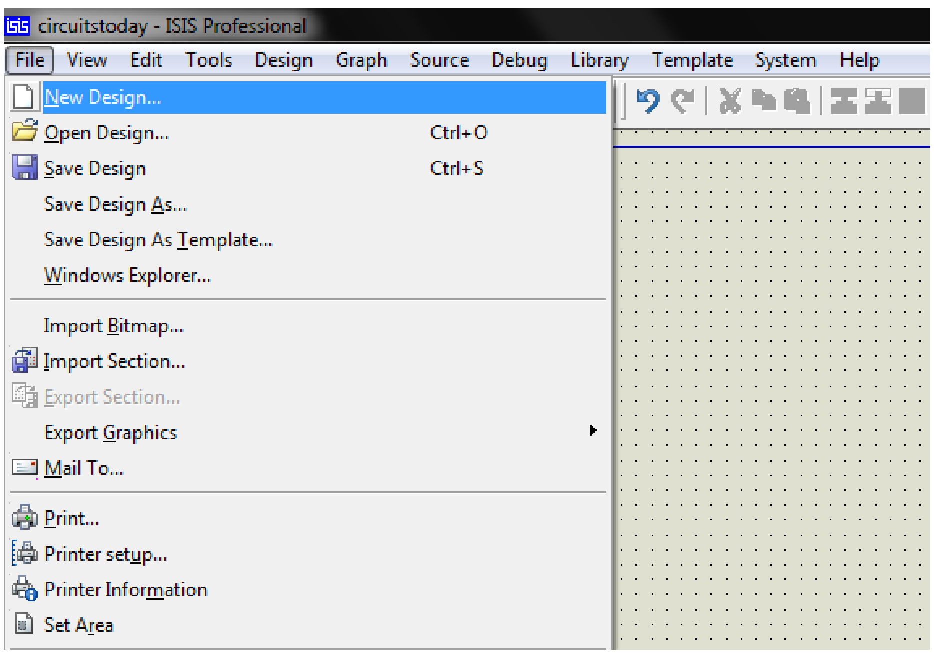


Fig. 3.2.7.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit

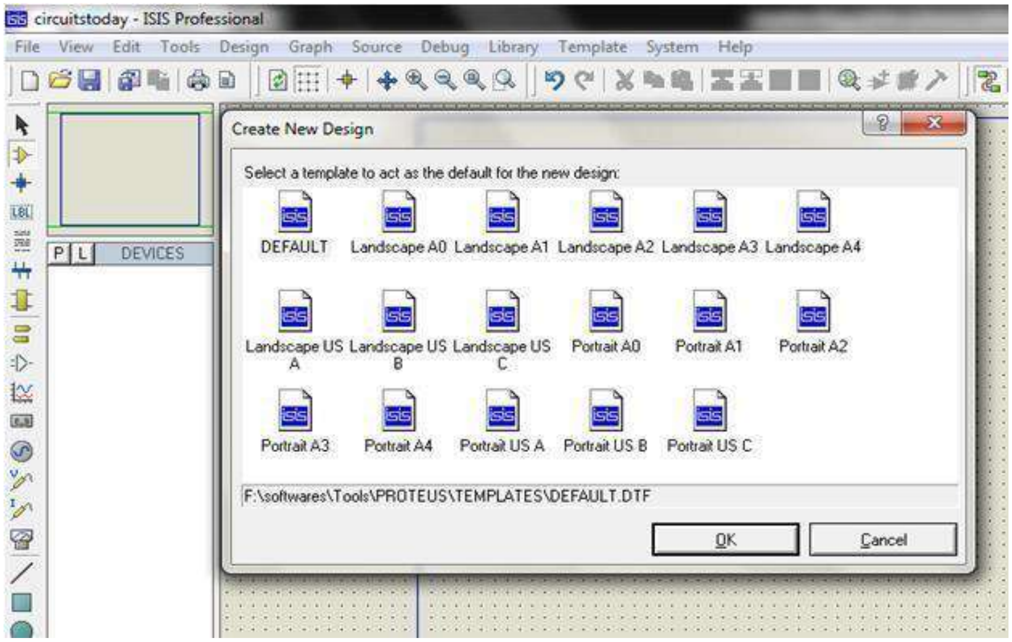


Fig. 3.2.7.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

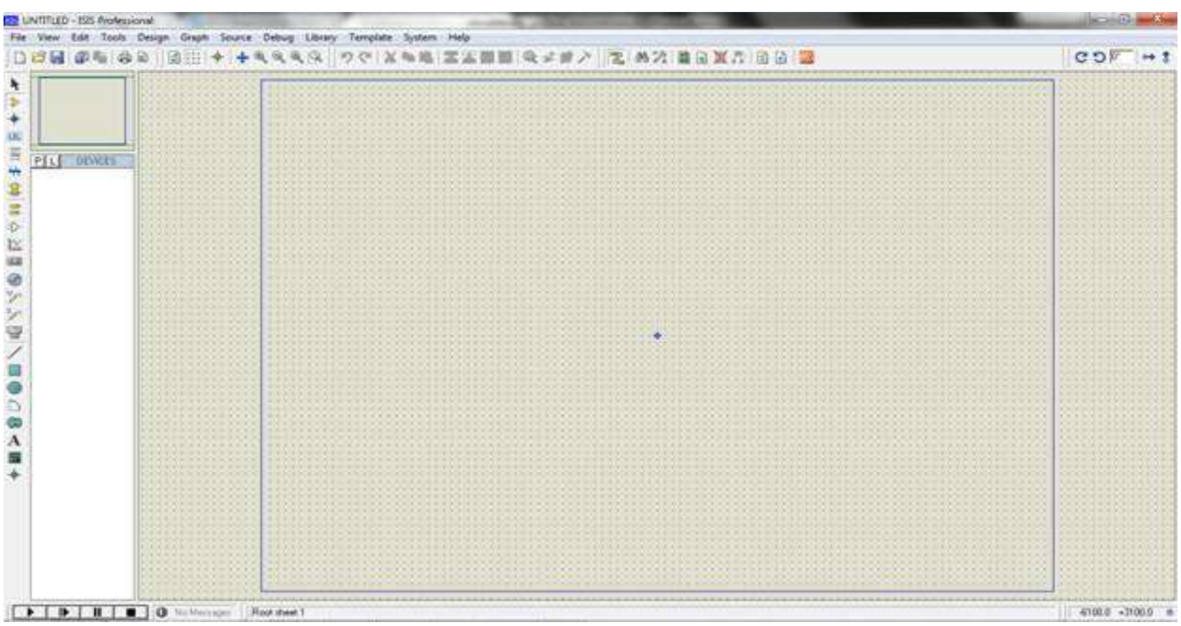


Fig. 3.2.7.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

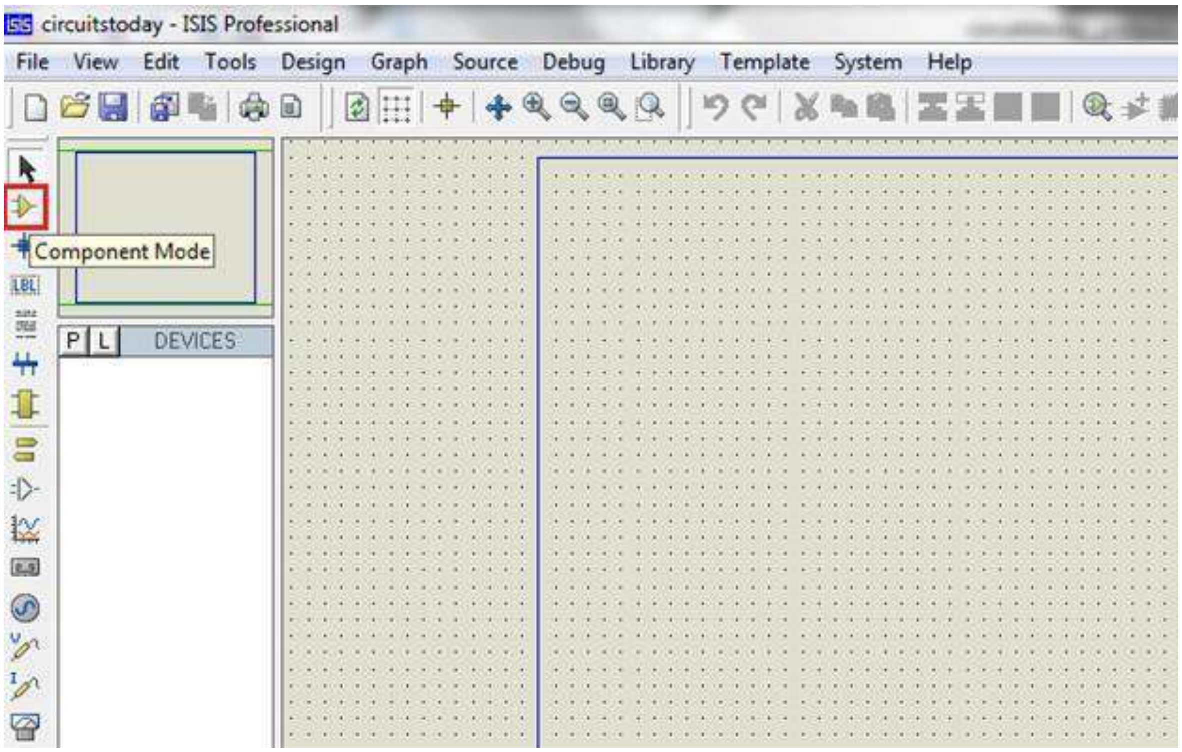


Fig. 3.2.7.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

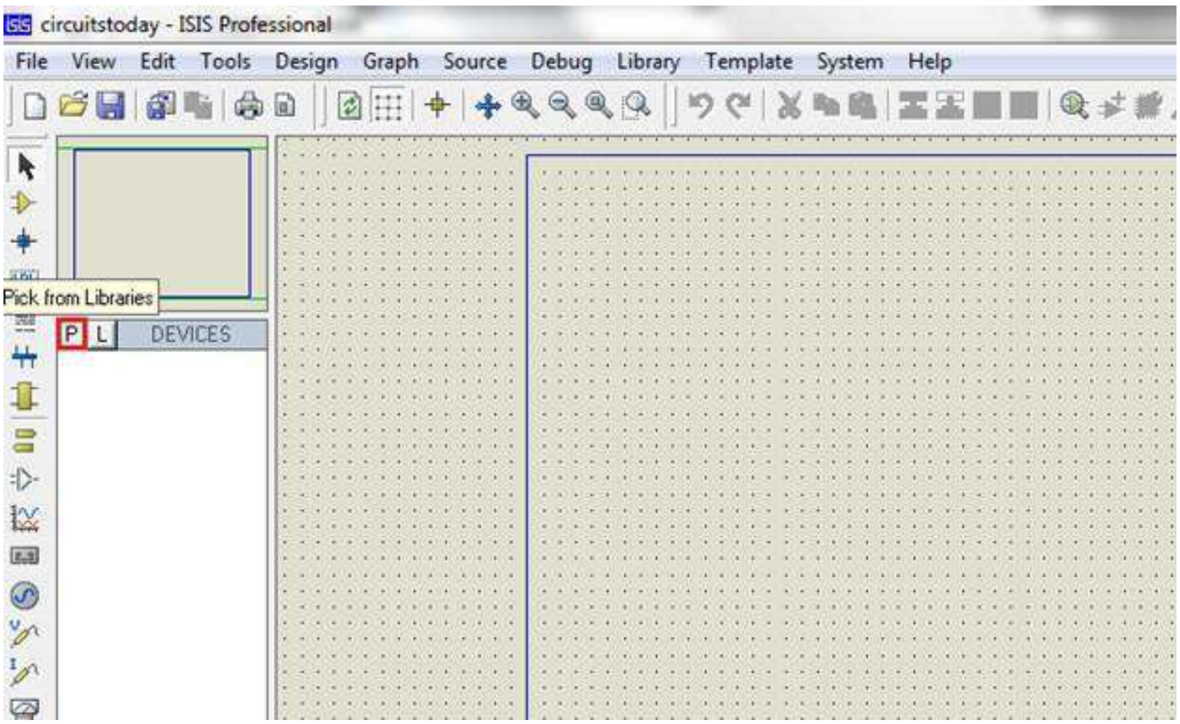


Fig. 3.2.7.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

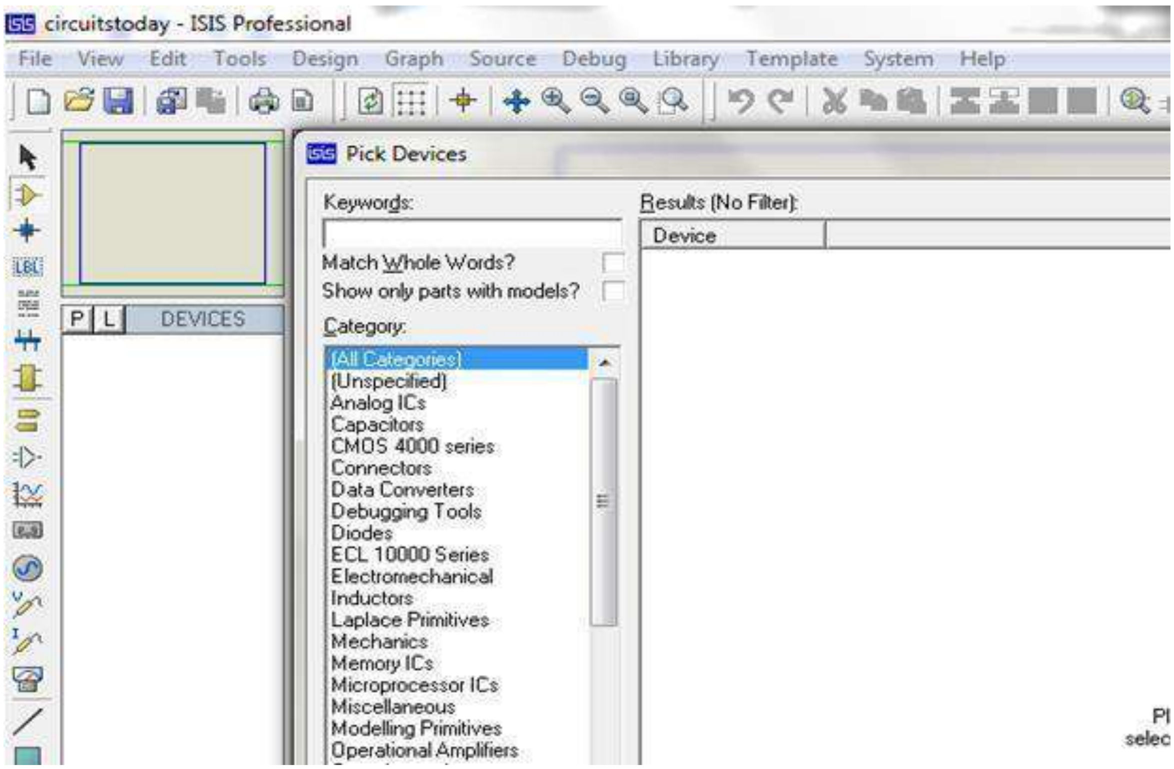


Fig. 3.2.7.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

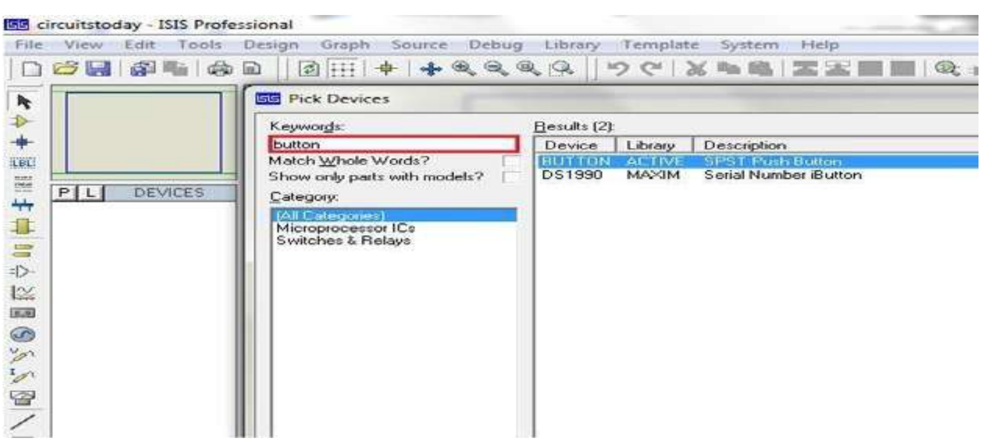


Fig. 3.2.7.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

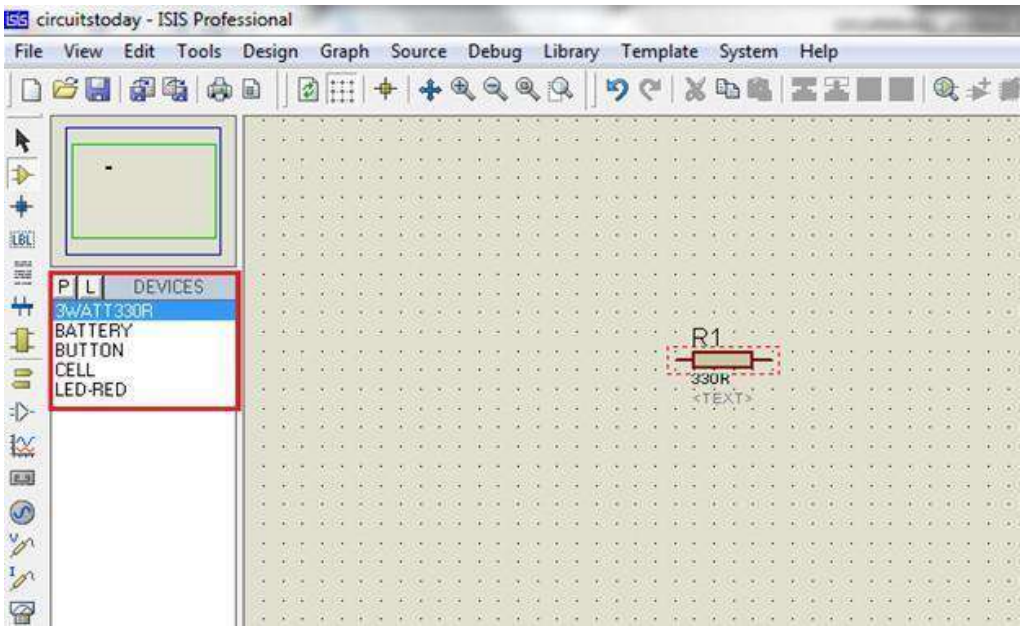


Fig. 3.2.7.8 Component Selection

Place all the required components and route the wires i.e., make connections. Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

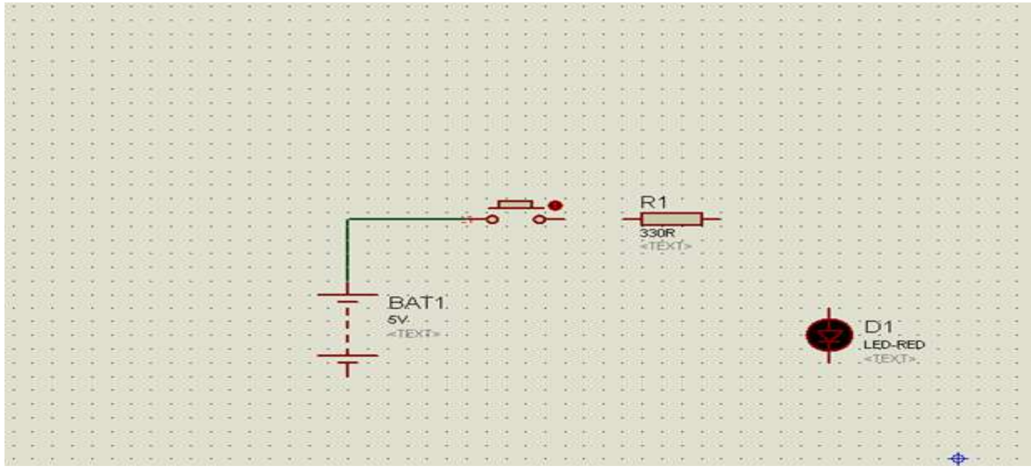


Fig. 3.2.7.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

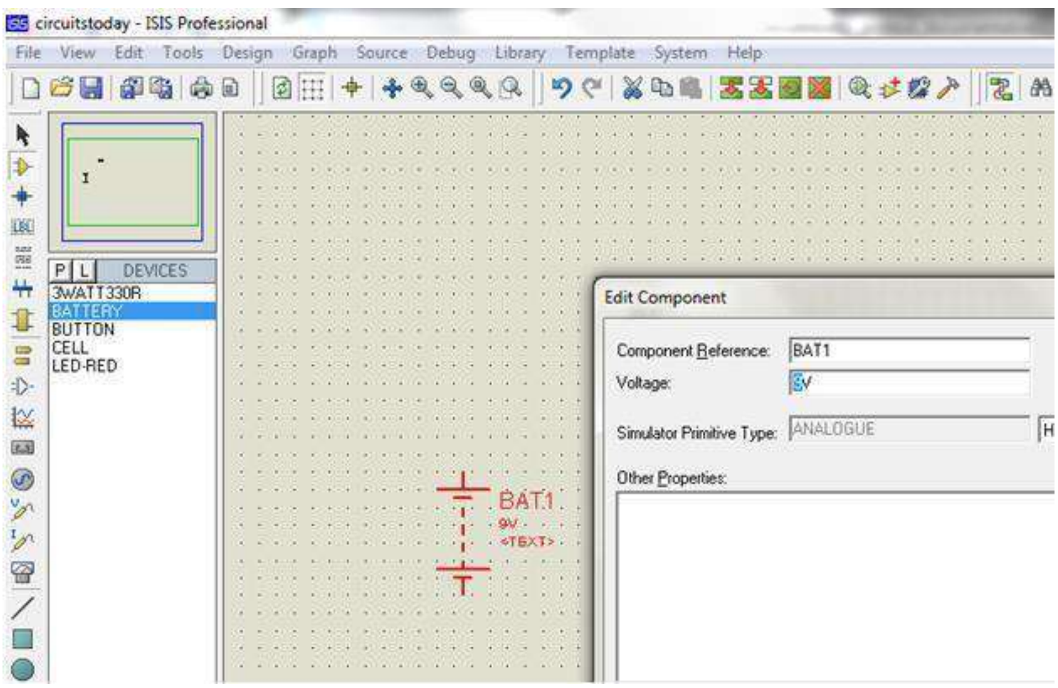


Fig. 3.2.7.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

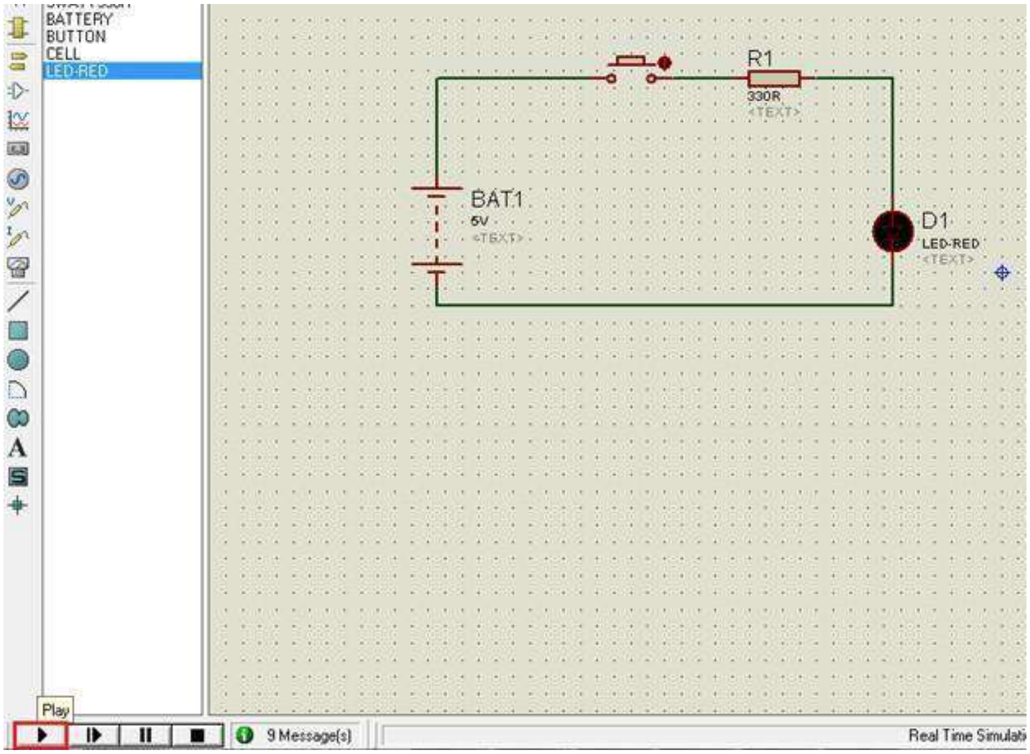


Fig. 3.2.7.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

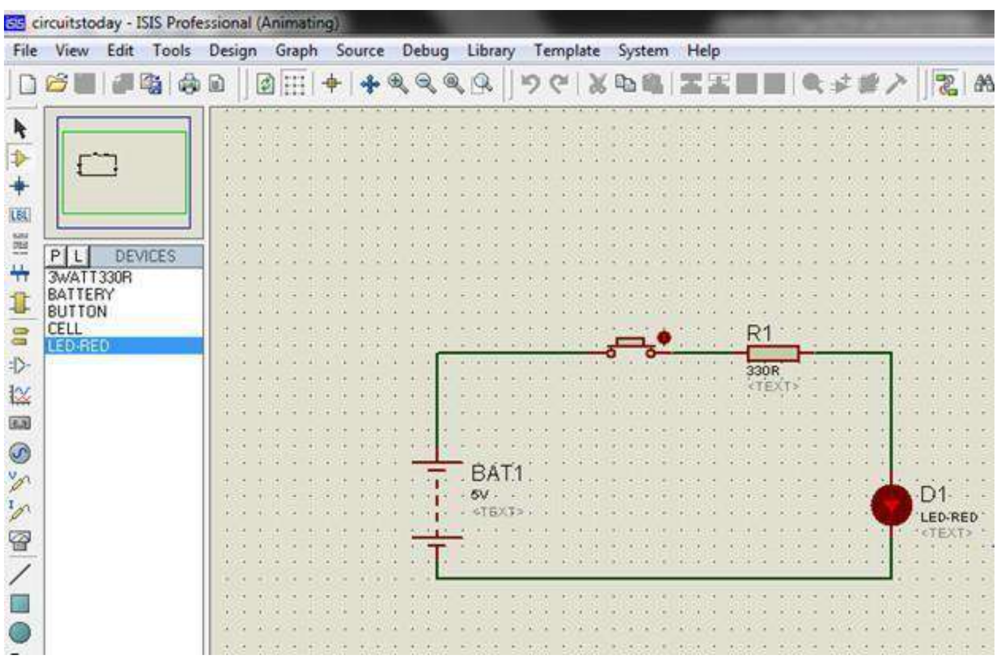


Fig . 3.2.7.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

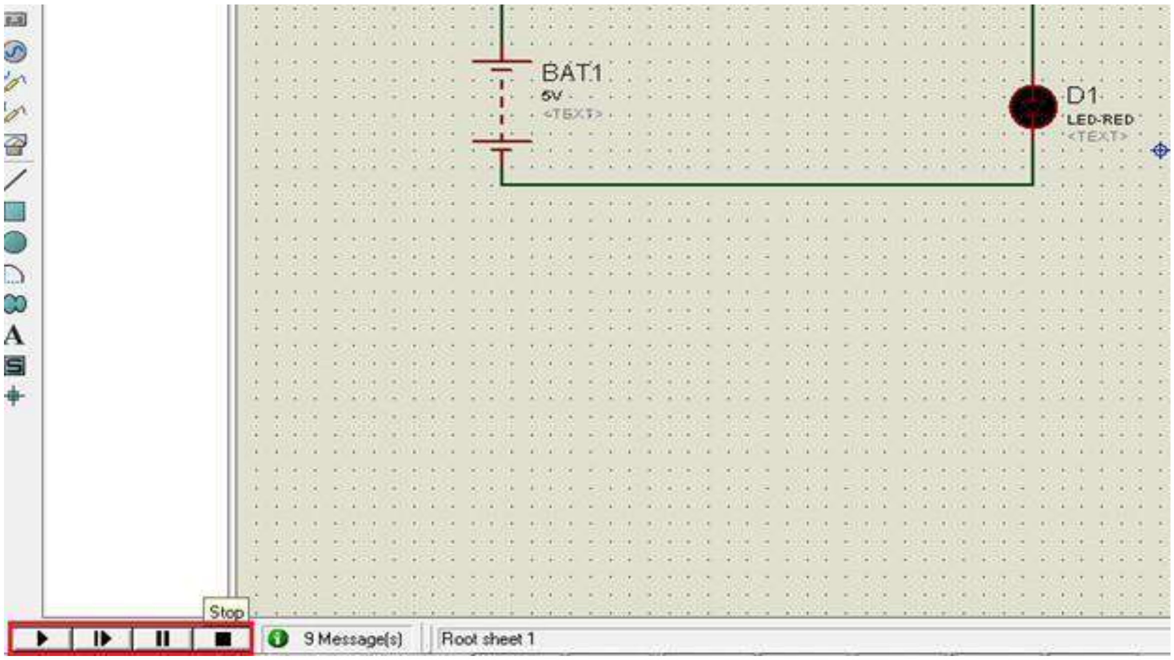


Fig. 3.2.7.13 Simulation Step-Pause-Stop Buttons

According to its developers, Things Speak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. Things Speak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates.

Things Speak was originally launched by iot Bridge in 2010 as a service in support of IoT applications.

Things Speak has integrated support from the numerical computing software MATLAB from MathWorks, allowing Things Speak users to analyse and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license from Math works.[citation needed].

Things Speak has a close relationship with Math works, Inc. In fact, all of the Things Speak documentation is incorporated into the Math works' Matlab documentation site and even enabling registered Math works user accounts as valid login credentials on the Things Speak website. The terms of service and privacy policy of ThingSpeak.com are between the agreeing user and Math works, Inc.

Setup Things Speak

Create a free MathWorks account or sign into Things Speak using an existing account.

Select the Things Speak channel you want your data to stream into. See Collect Data in a New Channel for help creating a new channel.

Record the following for the selected channel:

Channel ID, which is listed at the top of the channel view.

Write API key, which can be found on the API Keys tab of your channel view.

3.2.8.1 TEMPERATURE SENSOR:

Temperature Sensor which converts temperature value into electrical signals. We used IC called LM 35 as a temperature sensor. LM35 series sensors are precision integrated-circuit temperature sensors whose output voltage is linearly proportional to the Celsius temperature. The LM35 requires no external calibration since it is internally calibrated. . The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°C over a full −55 to +150°C temperature range.

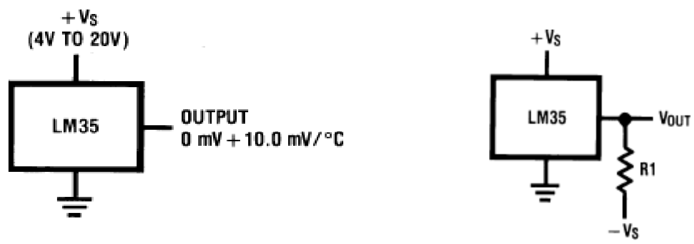
The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 μA from its supply, it has very low self-heating, less than 0.1°C in still air.

Features:

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteed (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications

- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 μA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±1⁄4°C typical
- Low impedance output, 0.1 W for 1 mA load

Figure 3.2.8.1: Temperature Sensor



The characteristic of this LM35 sensor is:

For each degree of centigrade temperature it outputs 10milli volts. ADC accepts the output from LM35 and converts that data into digital form which is sent to microcontroller for further processing.

TEMPERATURE SENSING CIRCUIT:

The methods of temperature measurement may be divided into two main classes according as the exchange of heat between the testing body and the hot system takes place by contact, or by radiation across a space. In the contact methods, thermometers or thermocouples are used and they are immersed in solids or liquids. The thermodynamic equilibrium between the hot body and the testing body is established by material contact.

In the non-contact methods, the thermodynamic equilibrium is established by the radiation emitted as excited atom and molecules in the hot body return to the ground state.

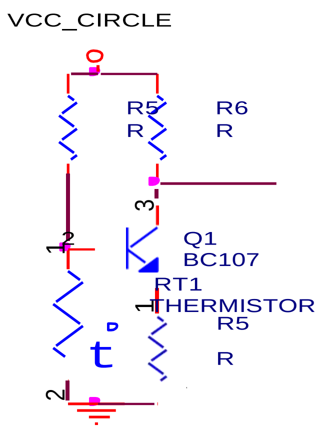
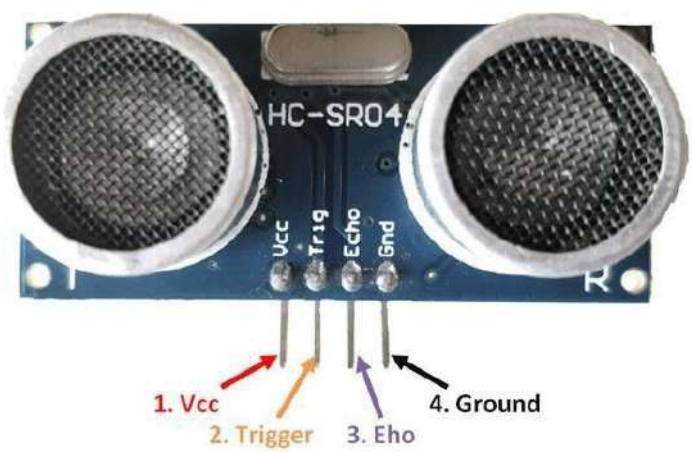


Figure 3.2.8.1(A). Temperature Sensing Circuit

3.2.9.1 ULTRA SONIC SENSOR :

HC-SR04 Ultrasonic Sensor



Ultrasonic Sensor HC SR04

Ultrasonic Sensor Pin Configuration

Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

3.2.9.2 HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz

3.2.9.3 HC-SR04 Ultrasonic Sensor - Working

As shown above the **HC-SR04 Ultrasonic (US) sensor** is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that,**Distance = Speed × Time.**The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below

Fig3.2.9.3 Ultrasonic Sensor

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

3.2.9.4 How to use the HC-SR04 Ultrasonic Sensor:

HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used.

Power the Sensor using a regulated +5V through the Vcc ad Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor.

The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information the distance is measured as explained in the above heading.

Applications:

- Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
- Used to measure the distance within a wide range of 2cm to 400cm
- Can be used to map the objects surrounding the sensor by rotating it
- Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water

2D model of the ultra sonic

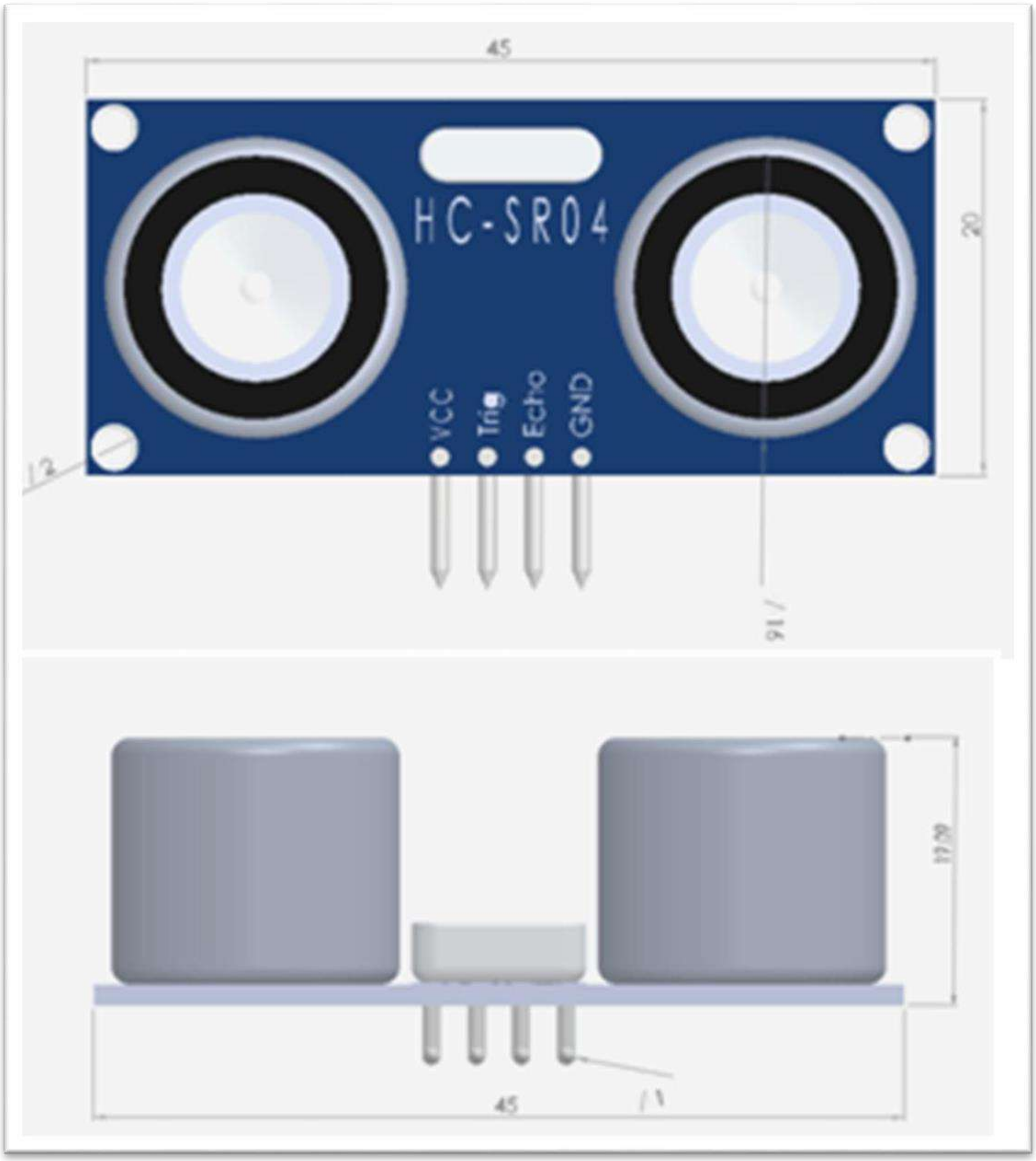


Fig3.2.9.5:2D model of the ultra sonic

3.3.1 Flow Sensor:

Flow sensor to detect leakage in a pipe is used for flow measurement. Precise measurement of the flow is an important step in qualitative and economic terms. The panel sits next to the water line and includes a panel to calculate the volume of water that has passed through. An integrated Hall-Effect magnet sensor generates an electric pulse with every turn.

Water Flow Sensor is an instrument that uses Hall effect to measure linear, nonlinear, The mass or volumetric flow rate of a liquid or a gas. The flow sensor consists of a plastic valve body, a rotor, and a hall-effect sensor. When water or gas flows through the rotor, the rotor rotates, the speed of the rotor changes with different rate of flow. The hall-effect sensor records these rotations as pulses and according to the pulses outputs the amount of water or gas flowing through the sensor. It operates on a maximum working voltage of 5V and minimum of 4.5V.

Working Principle

Water flow sensor consists of a plastic valve from which water can pass. A water rotor along with a hall effect sensor is present the sense and measure the water flow.

When water flows through the valve it rotates the rotor. By this, the change can be observed in the speed of the motor. This change is calculated as output as a pulse signal by the hall effect sensor. Thus, the rate of flow of water can be measured.

The main working principle behind the working of this sensor is the Hall effect. According to this principle, in this sensor, a voltage difference is induced in the conductor due to the rotation of the rotor. This induced voltage difference is transverse to the electric current.

When the moving fan is rotated due to the flow of water, it rotates the rotor which induces the voltage. This induced voltage is measured by the hall effect sensor and displayed on the LCD display.

The water flow sensor can be used with hot waters, cold waters, warm waters, clean water, and dirty water also. These sensors are available in different diameters, with different flow rate ranges.

These sensors can be easily interfaced with microcontrollers like Arduino. For this, an Arduino microcontroller board for processing, a Hall effect water flow sensor, a 16×2 LCD display, and Breadboard connecting wires are required. The sensor is placed at the water source inlet or at the opening of the pipe.

The sensor contains three wires. Red wire to connect with supply voltage. Black wire to connect to ground and a yellow wire to collect output from Hall effect sensor. For supply voltage 5V to 18V of DC is required.

3.3.2 ESP8266 WIFI MODULE:

3.3.2.1 INTRODUCTION

The **ESP8266 Wi-Fi Module** is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your **Wi-Fi** network. The **ESP8266** is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

3.3.2.2 CHARACTER CONFIGURATION:

- 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2),
- general-purpose input/output (16GPIO),
- Inter-Integrated Circuit (I^2C) serial communication protocol,
- analog-to-digital conversion (10-bitADC)
- Serial Peripheral Interface (SPI) serial communication protocol,
- I^2S (Inter-IC Sound) interfaces with DMA(Direct Memory Access) (sharing pins with GPIO),
- UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2),and
- Pulse-width modulation (PWM).

3.3.2.3 APPLICATIONS:

- Smart Home Appliances
- Improved Home safety and security
- Home air quality and water quality monitoring

- Natural Language-based voice assistants.

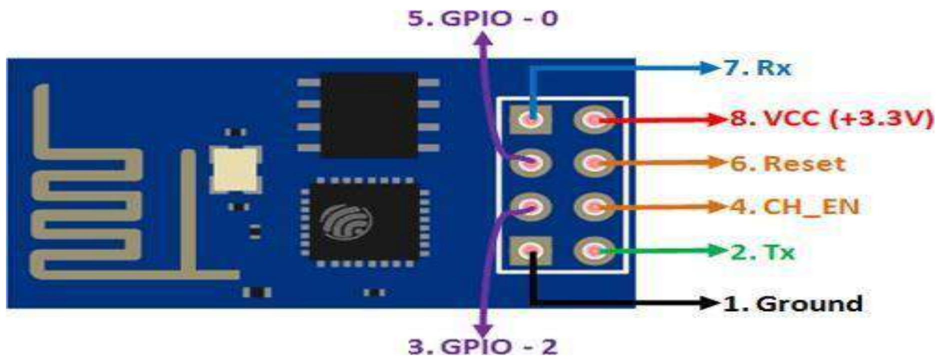


Fig.3.3.2.3 ESP8266 wifi module pin out

3.3.2.4 STRUCTURE AND CONFIGURATION:

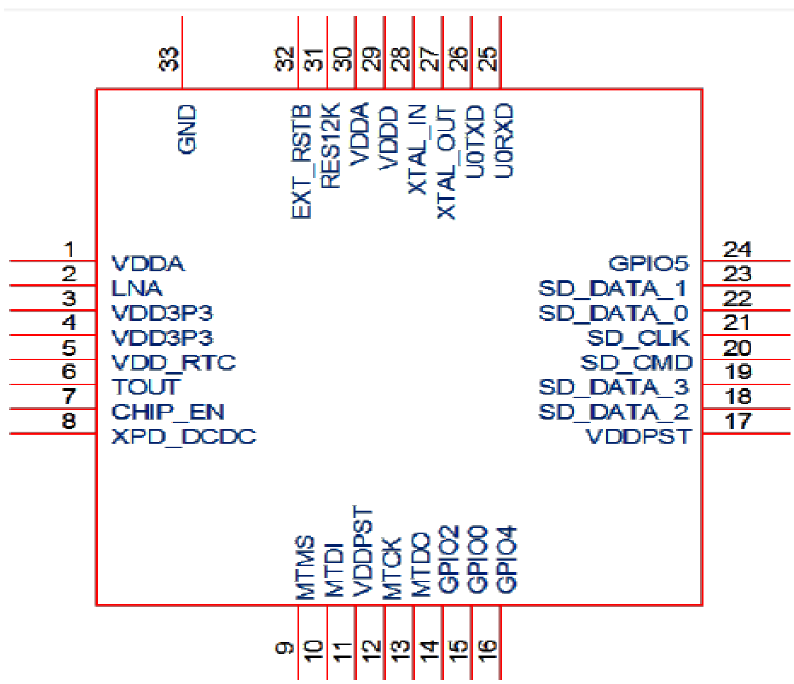


Fig.3.3.2.4 Structure and configuration

3.4.2 ESP8266 WIFI MODULE

The ESP8266 WiFi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all WiFi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existance interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support. In the Documents section below you will find many resources to aid you in using the ESP8266, even instructions on how to transform this module into an IoT (Internet of Things) solution.

CHAPTER-4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM OF PROJECT:

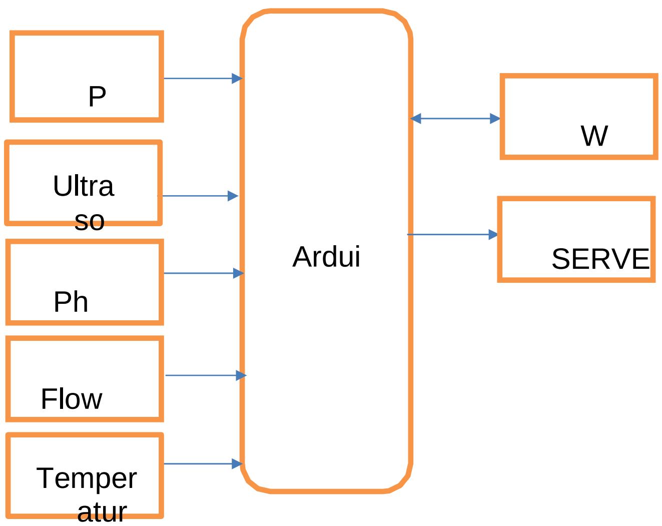


Fig4.0:Block diagram of project

4.2 FLOW CHART

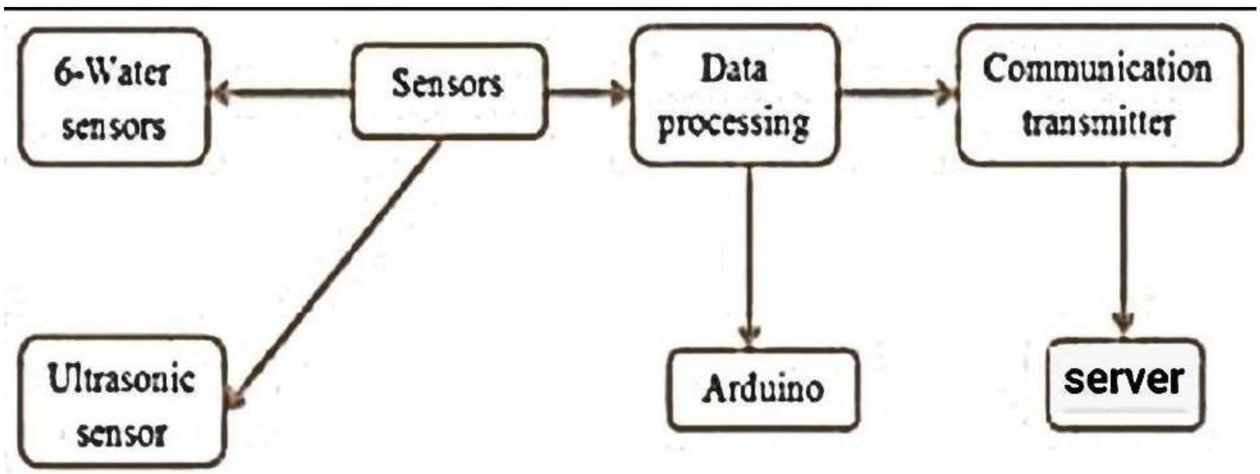


Fig.4.2 Flow Chart

4.3 IMPLEMENTATION STAGES:

In the figure we, notice that using server we can observe the readings in the Arduino here, in the display we can clearly identify that readings for our project in which D1-ultra sonic sensor, F-Flow sensor, PH - sensor, T-temperature. All the information is directly send this information to the Arduino through server, here we can clearly identify the readings or values of the components.

4.4 PROJECT CODE

```
#include <LiquidCrystal.h>
 LiquidCrystal lcd(13,12,11,10,9,8);
 #include <SoftwareSerial.h>
 SoftwareSerial mySerial(2,3);
  //adc
 int temp = A0,flow = A1,ultrasonic = A2,ph = A3;
char res[130];
 //pins

////////////////////
void serialFlush(){
  while(Serial.available() > 0) {
    char t = Serial.read();
  }
}

void myserialFlush(){
  while(mySerial.available() > 0) {
  char t = mySerial.read();
  }
}

char check(char* ex,int timeout)
{
```

```
int i=0;
int j = 0,k=0;
while (1)
{
  sl:
  if(mySerial.available() > 0)
  {
    res[i] = mySerial.read();
    if(res[i] == 0x0a || res[i]=='>' || i == 100)
    {
      i++;
      res[i] = 0;break;
    }
    i++;
  }
  j++;
  if(j == 30000)
  {
    k++;
    //Serial.println("kk");
    j = 0;
  }
  if(k > timeout)
  {
   // Serial.println("timeout");
    return 1;
   }
}//while 1
if(!strncmp(ex,res,strlen(ex)))
{
  //Serial.println("ok..");
```

```
    return 0;
   }
  else
  {
   // Serial.print("Wrong  ");
   // Serial.println(res);
    i=0;
    goto sl;
   }
}
char buff[200],k=0;
int  phvalue=0;
void upload1();


const char* ssid = "project";
const char* password = "project1235";
int T;
int tt;
void setup() {
  int  i=0;
  char ret;
    pinMode (temp, INPUT);
    pinMode (flowsensor, INPUT);
    pinMode (echopin, INPUT );
    pinMode (Ph,INPUT);
    pinMode (temp, OUTPUT);
    pinMode (flowsensor, OUTPUT);
    pinMode (trigpin, OUTPUT);
    pinMode (ph,OUTPUT);
```

```
  Serial.begin(9600);
  mySerial.begin(115200);
  lcd.begin(16,2);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("WELCOME");
  delay(3000);
//  serialFlush();
//gsm
st:
  mySerial.println("ATE0");
  Serial.println("ATE0");
  ret = check((char*)"OK",50);
  mySerial.println("AT");
  Serial.println("AT");
  ret = check((char*)"OK",50);
  if(ret != 0)
  {
    delay(1000);
   goto st;
  }

  lcd.clear();lcd.setCursor(0, 0);lcd.print("CONNECTING");
  mySerial.println("AT+CWMODE=1");
  Serial.println("AT+CWMODE=1");
   ret = check((char*)"OK",50);
 cagain:

  myserialFlush();
  Serial.print("AT+CWJAP=\"");
  mySerial.print("AT+CWJAP=\"");
  mySerial.print(ssid);
  Serial.print(ssid);
```

```
  mySerial.print("\",\"");
  Serial.print("\",\"");
  mySerial.print(password);
  Serial.print(password);
  mySerial.println("\"");
  Serial.println("\"");
  if(check((char*)"OK",300))goto cagain;
  mySerial.println("AT+CIPMUX=1");
  Serial.println("AT+CIPMUX=1");
  delay(1000);

 lcd.clear();lcd.setCursor(0, 0);lcd.print("WAITING");
}
void flow () // Interrupt function
{
   flow_frequency++;
}
void setup()
{
   pinMode(flowsensor, INPUT);
   digitalWrite(flowsensor, HIGH); // Optional Internal Pull-Up
   Serial.begin(9600);
   lcd.begin(16, 2);
   attachInterrupt(digitalPinToInterrupt(flowsensor), flow, RISING); // Setup Interrupt
   lcd.clear();
   lcd.setCursor(0,0);
   lcd.print("Water Flow Meter");
   lcd.setCursor(0,1);
   lcd.print("Circuit Digest");
   currentTime = millis();
   cloopTime = currentTime;
```

```
}
void loop ()
{
  currentTime = millis();
  // Every second, calculate and print litres/hour
  if(currentTime >= (cloopTime + 1000))
  {
   cloopTime = currentTime; // Updates cloopTime
   if(flow_frequency != 0){
     // Pulse frequency (Hz) = 7.5Q, Q is flow rate in L/min.
     l_minute = (flow_frequency / 7.5); // (Pulse frequency x 60 min) / 7.5Q = flowrate in L/hour
     lcd.clear();
     lcd.setCursor(0,0);
     lcd.print("Rate: ");
     lcd.print(l_minute);
     lcd.print(" L/M");
     l_minute = l_minute/60;
     lcd.setCursor(0,1);
     vol = vol +l_minute;
     lcd.print("Vol:");
     lcd.print(vol);
     lcd.print(" L");
     flow_frequency = 0; // Reset Counter
     Serial.print(l_minute, DEC); // Print litres/hour
     Serial.println(" L/Sec");
   }
   else {
     Serial.println(" flow rate = 0 ");
     lcd.clear();
     lcd.setCursor(0,0);
     lcd.print("Rate: ");
```

```
     lcd.print( flow_frequency );
     lcd.print(" L/M");
     lcd.setCursor(0,1);
     lcd.print("Vol:");
     lcd.print(vol);
     lcd.print(" L");
    }
   }
}
#define echopin 9
#define trigpin 8

int maximumRange = 105;
long duration;
float distance;

void setup()
{
  Serial.begin (9600);
  pinMode (trigpin, OUTPUT);
  pinMode (echopin, INPUT );
  pinMode (A0, OUTPUT); //0% indicator


}
void loop ()
{
    digitalWrite(trigpin,LOW);
    delayMicroseconds(2);
    digitalWrite(trigpin,HIGH);
    delayMicroseconds(10);
    duration=pulseIn (echopin,HIGH);
```

```
   distance= duration/58.2;
   delay(200);
   Serial.print(distance);
   Serial.println("cm");
   actual_distance = maximumRange - distance;
   v = ((3.14*(r*r))*(actual_height)); // formula to calculate volume in cubic cm

 if (distance < 0) // LEAK DETECTED
   {
   digitalWrite(A2,HIGH);
   //digitalWrite(6,LOW);
   }
   else
   {
   digitalWrite(A2,LOW);
   }
}

void loop() {
  for(int i=1; i<=10; i++) {
      String phdata = Serial.readStringUntil(':');
      Serial.println(phdata);
      if(phdata != ""){
         String ph = Serial.readStringUntil('$');

         Serial.println(ph);
         phvalue=ph.toFloat();
         Serial.println();
         Serial.println("PH Value");
         Serial.println(phvalue);
         delay(1000);}
```

```
  }
  int td = analogRead(temp)/4;
  int fd = digitalRead(flow);
  int ud = analogRead(ultrasonic);

  Serial.print(td);
  Serial.print("\r\n");
  Serial.print(ud);
  Serial.print("\r\n");
  lcd.setCursor(0, 0);lcd.print("T:");lcd.print(td);lcd.print("    ");
  lcd.setCursor(5, 0);lcd.print("F:");lcd.print(sd);lcd.print("    ");
  lcd.setCursor(10, 0);lcd.print("U:");lcd.print(wd);lcd.print("    ");
  lcd.setCursor(0, 1);lcd.print("PH:");lcd.print(phvalue);lcd.print("    ");
  delay(1000);
///////////////// ldr1 start //////////
if(td > 35 )
{
  lcd.clear();
  lcd.setCursor(0, 0);lcd.print("HIGH TEMPRATURE");
  delay(3000);
  upload1(td,fd,ud,phvalue);
  delay(500);
}

if(fd > 100 )
{
  lcd.clear();
  lcd.setCursor(0, 0);lcd.print("WATER FLOW HIGH");
  delay(3000);

 upload1(td,fd,ud,phvalue);
```

```
  delay(500);

}
if(ud < 105 )
{
  lcd.clear();
  lcd.setCursor(0, 0);lcd.print("NO LEAKAGE");
  delay(3000);
 upload1(td,fd,ud,phvalue);
  delay(500);

}
if(Pd < 5 )
{
  lcd.clear();
  lcd.setCursor(0, 0);lcd.print("ACIDIC");
  delay(3000);

 upload1(td,sd,wd,phvalue);
  delay(500);

}
else
{
 delay(4000);

upload1(td,fd,ud,phvalue);

}

}//loop
```

```
char bf2[100];
void upload1(unsigned char *chr ,unsigned char *chr1,unsigned char *chr2,unsigned char
*chr3)
{
  delay(2000);
  lcd.clear();lcd.setCursor(0, 1);lcd.print("UPLOADING");
  myserialFlush();
  mySerial.println("AT+CIPSTART=4,\"TCP\",\"api.thingspeak.com\",80");
 // Serial.println("AT+CIPSTART=4,\"TCP\",\"embeddedspot.top\",80");
      delay(8000);
      sprintf(buff,"GET
https://api.thingspeak.com/update?api_key=ZHR6DIPXM6G1SBYN&field1=0=%u&field2=%
u\r\n\r\n",chr,chr1,chr2,chr3);
      //sprintf(buff,"GET
http://embeddedspot.top/iot/storedata.php?name=sensors010&s1=%u\r\n\r\n",chr);
      myserialFlush();
      sprintf(bf2,"AT+CIPSEND=4,%u",strlen(buff));
      mySerial.println(bf2);

         delay(5000);

          myserialFlush();
          mySerial.print(buff);
          Serial.print(buff);
              delay(2000);

              mySerial.println("AT+CIPCLOSE");
              Serial.println("AT+CIPCLOSE");
              lcd.setCursor(0, 1);lcd.print("UPLOADDED");  lcd.clear();
}
```

4.5 RESULTS:

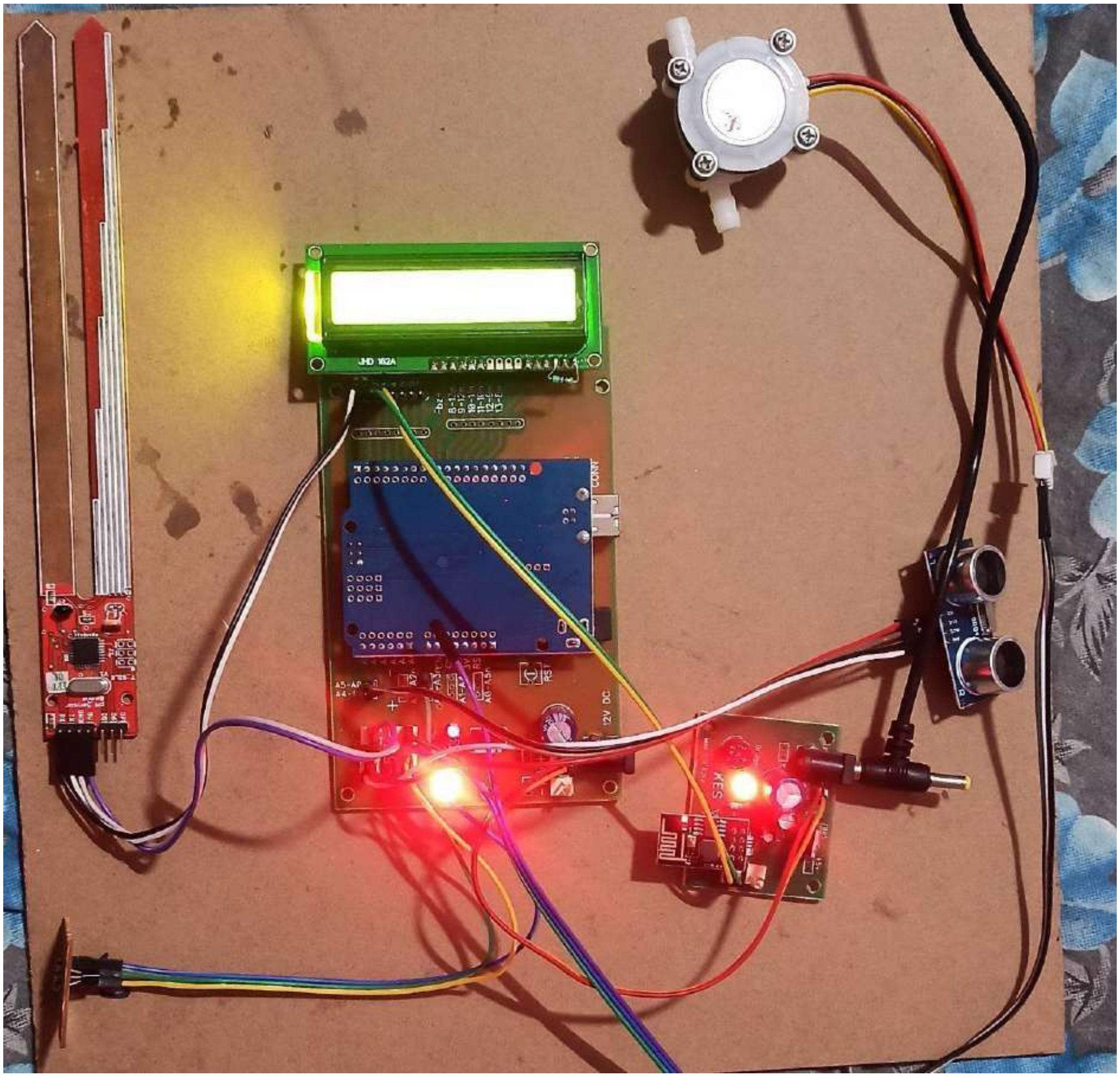


Fig.4.5 Final Output

CHAPTER-5

PROJECT TESTING

5.1 OVERVIEW OF TESTING METHODS

- As people continues to lay more pipelines, the asset base grows, which translates to more potential points of failure. As we're adopting a proactive and systematic approach to manage the asset base in order to optimize the lifespan of existing pipelines and pre-empt failures of critical pipelines To facilitate the implementation of the new asset management system, we will leverage widely on condition assessment technologies to retrieve more dynamic and real time data. The challenge is that pipelines are buried, lined & have undulating profiles. In addition, the network is interconnected & "live" 24/7 reliable supply is required. This makes it difficult to implement condition assessment technologies on a large scale. However, once these challenges are surmounted, these data can be uploaded into an asset management software to improve the identification of high-risk pipelines so that pipeline renewal works can be prioritized. With a better understanding of the circumstances leading to pipe failures, we can also improve pipeline designs to prevent future failures. Leveraging on a Smart Water Grid system for asset management thus allows us to plan, design, operate and manage assets cost-effectively for safe and reliable supply to customers.

- Our project goal is to monitor and model network water quality in real-time, so as to detect contamination early and control its spread to minimize impact to customers. There is a need to move away from depending on customers to act as sensors for water quality issues like discolored water. Furthermore, in today's volatile socio-political climate, we need to be even more vigilant to deter and prevent acts of sabotage that may threaten the quality of the water supply.

- We’re exploring how real time information and feedback from smart water-efficient gadgets/fittings & technologies could drive water consumption behavioral changes of domestic customers. Trust and data protection is important as water consumption data could be sensitive and personal. In addition, cost effectiveness is a major consideration for any home water management solutions and it is often challenging to justify the cost effectiveness of smart water-efficient gadgets

- However, with increasing public expectations, there is a need to find methods to further reduce response time towards leak incidents. We aims to deploy more advanced leak detection equipment/systems to continuously monitor and analyse leaks, thereby improving response time and minimizing impact to customers. With more leak monitoring systems installed island-wide, there can be significant reduction in the manpower which is normally required to perform the routine site inspection.

CHAPTER-6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

In order to save water for the future world, the IoT Based Water Saving and Leakage Detection and Manager for apartments and homes has been built. The findings of this analysis International Journal of Advanced Science and Technology Vol. 29, No. 9s, (2020), pp. 683-689 ISSN: 2005-4238 IJAST Copyright ⓒ 2020 SERSC 688 study are presented and discussed in various parts on the basis of the framework for research methodology. Important results of each study, including the most promising wearable devices and sensors for building safety surveillance applications and trends are presented. We have seen how this architecture can be used to handle actual water management systems. However, organizational processes need to be clearly defined to resolve other specific problems, such as physical network specification or identity mapping. In conclusion, we explain the scenario we identify for validation and the list of features that we will test at this station established an experimental station. We may conclude that IoT adoption and water management companies can promote exposure to a wider global market and will bring new benefits to support systems decisions, monitoring, water storage, and water energy ties. Future research will define the test carried out and will focus on solving coordination issues in multiple recipes using the same physical resources, taking into account priority and conditional output and also optimization of processes.

6.2FUTURE ENHANCEMENT

- In future we use IOT concept in this project
- Detecting the more parameters for most secure purpose
- Increase the parameters by addition of multiple sensors
- By interfacing relay we controls the supply of water

PUBLICATION

- Submitted Paper at Online Mega International Conference on “Smart Modernistic in Electronics and Communication” (ICSMEC – 21).

- Got Acceptance for the Paper with Paper ID (ICSMEC21-0024).

REFERENCES

[1] Bhad Vidya, Kale Poonam, Gavhale Priyanka, Darekar Gaurav, A.S Chandgude, ”Water Level Monitoring System In Real Time Mode Using WSN”, International Journal of Emerging Technology and Advanced Engineering (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 9, September 2016.

[2] Mithila Barabde, Shruti Danve, “Real Time Water Quality Monitoring System”, International Journal of Innovative Research in Computer and Communication Engineering Vol. 3, Issue 6, June 2015.

[3] Jaytibhatt, jigneshpatoliya, ”IoT based water quality monitoring system”, International Journal Of Industrial Electronics And Electrical Engineering Volume-4, Issue-4, Apr.- 2016.

[4] Cho Zin Myint, Lenin Gopal, and Yan Lin Aung, “Reconfigurable Smart Water Quality Monitoring System in IoT Environmet”, ACIS 16th International Conference on Computer and Information Science (ICIS), IEEE xplore, ISBN: 978-1-5090-5507-4.

[5] Anthony Faustine , Aloys N. Mvuma, Hector J. Mongi, Maria C. Gabriel, Albino J. Tenge, Samuel B. Kucel, “Wireless Sensor Networks for Water Quality Monitoring and Control within Lake Victoria Basin: Prototype Development”, Wireless Sensor Network, 2014, 6, 281-290.

[6] Ali M. Sadeghioon, Nicole Metje, David N. Chapman and Carl J. Anthony, “SmartPipes: Smart Wireless Sensor Networks for Leak Detection in Water Pipelines”, Journal of Sensor and Actuator Networks ISSN 2224-2708.

[7] A EjahUmraeni Salam, Muh.Tola1, Mary Selintung and Farouk Maricar," On-Line Monitoring System Of Water Leakage Detection In Pipe Networks With Artificial Intelligence", ARPN Journal of Engineering and Applied Sciences VOL. 9, NO. 10, OCTOBER 2014.

[8] J.Navarajan, B. Aswinkumar, S. Venkatesh , T. Jayachandran, "Detection of Water Pollution and Water Management Using Smart Sensors with IOT", International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 04 | Apr - 2017.

[9] Behera, S. K., & Gupta, M. K. (2019). Implementation of IOT for energy management. Test Engineering and Management, 81(11-12), 4856-4860.

[10] Goar, V. K., Tanwar, P., & Kuri, M. (2019). IoT based climate-smart agriculture. Test Engineering and Management, 81(11-12), 6620-6624.

[11] S. Geetha and S. Gouthami ,"Internet of things enabled real time water quality monitoring system", Department of Electrical and Electronics Engineering", (2016) 2: 1. https://doi.org/10.1186/s40713-017-0005-y.

[12] M. Thilagaraj, N.N. Swetha, R. Pugazhendhi, R. Rahul, "Finger Vein Based Bank Security System," International Journal of Control and Automation, Vol 13, No 3, 2020, pp 01-08.International Journal of Advanced Science and Technology Vol. 29, No. 9s, (2020), pp. 683-689 ISSN: 2005-4238 IJAST Copyright ⓒ 2020 SERSC 689

[13] M. Thilagaraj, M. Pallikonda Rajasekaran, "Classification of Non alcoholic and alcoholic based EEG Signal using Fuzzy Neural Network Classifier", Journal of Advanced Research in Dynamical and Control Systems – Vol -9, Sp-16/2017 pp 671- 680

[14] M. Thilagaraj and M. Pallikonda Rajasekaran, "Epileptic Seizure Mining via Novel Empirical Wavelet Feature with J48 and KNN Classifier", Intelligent Engineering Informatics, Advances in Intelligent Systems and Computing 695, https://doi.org/10.1007/978-981-10-7566-7_23, 2018

[15] M. Thilagaraj, M. Deepak, M. Mohanraj, K. Kannappan, M. Pallikonda Rajasekaran, "Modern day Traffic Light Monitoring System with Implementation of Internet of Things (IoT),"International Journal of Control and Automation, Vol 13, No 3, 2020, pp 17-22.

[16] M. Pradeep Kumar, M. Thilagaraj, S. Sakthivel, C. Maduraiveeran, M. Pallikonda Rajasekaran and S. Rama, "Sign Language Translator Using LabVIEW Enabled with Internet of Things, Smart Innovation, Systems and Technologies 104, https://doi.org/10.1007/978-981-13-1921-1_59, 2018

BIBILOGRAPHY

© 2015 International Institute for Sustainable Development Published by the International Institute for Sustainable **Development INTERNATIONAL INSTITUTE FOR SUSTAINABLE DEVELOPMENT** The International Institute for Sustainable Development (IISD) is one of the world's leading centres of research and innovation. The Institute provides practical solutions to the growing challenges and opportunities of integrating environmental and social priorities with economic development. We report on international negotiations and share knowledge gained through collaborative projects, resulting in more rigorous research, stronger global networks, and better engagement among researchers, citizens, businesses and policymakers. IISD is registered as a charitable organization in Canada and has 501(c)(3) status in the United States. IISD receives core operating support from the Government of Canada, provided through the International Development Research Centre (IDRC) and from the Province of Manitoba. The Institute receives project funding from numerous governments inside and outside Canada, United Nations agencies, foundations, the private sector, and individuals. Water Quality Monitoring System Design September 2015 Written by Carter Borden and Dimple Roy Head Office 111 Lombard Avenue, Suite 325 Winnipeg, Manitoba .

APPENDICES

The project shows functionalities of leakages and the quality of water using the concept of 'Internet of things'(IOT) based on Arduino in reference to a survey conducted by Government of USGS in the year 2018 on Monitoring Quality of Water. We examine and report on the conditions of the leakage of the pipelines based on the local water quality , to provide them optimal solution and we consent with the idea of making a System for Monitoring the Water Aspects and Leaks.

A

MAJOR PROJECT REPORT

On

Automatic Power Saving With The Help Of IOT

Submitted by

Mr. Astik Achary (16K81A0402)

Mr. Buddavarapu Krishna Vardhan (16K81A0467)

Mr. B. Shravan Babu (16K81A04J5)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under The Esteemed Guidance of

Mr. G. Ramesh Reddy

Associate Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST. MARTIN'S ENGINEERING COLLEGE

An Autonomous Institute

Dhulapally, Secunderabad – 500 100

JUNE 2021

BONAFIDE CERTIFICATE

This is to certify that the project entitled “AUTOMATIC POWER SAVING WITH THE HELP OF IOT”, is being submitted by **Mr. Astik Achary (16K81A0402), Mr. Buddavarapu Krishna Vardhan (16K81A0467) and Mr. B. Shravan Babu (16K81A04J5)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** is recorded of bonafide work carried out by her. The result embodied in this report have been verified and found satisfactory.

PROJECT GUIDE	**HEAD OF THE DEPARTMENT**
Mr. G. Ramesh Reddy	**Dr. B. HARI KRISHNA**
Associate Professor	**Professor**
Department of ECE	**Department of ECE**

Internal Examiner External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

\

THIS IS TO CERTIFY THAT ASTIK ACHARY WITH ROLL NO.16K81A0402, BUDDAVARAPU KRISHNA VARDHAN WITH ROLL NO.16K81A0467 AND B.SHRAVAN BABU WITH ROLL NO.16K81A0402 OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, SHE HAS SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "AUTOMATIC POWER SAVING WITH THE HELP OF IOT" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH HER SUCCESS IN HER FUTURE ENDEAVORS.

Lasya IT Solutions Pvt. Ltd. ★ Kompally-500014 ★

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

I, the student of **Bachelor of Technology** in Department of ELECTRONICS AND COMMUNICATION ENGINEERING, session: <2017 – 2021>, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled "AUTOMATIC POWER SAVING WITH THE HELP OF IOT" is the outcome of my own bonafide work and is correct to the best of my knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

Mr. Astik Achary (16K81A0402)

Mr. Buddavarapu Krishna Vardhan (16K81A0467)

Mr. B. Shravan Babu (16K81A04J5)

ACKNOWLEDGEMENTS

The satisfaction and mpletion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success

First and foremost, I would like to express our deep sense of gratitude and indebtedness to my **College Management** for their kind support and permission to use the facilities available in the Institute.

I especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

I would like to express my gratitude to **Dr.B.HARI KRISHNA**, **Head &Professor**, Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. I am proud to record that I had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

I would like to express my gratitude to my dept. project coordinator **Dr.A.ANAND, Professor** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

.

I sincerely thank my internal guide Mr. G. Ramesh Reddy, Assistant professor, De partment of ECE and my Class project Coordinator Ms. P. Pushpa, Assistant Professor , Department of ECE for guidance and encouragement in carrying out this project.

Finally, I am highly grateful and obliged to my project guide **Mr.S.P.MANIKANTA, Associate Professor,** for his advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

Astik Achary (16K81A0402)

Buddavarapu Krishna Vardhan (16K81A0467)

B. Shravan Babu (16K81A04J5)

ABSTRACT

Humans have been communicating with each other for ages in order to solve problems and do complex work but with the advancement of technology. IOT or internet of things promises a great future for the communication between machines that can be implemented to perform many tasks that can benefit the human community. Many problems are saving electricity, which is one of the major responsibilities of everyone yet few people could actually do so be it carelessness or lack of time. This paper aims to solve the problem of electricity wastage in school and college classrooms by using an automatic electricity control system for a room made by applying IOT sensors and microcontrollers.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY OF COMMONLY USED TERMS:

Active Tag – An RFID tag that uses a battery to power its microchip and communicate with a reader. Active tags can transmit over the greatest distances (100+ feet). Typically they can cost $20.00 or more and are used to track high value goods like vehicles and large containers of merchandise.

Agile Reader – A reader that can read different types of RFID tags – either made by different manufacturers or operating on different frequencies.

Antenna – A device for sending or receiving electromagnetic waves.

Anti-Collision – A feature of RFID systems that enables a batch of tags to be read in one reader field by preventing the radio waves from interfering with one another. It also prevents individual tags from being read more than once.

Automatic Data Capture (ADC) – Methods of collecting data and entering it directly into a computer system without human intervention. Automatic Identification (Auto-ID) Refers to any technologies for capturing and processing data into a computer system without using a keyboard. Includes bar coding, RFID and voice recognition.

Auto-ID Center – A group of potential RFID end users, technology companies and academia. The Auto-ID center began at the Massachusetts Institute of Technology (MIT) and is now a global entity. It is focused on driving the commercialization of ultra-low cost RFID solutions that use Internet like infrastructure for tracking goods throughout the global supply chain. The Auto-ID Center organization is now EPC global.

Electronic Product Code (EPC)

A standard format for a 96-bit code that was developed by the Auto-ID Center. It is designed to enable identification of products down to the unique item level. EPC's have memory allocated for the product manufacturer, product category and the individual item. The benefit of EPC's over traditional bar codes is their ability to be read without line of sight and their ability to track down to the individual item versus at the SKU level.

EPC global – The association of companies that are working together to set standards for RFID in the retail supply chain. EPC global is a joint venture between EAN International and the Uniform Code Council, Inc.

Radio Frequency Identification (RFID)

A method of identifying items uniquely using radio waves. Radio waves do not require line of site and can pass through materials like cardboard and plastic but not metals and some liquids.

Read Range

The distance from which a reader can communicate with a tag. Several factors including frequency used orientation of the tag, power of the reader and design of the antenna affect range.

Reader

Also called an interrogator. The RFID reader communicates via radio waves with the RFID tag and passes information in digital form to the computer system. Readers can be configured with antennas in many formats including handheld devices, portals or conveyor mounted.

Read Only Tags

Tags that contain data that cannot be changed. Read only chips are less expensive than read-write chips

Read-Write Tags

RFID chips that can be read and written multiple times. Read/Write tags can accept data at various points along the distribution cycle. This may include transaction data at the retail point of sale. They are typically more expensive than read only tags but offer more flexibility.

RFID Transponder

Another name for a RFID tag. Typically refers to a microchip that is attached to an antenna, which communicates with a reader via radio waves. RFID tags contain serial numbers that are permanently encoded, and which allow them to be uniquely id

"WORM" Chip (Write Once Read Many) and then becomes "Read only" afterwards

CHAPTER-1
INTRODUCTION

1.1 OVERVIEW OF THE PROJECT:

With the growth in the number of vehicles the need for expansive roads catering to thousands of vehicles moving across India has become inevitable. However, considering the present situation the current toll system has several drawbacks. Due to the limited number of toll booths and slow collection process, the average waiting time per vehicle is 10 minutes. This results in losses worth thousands of crores of Rupees in terms of fuel wastage. This long wait time often results in drivers getting irritated resulting in verbal spats and physical fights among people and the toll attendants. Several such incidents have been reported in the press with some of these fights even resulting in the death of the toll plaza attendants. In addition, there are numerous cases of toll plaza accidents which happen due to the sudden lane changing by drivers for faster clearance. The major reason behind this is that, the security at the tolls is insufficient and it is beyond the traffic polices control to manage the vast number of vehicles. We keep hearing of many such incidents at toll plazas which mostly occur due to negligence either on the people’s side or due to lack of control from the government agencies including the police. In case of events, where lives are lost, such losses are a life shattering experience.

As is well known, in such a scenario, the general public is a little hesitant in taking responsibilities of any such incident. Hence the government has to come up with an effective plan which bridges the gap between the toll management and the public expectation of the service that they experience. Introduction of an elective toll plaza operation plan by the government, its strict implementation and monitoring which would result in a more efficient and a more responsive and efficient system could be a good option for easing the challenges associated with the existing tolling process.

The requirements for new web applications supporting different types of devices an purposes are continuously growing. The main advantages of web application development as well as popular development features covering integration with different technologies are considered.

Integration and possibilities of application of cloud-based web applications in real scenarios with different embedded Internet of Things (IoT) devices are considered and described. The design and implementation of a cloud-based web application supporting vehicle toll payment system using IoT device is presented and described. The development framework as well as featured and popular technologies used to realize a vehicle toll payment by IoT device are described. The concept of vehicle toll payment over an online payment system is also described. Processing, monitoring and control in the cloud-based web application of such payments using IoT devices are described and presented. Also, system allows to store data of all the vehicles passed at particular time intervals for later reference and surveillance. This system thus automates the entire toll booth collection and monitoring process with ease using RFID plus IoT based system.

1.2 OBJECTIVES OF THE STUDY:

The use of automated toll collection system in many metropolitan cities would be an efficient step towards the overcrowding of the city highways in heavy congestion of traffic. As we all know, transportation is the strength of our country's economy. There are various implementation, protocols in wireless sensor network and components such as RFID thus enabling reduction in operation costs and motivating cashless transactions. In case of manual toll collection system time consumption is much far worse as well as fuel depletion and most important is the environment, the amount of air pollution that is created at the toll booth site is at high level, so our developed system will reduce time wastage and not only reduce air pollution but also conserve fuel. The sole purpose of this paper is to reduce the hardships caused by manual toll collection system and pass the subject's vehicle through toll barrier in a matter of few seconds without halt.

Each and every day more and more vehicles are increasing rapidly and the graph of the rate of buying vehicles is exponential, which has become a major problem at the site of toll booths due to heavy traffic causing endless number of problems such as high petrol/diesel consumption leading to depletion of hydrocarbon deposits below earth's crust and also death causalities due to heavy traffic. The sole purpose of this paper is to motivate cashless transactions by installing automated e-toll collection system and the technology that we used is the use of RFID readers/tags. Basically to tackle this problem, the use of RFID tags that must be uniquely fixed onto subject's vehicle and RFID reader module must be fixed at e-toll tooth. When a subject's vehicle passes through the gate, the reader will detect an incoming frequency of 125 kHz of the RFID tag and read a unique no that has been assigned by govt. authority and the toll fee will be deducted from linked bank a/c or the e-wallet of system. This system is capable of saving time as well as fuel conservation which can save a lot of individual’s economy. This particular system is far much better and very efficient towards people as they will not stay in a long and lengthy queue thus automated e-toll system will eliminate the hardships of people parking vehicles in a long queue. RFID has the potential of eliminating corruption at local level and also reduce operational costs as well as errors in human operations. WSN’s i.e. wireless sensor networks are basically used in different scenarios such as home, office, healthcare, agriculture and also at toll collection plaza which can capture and transmit data from all incoming vehicles and outgoing vehicles because of their consistent and distinctive properties.

1.3 SCOPE OF THE STUDY:

- **IOT (internet of things)** based toll booth manager system is such type of a system that provides the automatic toll collection process with the help of internet resources and RFID (radio frequency identification) technology.

- If we talk about accountability, then manually toll collection process is very complicated because it is not so much easy to manage hundred and thousands of vehicles at same time.

- It required more man power as well as with more toll collection booth, which could be so much costly and difficult for toll collection department.

- On other hand, so much time is wasted by the vehicle owners when they are stand in so much long toll collection row.

- To manage all these above situations and to facilitate the vehicles owners, so many toll booth manager systems are available in market but that one’s are so much costly as well as they are not so much efficient, reliable, precise and friendly to use.

- Here we have proposed a system that is called IOT based toll booth manager system which is so much efficient, reliable and friendly to use. In this system every vehicle owner possess a RFID based card, which would be scanned this card during pass through toll collection both.

1.4 MATERIAL REQUIREMENT:

1.4.1 HARDWARE REQUIREMENT:

- Arduino uno
- LCD
- Power Supply
- RFID
- Buzzer
- WIFI
- L293D
- Motor

1.4.2 SOFTWARE REQUIREMENT:

- Proteus software
- Arduino software

1.5 PROCUMENT OF EQUIPMENT:

We brought all the required hardware components in an electronic manufacturing shop and before assembling and interfacing hardware components to Arduino UNO micro-controller, we connected individual components to Arduino in Proteus Software and understood working of every individual component in Proteus Software.

Now, we assembled all components and interfaced to Arduino UNO micro-controller using connecting wires through bread board and we wrote program to work all the hardware components in Arduino IDE we understood the working of all components in a practical way.

We were able to monitor the toll booth system with better efficiency, with less time delay and the results were also satisfactory and we successfully got the output in mobile telnet app and LCD screen. We would like to improve the project in future for further developments.

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE REVIEW ON RESEARCH AREA:

The ATC system in the paper "A review on automated toll collection system" the two types of study were compared namely the RFID based study and Leach-C based study showed promising and efficient way to develop the system but it also showed us lack of a point where the Leach-C system worked on digital image processing where the number plate of subject's vehicle was scanned using high-tech camera. Using the RFID based implementation and eliminating the DIP based implementation to make system better time efficient. The survey done in the paper named "A 5.8 GHz ISM band Microstrip antenna for RFID applications" showed us the path using and developing RFID antenna for scanning the details of subject's vehicles from a handy distance thus enhancing the distance of vehicle and RFID reader would cover a large area. Smart Toll Collection System based on IoT (IJSTE/ Volume 3 / Issue 12 / 002) All rights reserved by www.ijste.org 10 The review was done of the paper "Automated toll collection system using RFID" Enlighted us the various components that can used while developing our own system in time saving and in efficient way so as to make a hybrid system that is mentioned in the paper "A review on automated toll collection system". "A Survey in Image Mining" showed a path towards machine learning and data mining concepts and helped us in mining databases in the form of people's feedback.

2.2 REVIEW ON RELATED LITERATURE:

The first person to propose electronic toll system was William Vickrey. He proposed this system for Washington Metropolitan Area. He was also the recipient of Nobel Economics prize in 1959 [5] The transponder and reader technology spread all around Norway. The system was broached in the Bergen (1986). World"s first completely automated electronic toll collection system was introduced in Trondheim (1991). Norway too has electronic fee collection EFC. [5] In different places in the United States like California, Pennsylvania, Texas, Delaware and Florida, vehicles can pass through electronic toll collecting booths directly. For the same no of cars, operations in electronic toll booths take place faster as the users are preregistered. Toll collecting gates are usually broader than the original width of the road but this made it possible to fit them into smaller narrower roads. Although these limitations are present, if the time taken for operation at the toll booth is reduced, then the speed of operation can be increased. The greater the speed of operation, less number of total lanes was required and as more number of lanes started getting automated, there was a reduction in the overall short term cost of the project. Also if the long term cost of the project is considered, larger number of vehicles started using electronic toll and vehicles using manual toll reduced. [5] According to references, the methodology proposes use of Infrared sensors to gather real time data from the waste bins and that of the microcontroller board to communicate this information to the waste managers.

These sensors collect information about the objects, their surroundings and communicate this information to other stations, linked through wired or wireless networks. The system will check using the IR sensors when the bin becomes full and will notify the waste managers upon filling up of the bins and will provide an optimal and effective collection route. [10] According to reference, The initial technical realization of IoT was achieved by utilizing RFID (Radio Frequency Identification) technology, which was limited to identification, object tracking and extracting information of specific objects. However, The implemented IoT based system performs sensing, actuating, data gathering, storing, and processing by connecting physical or virtual devices to the Internet. [10] According to reference, To facilitate vehicle monitoring, toll collection and faithful vehicle authentication on the highways and to have an efficient usage of communication link between RF Modems over a wireless channel, a module is proposed. There are two types of implementation modules-the Vehicle Module (Active RFID Tag) and the Base Module. Microcontroller contains user-specific information associated with vehicle, such as the owners information with his billing address, vehicle engine number and vehicle registration number. The base module allows the base module to check the activities of vehicles in range, including the vehicles in range, their status, and the detailed information about any registered vehicle.[6] According to reference, Open Road Tolling uses video evidence to identify vehicle usage of a toll facility without the use of toll booths for toll collection without having to stop or even slow down to pay the toll.

The application, for Open Road Tolling, utilizes pattern recognition and image processing methods. This paper presents Open Road Tolling (ORT) using number plate recognition. The proposed Number Plate Recognition (NPR) techniques consist of mainly two modules: histogram based number plate localization and number plate recognition using template matching, thus making it simple & faster. AVI technology uses Laser or Radio Frequency (RF) method. Laser systems uses barcode attached to the vehicle & read by vehicle scanner as the vehicle passes through the toll lane. [7] According to reference, The German Federal

Ministry of Transport had conducted a fee collection experiment on German motorways called ChipTicket systems. The ChipTicket system assumes that vehicles are equipped with a chip card, and an in-vehicle unit. The in-vehicle unit contains transmitting and receiving facilities for communication with the toll stations - the so-called charge collection stations. The chip card is a plastic card contains an integrated microprocessor. When a driver passes a charge collection station, payment is made automatically by the vehicle the chip ticket is made and the fee is registered and stored in the station computer.[8]

A microcontroller has been programmed to operate a remotely operated positioning system of a satellite. Earlier, if we wanted to get an exact angle of the satellite, it needed manual adjustment. To overcome this drawback, this paper was aimed at developing a system to remotely operate the satellite. The IR signal from the remote (Transmitter) is sent and is received by the IR sensor (Receiver) which has been interfaced with the main microcontroller. The data from the transmitter is sent in an encoded format which is received by the receiver sensor and is the suitably sent to the main microcontroller. Corresponding signal is sent by the microcontroller to the motor driver which in turn rotates the motor and thereby the satellite accordingly. [9] According to reference, the ESP826 is a Wi-Fi module that has very efficient on chip storage and processing capabilities with lower cost and higher quality. ESP is based on IEEE 802.11 Wi-Fi protocol and it supports various network protocols. For connecting the machine to server a level shifter is used. [12]

This system on chip (SOC) design has been provided with a TCP/IP protocol stack which is used to provide Wi-Fi connection to any microcontroller that has been used by the system. There are three different methods to implement ESP8266. Using AT commands for communication is the simplest way that can be used. AT commands can be sent via the computer through a USB to serial adapter cable to the controller for setup and testing purposes. The second way is peripheral mode by interfacing compatible microcontroller with the module. The third method can be implemented by directly programming the GPIO pins of the module and interfacing them with external peripherals and sensors. An ESP8266 device can be used as an access point or as a station or both at the same time. Usually the access point also has a network data connection.

Therefore itcan act as a bridge between the wireless network and the TCP/IP network which is the internet. Once it has been finalized which mode the device will be used in, then we set a global mode which indicates the state in which the device will be used (As an access point or a station or both). Wi-Fi module is connected to the controller through this level shifter and have used SPI protocol.. Here considering the need for consistency of data we have utilized

TCP/IP network protocol for communicating with the server [11] The ESP8266 is designed to occupy minimal PCB area, without compromising on providing maximum inbuilt features. Cost Effective - Among the major advantages of the module include its price. You can get a module at lower than $5. Power - The power saving architecture operates in 3 modes: active mode, sleep mode and deep sleep mode, thus yielding very power efficient system. Ease of Interfacing - Module can be easily calibrated when required by reprogramming it for interfacing wide range of sensors.[13]

There are various of protocol is used in IoT (Internet of Things) devices. One of the most useful protocols is MQTT (Message Queuing Telemetry Transport). Second basic protocol is TCP. MQTT is broker based publishing/subscribing messaging protocol. It is open, lightweight, and very easy to implement. Moreover it has low bandwidth protocol.[14] The ATmega328 is the microcontroller used for this project. It shows important specifications like 14 digital input/output pins, 6 analog inputs, a USB connection, a 16 MHz ceramic resonator, an ICSP header, a power jack, and a reset button. The Arduino can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an ACto-DC adapter (wall-wart) or battery. The board can operate on an external supply of 6 to 20 volts. However it suffers form possibilities of overheating which may damage the board. The recommended range is 7 to 12 volts.[21] Controller –ATmega328 controller has been used which is an 8-bit microcontroller based on AVR. It has built-in communication protocol like I2C and UART and many other essential features. Real Time Clock-RTC module of DS1307 is connected to ATmega328 through two wire interface for reading the on and off time of the machine. Level shifter: MOSFET based level shifter is needed to drop down voltage in between, as ATmega328 operate on 5V and ESP-12 on 3.3V.[12]

2.3 CONCLUSION ON REVIEWS:

The ATC system in the paper “A review on automated toll collection system” the two types of study were compared namely the RFID based study and Leach-C based study showed promising and efficient way to develop the system but it also showed us lack of a point where the Leach-C system worked on digital image processing where the number plate of subject’s vehicle was scanned using high-tech camera. Using the RFID based implementation and eliminating the DIP based implementation to make system better time efficient. The survey done in the paper named “A 5.8 GHz ISM band Microstrip antenna for

RFID applications” showed us the path using and developing RFID antenna for scanning the details of subject’s vehicles from a handy distance thus enhancing the distance of vehicle and RFID reader would cover a large area.

The review was done of the paper “Automated toll collection system using RFID” Enlighted us the various components that can used while developing our own system in time saving and in efficient way so as to make a hybrid system that is mentioned in the paper “A review on automated toll collection system”. “A Survey in Image Mining” showed a path towards machine learning and data mining concepts and helped us in mining databases in the form of people’s feedback. The survey done in the paper named ATC turned out to be good step for developing ATC as we tried the best option in choosing the components in terms of reducing operation costs, time saving, flexible and also added the theft detection module in our system enabling security on high alert and detecting of stolen vehicles. During the period of learning and reviewing we also tried in improving and eliminating the costs of GSM kits embedded on the micro-controller board by replacing by use of private SMS gateway or govt. running GSM Company, which has reduced the operation cost much and has made our system more independent of using GSM kit.

CHAPTER-3
PROJECT DESIGN

3.1 OVERVIEW OF THE DESIGN:

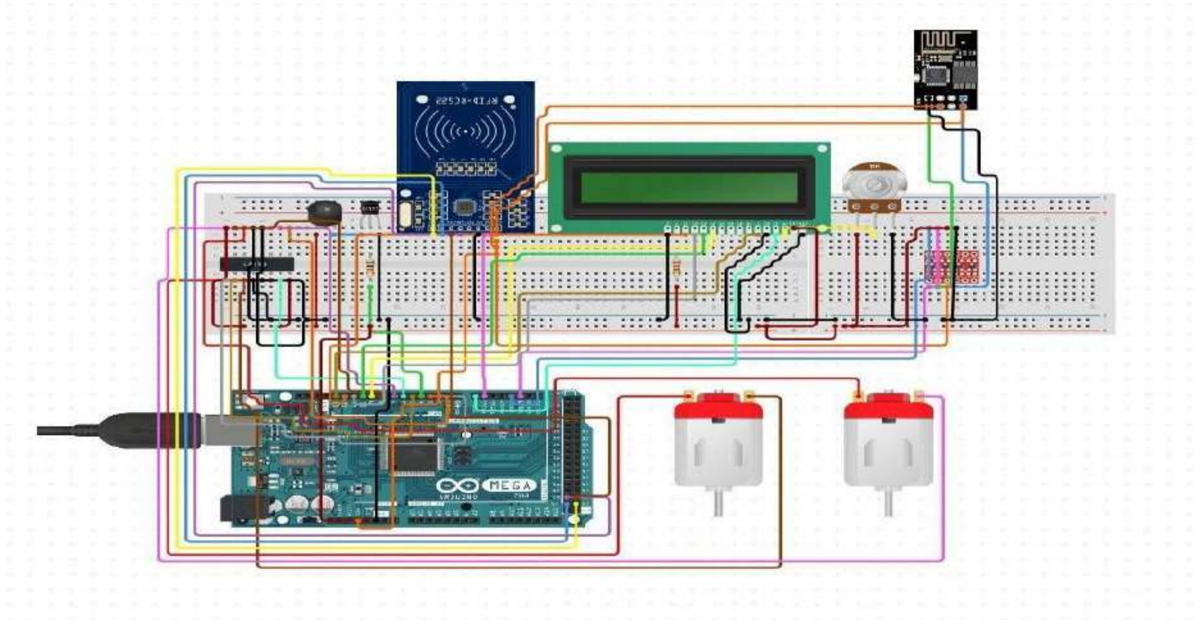


Fig.3.1 Overview of the design

3.2 EQUIPMENT ANALYSIS:

3.2.1 ARDUINO:

3.2.1.1 INTRODUCTION:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.

- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

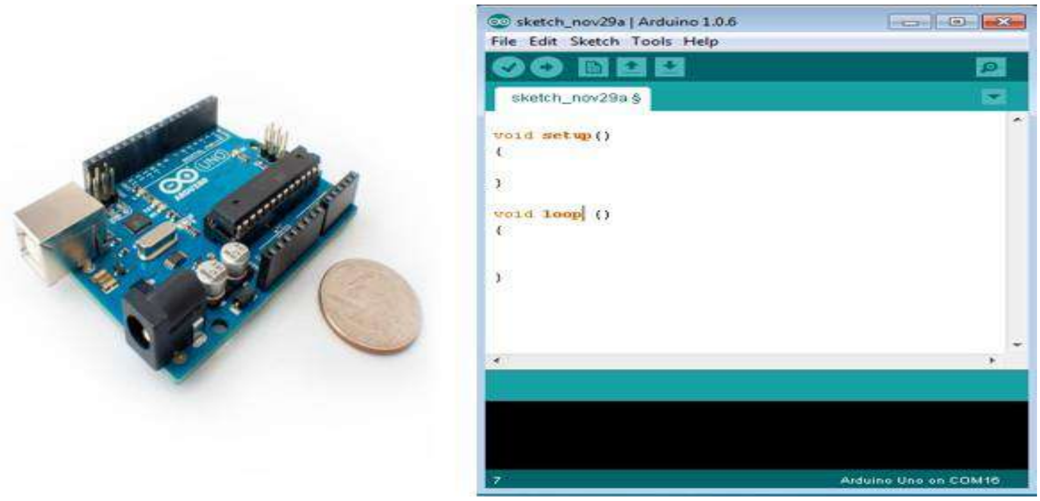


Fig 3.2.1.1 Arduino Uno

3.2.1.2 BOARD TYPES

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega 16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega 16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header

Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.2.1.2.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	**Operating Volt**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 3.2.1.2.2 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 3.2.1.2.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

Table 3.2.1.2.4 Arduino boards based on AT91SAM3X8E microcontroller

3.2.1.3 BOARD DESCRIPTION:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

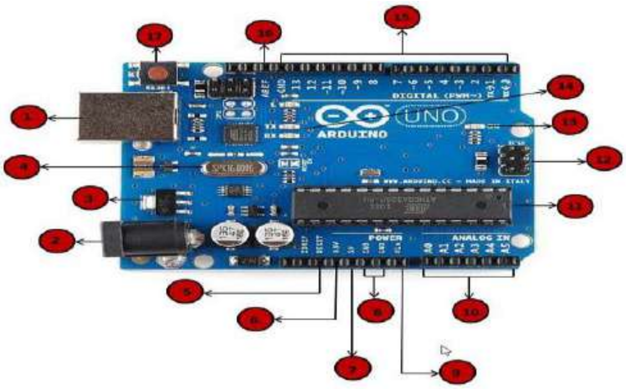


Fig.3.2.1.2 Board Description

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt

	• 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in

	two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

3.2.1.4 ARDUINO FAMILY

Arduino makes several different boards, each with different capabilities. In addition, part of being open-source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

Fig.3.2.1.3: Arduino Family

3.2.1.5 SHIELDS:

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

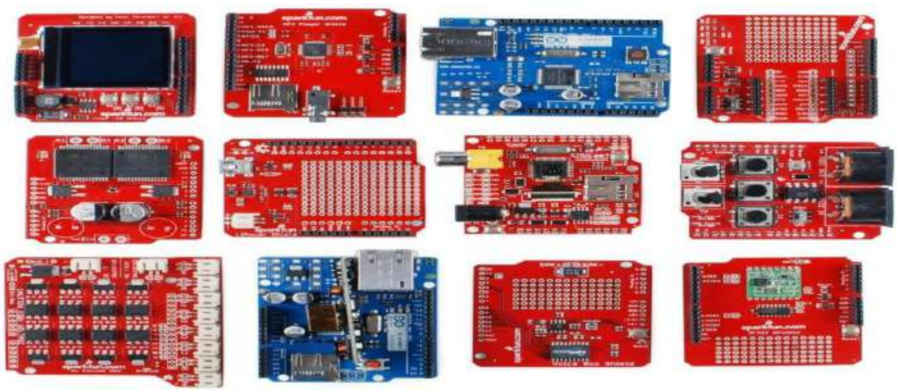


Fig.3.2.1.4: Arduino Shields

3.2.1.6 PIN DESCRIPTION OF ATMEGA328:

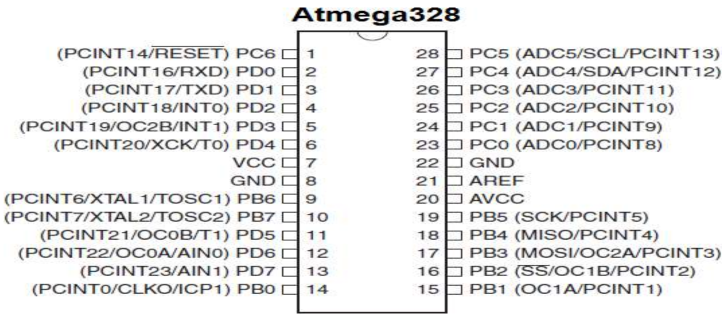


Fig.3.2.1.5 Pin description of ATMEGA328

3.2.1.7 APPLICATIONS:

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

3.2.2 16*2 LCD SCREEN DISPLAY

3.2.2.1 INTRODUCTION

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

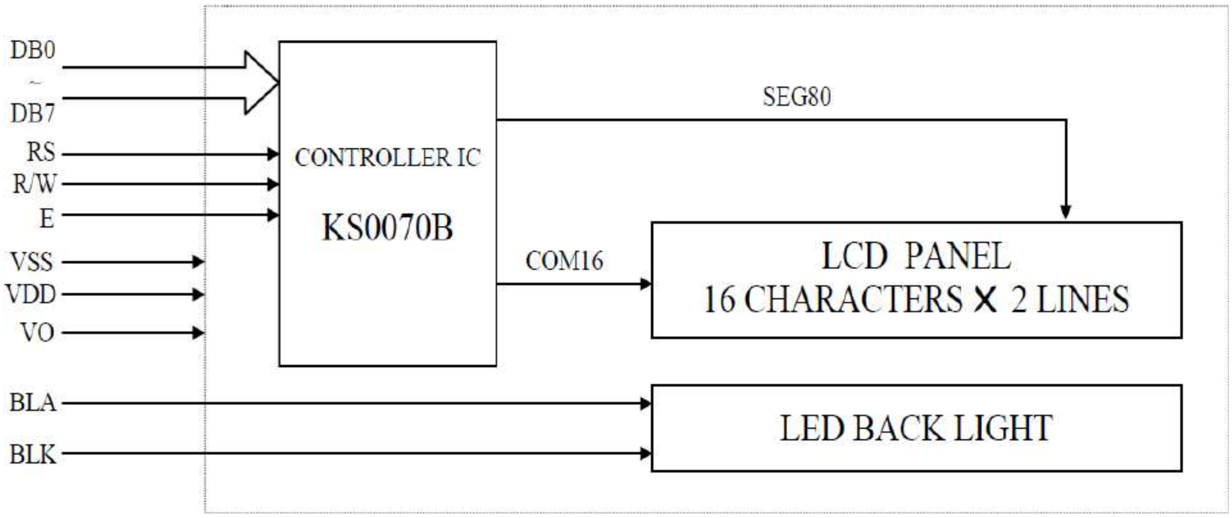


fig.3.2.2.1 Lcd block diagram

3.2.2.2 DISPLAY DATA RAM:

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.

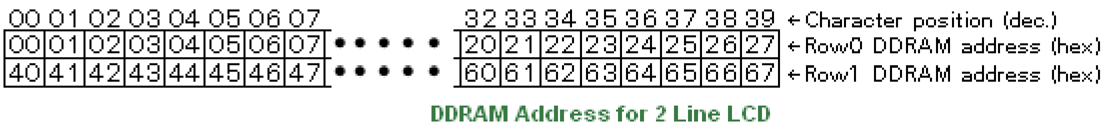


Fig.3.2.2.2 DDRAM addresses of 2 Line LCD

3.2.2.3 CHARACTER GENERATOR ROM:

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM.

3.2.2.4 BUS FLAG:

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e. BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

3.2.2.5 INSTRUCTION REGISTER (IR) AND DATA REGISTER (DR):

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g. LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD.
Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

3.2.2.6 16 X 2 ALPHANUMERIC LCD MODULE FEATURES:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

3.2.2.7 CIRCUIT DESCRIPTION:

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

3.2.3 BUZZER

3.2.3.1 MAGNETIC TRANSDUCER

Magnetic transducers contain a magnetic circuit consisting of a iron core with a wound coil and a yoke plate, a permanent magnet and a vibrating diaphragm with a movable iron piece. The diaphragm is slightly pulled towards the top of the core by the magnet's magnetic field. When a positive AC signal is applied, the current flowing through the excitation coil produces a fluctuating magnetic field, which causes the diaphragm to vibrate up and down, thus vibrating air. Resonance amplifies vibration through resonator consisting of sound hole(s) and cavity and produces a loud sound.

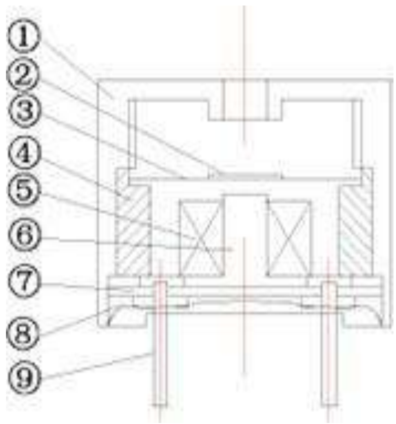
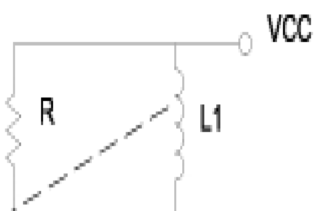


Fig.3.2.3.1 Magnetic Transducer

3.2.3.2 MAGNETIC BUZZER(SOUNDER)

Buzzers like the TMB-series are magnetic audible signal devices with built-in oscillating circuits. The construction combines an



oscillation circuit unit with a detection coil, a drive coil and a magnetic transducer. Transistors, resistors, diodes and other small devices act as circuit devices for driving sound generators. With the application of voltage, current flows to the drive coil on primary side and to the detection coil on the secondary side. The amplification circuit, including the transistor and the feedback circuit, causes vibration. The oscillation current excites the coil and the unit generates an AC magnetic field corresponding to an oscillation frequency. This AC magnetic field magnetizes the yoke comprising the magnetic circuit. The oscillation from the intermittent magnetization prompts the vibration diaphragm to vibrate up and down, generating buzzer sounds through the resonator.

3.2.3.3 SPECIFICATIONS:

Fig.3.2.3.2 Magnetic Buzzer

Rated Voltage: A magnetic buzzer is driven by 1/2 square waves (V o-p).

Operating Voltage: For normal operating. But it is not guaranteed to make the minimum Sound Pressure Level (SPL) under the rated voltage.

Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.

Direct Current Resistance: The direct current resistance is measured by ammeter directly.

Sound Output: The sound output is measured by decibel meter. Applying rated voltage and 1/2 square waves, and the distance of 10 cm.

Rated Frequency: A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.

Operating Temp. : Keep working well between -30℃ and +70℃.

How to choose:

Driving methods: AX series with built drive circuit will be the best choice when we cannot provide frequency signal to a buzzer, it only needs direct current.

Dimension: Dimension affects frequency, small size result in high frequency.

Voltage: Depend on V o-p (1/2 square waves)

Fixed methods: From the highest cost to the lowest- DIP, wires/ connector, SMD.

Soldering methods: AS series is soldered by hand, the frequency is lower because of the holes on the bottom. On the other hand, we suggest AC series for the reflow soldering, the reliability is better.

3.2.3.4 HOW TO CHOOSE A BUZZER:

There are many different kinds of buzzer to choose, first we need to know a few parameters, such as voltage, current, drive method, dimension, mounting type, and the most important thing is how much SPL and frequency we want.

Operating voltage: Normally, the operating voltage for a magnetic buzzer is from 1.5V to 24V, for a piezo buzzer is from 3V to 220V. However, in order to get enough SPL, we suggest giving at least 9V to drive a piezo buzzer.

Consumption current: According to the different voltage, the consumption current of a magnetic buzzer is from dozens to hundreds of mill amperes; oppositely, the piezo type saves much more electricity, only needs a few mill amperes, and consumes three times current when the buzzer start to work.

Driving method: Both magnetic and piezo buzzer have self drive type to choose. Because of the internal set drive circuit, the self drive buzzer can emit sound as long as connecting with the direct current. Due to the different work principle, the magnetic buzzer need to be driven by 1/2 square waves, and the piezo buzzer need square waves to get better sound output.

Dimension: The dimension of the buzzer affects its SPL and the frequency, the dimension of the magnetic buzzer is from 7 mm to 25 mm; the piezo buzzer is from 12 mm to 50 mm, or even bigger.

Connecting way: Dip type, Wire type, SMD type, and screwed type for big piezo buzzer are usually seen.

Sound Pressure Level (SPL): Buzzer is usually tested the SPL at the distance of 10 cm, if distance double, the SPL will decay about 6 dB; oppositely, the SPL will increase 6 dB when the distance is shortened by one time. The SPL of the magnetic buzzer can reach to around 85 dB/ 10 cm; the piezo buzzer can be designed to emit very loud sound, for example, the common siren, are mostly made of piezo buzzer.

3.2.3.5 INTRODUCTION OF PIEZO BUZZER:

Piezo buzzers are simple devices that can generate basic beeps and tones. They work by using a piezo crystal, a special material that changes shape when voltage is applied to it. If the crystal pushes against a diaphragm, like a tiny speaker cone, it can generate a pressure wave which the human ear picks up as sound. Simple change the frequency of the voltage sent to the piezo and it will start generating sounds by changing shape very quickly!

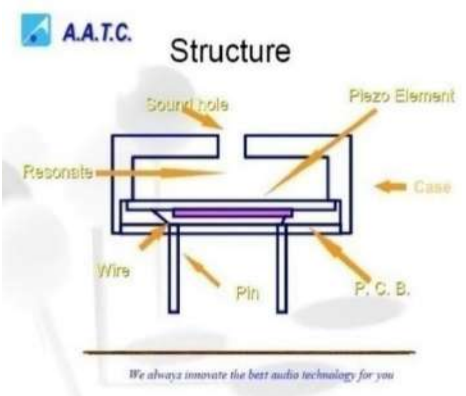


Fig.3.2.3.5 piezo buzzer

3.2.3.6 SPECIFICATIONS:

Rated Voltage: A piezo buzzer is driven by square waves(Vp-p).

Operating Voltage: For normal operating. But it is not guaranteed to make the minimum SPL under the rated voltage.

Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.

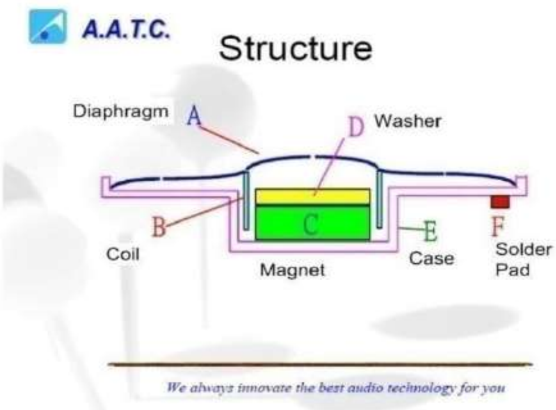


Fig. 3.2.3.6 structure of piezo buzzer

Capacitance: A piezo buzzer can make higher SPL with higher capacitance, but it consumes more electricity.

Sound Output: The sound output is measured by decibel meter.

Applying rated voltage and square waves, and the distance of 10 cm.

Rated Frequency: A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.

Operating Temp.: Keep working well between -30℃ and +70℃.

3.2.4 L293D

3.2.4.1 INTRODUCTION

L293D is basically a high current dual motor driver/controller Integrated Circuit (IC). It is able to drive load having current up to 1A at the voltage ranging from 4.5V to 36V. Motor driver usually act as current amplifier because they receive a low current signal as an input and provides high current signal at the output.

L293D is basically a motor driver or controller. It has two built-in H-bridge circuits which are able to control two DC motors simultaneously in both clockwise and counter clockwise direction. It acts as an current high amplifier because it take low current signal at its input and provides higher current signal at the output in order to drives different load e.g. stepper motor & DC motors. Its features include large input voltage supply range, large output current high noise immunity input signals etc. Its common real-life applications include stepping motor drivers, relay drivers, DC motor drivers etc. L-293D motor driver/controller is shown in the figure given below.

Fig. 3.2.4.1 L293D Motor Driver

3.2.4.2 L293D PINS:

- L-293D has sixteen (16) pins, having different individual functions.
- All of these sixteen pins along with their serial no, are given in the table shown below.

3.2.4.3 L293D PIN FUNCTIONS:

- L 293D each pin has different tasks to perform when they are in working condition.
- The tasks assigned to each pin are given in the table shown below.

L293D Pins	
Pin. No	Pin Name
1	Enable 1,2
2	Input 1
3	Output 1
4	GND
5	GND
6	Output 2
7	Input 2
8	Vcc2
9	Enable 3,4
10	Input 3
11	Output 3
12	GND
13	GND
14	Output put 4
15	Input 4
16	Vcc1

Table 3.2.4.3 L293D Pin Functions

3.2.4.4 L293D PINOUT:

- Pinout diagram of any device presents the pins configuration through a completely labelled diagram.

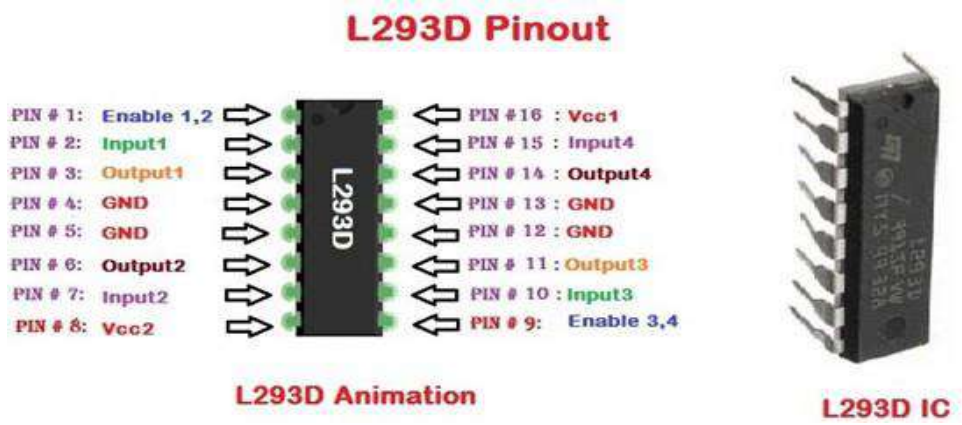


Fig. 3.2.4.4 L293D Pinout

3.2.5 MOTOR

Motors usually operates on this higher current. L-293D has to built-in H-Bridge driver circuits and is able to control two DC motors at a time in both clockwise and counter clockwise direction. It has two enable pins and they should be kept high in order to control the motor. By changing the polarity of applied signal motor can be rotated in either clockwise or counter clockwise direction. If L 293D enable pin is high, its corresponding driver will provide the desired out. If the enable pin is low, there will be no output. L-293D has different features including internal ESD protection, large voltage supply range, large output current per channel, high noise immunity input etc. L 293D plays a vital role in electronics era and has several different applications e.g. relay drivers, DC motor drivers, stepping motor drivers etc. The further detail about L 293D motor driver/controller will be given later in this tutorial.

3.2.6 SOFTWARE EXPLANATION:

- Arduino software
- Proteus simulation

3.2.6.1 ARDUINO SOFTWARE:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around thisopen-source platform, their contributions have added up to an incredible amount ofaccessible knowledge that can be of great help to novices and experts alike. Arduino wasborn at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed atstudents without a background in electronics and programming. As soon as it reached awider community, the Arduino board started changing to adapt to new needs andchallenges, differentiating its offer from simple 8-bit boards to products for IoTapplications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventuallyadapt them to their particular needs. The software, too, is open-source, and it is growingthrough the contributions of users worldwide.

3.2.6.2 WHY ARDUINO?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and
wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.Wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50.

- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as

well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under

a Creative Commons license, so experienced circuit designers c an make their own version Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money

- Getting Started with Arduino and Genuino products: -

Install the Arduino Software (IDE) on Windows PCs-

This document explains how to install the Arduino Software (IDE) on Windows machines.

✔ Download the Arduino Software (IDE)

✔ Proceed with board specific instructions.

3.2.6.3 HOW TO DOWNLOAD THE ARDUINO SOFTWARE (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

3.2.6.4 INSTALLATION:

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Fig.3.2.6.4.1 Standard USB Cable (A plug to B plug)

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Fig. 3.2.6.4.2 Standard USB Cable (an A to Mini-B plug)

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

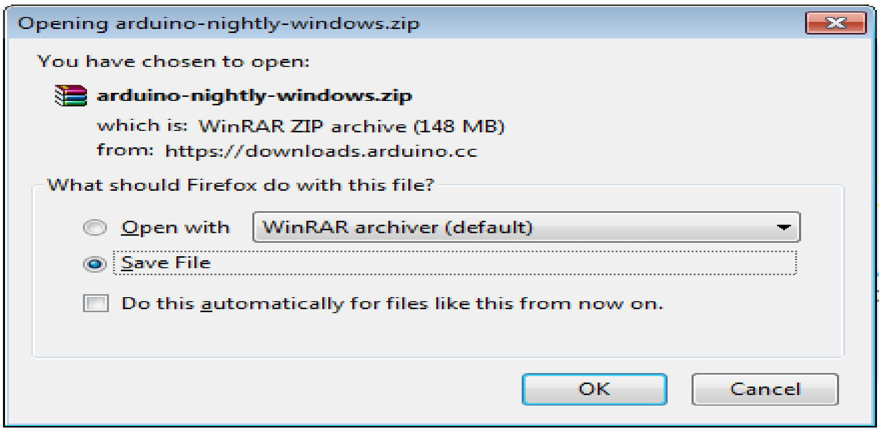


Fig. 3.2.6.4.3 Download Arduino IDE Software

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

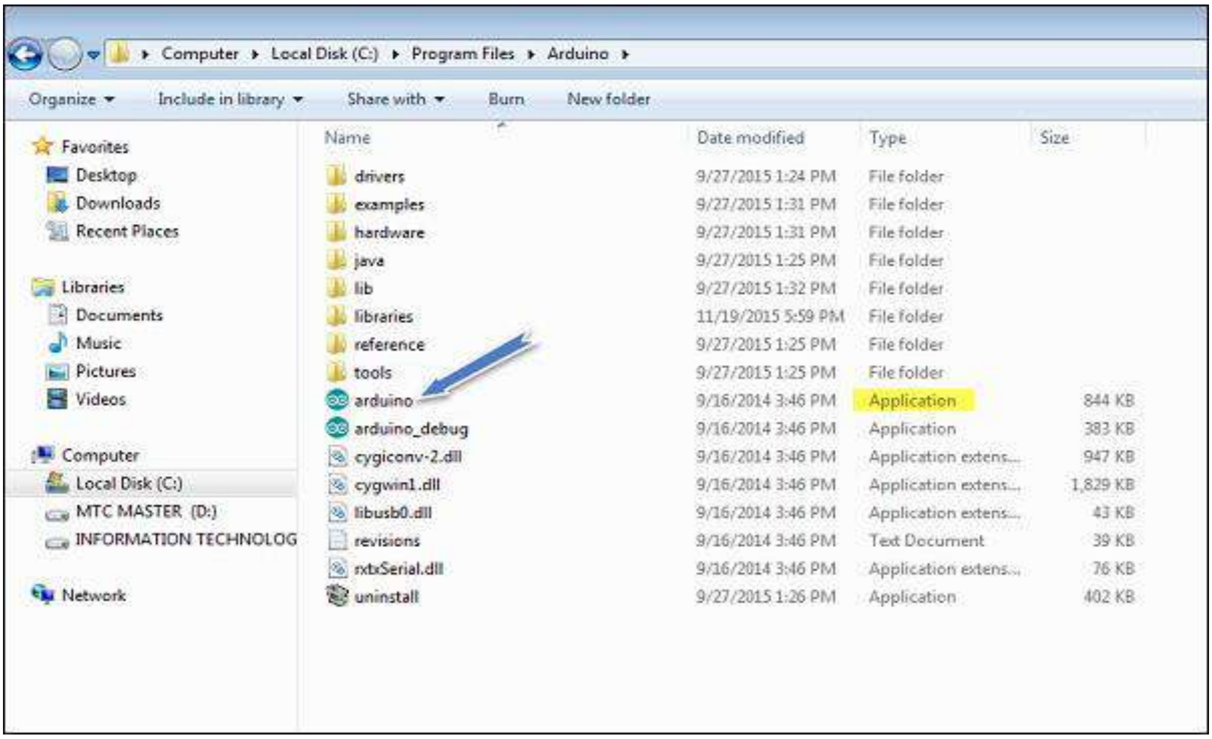


Fig. 3.2.6.4.4 Launch Arduino IDE

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**

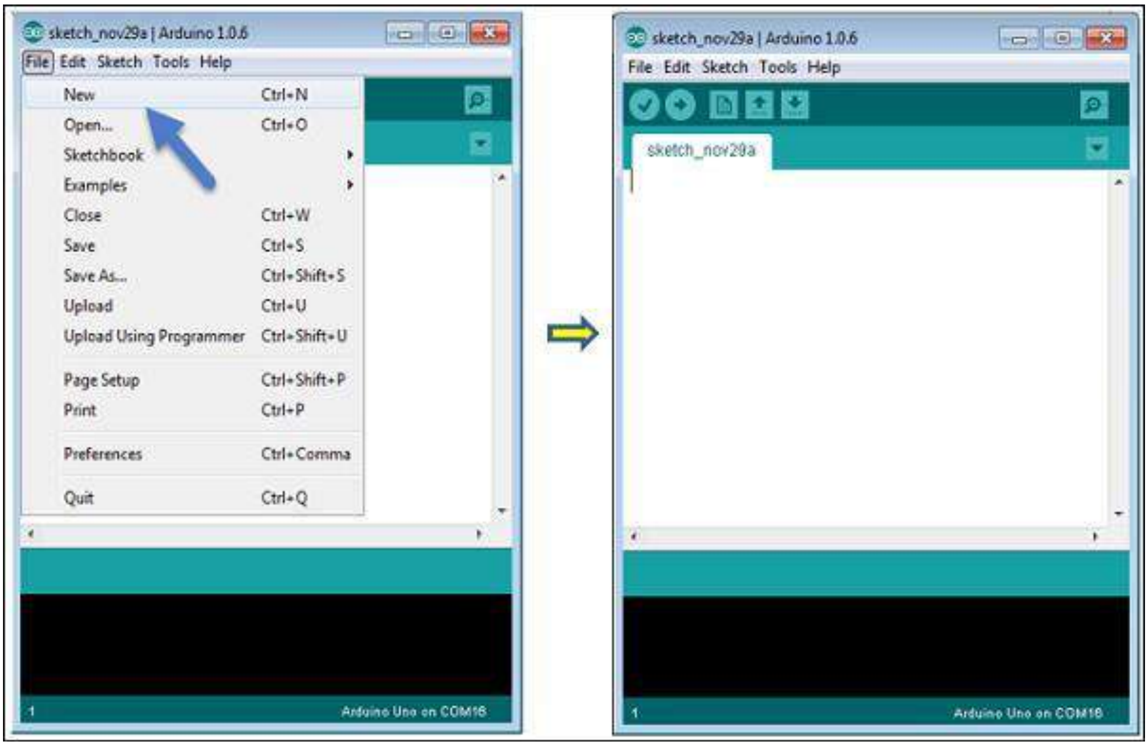


Fig. 3.2.6.4.5 Open your first project.

To open an existing project example, select File → Example → Basics → Blink.

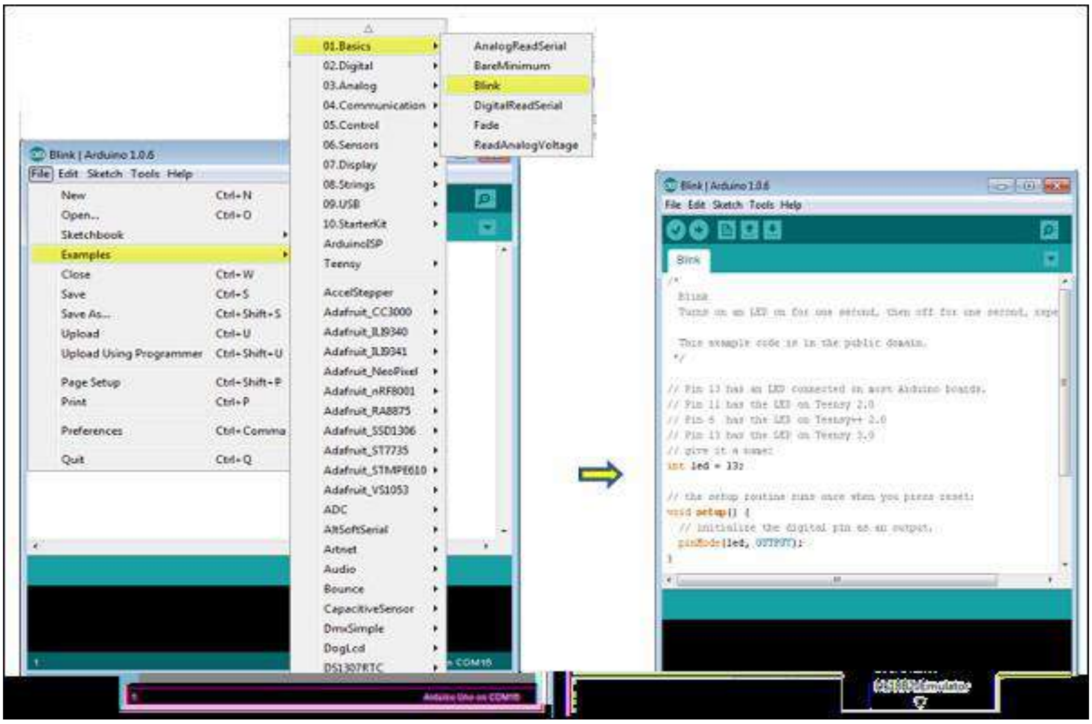


Fig. 3.2.6.4.6 Open your first program.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

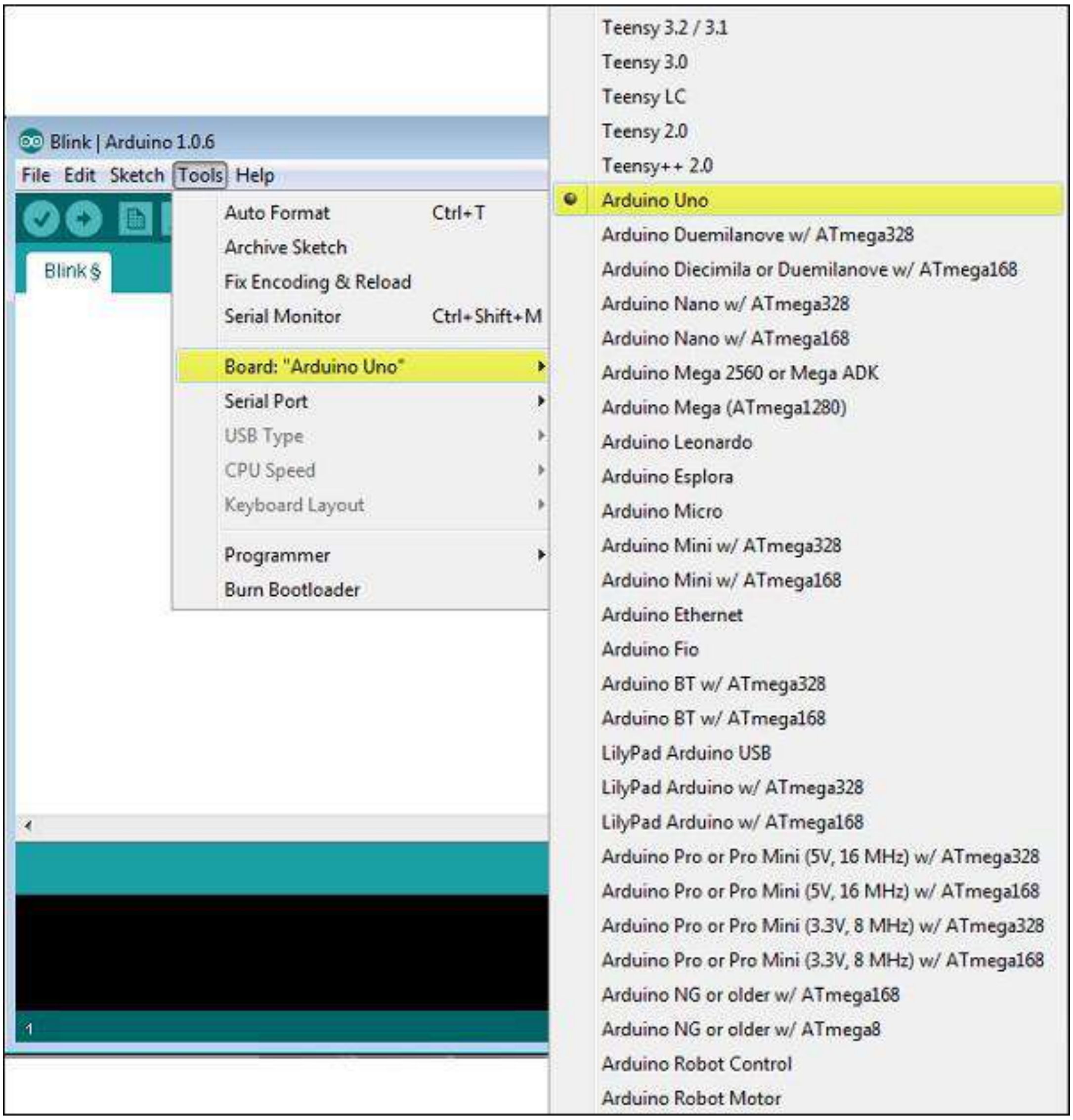


Fig. 3.2.6.4.7 Select your Arduino board.

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

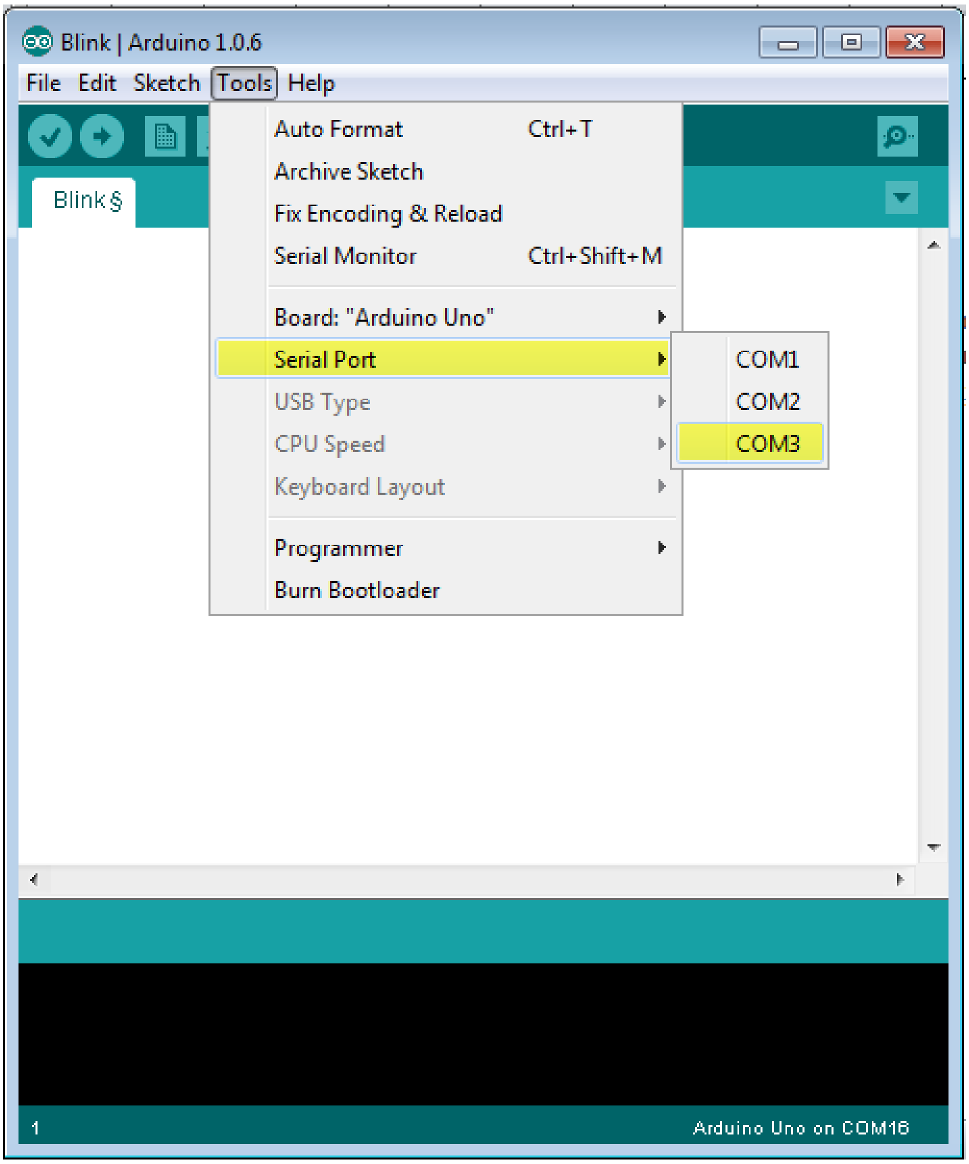


Fig. 3.2.6.4.8 Select your serial port.

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

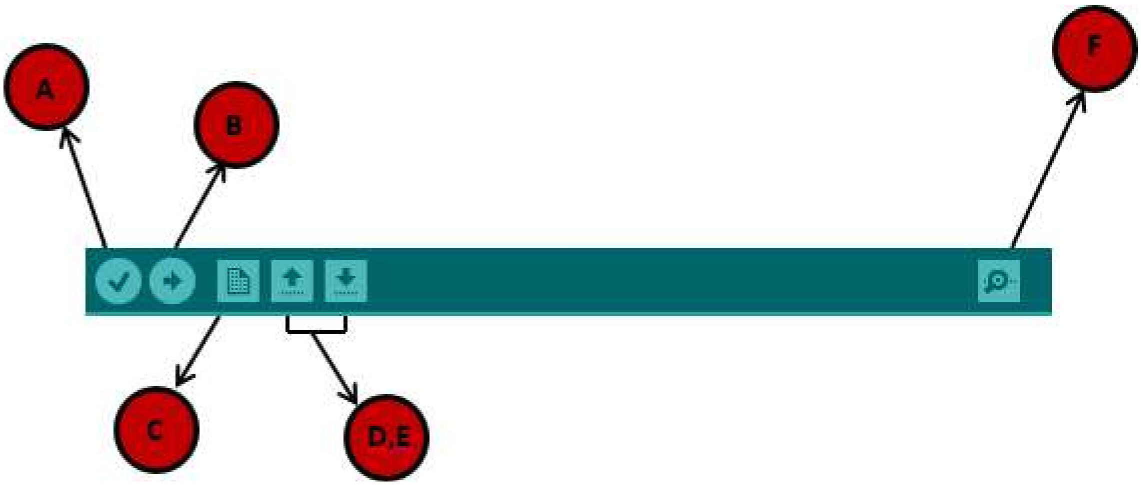


Fig. 3.2.6.4.9 Upload the program to your board

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from board & send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

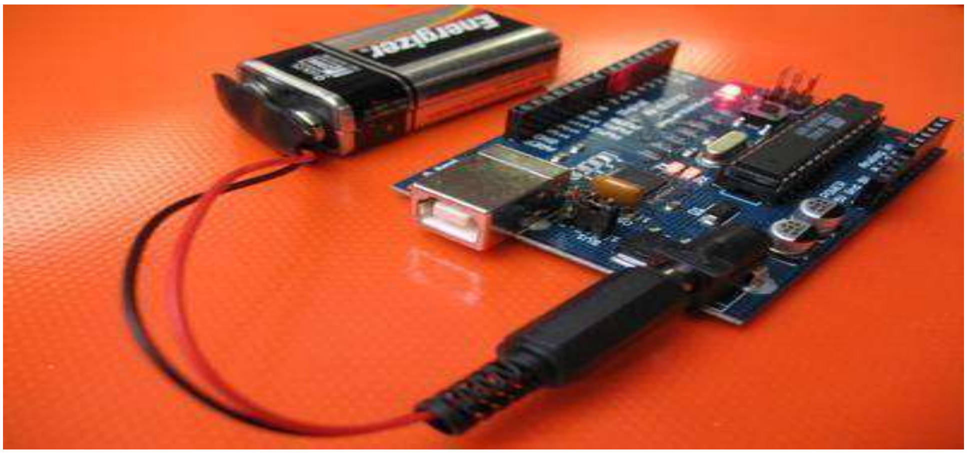


Fig. 3.2.6.4.10 Connecting a Battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called "sketches", but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
 {
   Serial.begin(9600);
   Serial.println("Hello World");
 }
void loop() {}
```

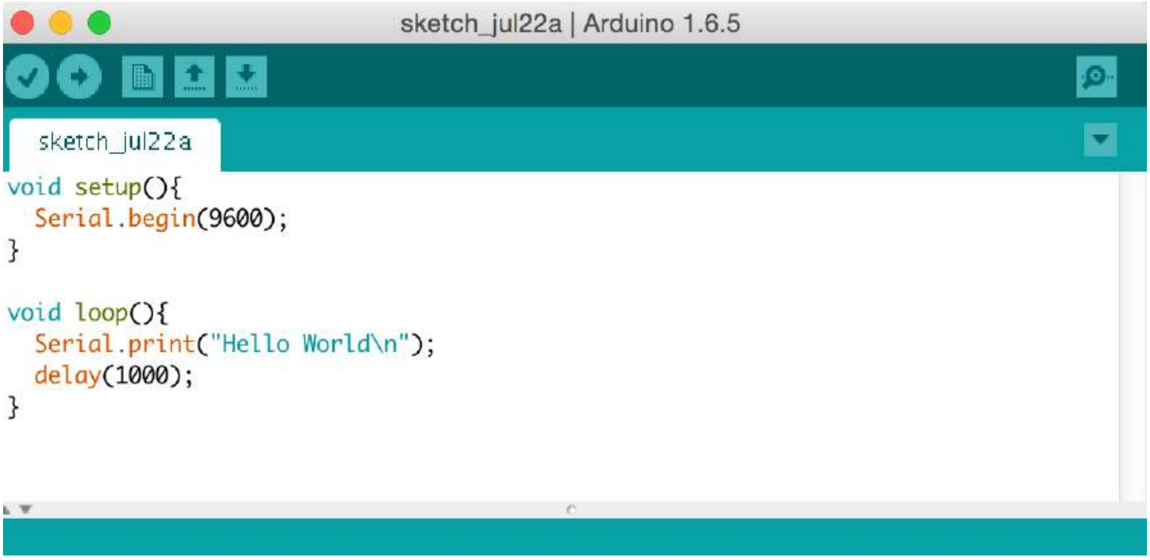


Fig. 3.2.6.4.11 Example Program

3.2.7 PROTEUS:

PROTEUS:

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline "From concept to completion".

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation. ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components. The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole

packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design:

Step 1: Open ISIS software and select New design in File menu

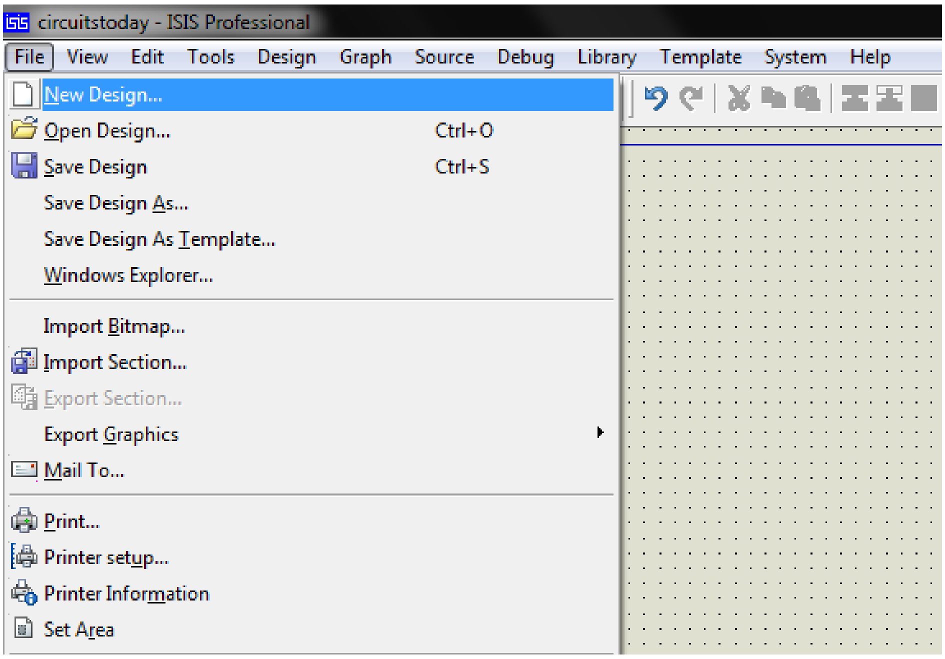


Fig. 3.2.7.1 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

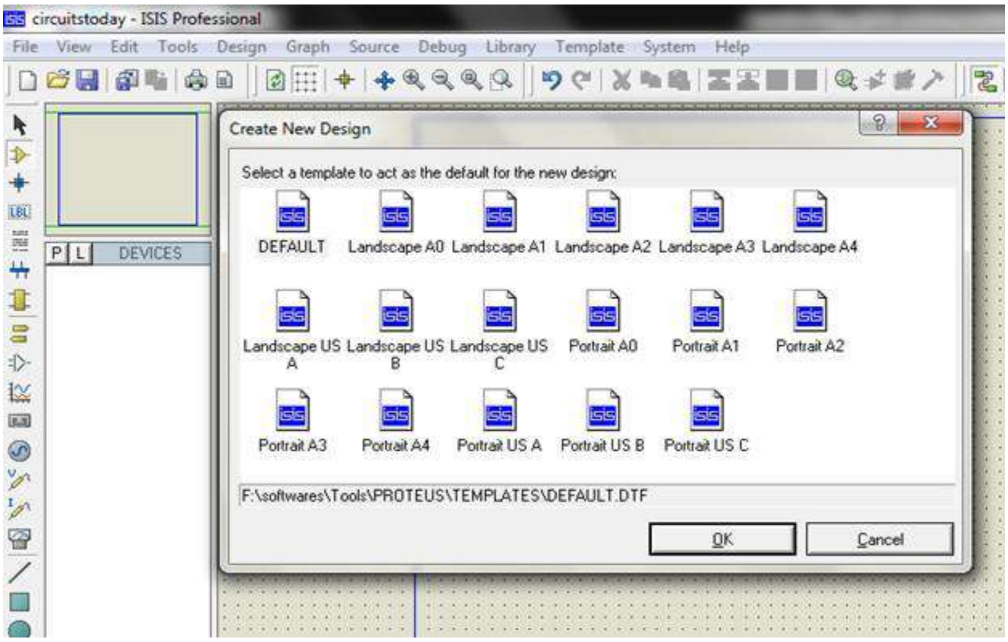


Fig. 3.2.7.2 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

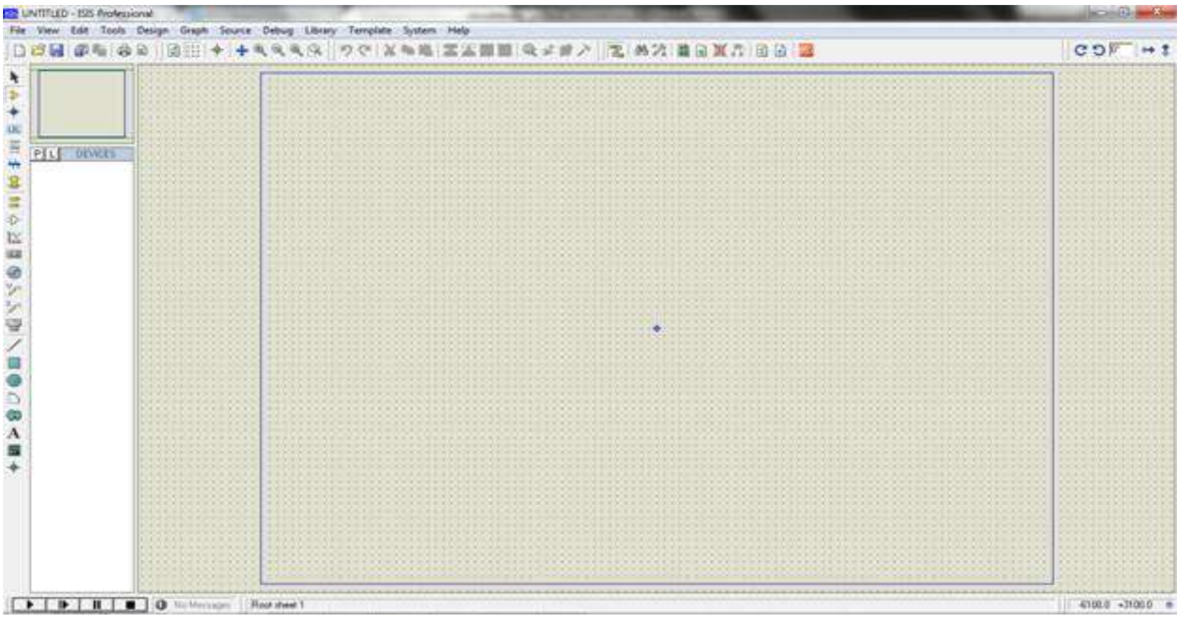


Fig. 3.2.7.3 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

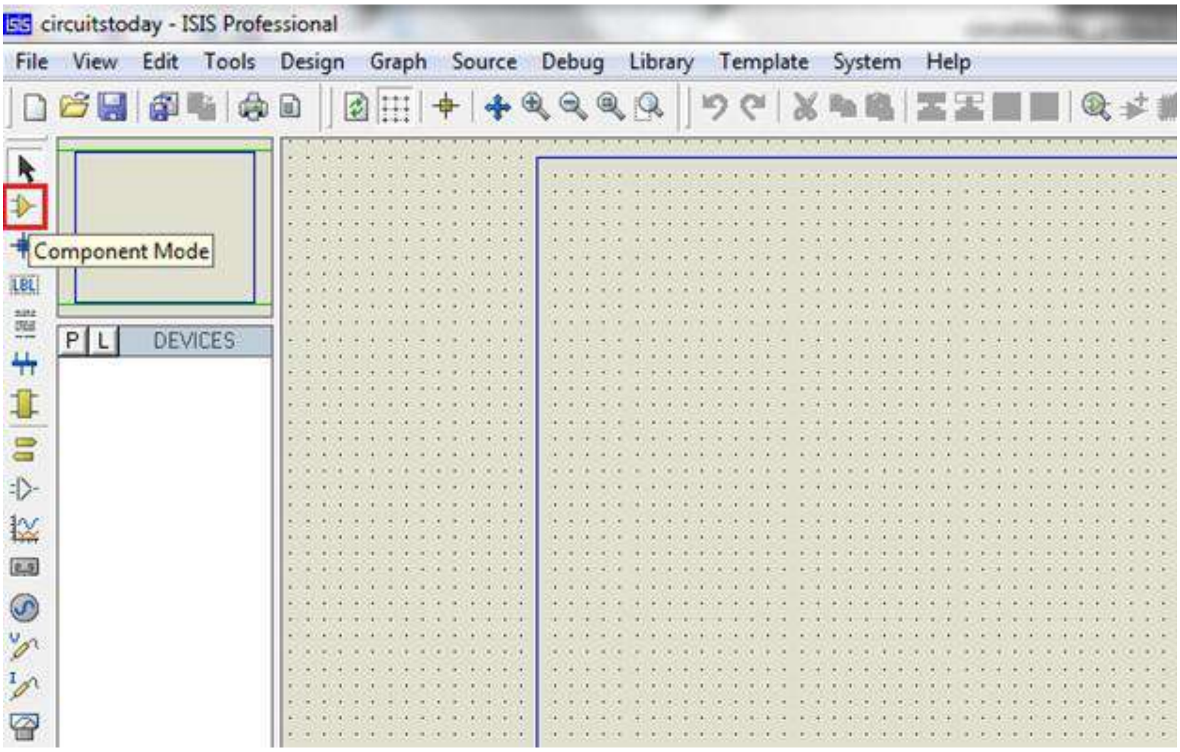


Fig. 3.2.7.4 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

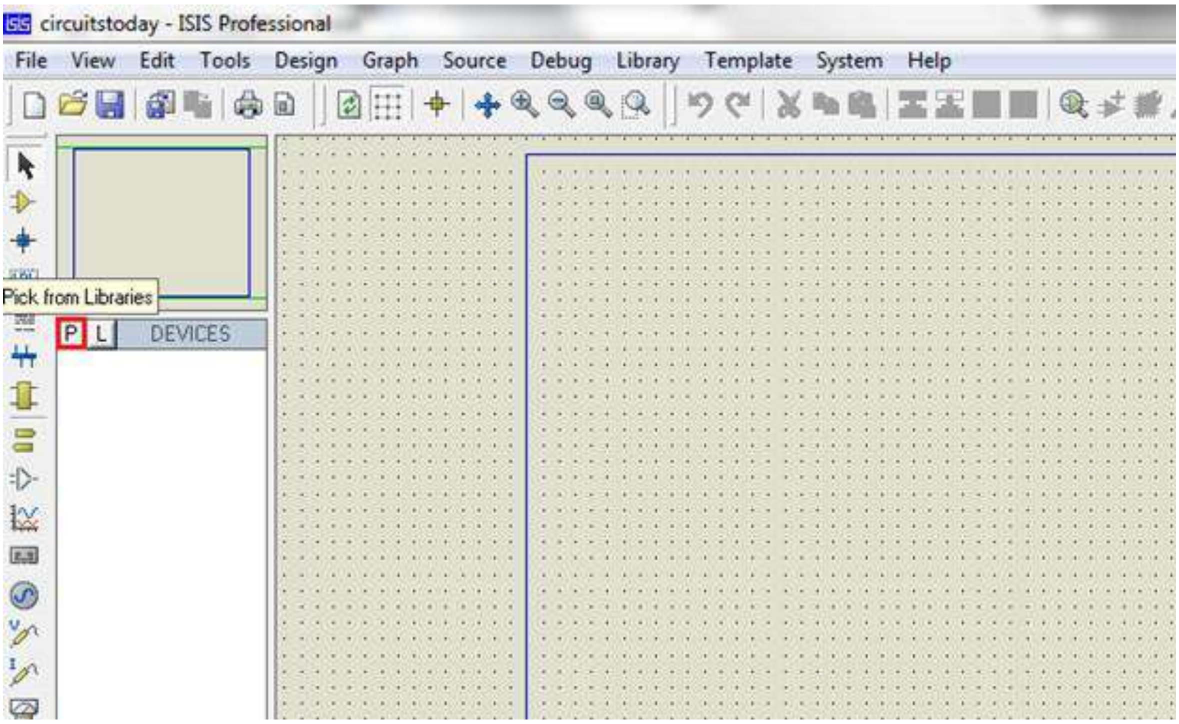


Fig. 3.2.7.5 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

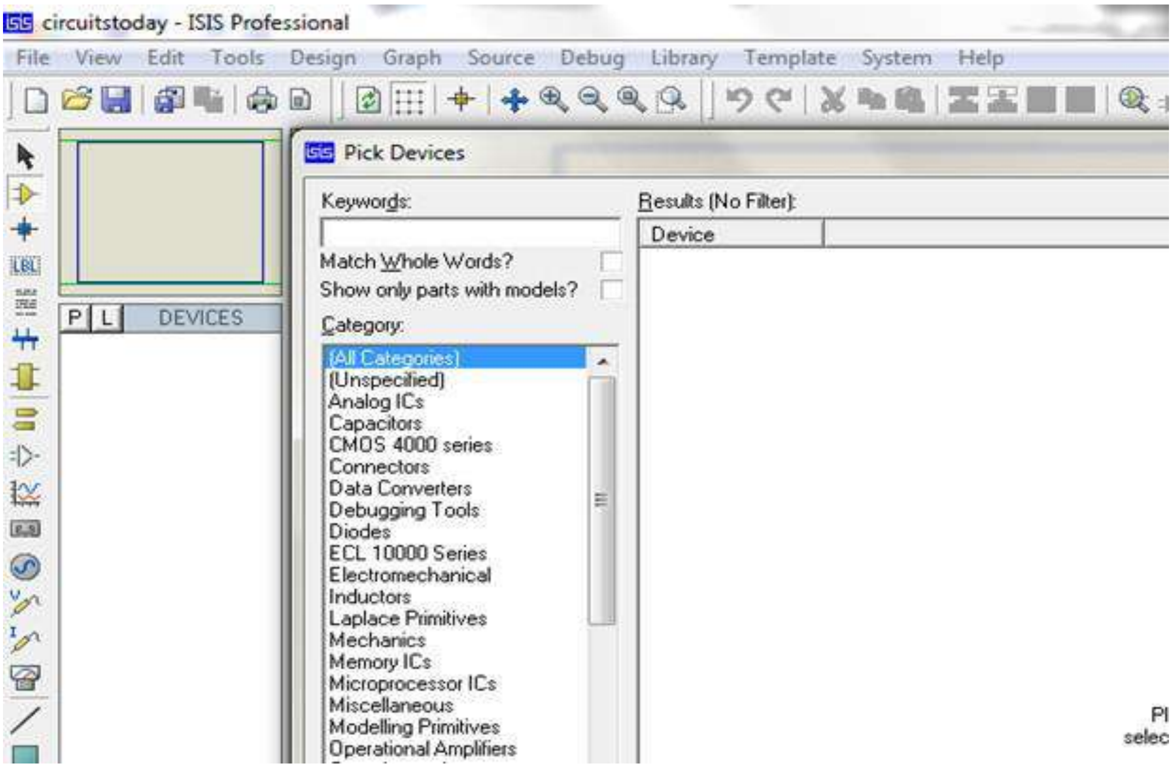


Fig. 3.2.7.6 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

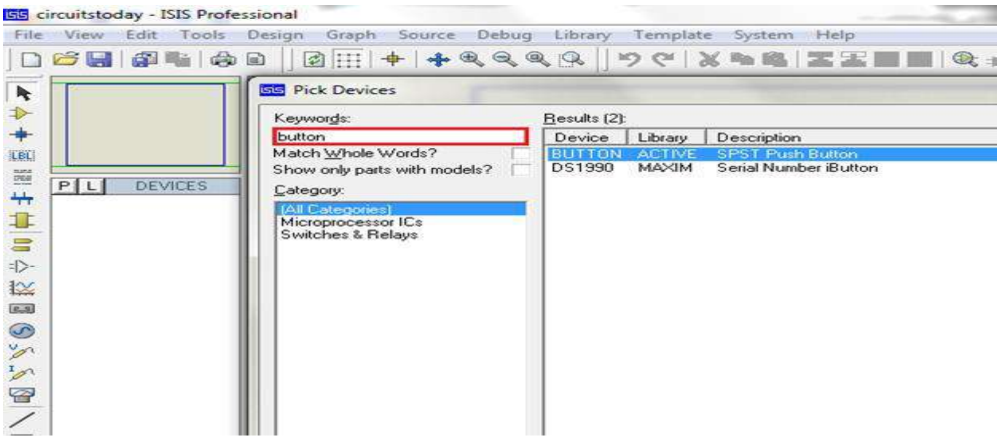


Fig. 3.2.7.7 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

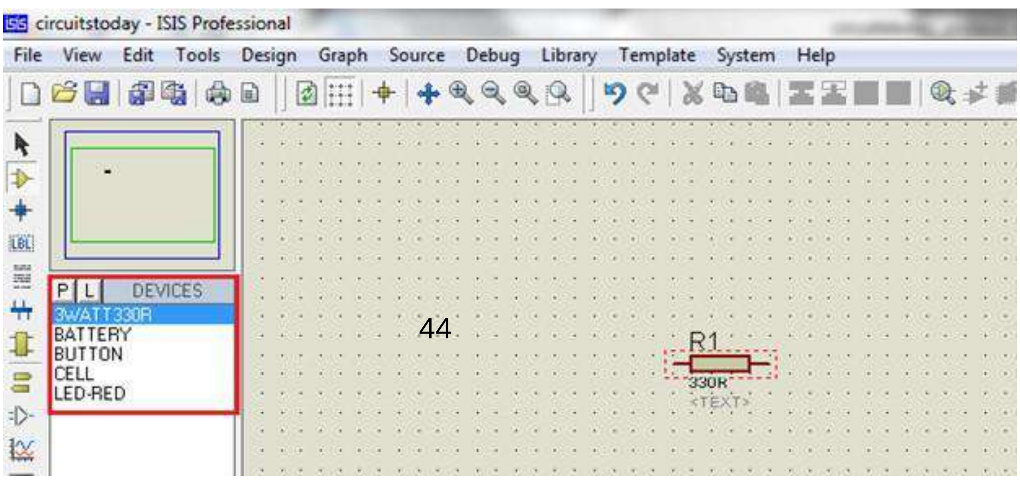


Fig. 3.2.7.8 Component Selection

Place all the required components and route the wires i.e., make connections. Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

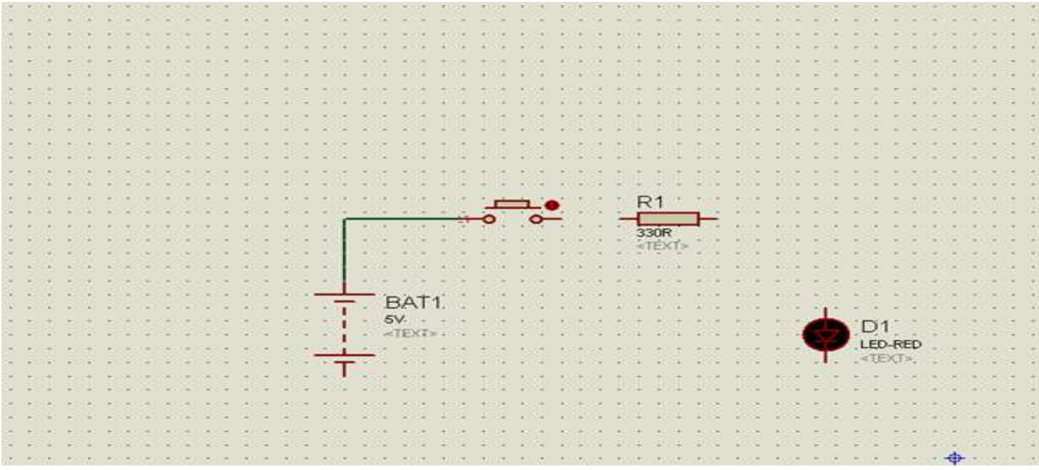


Fig. 3.2.7.9 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

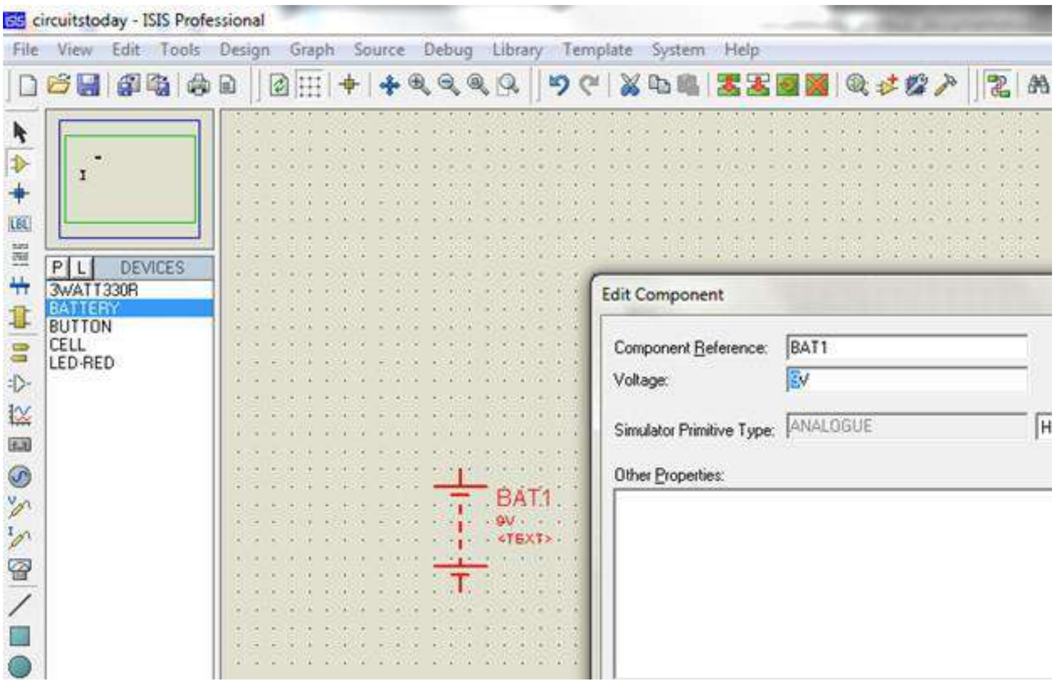


Fig. 3.2.7.10 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

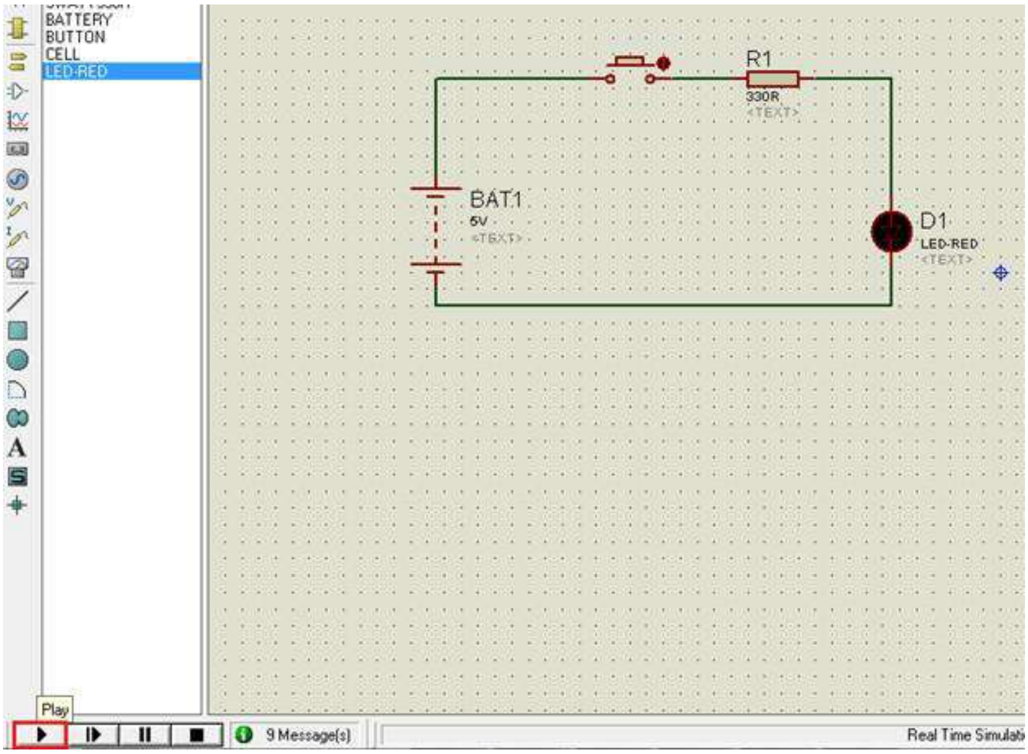


Fig. 3.2.7.11 Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

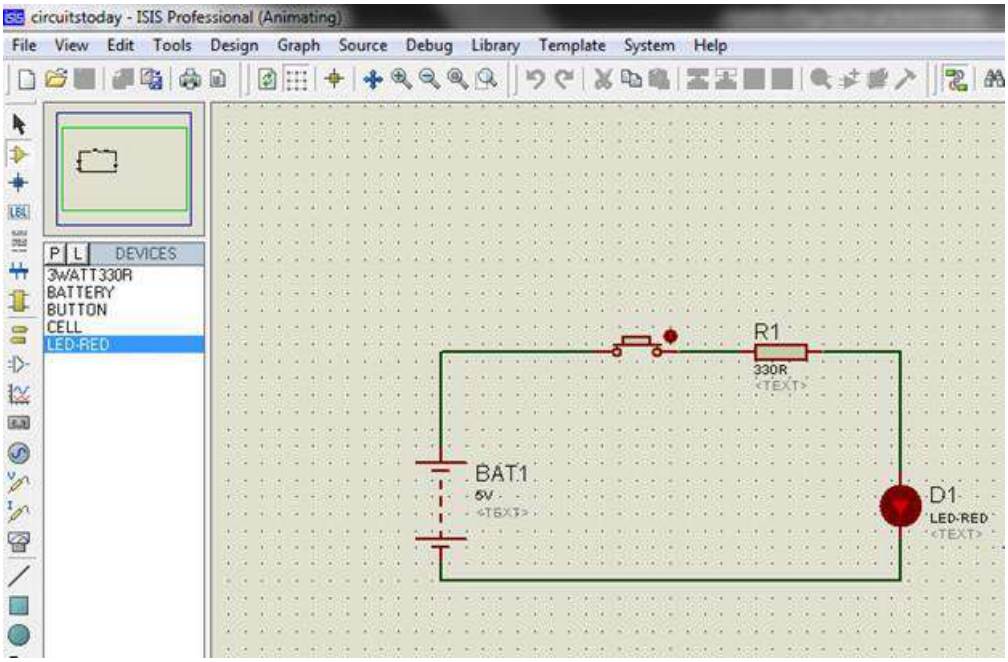


Fig . 3.2.7.12 Simulation Animating

Simulation can be stepped, paused or stopped at any time.

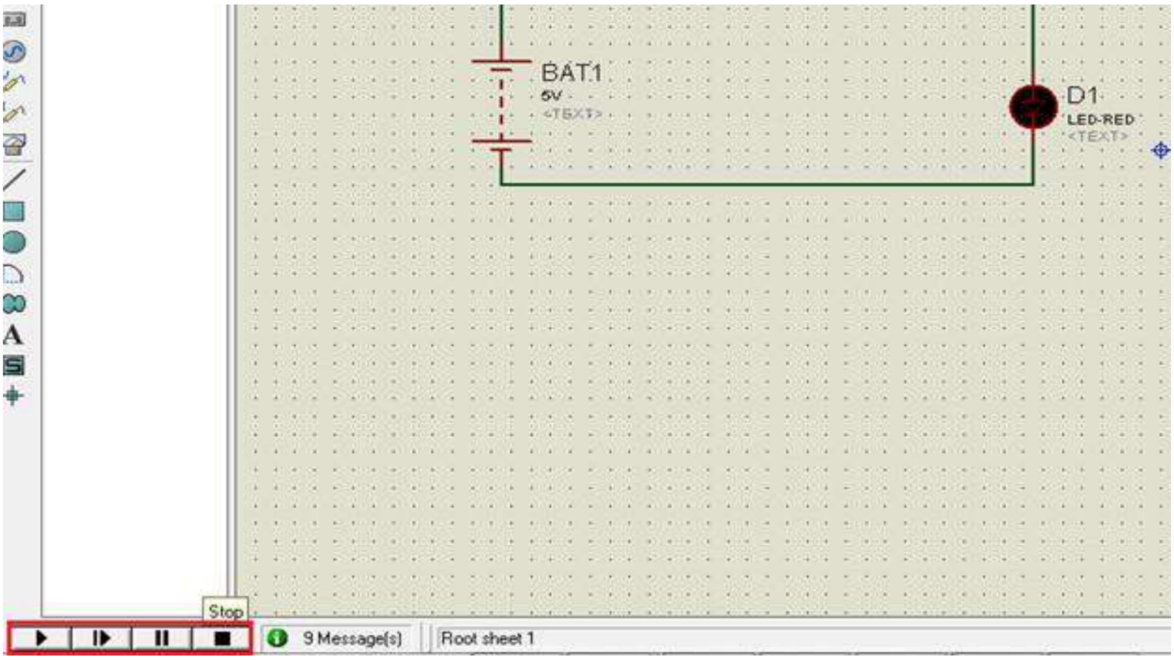


Fig. 3.2.7.13 Simulation Step-Pause-Stop Buttons

According to its developers, Things Speak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. Things Speak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates.

Things Speak was originally launched by iot Bridge in 2010 as a service in support of IoT applications.

Things Speak has integrated support from the numerical computing software MATLAB from MathWorks, allowing Things Speak users to analyse and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license from Math works.[citation needed].

Things Speak has a close relationship with Math works, Inc. In fact, all of the Things Speak documentation is incorporated into the Math works' Matlab documentation site and even enabling registered Math works user accounts as valid login credentials on the Things Speak website. The terms of service and privacy policy of ThingSpeak.com are between the agreeing user and Math works, Inc.

Setup Things Speak

Create a free MathWorks account or sign into Things Speak using an existing account.

Select the Things Speak channel you want your data to stream into. See Collect Data in a New Channel for help creating a new channel.

Record the following for the selected channel:

Channel ID, which is listed at the top of the channel view.

Write API key, which can be found on the API Keys tab of your channel view.

3.2.8 TELNET MOBILE APP:

Telnet is a client-server protocol, based on a reliable connection-oriented transport. Typically, this protocol is used to establish a connection to Transmission Control Protocol (TCP) port number 23, where a Telnet server application (telnetd) is listening. Telnet, however, predates TCP/IP and was originally run over Network Control Program (NCP) protocols.

Even though Telnet was an ad hoc protocol with no official definition until March 5, 1973,[3] the name actually referred to *Teletype Over Network Protocol* as the RFC 206 (NIC 7176) on Telnet makes the connection clear:[4]

The TELNET protocol is based upon the notion of a virtual teletype, employing a 7-bit ASCII character set. The primary function of a User TELNET, then, is to provide the means by which its users can 'hit' all the keys on that virtual teletype.[5]

Essentially, it used an 8-bit channel to exchange 7-bit ASCII data. Any byte with the high bit set was a special Telnet character. On March 5, 1973, a Telnet protocol standard was defined at UCLA[6] with the publication of two NIC documents: Telnet Protocol Specification, NIC 15372, and Telnet Option Specifications, NIC 15373.

Many extensions were made for Telnet because of its negotiable options protocol architecture. Some of these extensions have been adopted as Internet standards, IETF documents STD 27 through STD 32. Some extensions have been widely implemented and others are proposed standards on the IETF standards track (see below) Telnet is best understood in the context of a user with a simple terminal using the local Telnet program (known as the client program) to run a logon session on a remote computer where the user's communications needs are handled by a Telnet server program.

When Telnet was initially developed in 1969, most users of networked computers were in the computer departments of academic institutions, or at large private and government research facilities. In this environment, security was not nearly as much a concern as it became after the bandwidth explosion of the 1990s. The rise in the number of people with access to the

Internet, and by extension the number of people attempting to hack other people's servers, made encrypted alternatives necessary.

3.3 DEFINE THE MODULES:

3.3.1 RFID (RADIO FREQUENCY IDENTIFICATION) MODULE

3.3.1.1 INTRODUCTION:

Radio Frequency Identification (RFID) technology has been attracting considerable attention with the expectation of improved supply chain visibility for both suppliers and retailers. It will also improve the consumer shopping experience by making it more likely that the products they want to purchase are available.

Recent announcements from some key retailers have brought the interest in RFID to the forefront. This guide is an attempt to familiarize the reader with RFID technology so that they can be asking the right questions when considering the technology.

3.3.1.2 WHAT IS RFID?

RFID (Radio Frequency Identification) is a method of identifying unique items using radio waves. Typical RFID systems are made up of 2 major components: readers and tags. The reader, sometimes called the interrogator, sends and receives RF data to and from the tag via antennas. A reader may have multiple antennas that are responsible for sending and receiving the radio waves. The tag, or transponder, is made up of the microchip that stores the data, an antenna, and a carrier to which the chip and antenna are mounted.

RFID technology is used today in many applications, including security and access control, transportation and supply chain tracking. It is a technology that works well for collecting multiple pieces of data on items for tracking and counting purposes in a cooperative environment.

3.3.1.3 THREE PRIMARY FREQUENCY BANDS HAVE BEEN ALLOCATED FOR RFID USE.

Low Frequency (125/134 KHz):

Most commonly used for access control and asset tracking.

Mid-Frequency (13.56 MHz):

Used where medium data rate and read ranges are required.

Ultra-High-Frequency (850 MHz to 950 MHz and 2.4 GHz to 2.5 GHz): offer the longest read ranges and high reading speeds.

Applications for RFID within the supply chain can be found at multiple frequencies and different RFID solutions may be required to meet the varying needs of the marketplace. Many of today's RFID technologies cannot reliably cover areas wider than 4 to 5 feet, making them unsuitable for wide openings that are the norm in manufacturing, distribution and stor receiving dock environments. Since UHF (Ultra High Frequency) can cover portals up to 9 feet
wide it is gaining industry support as the choice bandwidth for inventory tracking applications including pallets and cases.
Technology providers are developing readers that work with multiple system protocols and frequencies so that users will be able to choose the RFID products that work best for their market and products.

3.3.1.4 RFID TAGS ARE FURTHER BROKEN DOWN INTO TWO CATEGORIES:

Active RFID Tags are battery powered. They broadcast a signal to the reader and can transmit over the greatest Distances (100+ feet). Typically, they can cost \$4.00 - \$20.00 or more and are used to track high value goods like vehicles and large containers of goods. Shipboard containers are a good example of an active RFID tag application

Passive RFID Tags do not contain a battery. Instead, they draw their power from the reader. The reader transmits a low power radio signal through its antenna to the tag, which in turn receives it through its own antenna to power the integrated circuit (chip). The tag will briefly converse with the reader for verification and the exchange of data. As a result, passive tags can transmit information over shorter distances (typically 10 feet or less) than active tags. They have a smaller memory capacity and are considerably lower in cost (\$1.00 or less) making them ideal for tracking lower cost items

There are two basic types of chips available on RFID tags, Read-Only and **Read-Write**. Read only chips are programmed with unique information stored on them during the manufacturing process. The information on read-only chips can never be changed.

With Read-Write chips, the user can add information to the tag or write over existing information when the tag is within range of the reader. Read-Write chips are more expensive that Read Only chips. Another method used is something called a "WORM" chip (Write Once Read Many). It can be written once and then becomes "Read only" afterwards. This is a desirable format since companies will be able to write an EPC (electronic product code) to the tag when the product is produced and packaged.

3.3.1.5 HOW WILL RFID AFFECT OUR INDUSTRY?

RFID is expected to provide huge advantages to manufacturers by offering the tools to better plan production and respond more quickly to market demand. It will facilitate automation of inventory counts and speed shipping and receiving at the distribution level. For retailers, it will help to reduce stock-outs, enable product tracking and potentially reduce theft and streamline the POS function. RFID will also open other merchandising opportunities and help with the overall consumer buying experience.

Due to the current cost of the technology (both tags and infrastructure), the initial phase of adoption for retailers is at carton and pallet marking applications. The current technology being adopted for carton and pallet labelling is passive UHF tags (850 MHz – 950 MHz). As the cost of tags and readers comes down, a wider adoption at the item marking level will develop.

In order for RFID to grow quickly, it is important that standards be developed so that the technology providers are working toward a common goal of providing low cost and compatible technologies. Not only will it drive down costs, but standards will also help users to reap the greatest benefit from their investment by providing value throughout the whole supply chain.

3.3.1.6 ORGANIZATIONS FOCUSED ON DEVELOPING RFID STANDARDS:

EPC global, Inc., a division of the Uniform Code Council, and its sponsors are working to standardize a new Electronic Product Code (EPC) as the next standard for identifying products. Their goal is not to replace existing bar code standards but to expand the information available down to unique identifiers for each marked item, and to enable more automatic reading. EPC utilizes the basic structure of the Global Trade Item Number (GTIN).

EPC global, Inc. has proposed open standards for tags and readers with the intention of bringing the costs down to a level where RFID tags could be applied to individual items. The work may lead to the creation of a new global Internet network that would allow companies to track items and enable end users to access the full benefits of RFID.

EPC global, Inc. has developed a specification for RFID tags to be used in the retail sector. The specification does not mandate what type of tag to be used but is intended to provide guidelines on data structure and how the tags should perform so that they can be used over a common platform.The specification requires that the chip contain an Electronic Product Code (EPC). The chip must be able to communicate according to an open standard and meet some minimum requirements so that it can be read by reading devices anywhere.

3.3.1.7 THE ELECTRONIC PRODUCT CODE (EPC):

The EPC is a number made up of a header and 3 sets of data as shown in the figure below. The header identifies the EPC version number – which will allow for different lengths or types of EPC later on. The second part of the number identifies the EPC manager – typically this would be the manufacturer of the item the EPC is attached to. The third part is called object class and refers to the exact type of product– most often the stock-keeping unit (SKU). The fourth series of numbers is the serial number that is unique to the item. (The second and third sets of data are similar in function to the numbers in UPC barcodes.)

ELECTRONIC PRODUCT CODE TYPE 1			
01.0000A89.00016F.000169DCO			
Header	EPC Manager	Object Class	Serial Number
8-bits	28-bits	24-bits	36-bits

Table 3.3.1.7 The Electronic Product Code (EPC)

Above is an example of a 96-bit EPC. It will allow sufficient capacity for 268 million companies. Each manufacturer will have the ability to create up to 16 million object classes with 68 billion serial numbers in each class. This should provide sufficient capacity to cover all products manufactured in the world for many years to come. As an interim step, the Auto-ID center is also proposing a 64-bit tag in order to minimize cost in the near term.

Potential Issues That Need Consideration When Choosing the Type of RFID and

Method for Application to Your Products or Packaging.

Enthusiasm within the RFID industry has resulted in much hype about the technology over the past several years. As a result, it is important to embrace the technology with a bit of caution. The following are some of the issues that require close scrutiny when investigating RFID:

Tag Cost – This should not to be confused with chip cost. Although the goal is to bring the cost of the tag (chip and antenna) down to 5 cents, this goal is in the future since it both assumes manufacturing breakthroughs and is predicated on consumption in the billions of tags per year. Today, the cost is closer to "less than 50 cents" for a read/write solution in high (millions) volume. Ultimate tag cost will also be very much dependent on the type of chip required (read only versus read/write), size of the antenna needed and how it is packaged to meet a specific application.

Tag Size – Tag size is dependent on the read range desired. Although the chips are very tiny, they will not operate without being mounted to an antenna. The size of the antenna will determine the read distance performance of the tag so understanding the size of the antenna needed for the application is more important than the size of the chip alone.

Infrastructure Cost – Much focus appears to be placed on the tag cost since it is a recurring expenditure. Reader cost and infrastructure costs for implementing RFID must also be looked at very closely as well. Both the software systems requirements and physical environment, in which RFID is intended to be used, are critical to the ultimate performance of a system and may require changes to accommodate using it effectively. As an example, RFID chips cannot be read through metal objects. Other forms of electromagnetic interference may also impede performance of he technology and require changes to the physical environment where RFID will be used. The number and types of readers will also be a major expenditure depending on your application.

Read Distances – Read distances for RFID are very much dependent on the frequency chosen for the application. Tag orientation also affects the read range as the range diminishes as the tag is rotated from

being perpendicular to the path to the reader. Reading reliability is quite good when labels are alone in a reader field like cases on a conveyor line, but less certain when the labels are randomly oriented as with labelled cases on a skid. The antenna size (both on the tag and the readers) will also be a determining factor. Hand held readers are not capable of using as much power as stationary readers and as result provide shorter read distances.

Government Regulation – Governments around the world regulate the use of the frequency spectrum. Different countries have already assigned certain parts of the spectrum for other uses and as a result, there is virtually no part of the spectrum that is available everywhere in the world for use by RFID. This means that a RFID tag may not work in all countries. As an example if you choose the Ultra High Frequency (UHF) frequency that

Operates at 915MHz in the U.S. and you ship your product to Europe, they may not be able to be read it since Europe operates in the UHF spectrum at 869 MHz. This is an important consideration when operating in a global environment.

Anti-Collision – This is an important feature of RFID chips/readers since it will allow multiple tags to be read while grouped in one reader field. It is not available on all RFID tags but is an important feature if you are planning to use RFID for inventory counts, shipping and receiving where multiple tags need to be read at the same time.

Privacy Issues – Consumer groups have expressed concern over the potential (real or imagined) privacy invasion that might result with widespread RFID item marking. These groups are pushing for legislation that will require manufacturers to advise consumers that the products contain RFID devices and must provide a means so that the devices can be disabled at point of purchase. These issues are most prevalent at the item marking level and will have little impact on the implementation of carton and pallet labelling.

3.3.2 ESP8266 WIFI MODULE

3.3.2.1 INTRODUCTION

The **ESP8266 Wi-Fi Module** is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your **Wi-Fi** network. The **ESP8266** is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

3.3.2.2 CHARACTER CONFIGURATION:

- 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2),

- general-purpose input/output (16GPIO),
- Inter-Integrated Circuit (I²C) serial communication protocol,
- analog-to-digital conversion (10-bitADC)
- Serial Peripheral Interface (SPI) serial communication protocol,
- I²S (Inter-IC Sound) interfaces with DMA(Direct Memory Access) (sharing pins with GPIO),
- UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2),and
- Pulse-width modulation (PWM).

3.3.2.3 APPLICATIONS:

- Smart Home Appliances
- Improved Home safety and security
- Home air quality and water quality monitoring
- Natural Language-based voice assistants.

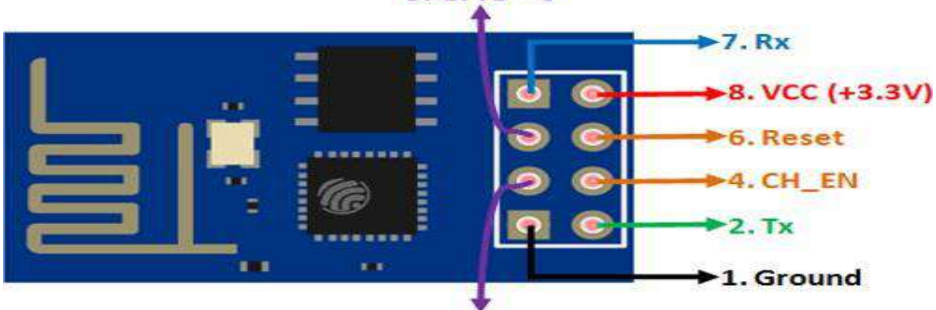


Fig.3.3.2.3 ESP8266 wifi module pin out

3.3.2.4 STRUCTURE AND CONFIGURATION:

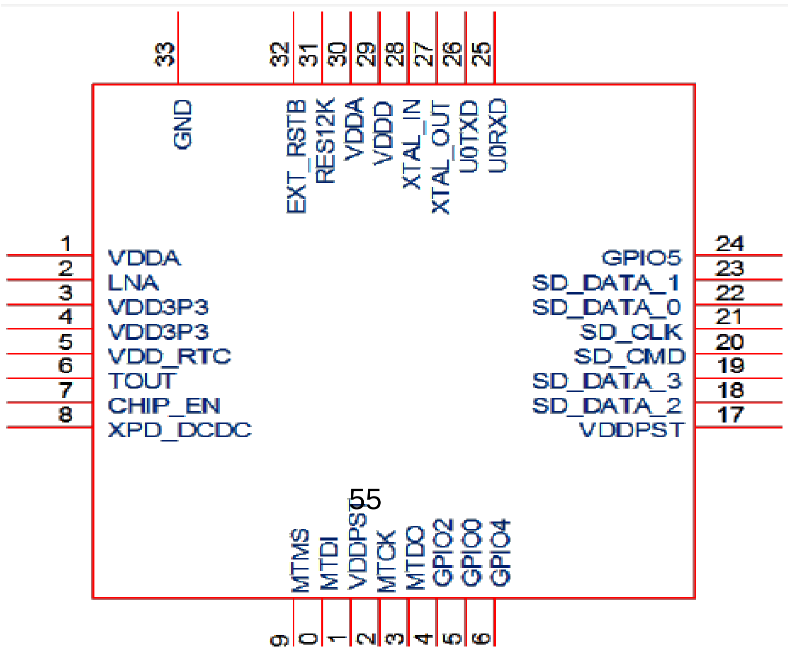


Fig.3.3.2.4 Structure and configuration

3.4 MODULE FUNCTIONALITIES:

3.4.1 RFID MODULE

3.4.1.1 Functionality of Radio Frequency Identification: The Transponder

The RFID transmitter is called a **transponder**, sometimes also a tag or simply a radio tag. It consists of an antenna (often a coil), a computer chip and a carrier material or housing. The chip is equipped with analog and digital circuits, and it also has a memory that can sometimes be written to several times. In some cases, the transponder is a semiconductor chip. Tags differ in several factors. Among other things, they have different sizes, lifetimes, clock frequencies, memory sizes or ranges.

The chips have a unique identity in the form of a **serial number** so that they can be clearly identified. The identity cannot be changed, which can prevent abuse. If it is a rewritable memory, the identity can be extended continuously and fully automatically by additional information (Auto-ID). In this respect, for example, previous locations of goods are interesting. RFID technology is also essential when it comes to choosing the right order picking procedure.

Tags usually do not have their **own energy supply**. In this case, the transponder is called passive. The reader delivers the necessary energy for data transmission to the tag. The antenna must be a coil that charges a capacitor by induction. Exceptions called active transponders have their own battery. Their use is worth it if the tag is to have a long range and various additional functions.

Transponders can be attached to or in the object to be marked. For example, they are stuck on (e.g. on pallets or cartons), embedded directly in objects (e.g. in screws, shoe soles, banknotes or documents) and even implanted in living beings (in pets behind the ear, in people's hands). Successful research and development have made it possible to use temperature-resistant transponders since 2006. Since then, they can even be cast in metals, which has various advantages: the risk of damage, wear and tear or loss is minimised and they are also invisible.

3.4.1.2 Functionality of Radio Frequency Identification: The Reader

A reader that must be in close range reads the tag's data using an alternating electromagnetic field or high-frequency radio waves. The receiver is also known as a **reader**. In addition to purely readout readers, there are also readers that can **write new information to the transponder's memory**. The transmitted radio waves are influenced by the tag because it reflects part of the energy. First, the unique identity and any other requested information are transmitted. In addition, different things can be stored.

The reader is a **combination of hardware and specially programmed software**. The **more range a reader should have, the more energy in the form of electrical current must be used for its use**. Typical ranges are between a few centimeters and a few meters. The smaller the range, the more precisely selected transponders can be addressed.

A longer range can be chosen, for example, if two objects should not be too close together. This is the case, for example, with containers containing flammable liquids and spark spraying equipment. Readers for radio frequency identification are available as handheld versions for manual operation, but can also take on various other forms, for example in the form of a gate through which goods flows.

3.4.2 ESP8266 WIFI MODULE

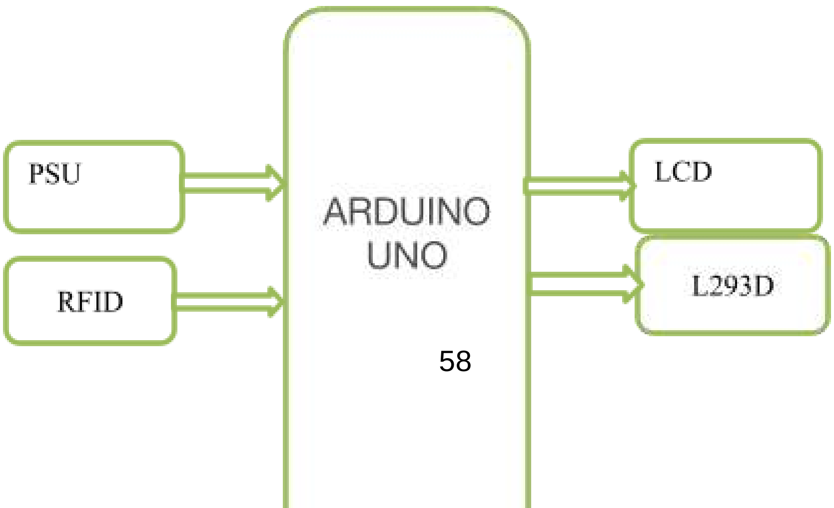
The ESP8266 WiFi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all WiFi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existance interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support. In the Documents section below you will find many resources to aid you in using the ESP8266, even instructions on how to transform this module into an IoT (Internet of Things) solution!

CHAPTER-4
PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM OF PROJECT:



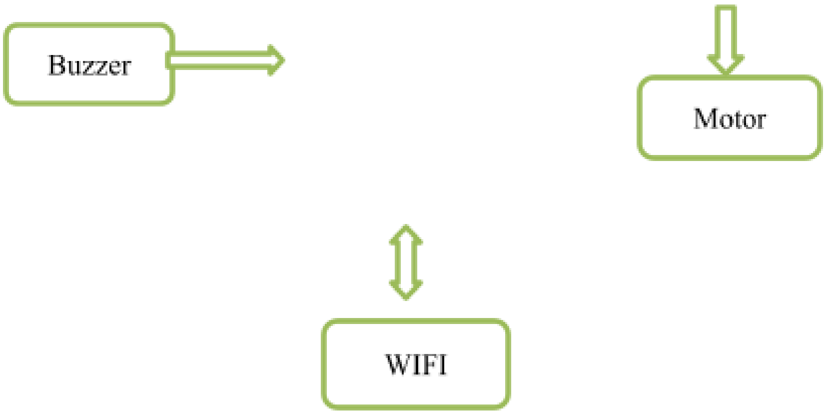


Fig.4.1: Block diagram of the project

4.2 FLOW CHART

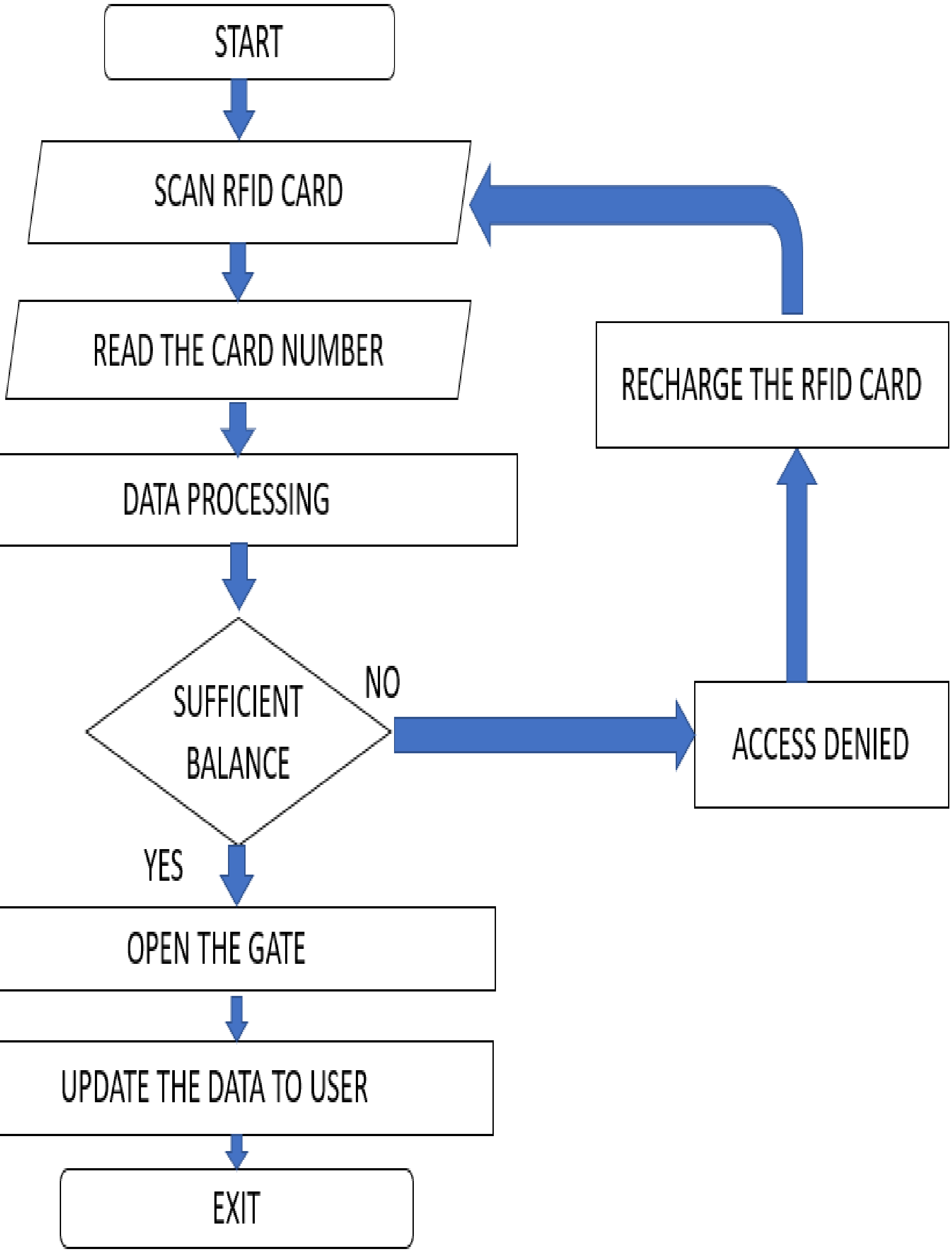


Fig.4.2 Flow Chart

4.3 IMPLEMENTATION STAGES

Stage-1: First on lcd we will get a message displaying waiting for link that indicates the wifi is not yet connected.

Fig.4.3.1 WAITING FOR LINK

Stage-2: In Mobile Telnet App, we will get a message like welcome when the wifi is connected to ESP8266 WIFI Module.

Fig.4.3.2 WELCOME

Stage-3: After the wifi is connected,we will get a message on LCD displaying Please Scan UR Card.

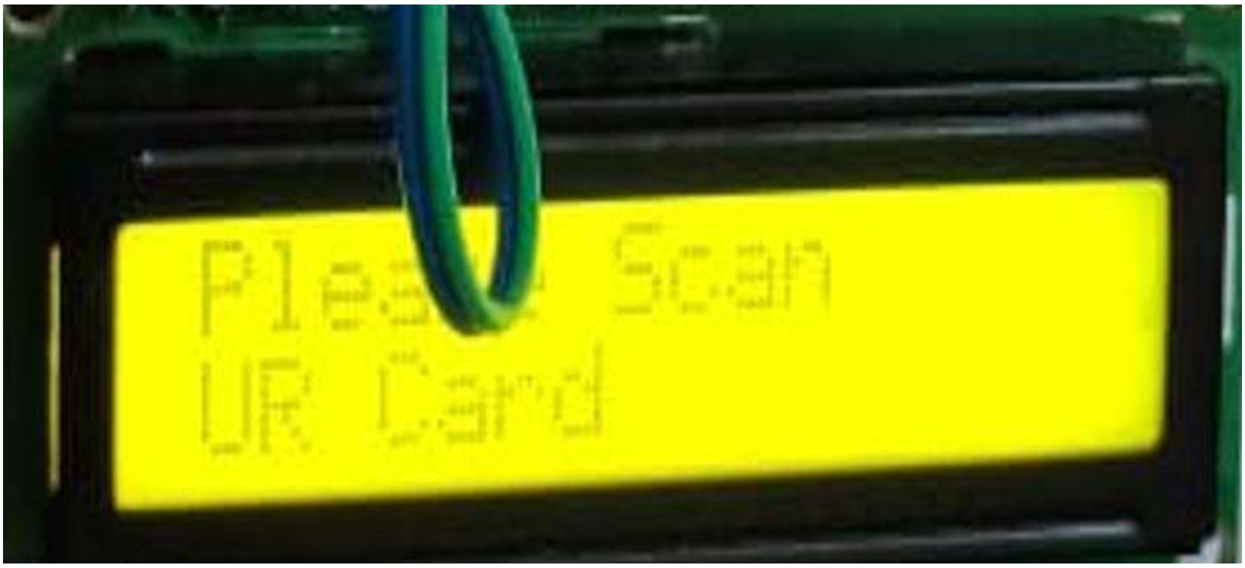


Fig.4.3.3 Please Scan UR Card

Stage-4: When we scan the card, if sufficient amount is available, we will get a message access granted for person and remaining balance will be displayed on LCD.

Fig.4.3.4 Access Granted

Stage-5: When we scan the card, if sufficient amount is not available, we will get a message access denied for person and balance will be displayed on LCD.

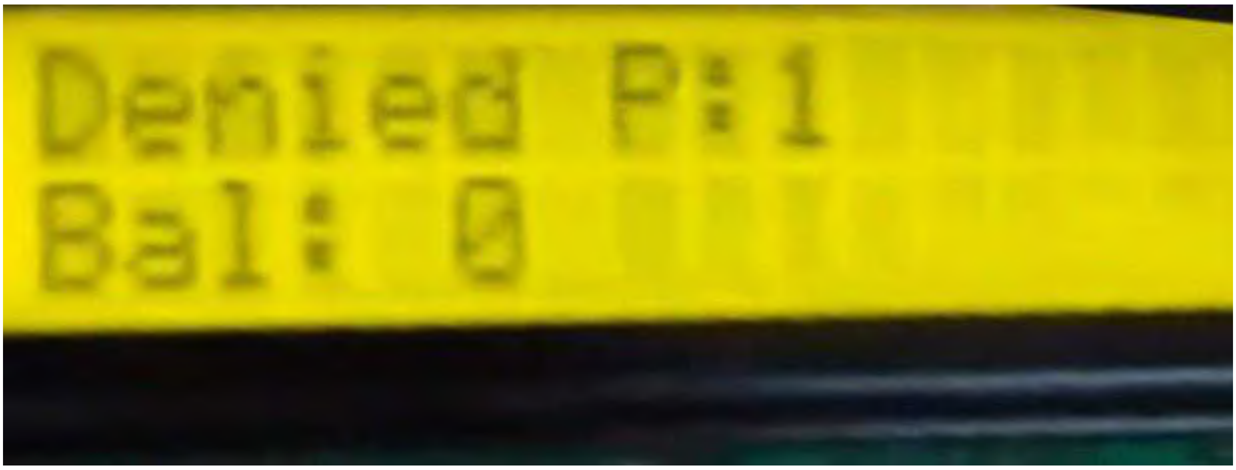


Fig.4.3.5 Access Denied

Stage-6: When card is recharged, amount will be added to card and display’s person recharge done on LCD Screen.

Fig.4.3.6 Recharge Done

4.4 PROJECT CODE

```
#include<SoftwareSerial.h>
SoftwareSerial mySerial(2,3);
#include<LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);
```

```
int m1=4,m2=5;

char res[130];
char buff[130];

void sendwifi(String chr,unsigned int len)
{
  mySerial.print("AT+CIPSEND=0,");
  mySerial.println(len-1);
  delay(2000);
  mySerial.print(chr);
  delay(2000);
}

char t;

void setup()
{
  pinMode(m1,OUTPUT);
  pinMode(m2,OUTPUT);
  digitalWrite(m1,LOW);
  digitalWrite(m2,LOW);
  lcd.begin(16,2);
  lcd.clear();lcd.setCursor(0,0);
  lcd.print("Welcome");
  delay(1000);
  Serial.begin(9600);
  mySerial.begin(115200);
  mySerial.print("AT\r\n");
  delay(1000);
  mySerial.print("ATE0\r\n");
  delay(1000);
    mySerial.print("AT+CWMODE=3\r\n");
    delay(1000);
```

```
    mySerial.print("AT+CWSAP=\"Project\",\"project1235\",5,0\r\n\r\n");
    delay(1000);
    mySerial.print("AT+CIPMUX=1\r\n");
    delay(1000);
    mySerial.print("AT+CIPSERVER=1,23\r\n");
    delay(1000);
    lcd.clear();lcd.setCursor(0, 0);lcd.print("WATING FOR LINK");
    while(1)
    {
      if(mySerial.available())
      {
         //if(Esp.find("0,LINK"))
        if(mySerial.find("0,CONNECT"))
        {
           lcd.clear();lcd.setCursor(0, 0);lcd.print("LINK FOUND");
         break;
        }

      }
    }
    delay(1000);
    sendwifi("WELCOME \r\n",11);
    delay(1000);
}
char tt;
int i=100,j=100,m=100;
char rfid1[15];
void loop()
{
  lcd.clear();lcd.setCursor(0,0);lcd.print("Please Scan");
  lcd.setCursor(0,1);lcd.print("UR Card");
  delay(1000);
 if(Serial.available())
 {
```

```
for(int k=0;k<12;k++)
{
 while(!Serial.available());
  rfid1[k] = Serial.read();
  //Serial.print(rfid[k]);
}
if(strncmp(rfid1,"5400C1374CEE",12)==0)
{
  i=i-20;
  if(i>=10)
   {
   lcd.clear();lcd.setCursor(0, 0);lcd.print("Access Grant P:1");
   lcd.setCursor(0,1);lcd.print("Bal: ");lcd.print(i);lcd.print("rs");
   delay(100);
   sprintf(buff,"Access Grant for Person: 1, Bal: %u\r\n",i);
   sendwifi(buff,strlen(buff));
   delay(100);
   digitalWrite(m1,LOW);
   digitalWrite(m2,HIGH);
   delay(1000);
   digitalWrite(m1,HIGH);
   digitalWrite(m2,LOW);
   delay(1000);
   digitalWrite(m1,LOW);
   digitalWrite(m2,LOW);
   }
if(i<=10)
   {
   lcd.clear();lcd.setCursor(0, 0);lcd.print("Denied P:1");
   lcd.setCursor(0,1);lcd.print("Bal: 0");//lcd.print(i);lcd.print("rs");
   delay(100);
   sprintf(buff,"Access Denied for Person: 1, Bal: %u\r\n",0);
   sendwifi(buff,strlen(buff));
   delay(100);
```

```
    digitalWrite(m1,LOW);
    digitalWrite(m2,LOW);
    delay(3000);
   }
 }
if(strncmp(rfid1,"5400C59D3C30",12)==0)
 {
  j=j-20;
  if(j>=10)
   {
    lcd.clear();lcd.setCursor(0, 0);lcd.print("Access Grant P:2");
    lcd.setCursor(0,1);lcd.print("Bal: ");lcd.print(j);lcd.print("rs");
    delay(100);
    sprintf(buff,"Access Grant for Person: 2, Bal: %u\r\n",j);
    sendwifi(buff,strlen(buff));
    delay(100);
    digitalWrite(m1,LOW);
    digitalWrite(m2,HIGH);
    delay(1000);
    digitalWrite(m1,HIGH);
    digitalWrite(m2,LOW);
    delay(1000);
    digitalWrite(m1,LOW);
    digitalWrite(m2,LOW);
   }
 if(j<=10)
   {
    lcd.clear();lcd.setCursor(0, 0);lcd.print("Denied P:2");
    lcd.setCursor(0,1);lcd.print("Bal: 0");//lcd.print(i);lcd.print("rs");
    delay(100);
    sprintf(buff,"Access Denied for Person: 2, Bal: %u\r\n",0);
    sendwifi(buff,strlen(buff));
    delay(100);
    digitalWrite(m1,LOW);
```

```
    digitalWrite(m2,LOW);
    delay(3000);
   }
  }
 if(strncmp(rfid1,"5400C5CF431D",12)==0)
  {
   m=m-20;
   if(m>=10)
   {
    lcd.clear();lcd.setCursor (0, 0);lcd.print("Access Grant P:3");
    lcd.setCursor(0,1);lcd.print("Bal: ");lcd.print(m);lcd.print("rs");
    delay(100);
    sprintf(buff,"Access Grant for Person: 3, Bal: %u\r\n",m);
    sendwifi(buff,strlen(buff));
    delay(100);
    digitalWrite(m1,LOW);
    digitalWrite(m2,HIGH);
    delay(1000);
    digitalWrite(m1,HIGH);
    digitalWrite(m2,LOW);
    delay(1000);
    digitalWrite(m1,LOW);
    digitalWrite(m2,LOW);
   }
  if(m<=10)
   {
    lcd.clear();lcd.setCursor(0, 0);lcd.print("Access Denied P:3");
    lcd.setCursor(0,1);lcd.print("Bal: 0");//lcd.print(i);lcd.print("rs");
    delay(100);
    sprintf(buff,"Access Denied for Person: 3, Bal: %u\r\n",0);
    sendwifi(buff,strlen(buff));
    delay(100);
    digitalWrite(m1,LOW);
    digitalWrite(m2,LOW);
```

```
    delay(3000);
   }
 }
}

if(mySerial.available()> 0)
   {
     tt = mySerial.read();

 if( tt == '*')
 {
  i=i+100;
  lcd.clear();lcd.setCursor(0,0);lcd.print("Person 1");
   lcd.setCursor(0,1);lcd.print("Recharge Done");
   delay(500);
   sprintf(buff,"Recharge Done for 1 Rs: %u\r\n",100);
   sendwifi(buff,strlen(buff));
   delay(2000);
 }
 if( tt == '#')
 {
  j=j+100;
  lcd.clear();lcd.setCursor(0,0);lcd.print("Person 2");
   lcd.setCursor(0,1);lcd.print("Recharge Done");
   delay(500);
   sprintf(buff,"Recharge Done for 2 Rs: %u\r\n",100);
   sendwifi(buff,strlen(buff));
   delay(2000);
 }
   if( tt == '@')
 {
  m=m+100;
  lcd.clear();lcd.setCursor(0,0);lcd.print("Person 3");
   lcd.setCursor(0,1);lcd.print("Recharge Done");
```

```
    delay(500);
    sprintf(buff,"Recharge Done for 3 Rs: %u\r\n",100);
    sendwifi(buff,strlen(buff));
    delay(2000);
  }
 }
}
```

4.5 RESULTS

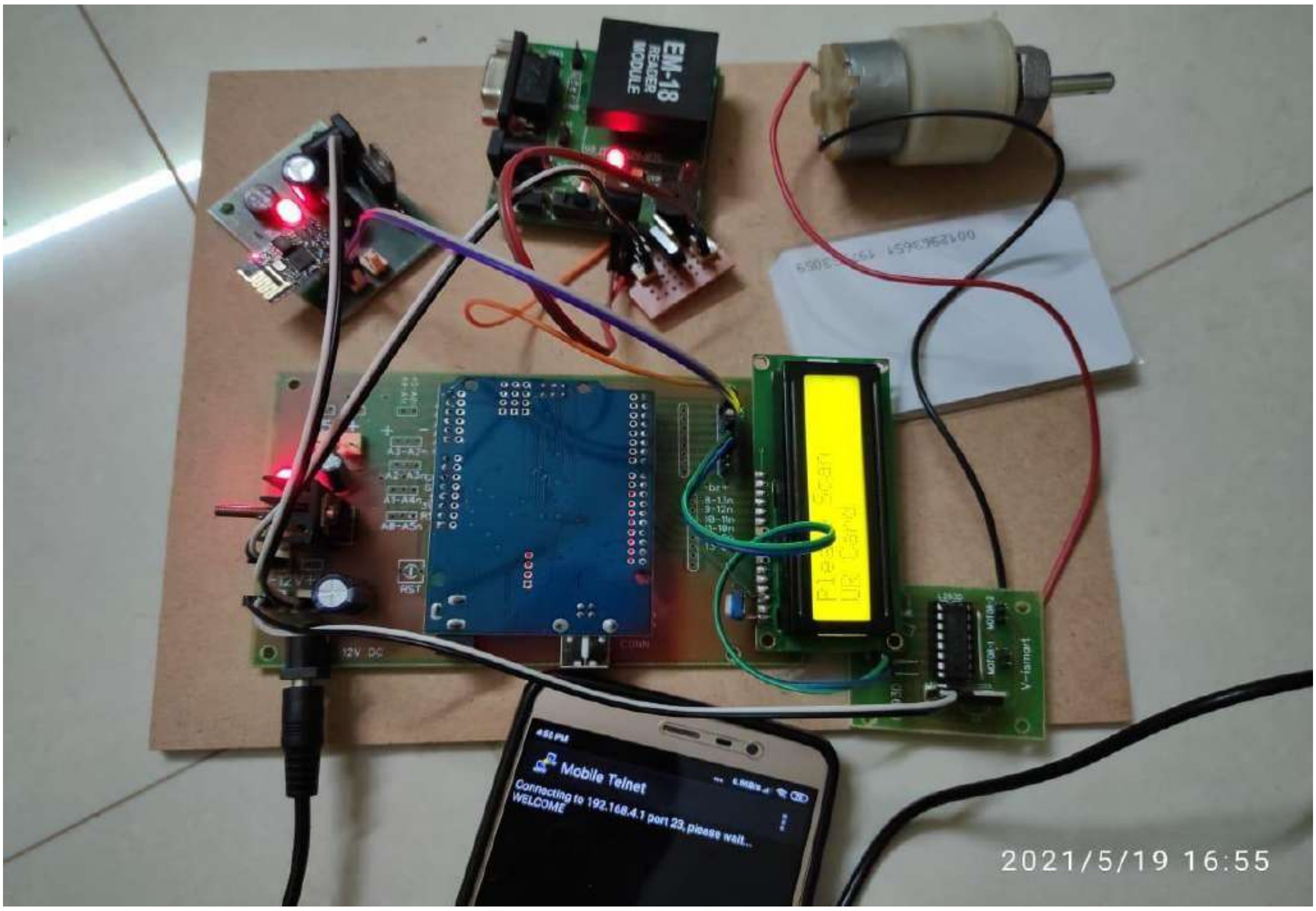


Fig.4.5 Final Output

CHAPTER-5
PROJECT TESTING

5.1 OVERVIEW OF TESTING METHODS

Today, governments want to take these ETC solutions one step further, with more sophisticated systems that not only eliminate the need for toll booths, but also enable governments to charge for road usage based on a variety of factors, including time of day, vehicle weight, distance driven, type of road and the pollution produced by the vehicle. These new systems enable governments to implement programs that charge tolls that better reflect the vehicle's impact on road infrastructure, traffic and the environment, helping them:

- Reduce congestion during peak periods of the day by lowering tolls during low-traffic periods;
- Reduce congestion in specific areas by lowering tolls on specific roads;
- Encourage individuals and companies to purchase low-carbon-emission vehicles by charging higher tolls for vehicles that pollute more; and
- Transition the source of revenue financing road infrastructure from fuel-based taxes to toll on road usage, which will include any type of vehicle (from electric vehicles with small tolls to high-gas-consumption vehicles with high tolls).

Security: IoT is a combination of physical objects ("things"), sensors, embedded software and keeps a goal of effective connectivity and faithful data exchange. The current graphs imply that IoT market growing at remarkable rate. However, some of these devices suffer from limited memory, power consumption and processing power. These issues may cause IoT to become penetrable and hence security is at utmost priority to IoT domain.

As stated in a 2015 report by Hewlett-Packard on IoT research [3]:

1. Per 100, 70 devices still use un-encrypted network services
2. Per 100, 90 devices collected at least one piece of personally identifiable information (via device, cloud or mobile app)
3. Per 100, 70 devices (with cloud and mobile app components) enabled an attacker to identify valid user accounts through enumeration.
4. 80% of devices (with cloud and mobile app components) failed to require passwords of sufficient complexity
5.Behind every 10 device every 6 devices user interfaces (UI) were vulnerable

Following are the vulnerabilities of the IoT as per The Open Web Application Security

Project (OWASP)

1) Insufficient protection compositions.

2) Dubiously secured Software/Firmware

3) Doubtful Network Services

4) Doubtful Web Interface

5) Worst Physical Security

6) Doubtful Cloud Interface

7) Doubtful Mobile Interface

8) Privacy Concerns

9) Doubtful Authentication/Authorization

10) Inadequate Transport Encryption Ongoing approaches to secure IoT have attempted to put a grip on communication protocol-based mechanisms, such as encryption for data-at-rest or in-transit.

But this itself is doubtful if the respective endpoints themselves are capable of being modified either by local access or remote connections. Gartner claims that by year 2020 more than 25% of identified attacks in a particular company will be on IoT devices or systems, even though IoT will only contribute to less than 10% of IT security budgets.

The testing methods of our project that is iot based toll booth manager systemThe requirements for testing this project is mobile app,wifi connection.First on lcd we will get a message displaying waiting for link that indicates the wifi is not yet connected.In Mobile Telnet App, we will get a message like welcome when the wifi is connected to ESP8266 WIFI Module.After the wifi is connected,we will get a message on LCD displaying Please Scan UR Card.When we scan the card, if sufficient amount is available, we will get a message access granted for person and remaining balance will be displayed on LCD.When we scan the card, if sufficient amount is not available, we will get a message access denied for person and balance will be displayed on LCD.When card is recharged, amount will be added to card and display’s person recharge done on LCD Screen.

CHAPTER-6
CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

- The sole purpose of this project was to promote cashless transactions under Digital India initialization. Smooth trafficking at the toll plaza site corruption less transactions as well as reducing air pollution, stagnant long queue traffic will be reduced.

- The system has successfully overcome the drawbacks of the existing system by reducing the man power at the toll booth. It provides easy way of toll collection and maintenance of the information in a reliable and efficient manner.

- We aim to show that Toll booth can be completely managed by „Internet of Things" technology with the help of RFID reader. We present a related literature and market survey to show need of such design. We now discuss the potential directions for improving the design presented in this paper.

- **High data rates**: While the data rates of transmission and reception of data by Wi-Fi module and RFID reader are sufficient for our target application scenarios, we believe that with better access to the hardware functionality of RFID sensors, one can achieve higher data rates.

6.2 FUTURE ENHANCEMENT

- We can add GPS to our project to know live location, if anyone skips the toll booth.

- Voice System can be added in this project to indicate if the card is invalid or has insufficient balance.

- Cloud data can be handled using android application. Person will be granted permissions to register, recharge and renew toll account with the help of android application. Moreover, security system needs to be integrated within this module. Thus, when module is implemented on a greater level, it will result in cost reduction and increase in reliability.

PUBLICATION

Submitted Paper in the Conference ICISECT - 21 with Paper ID (ICISECT21-0003) and got Acceptance for the Paper.

REFERENCES

[1]. Sridevi, Vasantha Pai, Rakesh Y," IoT Based Unmanned Toll Booth Monitoring System" International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, (IJIREEICE), Volume 5, ISSUE 5, 2017.

[2]. R.Brindha, A.Jayanthi, A.Kalaivani, "IoT based Electronic Toll Collection System for Vehicles using Barcode System", International Journal of Electronics, Electrical and Computational System (IJEECS), Volume 7, Issue 3 March 2018.

[3]. Ganesh K. Andurkar, Vidya R. Ramteke, ‘‘Smart Highway Electronic Toll Collection System”, International Journal of Innovative Research in Computer and Communication Engineering, Volume 3, Issue 5, May 2016.

[4]. P. S. Hanwate, Ashlesh Mandke, Narin Meher,” Smart Toll Collection System based on IoT”, IJSTE - International Journal of Science Technology & Engineering, Volume 3, Issue 12, June 2017.

[5]. Rana Pratap, Sanjeev, Niraj Kumar Tiwary, “IoT Based Smart Toll Booth System over Vahan 4.0 Cloud Database", International Journal of Science & Engineering (IJSE), Volume 4, Issue 8,2018.

[6]. Geethanjali N, Chiranth R, Chaitra J M, “IOT Based Intelligent Toll Gate System using Raspberry PI”, International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume 6, Issue 2,2018.

[7]. Satyasrikanth P, Mahaveer Penna, Dileep Reddy Bolla, ‘‘AUTOMATIC TOLL COLLECTION SYSTEM USING RFID”, International Journal of Computer Science and Mobile Computing, Volume 5, Issue 8, August- 2016.

[8]. Allen Francis Antony, Anju P, Gautham Raj P,” Design of an Integrated Speed Tracking and Toll Management System for Express Highways”, Proceedings of the IEEE 2017 International Conference on Computing Methodologies and Communication (ICCMC).

[9]. Amit Hatekar, Nikita Kokal, Tarun Jeswani, Pratik Lalwani,” RFID Based Automatic Toll Collection System

BIBILOGRAPHY

[1] Sighila. P, VinithaValsan ,Preethibha .C,\ IoT Based RFID Gate Automation System", International Journal of Engineering Trends and Technology (IJETT), Vol. 36, Issue 9, June 2016.

[2] Ganesh K. Andurkar, Vidya R. Ramteke, \ Smart Highway Electronic Toll Col-lection System", International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certied Organization), Vol. 3, Issue 5, May 2015.

[3] Arduino Mega Front view",http://www.electroschematics.com/7963/ arduino-mega-2560-pinout/

[4] RFID Reader",https://communities.intel.com/thread/57844

[5] RFID",https://www.researchgate.net/publication235332138 Development of Java based RFID application programmable interface for heterogeneous RFID system

[6] LCD",https://www.researchgate.net/publication/235332138 Development of Java based RFID application programmable interface for heterogeneous RFID system

[7] GSM SIM900A Module",www.forum.researchdesignlab.com/datasheet/ modules/GSMsim900.pdf

[8] ESP 8266-(ESP 01)",www.sparkfun.com/products/13678

[9] Stepper motor",http://www.instructables.com/id/BYJ48-Stepper-Motor/

APPENDICES

This project shows, that the functionalities of the Toll booth can be completely managed using the 'Internet of Things' concept based on the RFID technology. In reference to survey of annual toll collection on toll plazas, conducted by government of Maharashtra in year 2010, we examine and report on the conditions of total time spent, for money transaction on toll plaza, by a particular vehicle and further evaluate total fuel wastage as well as human errors involvement, while doing so. Finally, to provide an optimal solution, we consent with the idea of making toll plazas completely automated, with assistance of IoT technology.

FULLY AUTOMATED SOLAR GRASS CUTTER ROBOT

A PROJECT REPORT

Submitted by

1) Mr. D.Rahul Kumar (16K81A0471)

2) Mr. M.Goutham (16K81A0495)

3) Mr. M.A.Raqueeb (16K81A0496)

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

Electronics & Communication Engineering

Under The Esteemed Guidance of

Mr.N.Vishwanath

Associate Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST.MARTIN'S ENGINEERING COLLEGE
(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

MAY 2021

St.MARTIN'S ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

An UGC Autonomous Institute

Dhulapally, Secunderabad-500 100

NBA & NAAC A+ Accredited

Department of Electronics & Communication Engineering

CERTIFICATE

Certified that this project report titled "*FULLY AUTOMATED SOLAR GRASS CUTTER ROBOT*", is a *bona fide* work of **1.Mr. D.Rahul Kumar (16K81A0471), 2.Mr. M.Goutham (16K81A0495), 3.Mr. M.A.Raqueeb (16K81A0496)** who carried out the work under my supervision, for the partial fulfillment of the requirements for the award of the degree of *Bachelor of Technology* in Electronics & Communication Engineering. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion.

Place: Secunderabad

Date:

Dr..B.HariKrishna,
B.Tech. M.Tech, Ph.D.,
Professor & HOD
Department of ECE
St.Martin's Engineering College,

COUNTERSIGNED

HEAD
Department of Electronics & Communication Engineering
St.Martin's Engineering College

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT ABDUL RAQUEEB MOHAMMAD WITH ROLL NO.16K81A0496, D RAHUL KUMAR WITH ROLL NO.16K81A0471, METTA GOUTHAM WITH ROLL NO.16K81A0495, OF B.TECH – IV YEAR, ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT OF ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT LASYA IT SOLUTION PVT. LTD, KOMPALLY.

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "FULLY AUTOMATED SOLAR GRASS CUTTER" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR.

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD.

DECLARATION

We, the students of '**Bachelor of Technology in Department of Electronics and Communication Engineering**', session: 2020 – 2021, **St. Martin's Engineering College**, hereby declare that the project work entitled '**Fully Automated Solar Grass Cutter Robot**' is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

D.Rahul Kumar	M.Goutham	M.A.Raqueeb
16K81A0471	16K81A0495	16K81A0496

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded effects with success.

First and foremost, I would like to express our deep sense of gratitude and indebtedness to my **College Management** for their kind support and permission to use the facilities available in the Institute.

I especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St.Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

I would like to express my gratitude to **Dr.B.HARI KRISHNA**, **Head &Professor**, Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. I am proud to record that I had the opportunity to work with an exceptionally experienced Professor like him. The time spent with him will remain in my memory for years to come.

I would like to express my gratitude to my department. Project coordinator **Dr.A.ANAND, Professor** for his involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We are highly grateful and obliged to **our college level Project Coordinators**, **Ms.P.Pusha Assistant Professor,** Department of Electronics & Communication Engineering for their advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

Finally, I am highly grateful and obliged to my project guide **Mr.N.Vishwanath, Associate Professor,** for his advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

D.Rahul Kumar	M.Goutham	M.A.Raqueeb
16K81A0471	16K81A0495	16K81A0496

TABLE OF CONTENTS

ABSTRACT

The fully automated solar grass cutter is a fully automated grass cutting robotic vehicle powered by solar energy that also avoids obstacles and is capable of fully automated grass cutting without the need of any human interaction. The system uses 6V batteries to power the vehicle movement motors as well as the grass cutter motor. We also use a solar panel to charge the battery so that there is no need to charge it externally. The grass cutter and vehicle motors are interfaced to an Arduino family microcontroller that controls the working of all the motors. It is also interfaced to an ultrasonic sensor for object detection. The microcontroller moves the vehicle motors in forward direction in case no obstacle is detected. On obstacle detection the ultrasonic sensor monitors it and the microcontroller thus stops the grass cutter motor to avoid any damage to the object/human/animal whatever it is. Microcontroller then turns the robot as long as it gets clear of the object and then moves the grass cutter in forward direction again

CHAPTER-1

INTRODUCTION

1.1. Introduction to the project:

Present days the very large lawns in colleges, public parks, botanical gardens are kept up manually. To cut the grass the gardener uses hands and scissors. But it is not uniform as well as it is not an easy task to finish within the time limit. Most of the lawn mowers are used with electrical energy, gasoline and also need a lot of human effort. In present days the energy sources, electricity and the other non-renewable sources are becoming short and in the future, they may get exhausted. So we focused on renewable resources i.e. solar energy. The solar grass cutter is a fully automated grass cutting vehicle powered by solar energy. It cuts the grass at a very high RPM.The solar grass cutting machine is a fully automated machine power-driven by solar energy. It also detects the obstacles in the path based on that changes the movement direction. It does not require any human interaction. The motors i.e vehicle and grass cutter motors are interfaced to a microcontroller. working of all motors completely controlled by the PICmicrocontroller. An ultrasonic sensor is incorporated into a microcontroller to detect the obstacles which are in the predefined path. The sensors are working as the eyes of an automated machine. Suppose if no sensors are equipped to the machine, the vehicle goes straight and may hit the objects and the vehicle damages. For object detection, we are using two ultrasonic sensors. If the object was detected by the left sensor the vehicle moves to the right with the help of the motor driver and cuts the grass. The smart grass cutter structure is an automated yard mover element. A grasscutter is an edge that is outfitted to an automated machine that takes into account that grass cutting at a very high RPM. The background has smart practicality that permits it to cover the total territory of the yard by distinguishing corners to make use of ultrasonic sensors. The end goal is to cover the whole region by moving the machine is loaded with blades. A microcontroller-based circuit is used in this productive based framework to accomplish the task. There are two batteries are used in this machine. In these two batteries, one is used to run the DC engines and the other one is used to control the engine of the grass cutter. Solar panels are used to charge the batteries which are loaded on the top of the machine.

Ultrasonic sensors are also used in this machine. The framework also makes use of a gyro sensor as a part of the request to accomplish consummate 180 degree turns keeping in mind the end goal to accomplish finish grass/cultivate scope. In this manner, this framework takes into deliberation a mechanized grass-cutting framework without the prerequisite for any human intervention.

1.2 Literature survey:

In this project, the author mainly focused on a grass cutter with helix-shaped cutting blades. Nowadays, grass cutters are functioned by using electrical energy as well as fuel. The main objective behind this design is a machine with a mower that is handy, durable, easy to control and it has a low maintenance cost. Particularly in our, we focused on agricultural needs. We developed an automated machine for agricultural field works i.e to cuts the crops in the different types of fields. There are different types of components are used to develop this simple automated machine. Those are the base frame, wheels, bearings, roller, and gear arrangements. A revolve cutting blade is fitted below the gear. The reel mover is rotated by the gear arrangements that tend to cut the crops or plants. The reel is made of several helix-shaped blades attached to a rotating shaft. The complete set up is positioned on a movable base that has wheel prearrangement. It is used to uphold and maintenance lawns in gardens, schools, colleges, etc.[1].

2.2. Paper-2: Fabrication and Analysis of Lawn Mower Revolving blades are used in grass cutter to cut the grass and garden land space at different lengths and different designs. A blade that rotates on a vertical axis is known as a cutter and a blade assembled on a horizontal axis is known as cylinder or reel cutters. There are different kinds of machines that are developed based on the requirements. For residential gardens and lawns, the compatible type is pushed by human beings. While, superior, autonomous,ride-on mowers are appropriate for large lawns. The biggest, multi-gang mowers dragged behind a tractor, are considered for large areas of grass such as golf courts metropolitan, public, community parks [2].

In 1827, the first grasscutter was developed by Edwin budding in Stroud, Gloucestershire. Primarily it is designed to cut the grass on the luxurious country parks, botanical gardens, and sports gardens. It became an alternative to the scythe. It was patented in 1830. Edwin Budding done an agreement with John Ferrabee dated 18 –May- 1830. As per the agreement between these two people the development cost paid by the Ferrabee obtained the patent letter and acquired rights to manufacture. He was sold the license to other industrialists in the manufacture of lawnmowers. The first chain-driven grass cutter is developed by Thomas Green in the year 1859. The production of lawnmowers commenced in the 1860s. By the year 1862, Farrabee's corporation was building eight models in different roller dimensions. The production was ceased in the year 1863. Until that he was built more than 5000 machines. A human-pushed grass-cutter was designed by Elwood McGuire of Richmond, Indiana in 1870. It was a very lightweight and profitable success. In the year 1899, a better-quality gyratory razor blade grasscutter was patented by John Burr. For better performance wheel alignment is altered in this machine. The Archimedean cutter was built by the Amariah Hills in the year 1871. About, 1900 the Ransomes' Automaton is the finest English instrument. It was available in two variants those are gear-driven and chain models. After world war-I, JP Engineering company situated in Leicester developed and produced popular chain-driven machines. The escalation in the fame of grassland games helped quickly the spread of the creation. Lawnmowers turn out to be a new resourceful alternate to the scythe and trained grazing creatures. In the year 1893, a steam-powered lawn mower was patented by James summer who lives in Lancashire. Petrol or kerosene was used as fuel for this machine. Afterward, several improvements, the pieces of machinery were sold by the Stott Fertilizer and Insecticide Company of Manchester. Later on, the Leyland Steam Motor Company took over the control of sales. Several industrialists go into the arena with gasoline-driven mowers afterward at the turn of the period. The roller drive lawn mower has altered very little meanwhile about 1930. In the United States, in the year 1919, Worthington assembled Gang mowers with various sets of blades. Then, Jacobsen Corporation bought this company but the name of Worthington is still cast on the structures of their gang machinery [2]. 2.3.

Paper-3: Modification of Solar Grass Cutting Machine Cutting grass can't be effortlessly done through elders and youngers. Grasscutter transforming using appliance creates sound pollution owing to the loud engine. and nearby air pollution also rises due to the combustion in the engine. Nearby air pollution as well increases because of the burning in the engine. Periodic maintenance is required for these types of machines for instance changing the oil. The photovoltaic lawnmower is ecologically friendly. Motor-powered machines are additionally dangerous and can't be situated without difficulty castoff through all thus we developed the automated grasscutter handy to function using the remote control. Also, the Battery operated grass cutter remains manual. These grass cutters are using the photovoltaic panels to charge the batteries using solar energy as well as charging done through Alternate Current. To increase the cutting efficiency of the machine is having the spiral cutting blades. It can be operated in two modes those are manual mode and automatic mode. The automatic mode of operation is controlled by remote control. Boxes are attached to the grass cutter to collect the grass cut by the blades automatically [3].

1.3 PROPOSED SYSTEM:

The source is determined commencing from solar energy using a solar panel that energizes the battery and makes use of for driving the process of the system. The microcontroller controls the entire system. A 40 kHz ultrasonic sensor, microcontrollers are used to automated the system. These are helpful for object detection. DC motors are used for wheel movement and cutting operations. An L298 driver circuit is used to achieve the compatibility of motors and the microcontrollers. Here, the output of the microcontroller enhances by using the driver circuit. The cutting action of the blade is delivered by a motor of 1000rpm. The previous technological know-how of grass slicing is hand-operated with the aid of the use of hand gadgets like cutters, scissors, these consequences in better human work, and abundant time necessary for accomplishing the work. Also in old techniques deficiency of uniformity of the remaining grasses. The usage of engine-powered machines will increase sound and air pollution. Additionally, these machines require high maintenance. Nowadays grass cutting machines are turning into very popular. In the olden model of grasscutter, the IC engine is used, and subsequently, due to the fact of ecological influence, air contamination level growths and ic engine grass cutters are also more costly. Looking after such a traditional grass cutter is more. Headed to cling away from these disadvantages we make a strategy to construct an innovative kind of photovoltaic power and this mannequin is additionally economical. The main goal behind this project is headed for reducing the grass which functions on photovoltaic power consequently except for electrical energy and diminishes manpower. In this project, a microcontroller is used for controlling quite several operations of the grass cutter. Also, the grass cutter operates automatically and much less usage of manpower, for this reason, it does now not require a professional individual to operate

CHAPTER-2

INTRODUCTION ABOUT EMBEDDED SYSTEMS

2.1 Introduction:

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.

The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases, a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes.

The applications software on such processors is sometimes referred to as firmware.

The simplest devices consist of a single microprocessor (often called a "chip"), which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine.

As the embedded system is the combination of both software and hardware

2.2 Block diagram of embedded system:

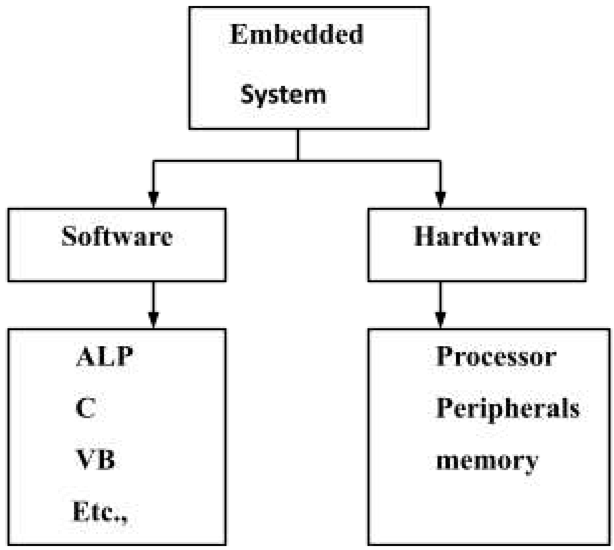


Fig.2.2: Block diagram of embedded system

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory

2.3 Applications of embedded systems:

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance and commercial
- Medical diagnostics, monitoring and life support
- Testing, monitoring and diagnostic systems

2.4 Micro Processor (µp):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Processors are classified into four types like:

- Microprocessor (µp)
- Micro controller (µc)
- Digital Signal Processor (DSP)
- Application Specific Integrated Circuits (ASIC)

2.5 Three Basic Elements of a Microprocessor:

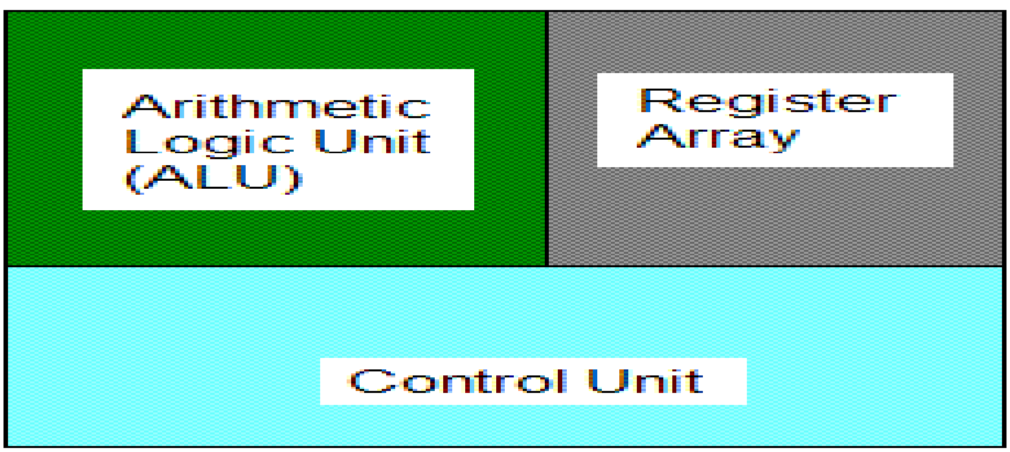


Fig.2.5: Three basic elements of a microprocessor

2.6 Harvard Architecture:

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

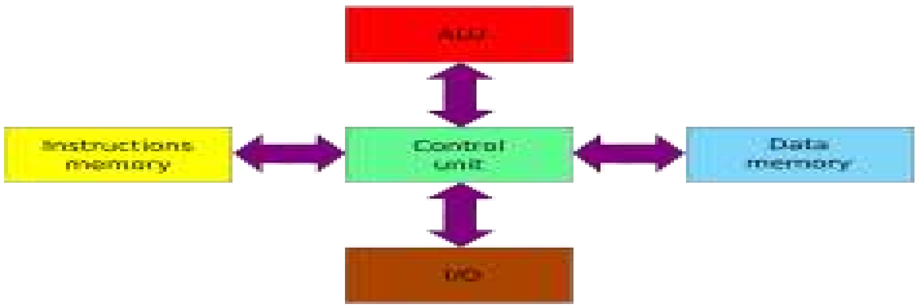


Fig.2.6: Harvard Architecture

2.6.1 Uses of the Harvard architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

- Digital signal processors (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behaviour must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing. Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.

- Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction prefetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

2.6.2 Von-Neumann Architecture

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. They cannot be performed at the same time. The von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A stored-program digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were an advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

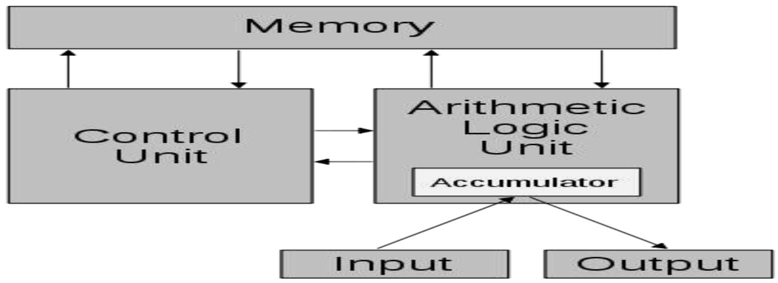


Fig.2.6.1: Schematic of the Von-Neumann Architecture.

CHAPTER-3

MICRO CONTROLLER UNIT

3.1 ATMEGA328

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The ATmega8 microcontroller contains 32 general purpose working registers. As shown in the below figure these registers are directly connected to ALU. Two registers can carry one single instruction consequently in one clock cycle.

Specifications:

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 Ma
DC Current on 3.3V Pin	50 Ma
Flash Memory	32 KB (0.5 KB is used for Boot loader)
SRAM	2 KB
EEPROM	1 KB

Frequency (Clock Speed)	16 MHz

Table 3.1: Atmega328 specifications

Arduino:

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

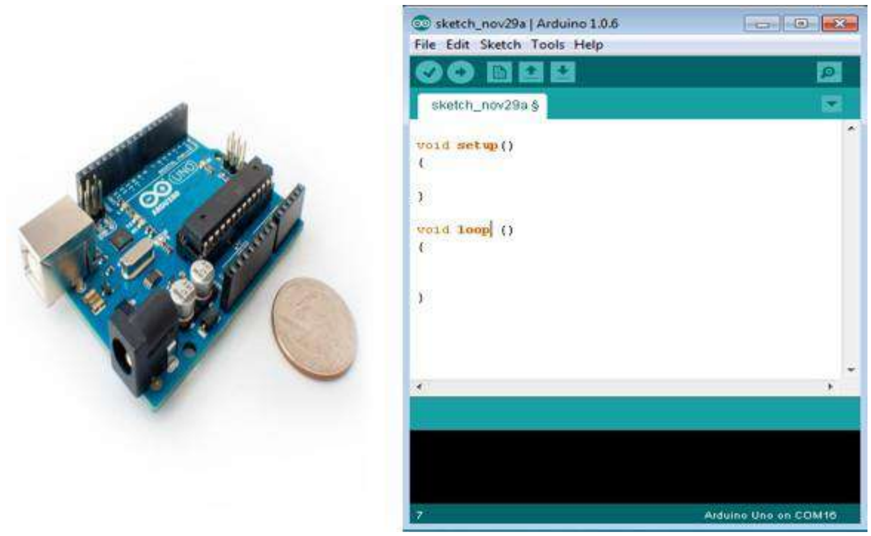


Fig 3.1 Arduino Uno

3.2 Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Table 3.1 Arduino boards based on ATMEGA328 microcontroller

Board Name	**Operating Volt**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**

Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATMega 16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATMega 16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Compatible Header

Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI-Compati
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI-Compatible Header

Table 3.2 Arduino boards based on ATMEGA32u4 microcontroller

Board Name	**Operating Volt**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

Table 3.3 Arduino boards based on ATMEGA2560 microcontroller

Board Name	**Operating Volt**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**

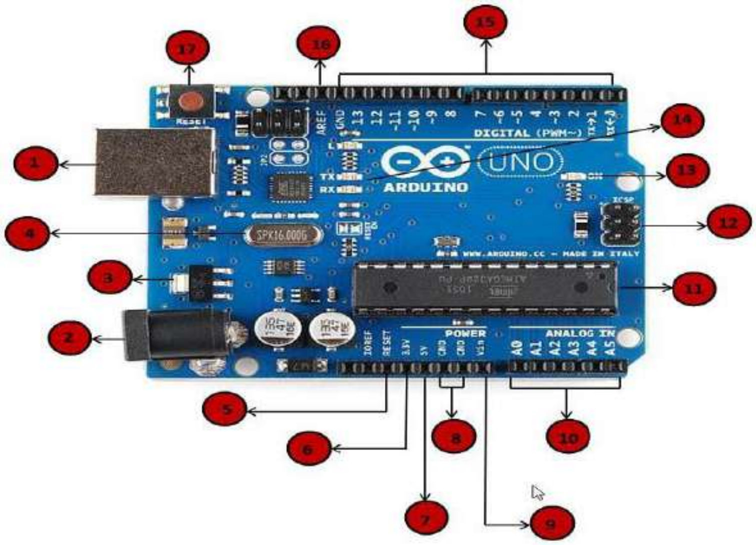
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATMega16U2B
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header

Table 3.4 Arduino boards based on AT91SAM3X8E microcontroller

Board Name	**Operating Volt**	**Clock Speed**	**Digital i/o**	**Analog Inputs**	**PWM**	**UART**	**Programming Interface**
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

3.2.1 Board Description:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.



1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5, 17	**Arduino Reset**

	You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6, 7, 8, 9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral

	Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

3.5 ARDUINO FAMILY

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of

Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project , check this guide for some helpful hints. Here are a few options that are well suited to someone new to the world of Arduino.

Fig.3.5: Arduino Family

3.6 SHIELDS

Additionally, there are these things called shields - basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen and much more.

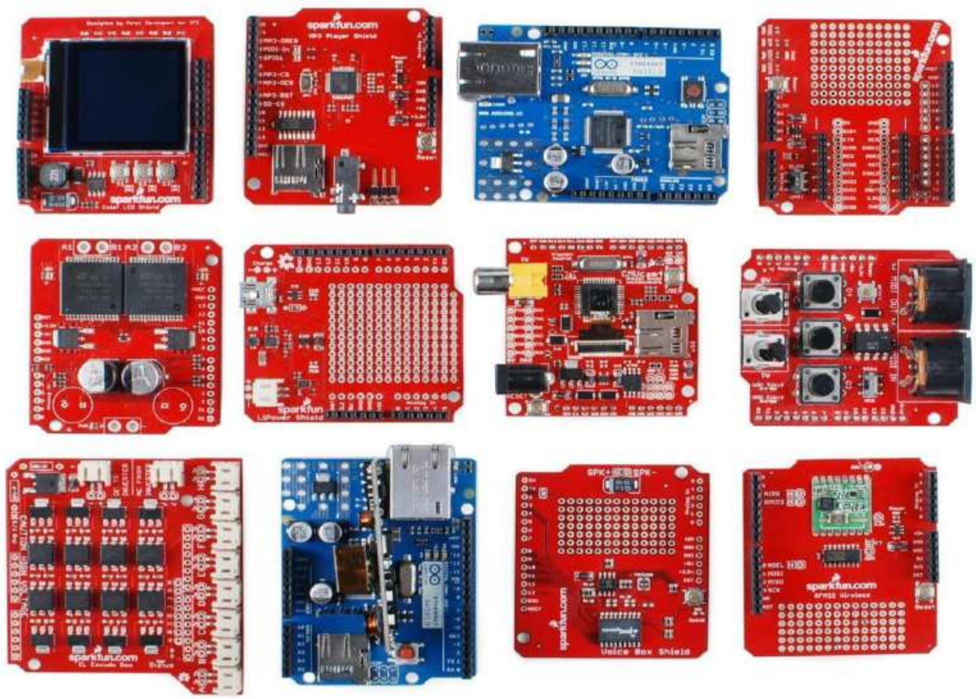


Fig.3.6: Arduino Shields

3.7 PIN DESCRIPTION OF ATMEGA328

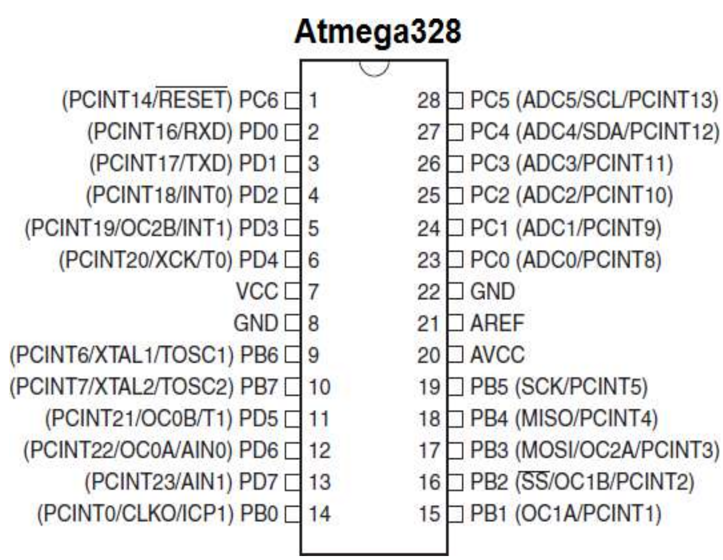


Fig.3.7: Pin description of ATMEGA328

ADVANTAGES OF ARDUINO

- It is cheap
- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
- For beginners, it is very simple to use.

APPLICATIONS

ATMEGA328 is commonly used in many projects and autonomous systems where a simple, low- powered, low- cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino UNO and Arduino Nano model

CHAPTER-4

POWER SUPPLY UNIT

4.1 INTRODUCTION:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as “Regulated D.C Power Supply”.

4.1.1 BLOCK DIAGRAM OF POWER SUPPLY:

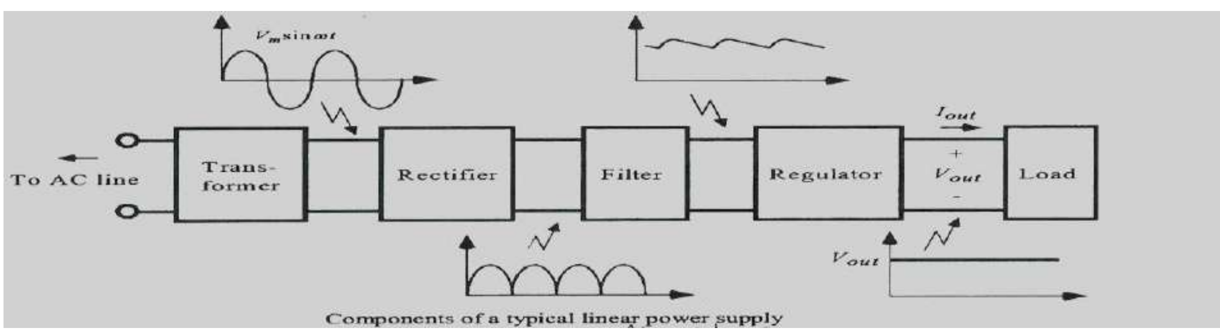


Fig.4.1.1: Block Diagram of Power Supply

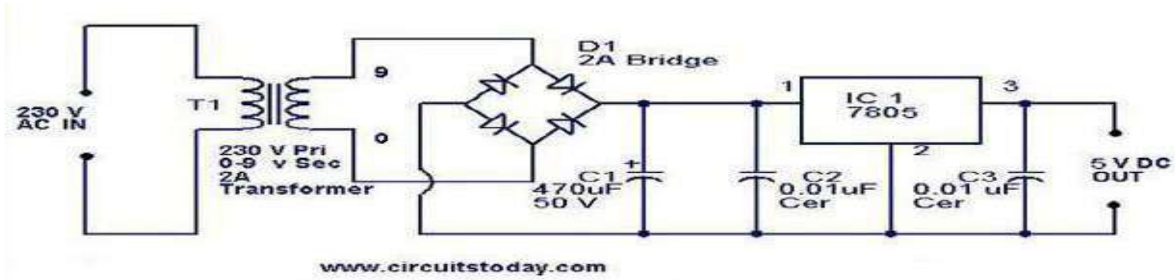


Fig.4.1.1(A): Schematic Diagram of Power Supply

4.1.2 DESCRIPTION OF POWER SUPPLY:

A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a a different form of energy – such as solar, mechanical, or chemical - into electrical energy.

A power supply provides components with electric power. The term usually pertains to devices integrated within the component being powered. For example, computer power supplies convert AC current to DC current and are generally located at the rear of the computer case, along with at least one fan. A power supply is also known as a power supply unit, power brick or power adapter.

4.2 TRANSFORMER:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. A varying current in the first or *primary* winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the *secondary* winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

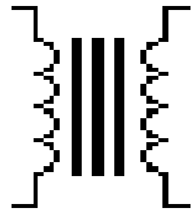


Fig.4.2: Transformer Symbol

(or)

Transformer is a device that converts the one form energy to another form of energy like a transducer.

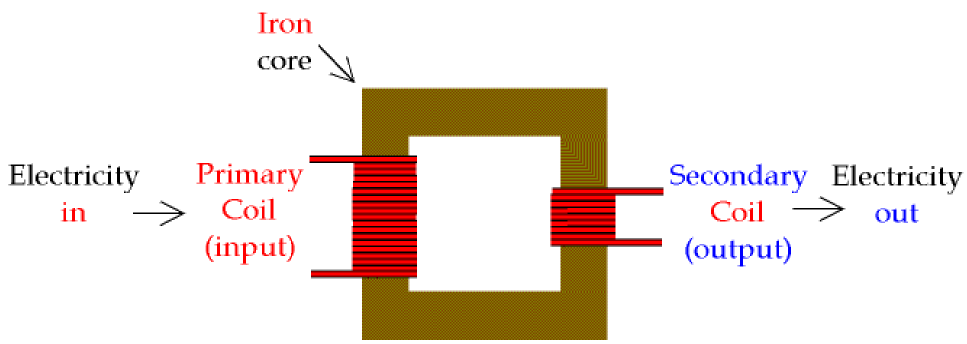


Fig.4.2(A): Transformer

4.2.1 Basic Principle of Transformer:

A transformer makes use of Faraday's law and the ferromagnetic properties of an iron core to efficiently raise or lower AC voltages. It of course cannot increase power so that if the voltage is raised, the current is proportionally lowered and vice versa.

From Faraday's Law

$$\frac{V_S}{V_P} = \frac{N_S}{N_P}$$

For ideal transformer

The voltage ratio is equal to the turns ratio, and power in equals power out.

From conservation of energy

$$P_P = V_P I_P = V_S I_S = P_S$$

Working of Transformer:

A transformer consists of two coils (often called 'windings') linked by an iron core, as shown in figure below. There is no electrical connection between the coils, instead they are linked by a magnetic field created in the core.

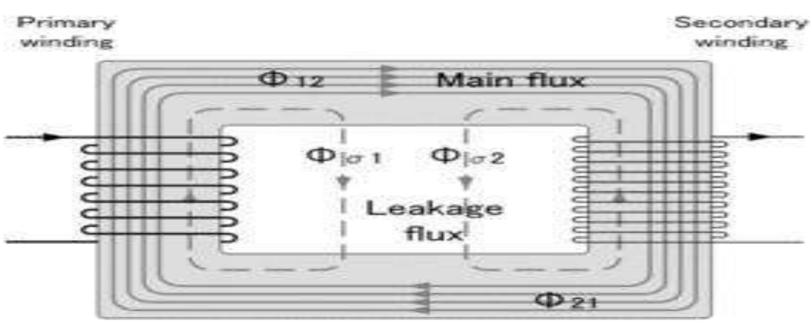


Fig.4.2.1: Basic Transformer

Transformers are used to convert electricity from one voltage to another with minimal loss of power. They only work with AC (alternating current) because they require a changing magnetic field to be created in their core. Transformers can increase voltage (step-up) as well as reduce voltage (step-down).

Alternating current flowing in the primary (input) coil creates a continually changing magnetic field in the iron core. This field also passes through the secondary (output) coil and the changing strength of the magnetic field induces an alternating voltage in the secondary coil. If the secondary coil is connected to a load the induced voltage will make an induced current flow. The correct term for the induced voltage is 'induced electromotive force' which is usually abbreviated to induced e.m.f. The iron core is laminated to prevent 'eddy currents' flowing in the core.

These are currents produced by the alternating magnetic field inducing a small voltage in the core, just like that induced in the secondary coil. Eddy currents waste power by needlessly heating up the core but they are reduced to a negligible amount by laminating the iron because this increases the electrical resistance of the core without affecting its magnetic properties.

Transformers have two great advantages over other methods of changing voltage:

1. They provide total electrical isolation between the input and output, so they can be safely used to reduce the high voltage of the mains supply.
2. Almost no power is wasted in a transformer. They have a high efficiency (power out / power in) of 95% or more.

4.2.2 Classification of Transformer:

- Step-Up Transformer
- Step-Down Transformer

Step-Down Transformer:

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage

applied to it. For instance, a step down transformer is needed to use a 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn's ratio of the two sets of windings determines the amount of voltage transformation.

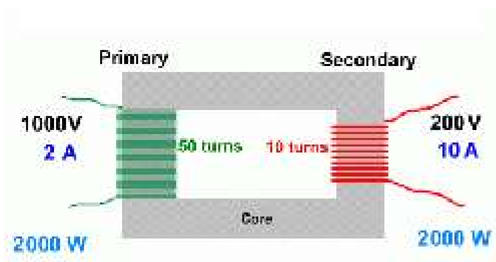


Fig.4.2.2: Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.

Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn's ratio" (except for single phase smaller than 1 kva which have compensated secondary). A practical application of this 2 to 1 turn's ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single phase steps down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages. (Note: single phase steps up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired.)

Step-Up Transformer:

A step up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input.

Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply. A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step-up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

The primary components for voltage transformation are the step up transformer core and coil. The insulation is placed between the turns of wire to prevent shorting to one another or to ground. This is typically comprised of Mylar, nomex, Kraft paper, varnish, or other materials. As a transformer has no moving parts, it will typically have a life expectancy between 20 and 25 years.

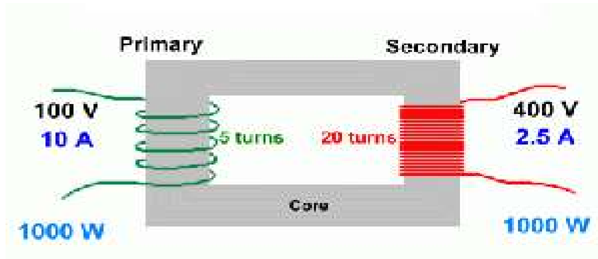


Fig.4.2.2(a): Step-Up Transformer

4.3 DIODES:

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

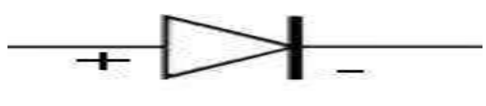


Fig.4.3: Diode Symbol

A **diode** is a device which only allows current to flow through it in one direction. In this direction, the diode is said to be 'forward-biased' and the only effect on the signal is that there will be a voltage loss of around 0.7V. In the opposite direction, the diode is said to be 'reverse-biased' and no current will flow through it.

4.4 Rectifier:

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. There are two different rectification circuits, known as **'half-wave'** and **'full-wave' rectifiers**. Both use components called **diodes** to convert **AC into DC**.

The Half-Wave Rectifier:

The half-wave rectifier is the simplest type of rectifier since it only uses one diode, as shown in figure.

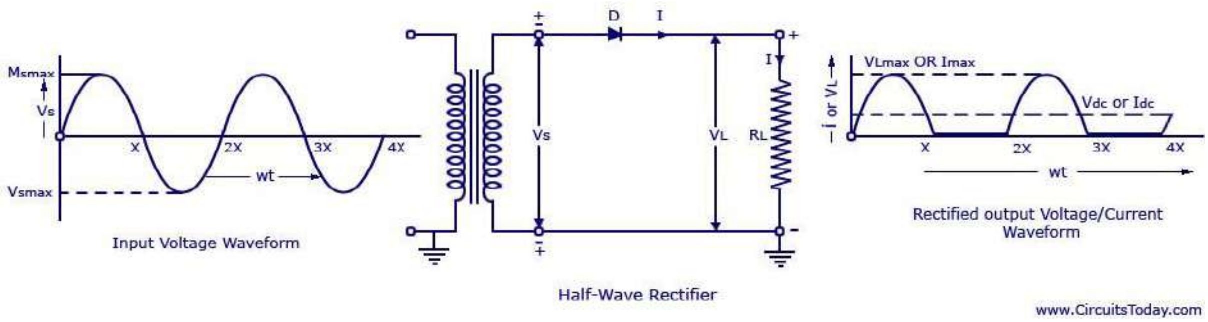


Fig.4.4(a): Half Wave Rectifier

Figure 2 shows the AC input waveform to this circuit and the resulting output. As you can see, when the AC input is positive, the diode is forward-biased and lets the current through. When the AC input is negative, the diode is reverse-biased and the diode does not let any current through, meaning the output is0V. Because there is a 0.7V voltage loss across the diode, the peak output voltage will be 0.7V less than Vs.

While the output of the half-wave rectifier is DC (it is all positive), it would not be suitable as a power supply for a circuit. Firstly, the output voltage continually varies between 0V and Vs-0.7V, and secondly, for half the time there is no output at all.

The Full-wave Rectifier:

The circuit in figure 3 addresses the second of these problems since at no time is the output voltage 0V. This time four diodes are arranged so that both the positive and negative parts of the AC waveform are converted to DC.

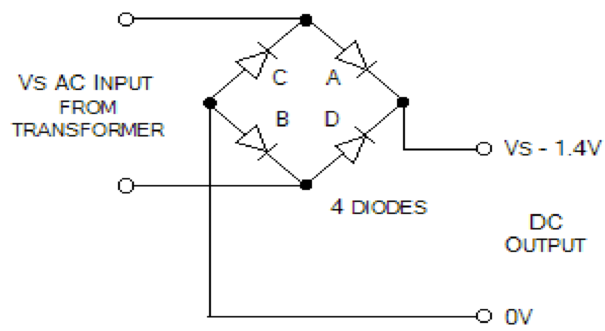


Fig.4.4(b): Full-Wave Rectifier

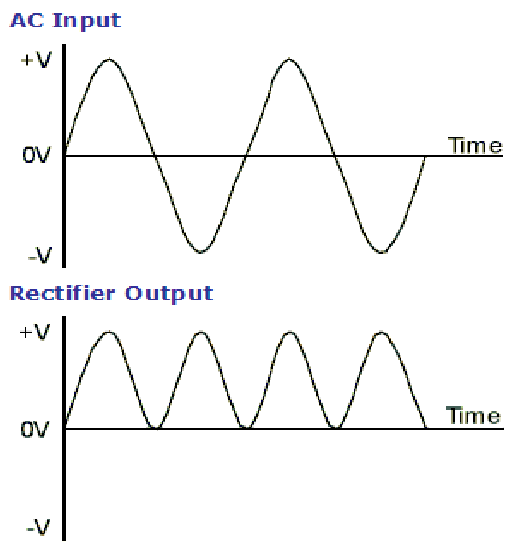


Fig.4.4(c): Full-Wave Rectification

When the AC input is positive, diodes A and B are forward-biased, while diodes C and D are reverse-biased. When the AC input is negative, the opposite is true - diodes C and D are forward-biased, while diodes A and B are reverse-biased. While the full-wave rectifier is an improvement on the half-wave rectifier, its output still isn't suitable as a power supply for most circuits since the output voltage still varies between 0V and Vs-1.4V. So, if you put 12V AC in, you will 10.6V DC out.

4.5 CAPACITOR FILTER:

The capacitor-input filter, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

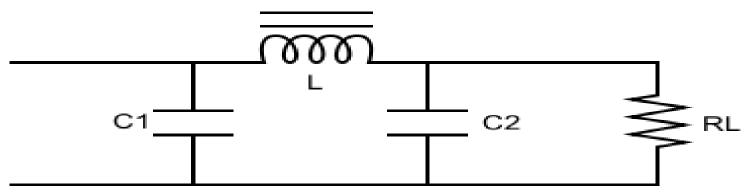


Fig.4.5: Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result, the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result, only the DC component appears across the load RL.

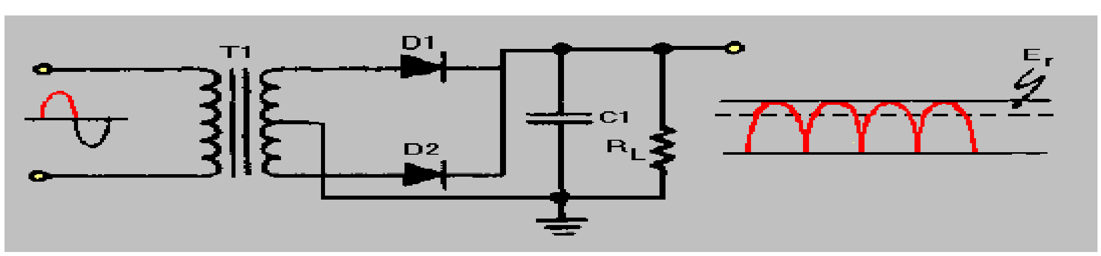


Fig.4.5(A): Centered Tapped Full-Wave Rectifier with a Capacitor Filter

4.6 VOLTAGE REGULATOR:

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. There are two types of regulator are they.

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:’78’ indicate the positive series and ‘xx’indicates the voltage rating. Suppose 7805 produces the maximum 5V.’05’indicates the regulator output is 5V.

79xx:’78’ indicate the negative series and ‘xx’indicates the voltage rating. Suppose 7905 produces the maximum -5V.’05’indicates the regulator output is -5V.

These regulators consist the three pins there are

Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

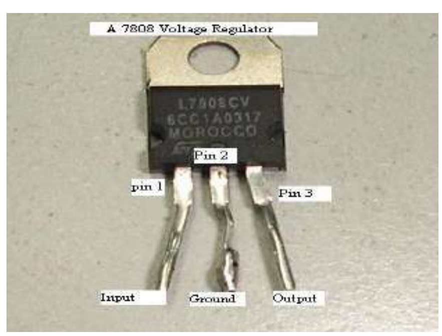


Fig.4.6: Regulator

CHAPTER-5

PROJECT DESCRIPTION

5.1 Hardware Specification:

- Battery
- LCD
- L293D
- Motors
- Arduino uno
- Solar panel
- Ultrasonic sensor
- IR Sensor

BLOCK DIAGRAM:

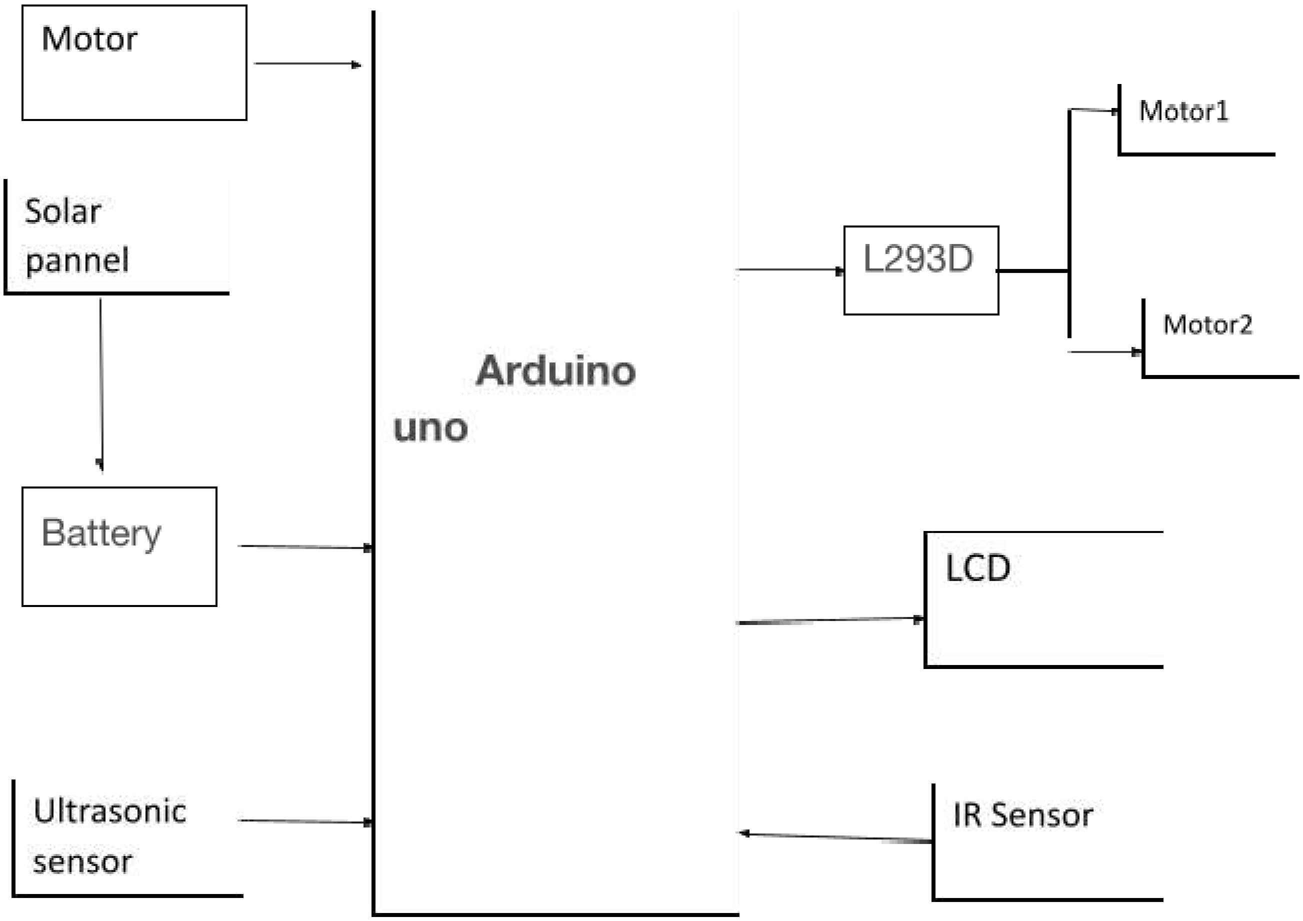


Fig.5.2: Block diagram of the project

Battery:

Batteries are a collection of one or more cells whose chemical reactions create a flow of electrons in a circuit. All batteries are made up of three basic components: an anode (the '-' side), a cathode (the '+' side), and some kind of electrolyte (a substance that chemically reacts with the anode and cathode).

When the anode and cathode of a battery is connected to a circuit, a chemical reaction takes place between the anode and the electrolyte. This reaction causes electrons to flow through the circuit and back into the cathode where another chemical reaction takes place. When the material in the cathode or anode is consumed or no longer able to be used in the reaction, the battery is unable to produce electricity. At that point, your battery is "dead."

Batteries that must be thrown away after use are known as **primary batteries**. Batteries that can be recharged are called **secondary batteries**.

Fig : Lithium Battery

Lithium polymer batteries, for example, can be recharged without batteries, your quadcopter would have to be tethered to the wall, you would have to hand crank your car, and your Xbox controller would have to plugged in all the time (like in the good old days). Batteries offer a way to store electrical potential energy in a portable container.

Fig : Batteries come in a variety of shapes, sizes, and chemistries.

History

The Term Battery

Historically, the word "battery" was used to describe a "series of similar objects grouped together to perform a function," as in a battery of artillery. In 1749, Benjamin Franklin first used the term to describe a series of capacitors he had linked together for his electricity experiments. Later, the term would be used for any electrochemical cells linked together for the purpose of providing electric power.

Battery of Leyden Jar "capacitors" linked together (Image courtesy of Alvinrune of Wikimedia Commons)

Invention of the Battery

One fateful day in 1780, Italian physicist, physician, biologist, and philosopher, Luigi Galvani, was dissecting a frog attached to a brass hook. As he touched the frog's leg with an iron scapel, the leg twitched. Galvani theorized that the energy came from the leg itself, but his fellow scientist, Alessandro Volta, believed otherwise.

Volta hypothesized that the frog's leg impulses were actually caused by different metals soaked in a liquid. He repeated the experiment using cloth soaked in brine instead of a frog corpse, which resulted in a similar voltage. Volta published his findings in 1791 and later created the first battery, the voltaic pile, in 1800.

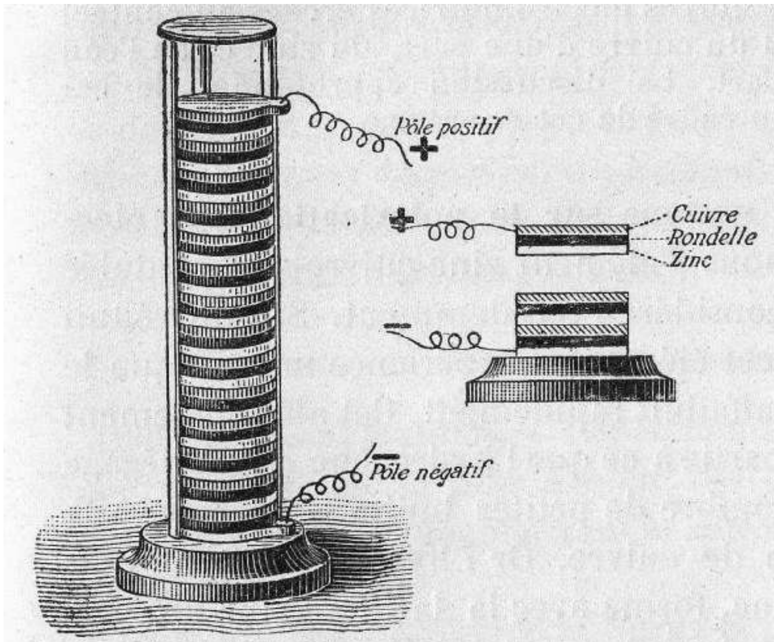


Fig. 283. — Pile de Volta.

The voltaic pile consisted of a stack of zinc and copper plates separated by cloth soaked in brine. Volta's pile was plagued by two major issues: the weight of the stack caused the electrolyte to leak out of the cloth, and the particular chemical properties of the components resulted in a very short life span (about an hour). The next two hundred years would be spent perfecting Volta's design and solving these issues.

Fixes to the Voltaic Pile

William Cruickshank of Scotland solved the leakage problem by laying the voltaic pile on its side to form the "trough battery."

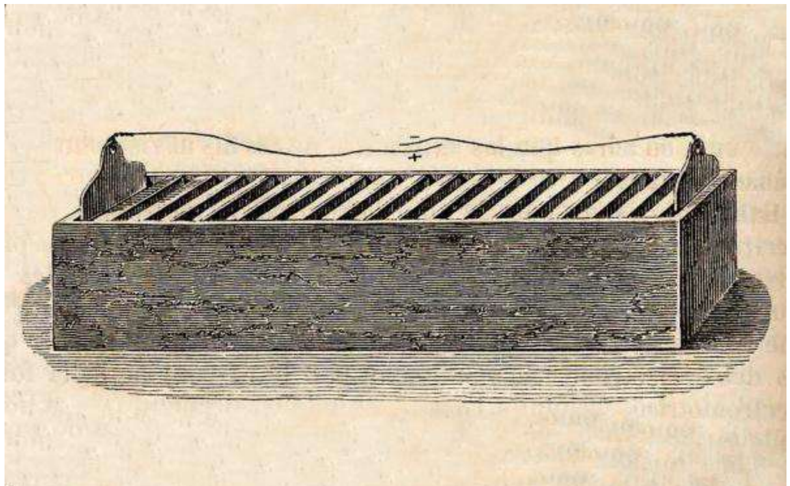


Fig : The trough battery solved the leakage problem of the voltaic pile

The second problem, short life span, was caused by the degradation of the zinc due to impurities and a build-up of hydrogen bubbles on the copper. In 1835, William Sturgeon discovered that treating the zinc with mercury would prevent degradation.

The British chemist John Frederic Daniell used a second electrolyte that reacted with the hydrogen, preventing buildup on the copper cathode. Daniell's two-electrolyte battery, known as the "Daniell cell," would become a very popular solution to providing power to the budding telegraph networks.

Fig : A collection of Daniell cells from 1836

The First Rechargeable Battery

In 1859, the French physicist Gaston Planté created a battery using two rolled sheets of lead submerged in sulfuric acid. By reversing the electrical current through the battery, the chemistry would return to its original state, thus creating the first rechargeable battery.

Later, in 1881, Camille Alphonse Faure improved Planté's design by forming the lead sheets into plates. This new design made the batteries easier to manufacture, and the lead acid battery saw wide-spread use in automobiles.

-> The design for the common "car battery" has been around for more than 100 years
(Image courtesy of Emilian Robert Vicol of Wikimedia Commons) <-

The Dry Cell

Up until the late 1800s, the electrolyte in batteries was in a liquid state. This made battery transportation a very careful endeavor, and most batteries were never intended to be moved once attached to the circuit.

In 1866, Georges Leclanché created a battery using a zinc anode, a manganese dioxide cathode, and an ammonium chloride solution for the electrolyte. While the electrolyte in the Leclanché cell was still a liquid, the battery's chemistry proved to be an important step for the invention of the dry cell.

Carl Gassner figured out how to create an electrolyte paste out of ammonium chloride and Plaster of Paris. He patented the new "dry cell" battery in 1886 in Germany.

These new dry cells, commonly called "zinc-carbon batteries," were massed produced and proved hugely popular until the late 1950s. While carbon is not used in the chemical reaction, it performs an important role as an electrical conductor in the zinc-carbon battery.

-> 3V zinc-carbon battery from the 1960s (Image courtesy of PhFabre of Wikimedia Commons) <- In the 1950s, Lewis Urry, Paul Marsal, and Karl Kordesch of the Union Carbide company (later known as "Eveready" and then "Energizer") replaced the ammonium chloride electrolyte with an alkaline substance, based on the battery chemistry formulated by Waldemar Jungner in 1899. Alkaline dry cell batteries could hold more energy than zinc carbon batteries of the same size and had a longer shelf life.

Alkaline batteries rose in popularity in the 1960s, overtook zinc-carbon batteries, and have since become the standard primary cell for consumer use.

-> Alkaline batteries come in many shapes and sizes (Image courtesy of Aney~commonswiki of Wikimedia Commons) <-

20th Century Rechargeable Batteries

In the 1970s, COMSAT developed the nickel-hydrogen battery for use in communication satellites. These batteries store hydrogen in a pressurized, gaseous form. Many man-made satellites, like the International Space Station, still rely on nickel-hydrogen batteries.

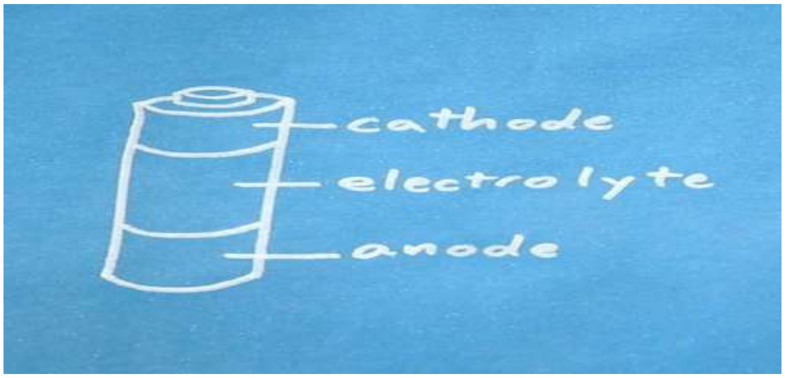
The research of several companies since the late 1960s resulted in the creation of the nickel-metal hydride (NiMH) battery. NiMH batteries were released to the consumer market in 1989, and provided a smaller, cheaper alternative to the rechargeable nickel-hydrogen cells.

Asahi Chemical of Japan built the first lithium-ion battery in 1985, and Sony created the first commercial lithium-ion battery in 1991. In the late 1990s, a soft, flexible casing was created for lithium-ion batteries and gave rise to the "lithium polymer" or "LiPo" battery.

The chemical reactions in the lithium polymer battery are essentially the same as those in the lithium-ion battery. Obviously, many more battery chemistries have been invented, manufactured, and become obsolete. If you would like to read more about modern, popular battery technologies, check out our Battery Technologies tutorial.

Components

Batteries are made up of three basic components: an **anode**, a **cathode**, and an **electrolyte**. A **separator** is often used to prevent the anode and cathode from touching, if the electrolyte is not sufficient. In order to store these components, batteries usually have some kind of **casing**.



OK, most batteries are not actually divided up in three equal sections, but you get the idea. A better cross-section of an alkaline cell can be found on Wikipedia.

Both the anode and cathode are types of **electrodes**. Electrodes are conductors through which electricity enters or leaves a component in a circuit.

Anode

Electrons flow out from the anode in a device connected to a circuit. This means that conventional "current" flows into an anode.

Fig: On batteries, the anode is marked as the negative (-) terminal

In a battery, the chemical reaction between the anode and electrolyte causes a build up of electrons in the anode. These electrons want to move to the cathode, but cannot pass through the electrolyte or separator.

Cathode

Electrons flow into the cathode in a device connected to a circuit. This means that conventional "current" flows out from a cathode.

Fig: On batteries, the cathode is marked as the positive (+) terminal

In batteries, the chemical reaction in or around the cathode uses the electrons produced in the anode. The only way for the electrons to get to the cathode is through a circuit, external to the battery.

Electrolyte

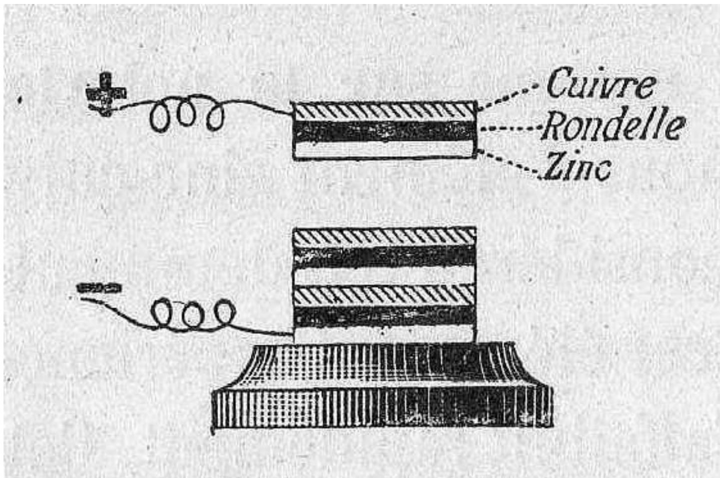
The electrolyte is the substance, often a liquid or gel, that is capable of transporting ions between the chemical reactions that happen at the anode and cathode. The electrolyte also inhibits the flow of electrons between the anode and cathode so that the electrons more easily flow through the external circuit rather than through the electrolyte.

-> Alkaline batteries can leak their electrolyte, potassium hydroxide, if subjected to high heat or reverse voltage (Image courtesy of Wiliam Davies of Wikimedia Commons)

<- The electrolyte is crucial in the operation of a battery. Because electrons cannot pass through it, they are forced to travel through electrical conductors in the form of a circuit that connect the anode to the cathode.

Separator

Separators are porous materials that prevent the anode and cathode from touching, which would cause a short circuit in the battery. Separators can be made from a variety of materials, including cotton, nylon, polyester, cardboard, and synthetic polymer films. Separators do not chemically react with either the anode, cathode, or electrolyte.



The voltaic pile used cloth or cardboard (separator) soaked in brine (electrolyte) to keep the electrodes apart. Ions in the electrolyte can be positively charged, negatively charged, and can come in a variety of sizes. Special separators can be manufactured that allow some ions to pass but not others.

Casing

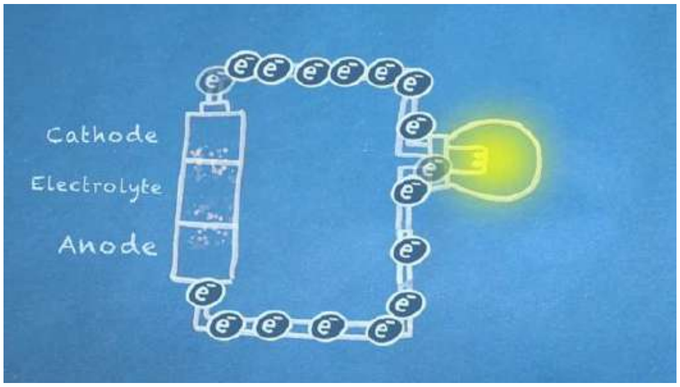
Most batteries need a way to contain their chemical components. Casings, otherwise known as "housings" or "shells," are simply mechanical structures meant to hold the battery's internals.

Fig :This lead-acid battery has a plastic casing

Battery casings can be made of almost anything: plastic, steel, soft polymer laminate pouches, and so on. Some batteries use a conducting steel casing that is electrically connected to one of the electrodes. In the case of the common AA alkaline cell, the steel casing is connected to the cathode.

Operation

Batteries generally require several chemical reactions in order to operate. At least one reaction occurs in or around the anode and one or more reactions occur in or around the cathode. In all cases, the reaction at the anode produces extra electrons in a process called **oxidation**, and the reaction at the cathode uses the extra electrons during a process known as **reduction**.



When the switch is closed, the circuit is complete, and electrons can flow from the anode to the cathode. These electrons enable the chemical reations at the anode and cathode.

In essence, we are separating a certain kind of chemical reaction, a reduction-oxidation reaction or redox reaction, into two separate parts. Redox reactions occur when electrons are transferred between chemicals. We can harness the movement of electrons in this reaction to flow outside the battery to power our circuit.

Anode Oxidation

This first part of the redox reaction, oxidation, occurs between the anode and electrolyte, and it produces electrons (marked as e^-).

Some oxidation reactions produce ions, such as in a lithium-ion battery. In other chemistries, the reaction consumes ions, like in the common alkaline battery. In either case, ions are able to flow freely through the electrolyte where electrons cannot.

Cathode Reduction

The other half of the redox reaction, reduction, occurs in or near the cathode. Electrons produced by the oxidation reaction are consumed during reduction.

In some cases, like lithium-ion batteries, positively charged lithium ions produced during the oxidation reaction are consumed during reduction. In other cases, like alkaline batteries, negatively charged ions are produced during reduction.

Electron Flow

In most batteries, some or all of the chemical reactions can occur even when the battery is not connected to a circuit. These reactions can impact a battery's shelf life.

For the most part, the reactions will only occur at full force when an electrically conductive circuit is completed between the anode and cathode. The less resistance between the anode and cathode, the more electrons are allowed to flow, and the quicker the chemical reactions occur.

Solar panel

The term **solar panel** is used colloquially for a photo-voltaic **(PV) module**.

A PV module is an assembly of photo-voltaic cells mounted in a frame work for installation. Photo-voltaic cells use sunlight as a source of energy and generate direct current electricity. A collection of PV modules is called a PV Panel, and a system of Panels is an Array. Arrays of a photovoltaic system supply solar electricity to electrical equipment.

The most common application of solar energy collection outside agriculture is solar water heating systems.[1]

Photovoltaic modules use light energy (photons) from the Sun to generate electricity through the photovoltaic effect. Most modules use wafer-based crystalline silicon cells or thin-film cells. The structural (load carrying) member of a module can be either the top layer or the back layer. Cells must be protected from mechanical damage and moisture. Most modules are rigid, but semi-flexible ones based on thin-film cells are also available. The cells are connected electrically in series, one to another to a desired voltage, and then in parallel to increase amperage. The wattage of the module is the mathematical product of the voltage and the amperage of the module.

A PV junction box is attached to the back of the solar panel and functions as its output interface. External connections for most photovoltaic modules use MC4 connectors to facilitate easy weatherproof connections to the rest of the system. A USB power interface can also be used.

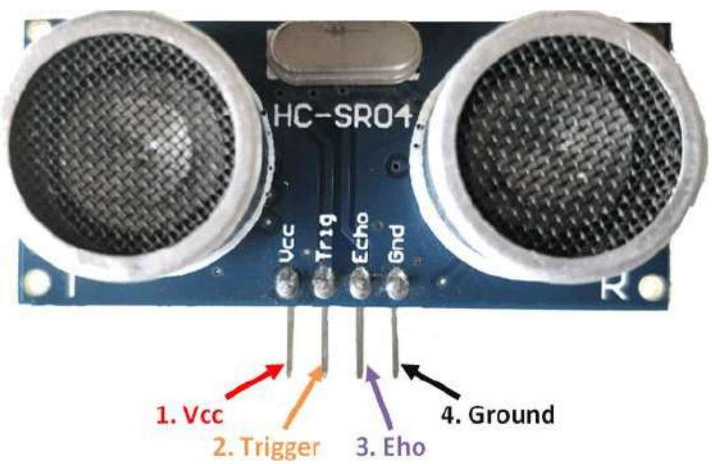
Module electrical connections are made in series to achieve a desired output voltage or in parallel to provide a desired current capability (amperes) of the solar panel or the PV system. The conducting wires that take the current off the modules are sized according to the ampacity and may contain silver, copper or other non-magnetic conductive transition metals. Bypass diodes may be incorporated or used externally, in case of partial module shading, to maximize the output of module sections still illuminated.

Some special solar PV modules include concentrators in which light is focused by lenses or mirrors onto smaller cells. This enables the use of cells with a high cost per unit area (such as gallium arsenide) in a cost-effective way.

Solar panels also use metal frames consisting of racking components, brackets, reflector shapes, and troughs to better support the panel structure.[2]

Ultrasonic sensor

HC-SR04 Ultrasonic Sensor



Ultrasonic Sensor HC SR04

Ultrasonic Sensor Pin Configuration

Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz

HC-SR04 Ultrasonic Sensor - Working

As shown above the **HC-SR04 Ultrasonic (US) sensor** is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

How to use the HC-SR04 Ultrasonic Sensor

HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used.

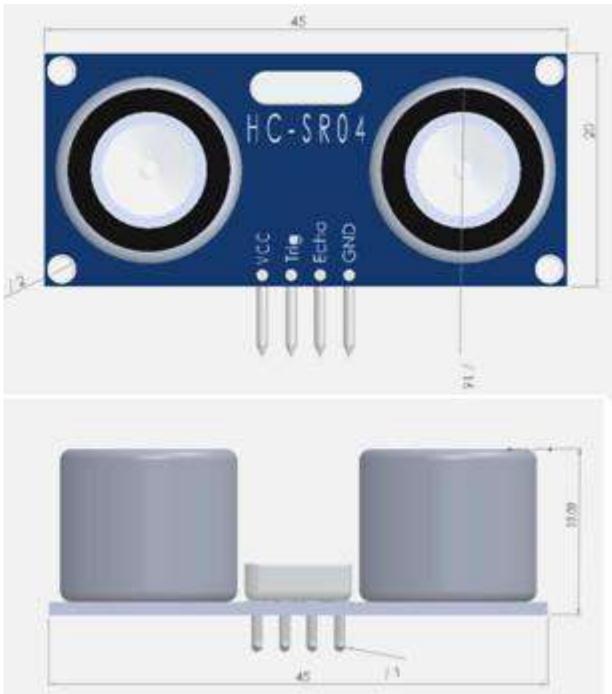
Power the Sensor using a regulated +5V through the Vcc ad Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor.

The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information the distance is measured as explained in the above heading.

Applications

- Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
- Used to measure the distance within a wide range of 2cm to 400cm
- Can be used to map the objects surrounding the sensor by rotating it
- Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water

2D model of the component



Motor

Motor is a device that creates motion, not an engine; it usually refers to either an electrical motor or an internal combustion engine.

It may also refer to:

- Electric motor, a machine that converts electricity into a mechanical motion
    - AC motor, an electric motor that is driven by alternating current
        - Synchronous motor, an alternating current motor distinguished by a rotor spinning with coils passing magnets at the same rate as the alternating current and resulting magnetic field which drives it

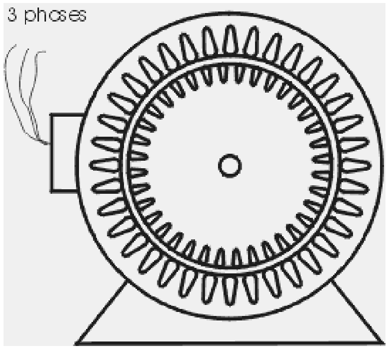
    - Induction motor, also called a squirrel-cage motor, a type of asynchronous alternating current motor where power is supplied to the rotating device by means of electromagnetic induction
- DC motor, an electric motor that runs on direct current electricity
    - Brushed DC electric motor, an internally commutated electric motor designed to be run from a direct current power source
    - Brushless DC motor, a synchronous electric motor which is powered by direct current electricity and has an electronically controlled commutation system, instead of a mechanical commutation system based on brushes
- Electrostatic motor, a type of electric motor based on the attraction and repulsion of electric charge
- Servo motor, an electric motor that operates a servo, commonly used in robotics
- Internal fan-cooled electric motor, an electric motor that is self-cooled by a fan, typically used for motors with a high energy density

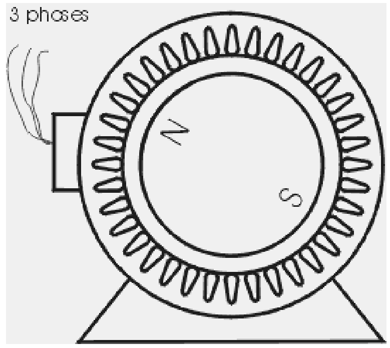
TYPES OF MOTORS

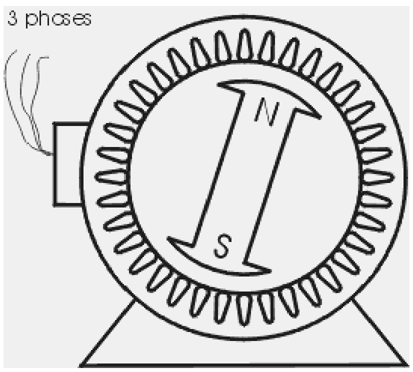
Industrial motors come in a variety of basic types. These variations are suitable for many different applications. Naturally, some types of motors are more suited for certain applications than other motor types are. This document will hopefully give some guidance in selecting these motors.

AC Motors

The most common and simple industrial motor is the three phase AC induction motor, sometimes known as the "squirrel cage" motor. Substantial information can be found about any motor by checking its (nameplate).







Advantages

- Simple Design
- Low Cost
- Reliable Operation
- Easily Found Replacements
- Variety of Mounting Styles
- Many Different Environmental Enclosures

Simple Design

The simple design of the AC motor -- simply a series of three windings in the exterior (stator) section with a simple rotating section (rotor). The changing field caused by the 50 or 60 Hertz AC line voltage causes the rotor to rotate around the axis of the motor.

The speed of the AC motor depends only on three variables:

1. The fixed number of winding sets (known as poles) built into the motor, which determines the motor's base speed.
2. The frequency of the AC line voltage. Variable speed drives change this frequency to change the speed of the motor.
3. The amount of torque loading on the motor, which causes slip.

Low Cost

The AC motor has the advantage of being the lowest cost motor for applications requiring more than about 1/2 hp (325 watts) of power. This is due to the simple design of the motor. For this reason, AC motors are overwhelmingly preferred for fixed speed applications in industrial applications and for commercial and domestic applications where AC line power can be easily attached. Over 90% of all motors are AC induction motors. They are found in air conditioners, washers, dryers, industrial machinery, fans, blowers, vacuum cleaners, and many, many other applications.

Reliable Operation

The simple design of the AC motor results in extremely reliable, low maintenance operation. Unlike the DC motor, there are no brushes to replace. If run in the appropriate environment for its enclosure, the AC motor can expect to need new bearings after several years of operation. If the application is well designed, an AC motor may not need new bearings for more than a decade.

Easily Found Replacements

The wide use of the AC motor has resulted in easily found replacements. Many manufacturers adhere to either European (metric) or American (NEMA) standards. (For Replacement Motors)

Variety of Mounting Styles

AC Motors are available in many different mounting styles such as:

- Foot Mount

- C-Face
- Large Flange
- Vertical
- Specialty

DC Motors

The brushed DC motor is one of the earliest motor designs. Today, it is the motor of choice in the majority of variable speed and torque control applications.

Advantages

- Easy to understand design
- Easy to control speed
- Easy to control torque
- Simple, cheap drive design

Easy to understand design

The design of the brushed DC motor is quite simple. A permanent magnetic field is created in the stator by either of two means:

- Permanent magnets
- Electro-magnetic windings

If the field is created by permanent magnets, the motor is said to be a "permanent magnet DC motor" (PMDC). If created by electromagnetic windings, the motor is often said to be a "shunt wound DC motor" (SWDC). Today, because of cost-effectiveness and reliability, the PMDC motor is the motor of choice for applications involving fractional horsepower DC motors, as well as most applications up to about three horsepower.

At five horsepower and greater, various forms of the shunt wound DC motor are most commonly used. This is because the electromagnetic windings are more cost effective than permanent magnets in this power range.

Caution: If a DC motor suffers a loss of field (if for example, the field power connections are broken), the DC motor will immediately begin to accelerate to the top speed which the loading will allow. This can result in the motor flying apart if the motor is lightly loaded. The possible loss of field must be accounted for, particularly with shunt wound DC motors.

Opposing the stator field is the armature field, which is generated by a changing electromagnetic flux coming from windings located on the rotor. The magnetic poles of the armature field will attempt to line up with the opposite magnetic poles generated by the stator field. If we stopped the design at this point, the motor would spin until the poles were opposite one another, settle into place, and then stop -- which would make a pretty useless motor!

However, we are smarter than that. The section of the rotor where the electricity enters the rotor windings is called the commutator. The electricity is carried between the rotor and the stator by conductive graphite-copper brushes (mounted on the rotor) which contact rings on stator. Imagine power is supplied:

The motor rotates toward the pole alignment point. Just as the motor would get to this point, the brushes jump across a gap in the stator rings. Momentum carries the motor forward over this gap. When the brushes get to the other side of the gap, they contact the stator rings again and -- the polarity of the voltage is reversed in this set of rings! The motor begins accelerating again, this time trying to get to the opposite set of poles. (The momentum has carried the motor past the original pole alignment point.) This continues as the motor rotates.

In most DC motors, several sets of windings or permanent magnets are present to smooth out the motion.

Easy to control speed

Controlling the speed of a brushed DC motor is simple. The higher the armature voltage, the faster the rotation. This relationship is linear to the motor's maximum speed.

The maximum armature voltage which corresponds to a motor's rated speed (these motors are usually given a rated speed and a maximum speed, such as 1750/2000 rpm) are available in certain standard voltages, which roughly increase in conjuntion with horsepower. Thus, the smallest industrial motors are rated 90 VDC and 180 VDC. Larger units are rated at 250 VDC and sometimes higher.

Specialty motors for use in mobile applications are rated 12, 24, or 48 VDC. Other tiny motors may be rated 5 VDC.

Most industrial DC motors will operate reliably over a speed range of about 20:1 -- down to about 5-7% of base speed. This is much better performance than the comparible AC motor. This is partly due to the simplicity of control, but is also partly due to the fact that most industrial DC motors are designed with variable speed operation in mind, and have added heat dissipation features which allow lower operating speeds.

Easy to control torque

In a brushed DC motor, torque control is also simple, since output torque is proportional to current. If you limit the current, you have just limited the torque which the motor can achieve. This makes this motor ideal for delicate applications such as textile manufacturing.

Simple, cheap drive design

The result of this design is that variable speed or variable torque electronics are easy to design and manufacture. Varying the speed of a brushed DC motor requires little more than a large enough potentiometer. In practice, these have been replaced for all but sub-fractional horsepower applications by the SCR and PWM drives, which offer relatively precisely control voltage and current. Common DC drives are available at the low end (up to 2 horsepower) for under US$100 -- and sometimes under US$50 if precision is not important.

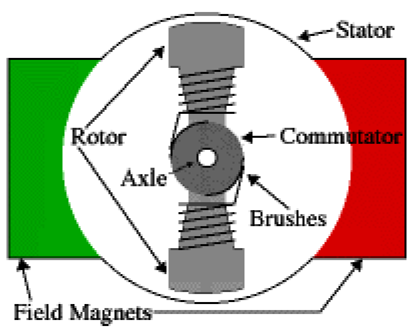
Large DC drives are available up to hundreds of horsepower. However, over about 10 horsepower careful consideration should be given to the price/performance tradeoffs with AC inverter systems, since the AC systems show a price advantage in the larger systems. (But they may not be capable of the application's performance requirments).

Disadvantages

- Expensive to produce
- Can't reliably control at lowest speeds
- Physically larger
- High maintenance
- Dust

WORKING OF DC MOTOR

In any electric motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. As you are well aware of from playing with magnets as a kid, opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion.



Principle

When a rectangular coil carrying current is placed in a magnetic field, a torque acts on the coil which rotates it continuously.

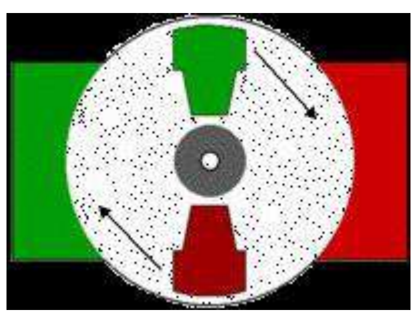
When the coil rotates, the shaft attached to it also rotates and thus it is able to do mechanical work.

Every DC motor has six basic parts -- axle, rotor (a.k.a., armature), stator, commutator, field magnet(s), and brushes. In most common DC motors (and all that BEAMers will see), the

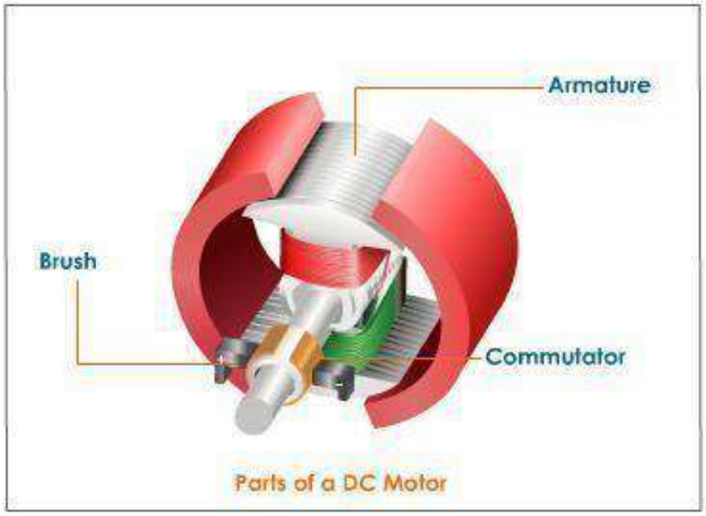
external magnetic field is produced by high-strength permanent magnets[1]. The stator is the stationary part of the motor -- this includes the motor casing, as well as two or more permanent magnet pole pieces. The rotor (together with the axle and attached commutator) rotate with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a common motor layout -- with the rotor inside the stator (field) magnets.

The geometry of the brushes, commentator contacts, and rotor windings are such that when power is applied, the polarities of the energized winding and the stator magnet(s) are misaligned, and the rotor will rotate until it is almost aligned with the stator's field magnets. As the rotor reaches alignment, the brushes move to the next commentator contacts, and energize the next winding. Given our example two-pole motor, the rotation reverses the direction of current through the rotor winding, leading to a "flip" of the rotor's magnetic field, driving it to continue rotating.

In real life, though, DC motors will always have more than two poles (three is a very common number). In particular, this avoids "dead spots" in the commutator. You can imagine how with our example two-pole motor, if the rotor is exactly at the middle of its rotation (perfectly aligned with the field magnets), it will get "stuck" there. Meanwhile, with a two-pole motor, there is a moment where the commutator shorts out the power supply (i.e., both brushes touch both commutator contacts simultaneously). This would be bad for the power supply, waste energy, and damage motor components as well. Yet another disadvantage of such a simple motor is that it would exhibit a high amount of torque "ripple" (the amount of torque it could produce is cyclic with the position of the rotor).



Construction and Working



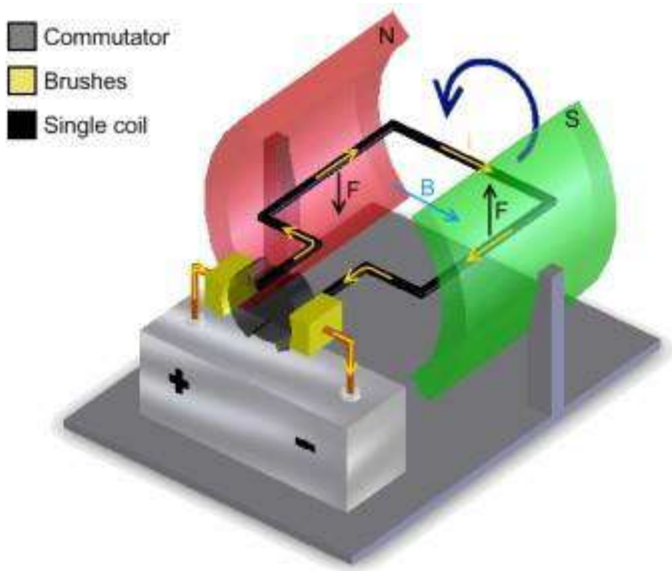
Parts of a DC Motor

Armature

A D.C. motor consists of a rectangular coil made of insulated copper wire wound on a soft iron core. This coil wound on the soft iron core forms the armature. The coil is mounted on an axle and is placed between the cylindrical concave poles of a magnet.

Commutator

A commutator is used to reverse the direction of flow of current. Commutator is a copper ring split into two parts C_1 and C_2. The split rings are insulated form each other and mounted on the axle of the motor. The two ends of the coil are soldered to these rings. They rotate along with the coil. Commutator rings are connected to a battery. The wires from the battery are not connected to the rings but to the brushes which are in contact with the rings.



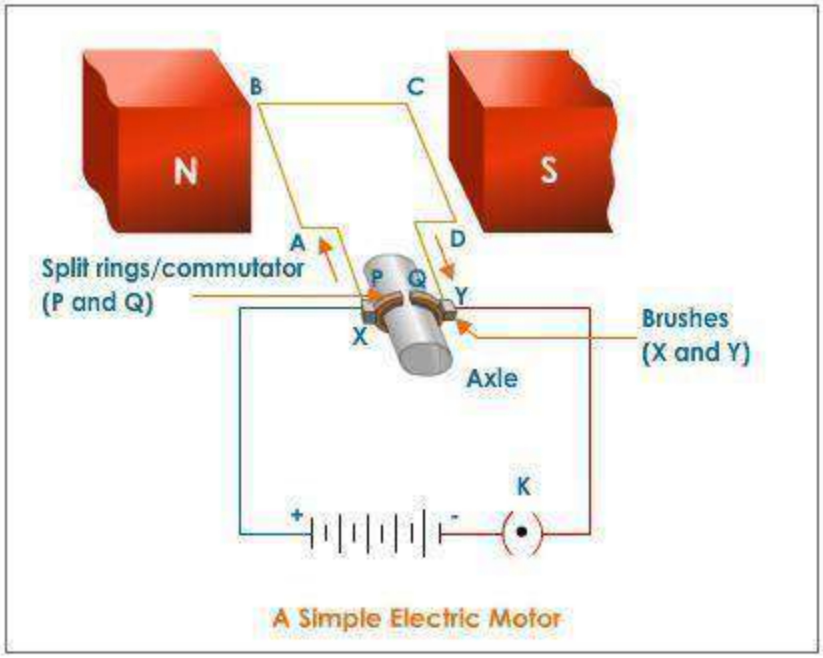
Brushes

Two small strips of carbon, known as brushes press slightly against the two split rings, and the split rings rotate between the brushes.

The carbon brushes are connected to a D.C. source.

Working of a DC Motor

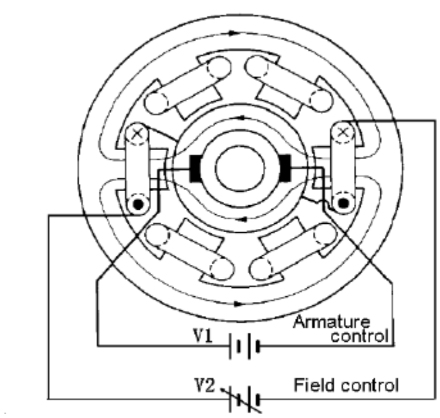
When the coil is powered, a magnetic field is generated around the armature. The left side of the armature is pushed away from the left magnet and drawn towards the right, causing rotation.

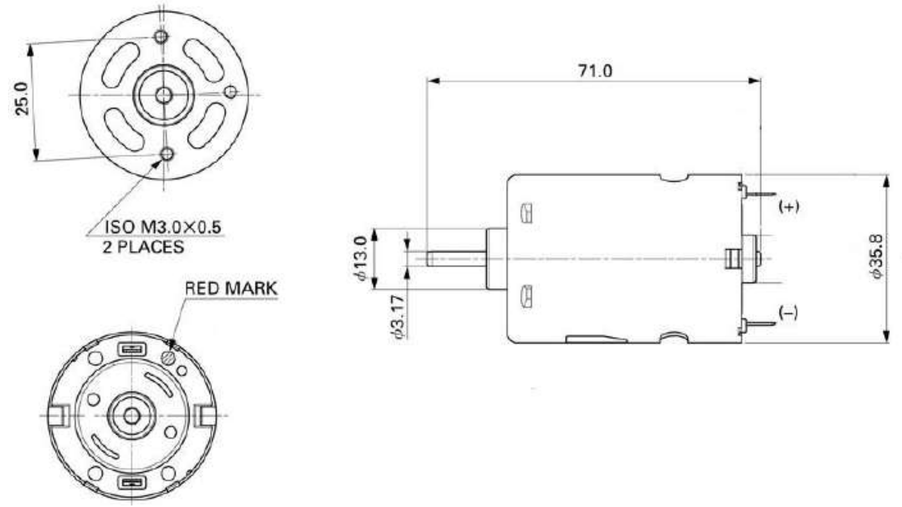


When the coil turns through 90^0, the brushes lose contact with the commutator and the current stops flowing through the coil.

However the coil keeps turning because of its own momentum.

Now when the coil turns through 180^0, the sides get interchanged. As a result the commutator ring C_1 is now in contact with brush B_2 and commutator ring C_2 is in contact with brush B_1. Therefore, the current continues to flow in the same direction.



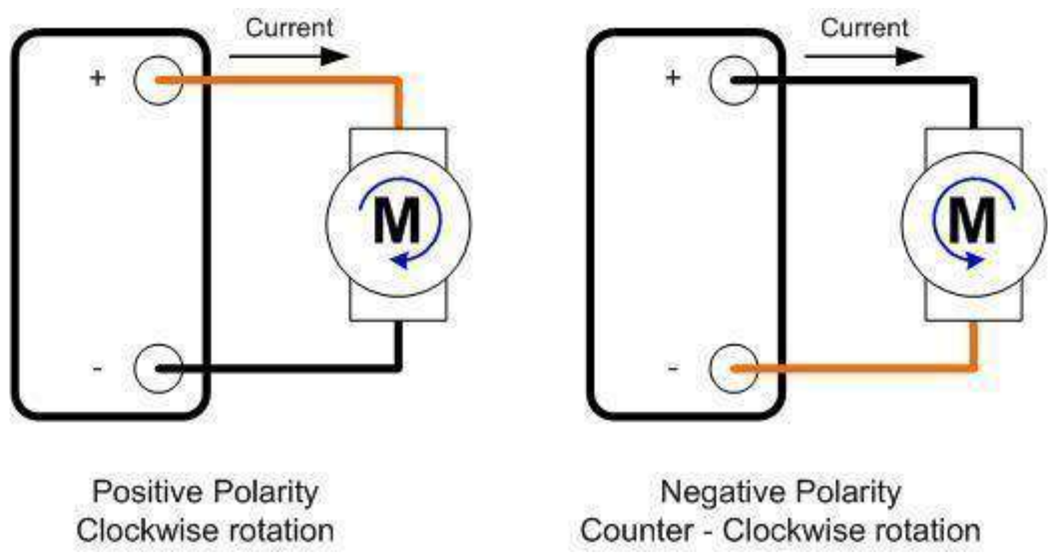


PARAMETERS OF THE DC MOTORS

1. Direction of rotation
2. Motor Speed
3. Motor Torque
4. Motor Start and Stop

Direction of Rotation

A DC Motor has two wires. We can call them the positive terminal and the negative terminal, although these are pretty much arbitrary names (unlike a battery where these polarities are vital and not to be mixed!). On a motor, we say that when the + wire is connected to + terminal on a power source, and the - wire is connected to the - terminal source on the same power source, the motor rotates clockwise (if you are looking towards the motor shaft). If you reverse the wire polarities so that each wire is connected to the opposing power supply terminal, then the motor rotates counter clockwise. Notice this is just an arbitrary selection and that some motor manufacturers could easily choose the opposing convention. As long as you know what rotation you get with one polarity, you can always connect in such a fashion that you get the direction that you want on a per polarity basis.



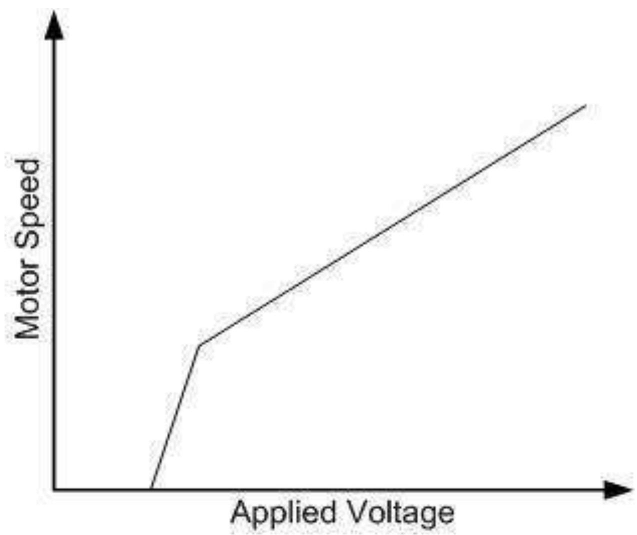
DC Motor Rotation vs Polarity

Facts:

- DC Motor rotation has nothing to do with the voltage magnitude or the current magnitude flowing through the motor.
- DC Motor rotation does have to do with the voltage polarity and the direction of the current flow.

DC Motor Speed

Whereas the voltage polarity controls DC motor rotation, voltage magnitude controls motor speed. Think of the voltage applied as a facilitator for the strengthening of the magnetic field. In other words, the higher the voltage, the quicker will the magnetic field
become strong. Remember that a DC motor has an electromagnet and a series of permanent magnets. The applied voltage generates a magnetic field on the electromagnet portion. This electromagnet field is made to oppose the permanent magnet field. If the electromagnet field is very strong, then both magnetic entities will try to repel each other from one side, as well as atract each other from the other side. The stronger the induced magnetic field, the quicker will this separation/attaction will try to take place. As a result, motor speed is directly proportional to applied voltage.



Motor Speed Curve

One aspect to have in mind is that the motor speed is not entirely lineal. Each motor will have their own voltage/speed curve. One thing I can guarantee from each motor is that at very low voltages, the motor will simply not move. This is because the magnetic field strength is not enough to overcome friction. Once friction is overcome, motor speed will start to increase as voltage increase.

The following video shows the concept of speed control and offers some ideas on how this can be achieved.

Motor Torque

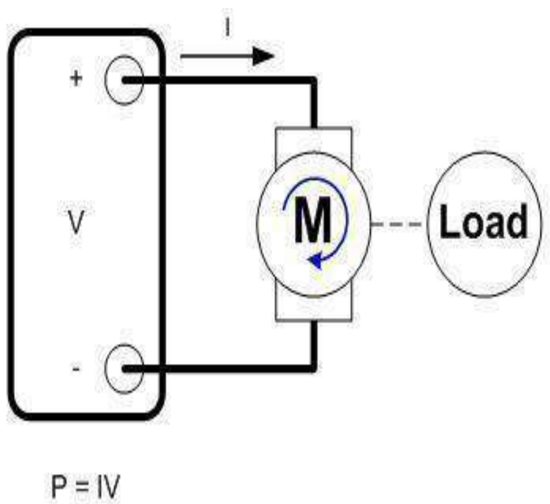
In the previous segment I kind of described speed as having to do with the strength of the magnetic field, but this is in reality misleading. Speed has to do with how fast the magnetic field is built and the attraction/repel forces are installed into the two magnetic structures. Motor strength, on the other hand, has to do with magnetic field strength. The stronger the electromagnet attracts the permanent magnet, the more force is exerted on the motor load.

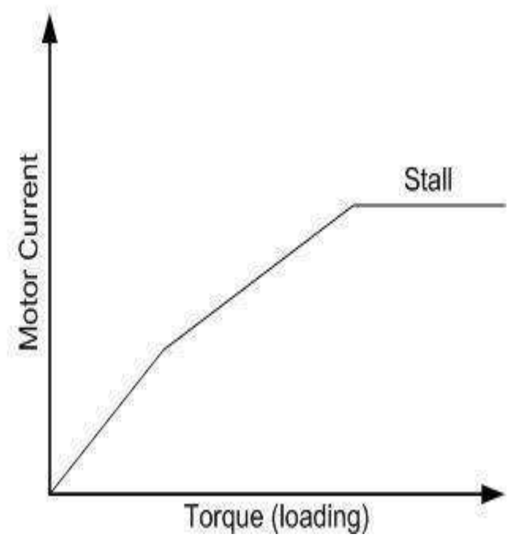
Per example, imagine a motor trying to lift 10 pounds of weight. This is a force that when multiplied by a distance (how much from the ground we are lifting the load) results in WORK. This WORK when exerted through a predetermined amount of time (for how long we are lifting the weight) gives us power. But whatever power came in, must come out as energy can not be created or destroyed. So that you know, the power that we are supplying to the motor is computed by

P = IV

Where P is power, I is motor current and V is motor voltage

Hence, if the voltage (motor speed) is maintained constant, how much load we are moving must come from the current. As you increase load (or torque requirements) current must also increase.





Motor Loading

One aspect about DC motors which we must not forget is that loading or increase of torque can not be infinite as there is a point in which the motor simply can not move. When this happens, we call this loading "Stalling Torque". At the same time this is the maximum amount of current the motor will see, and it is refer to Stalling Current. Stalling deserves a full chapter as this is a very important scenario that will define a great deal of the controller to be used. I promise I will later write a post on stalling and its intricacies.

Motor Start and Stop

You are already well versed on how to control the motor speed, the motor torque and the motor direction of rotation. But this is all fine and dandy as long as the motor is actually moving. How about starting it and stopping it? Are these trivial matters? Can we just ignore them or should we be careful about these aspects as well? You bet we should!

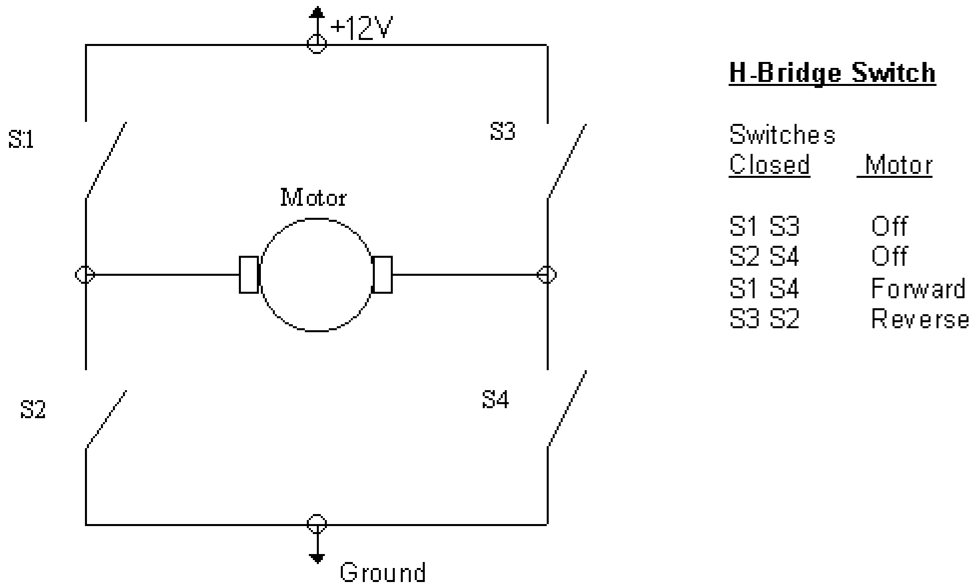
Starting a motor is a very hazardous moment for the system. Since you have an inductance whose energy storage capacity is basically empty, the motor will first act as an inductor. In a sense, it should not worry us too much because current can not change abruptly in an inductor, but the truth of the matter is that this is one of the instances in which you will see the highest currents flowing into the motor. The start is not necessarily bad for the motor itself as in fact the motor can easily take this Inrush Current. The power stage, on the other hand and if not properly designed for, may take a beating.

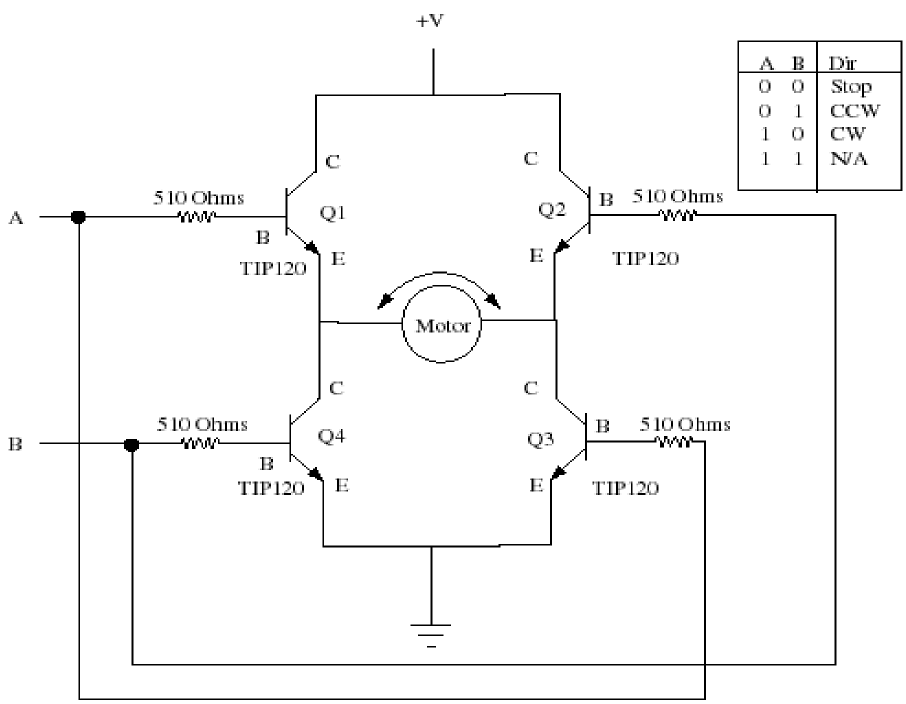
Once the motor has started, the motor current will go down from inrush levels to whatever load the motor is at. Per example, if the motor is moving a few gears, current will be proportional to that load and according to torque/current curves.

Stopping the motor is not as harsh as starting. In fact, stopping is pretty much a breeze. What we do need to concern ourselves is with how we want the motor to stop. Do we want it to coast down as energy is spent in the loop, or do we want the rotor to stop as fast as possible? If the later is the option, then we need braking. Braking is easily accomplished by shorting the motor outputs. The reason why the motor stops so fast is because as a short is applied to the motor terminals, the Back EMF is shorted. Because Back EMF is directly proportional to speed, making Back EMF = 0, also means making speed = 0.

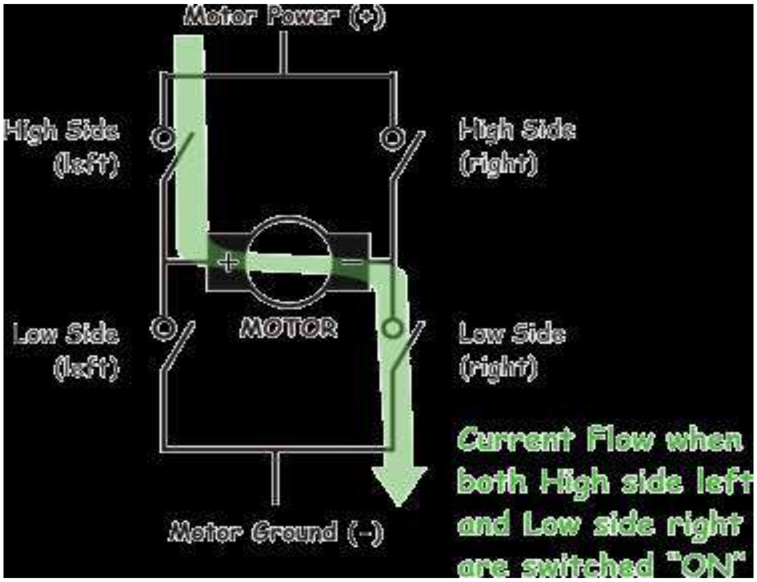
MOTORDRIVER CIRCUIT

The name "H-Bridge" is derived from the actual shape of the switching circuit which control the motion of the motor. It is also known as "Full Bridge". Basically there are four switching elements in the H-Bridge as shown in the figure below.





As you can see in the figure above there are four switching elements named as "High side left", "High side right", "Low side right", "Low side left". When these switches are turned on in pairs motor changes its direction accordingly. Like, if we switch on High side left and Low side right then motor rotate in forward direction, as current flows from Power supply through the motor coil goes to ground via switch low side right. This is shown in the figure below.



Similarly, when you switch on low side left and high side right, the current flows in opposite direction and motor rotates in backward direction. This is the basic working of H-Bridge. We can also make a small truth table according to the switching of H-Bridge explained above.

Truth Table

High Left	High Right	Low Left	Low Right	Description
On	Off	Off	On	Motor runs clockwise
Off	On	On	Off	Motor runs anti-clockwise
On	On	Off	Off	Motor stops or decelerates
Off	Off	On	On	Motor stops or decelerates

As already said, H-bridge can be made with the help of transistors as well as MOSFETs, the only thing is the power handling capacity of the circuit. If motors are needed to run with high current then lot of dissipation is there. So head sinks are needed to cool the circuit.

Now you might be thinking why i did not discuss the cases like High side left on and Low side left on or high side right on and low side right on. Clearly seen in the diagra, you don't want to burn your power supply by shorting them. So that is why those combinations are not discussed in the truth table.

L293D

L293D is basically a high current dual motor driver/controller Integrated Circuit (IC). It is able to drive load having current up to 1A at the voltage ranging from 4.5V to 36V. Motor driver usually act as current amplifier because they receive a low current signal as an input and provides high current signal at the output.

Motors usually operates on this higher current. L-293D has to builtin H-Bridge driver circuits and is able to control two DC motors at a time in both clockwise and counter clockwise direction. It has two enable pins and they should be kept high in order to control the motor. By changing the polarity of applied signal motor can be rotated in either clockwise or counter clockwise direction. If L 293D enable pin is high, its corresponding driver will provide the desired out. If the enable pin is low, there will be no output. L-293D has different features including internal ESD protection, large voltage supply range, large output current per channel, high noise immunity input etc. L 293D plays a vital role in electronics era and has several different applications e.g relay drivers, DC motor drivers, stepping motor drivers etc. The further detail about L 293D motor driver/controller will be given later in this tutorial.

Introduction to L293D

L293D is basically a motor driver or controller. It has two builtin H-bridge circuits which are able to control two DC motors simultaneously in both clockwise and counter clockwise direction. It acts as an current high amplifier because it take low current signal at its input and provides higher current signal at the output in order to drives different load e.g stepper motor & DC motors. Its features include large inpu voltage supply range, large output current high noise immunity input signals etc. Its common real life applications include stepping motor drivers, relay drivers, DC motor drivers etc. L-293D motor driver/controller is shown in the figure given below.

1. L293D Pins

- L-293D has sixteen (16) pins, having different individual functions.
- All of these sixteen pins along with their serial no, are given in the table shown below.

2. L293D Pin Functions

- L 293D each pin has different tasks to perform when they are in working condition.
- The tasks assigned to each pin are given in the table shown below.

L293D Pins	
Pin. No	Pin Name
1	Enable 1,2
2	Input 1
3	Output 1
4	GND
5	GND
6	Output 2
7	Input 2
8	Vcc2
9	Enable 3,4
10	Input 3
11	Output 3
12	GND
13	GND
14	Output put 4
15	Input 4
16	Vcc1

3. L293D Pinout

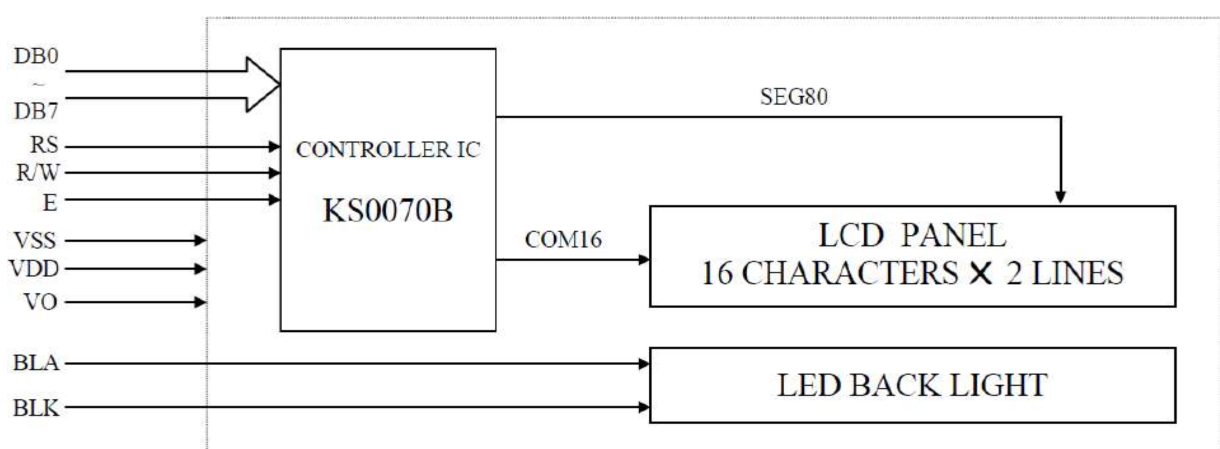
- Pinout diagram of any device presents the pins configuration through a completely labelled diagram.
- L293D pinout diagram is shown in the figure given below.

16 * 2 Alphanumeric LCD

Description

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. A liquid crystal display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Here the lcd has different memories to display data, those are discussed below.

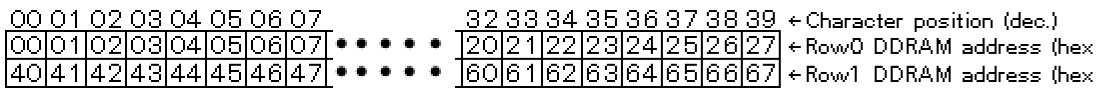
Block Diagram



Display Data RAM

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

Figure below will show you the DDRAM addresses of 2 Line LCD.



DDRAM Address for 2 Line LCD

Character Generator ROM

Now you might be thinking that when you send an ascii value to DDRAM, how the character is displayed on LCD? so the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. Userdefined character patterns are also available by mask-programmed ROM.

Busy Flag

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact ammount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

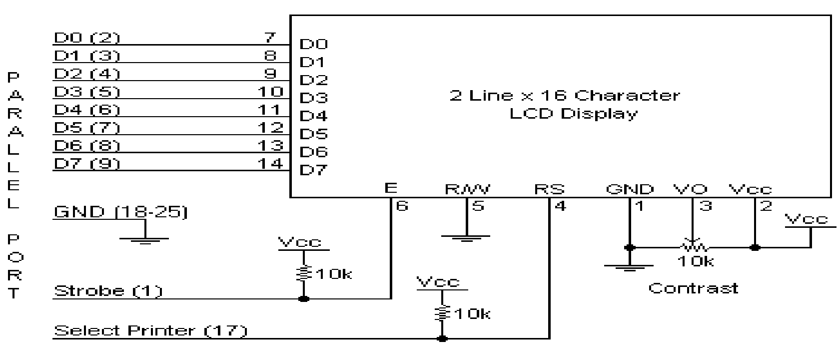
Instruction Register(IR) and Data Register(DR)

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD.
Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

16 x 2 Alphanumeric LCD Module Features

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers.

Schematic

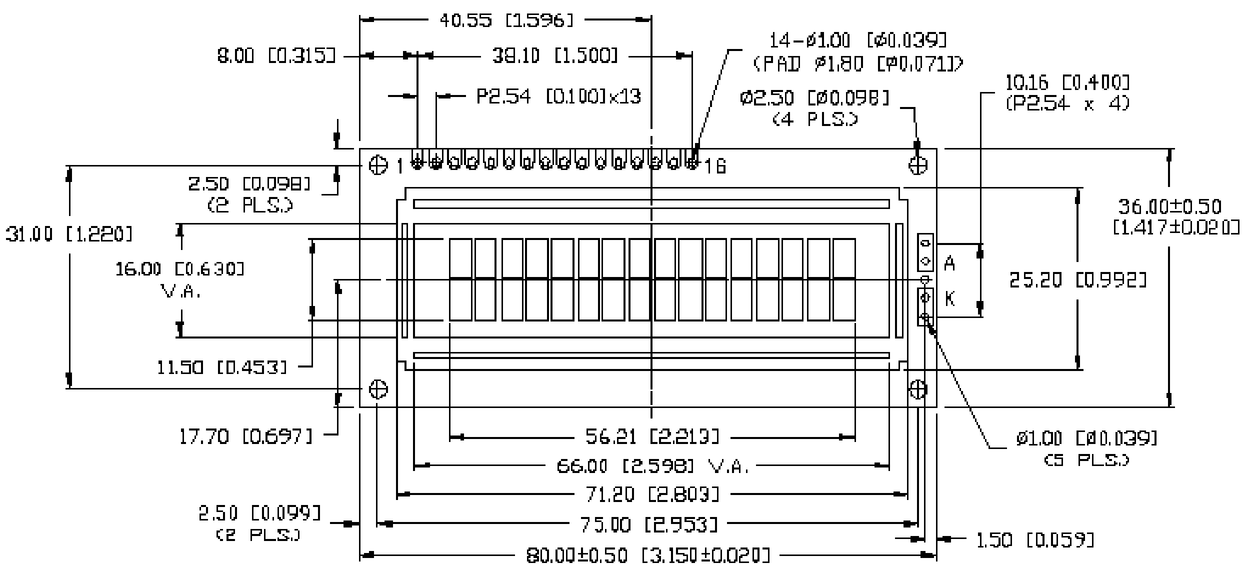


Specifications

Connector Pin Assignment:

Pin	*Symbol*	*Function*	*Pin*	*Symbol*	*Function*
14	DB7	Data Bus Line	6	E	Enable
13	DB6	Data Bus Line	5	R/W	Read/Write
12	DB5	Data Bus Line	4	RS	Register select input
11	DB4	Data Bus Line	3	Vcontrast	Contrast (0V for max contrast)
10	DB3	Data Bus Line	2	**GND (0V)**	Power supply
9	DB2	Data Bus Line	1	**Vcc +5V**	Power supply
8	DB1	Data Bus Line			
7	DB0	Data Bus Line			

Note1: Pin 1 is +5V and pin 2 is GND! This is different from most other HD44780 compatible models.
Note2: Pin 14 is marked with a “1” on the back of the PCB. Do not get confused by this.



Circuit Description

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the *R/W* line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

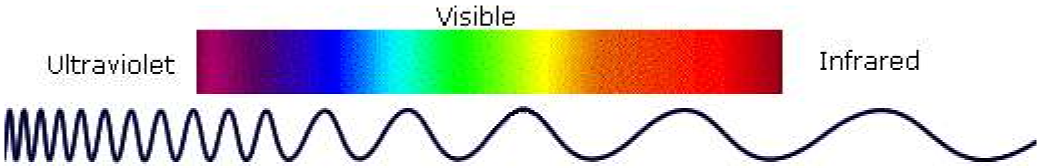
The 10k Potentiometer controls the contrast of the LCD panel. You can use a bench power supply set to 5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

INFRARED TECHNOLOGY (IR)

Technically known as "infrared radiation", infrared light is part of the electromagnetic spectrum located just below the red portion of normal visible light – the opposite end to ultraviolet. Although invisible, infrared follows the same principles as regular light and can be reflected or pass through transparent objects, such as glass. Infrared remote controls use this invisible light as a form of communications between themselves and home theater equipment, all of which have infrared receivers positioned on the front. Essentially, each time you press a button on a remote, a small infrared diode at the front of the remote beams out pulses of light at high speed to all of your equipment. When the equipment recognizes the signal as its own, it responds to the command.

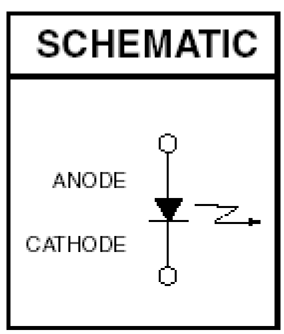
But much like a flashlight, infrared light can be focused or diffused, weak or strong. The type and number of emitters can affect the possible angles and range your remote control can be used from. Better remotes can be used up to thirty feet away and from almost any angle, while poorer remotes must be aimed carefully at the device being controlled.

The light our eyes see is but a small part of a broad spectrum of electromagnetic radiation. On the immediate high energy side of the visible spectrum lies the ultraviolet, and on the low energy side is the infrared. The portion of the infrared region most useful for analysis of organic compounds is not immediately adjacent to the visible spectrum, but is that having a wavelength range from 2,500 to 16,000 nm, with a corresponding frequency range from $1.9*10^{13}$ to $1.2*10^{14}$ Hz.(From http://hyperphysics.phy-astr.gsu.edu/hbase/ems3.html : the frequency of infrared ranges from 0.003 - 4 x 10^{14} Hz or about 300 gigahertz to 400 terahertz.).



Infrared imaging is used extensively for both military and civilian purposes. Military applications include target acquisition, surveillance, night vision, homing and tracking. Non-military uses include thermal efficiency analysis, remote temperature sensing, short-ranged wireless communication, spectroscopy, and weather forecasting. Infrared astronomy uses sensor-equipped telescopes to penetrate dusty regions of space, such as molecular clouds; detect cool objects such as planets, and to view highly red-shifted objects from the early days of the universe

IR LED QED234:



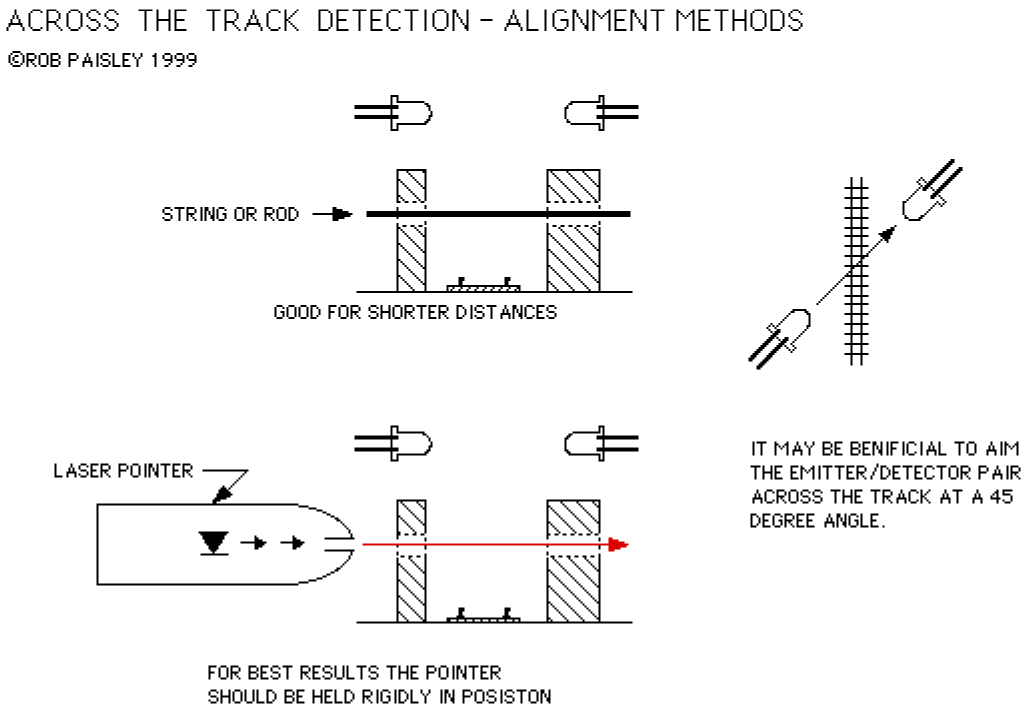
FEATURES:

- Wave length is 940 nm
- Chip material =GaAs with AlGaAs window
- Package type: T-1 3/4 (5mm lens diameter)
- Matched Photo sensor: QSD122/123/124
- Medium Emission Angle, 40°
- High Output Power
- Package material and color: Clear, untainted, plastic
- Ideal for remote control applications

Emitter/Detector Alignment:

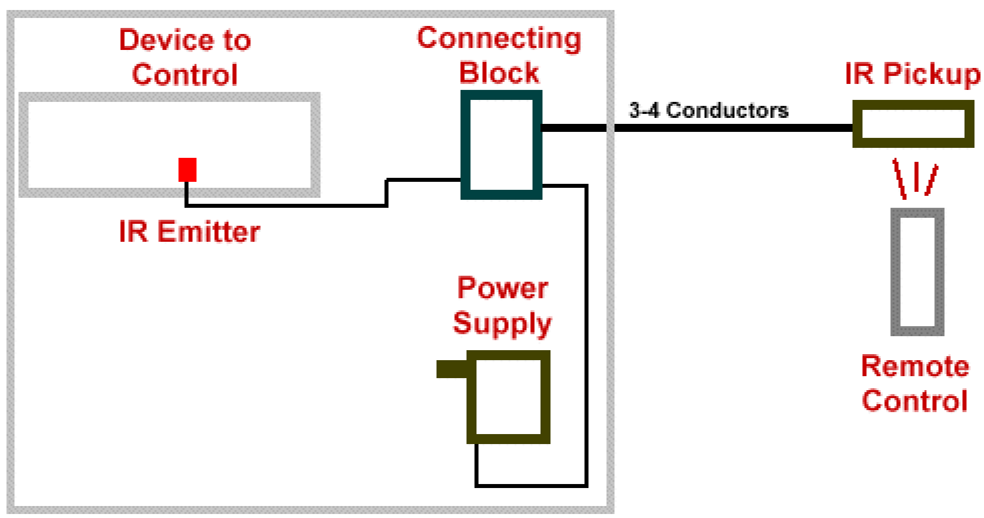
Good alignment of the emitter and detector is important for good operation, especially if the gap is large. This can be done with a piece of string stretched between and in line with LED and phototransistor. A length of dowel or stiff wire could be used to set the alignment. Another method that can be used for longer distances is a laser pointer shone through one of the mounting holes.

For best results the height of the "beam" should be at coupler height and at an angle across the tracks. The emitter could also be mounted above the track with the phototransistor placed between the rails in locations such as hidden yards. Placing the emitter and detector at an angle would again be helpful.



Emitter/Detector Alignment Methods

<u>A sample infrared remote control setup</u>:



A Simple Infrared Remote Control Setup

Infrared Receiver (Pickup)

This device picks up the infrared signal from your remote control just like a TV or VCR. It encodes the infrared signal into a signal suitable for transmission. Receivers must be located in the room you wish to use the remote control. The wire from the receiver to the connecting block needs at least three available conductors and can be several hundred feet long. Both quad wire and category 5wire work fine. See our IR receivers here.

Connecting Block

This is simply a place for all the parts to plug in or connect to. Connecting blocks are usually classified based on the number of outputs (how many IR emitters can connect to the block) Amplified connecting blocks can generally support more outputs. All connecting blocks can support many IR receivers wired in parallel. Connecting blocks are usually located near the equipment that is to be controlled, along with the power supply and emitters. See our connecting blocks here.

Infrared Emitters

IR Emitters generally "stick" onto the front of the device you want to control. Therefore you need one emitter for each device. "Dual" emitters have two emitters and one plug, so they only take up one jack of the connecting block. "Blink" emitters blink visibly as well as infrared, so they are easier to troubleshoot. All emitters come with long cords and extra double-stick tape. "Blast" style emitters, where one emitter blinks into several devices, are usually less reliable but can be used when the environment is tightly controlled and

Applications:

- Infrared Filters
- Night vision
- Thermograph
- Other imaging
- Tracking
- Heating
- Communications
- Spectroscopy
- Meteorology
- Climatology
- Astronomy
- Art history
- Biological systems
- Photobiomodulation
- Health hazard

CHAPTER -6

SOFTWARE EXPLANATION

6.1 SOFTWARE EXPLANATION:

Software Requirements

- Proteus simulation
- Arduino software
- Programming language

6.1.1 Arduino software:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals

- has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that

environment will be familiar with how the Arduino IDE works.

- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers c an make their own version
- Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money
- Getting Started with Arduino and Genuino products:-

Install the Arduino Software (IDE) on Windows PCs-

…This document explains how to install the Arduino Software (IDE) on Windows machines.

✓ Download the Arduino Software(IDE)

✓ Proceed with board specific instructions.

How to Download the Arduino Software (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

Installation:

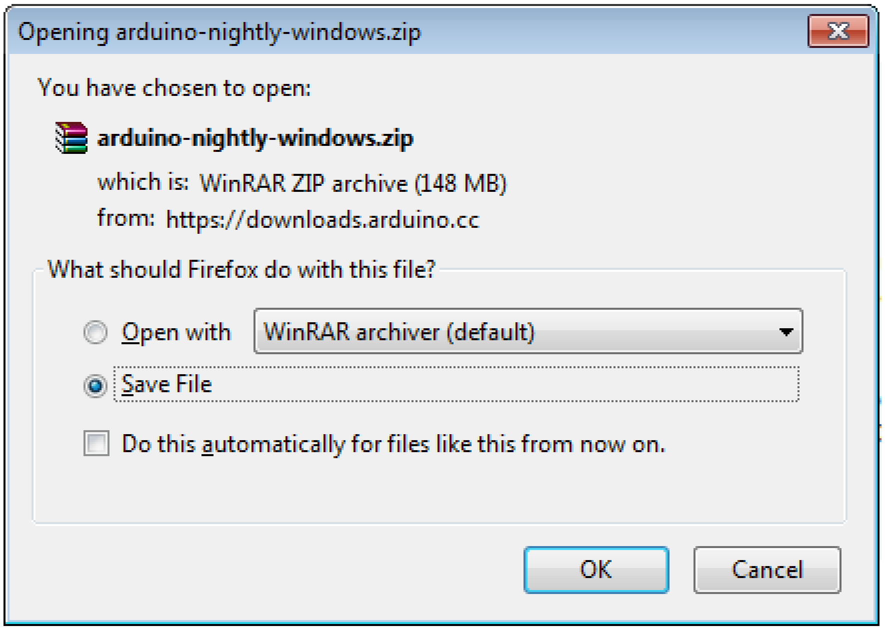
In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



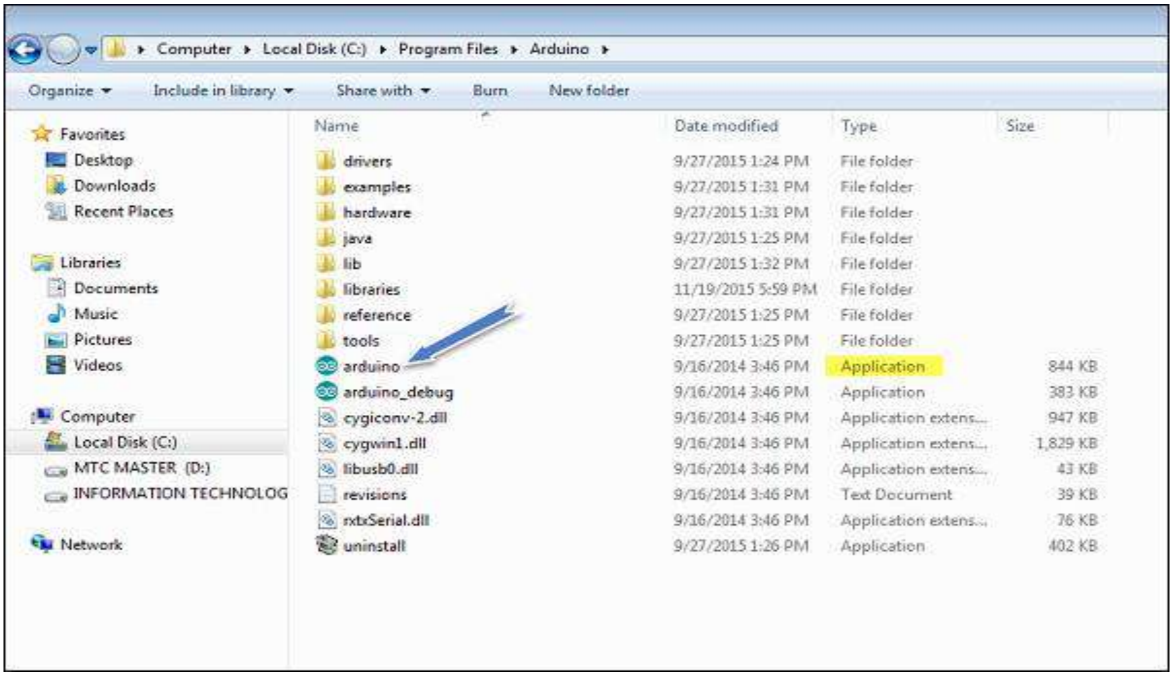
Step 3 – Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 – Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

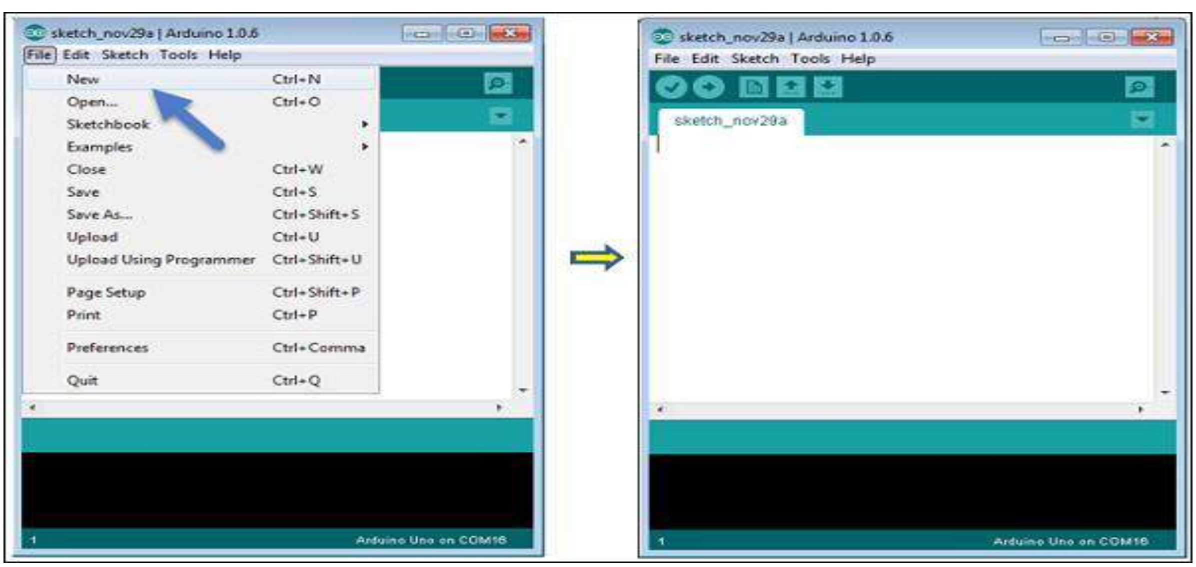


Step 5 − Open your first project.

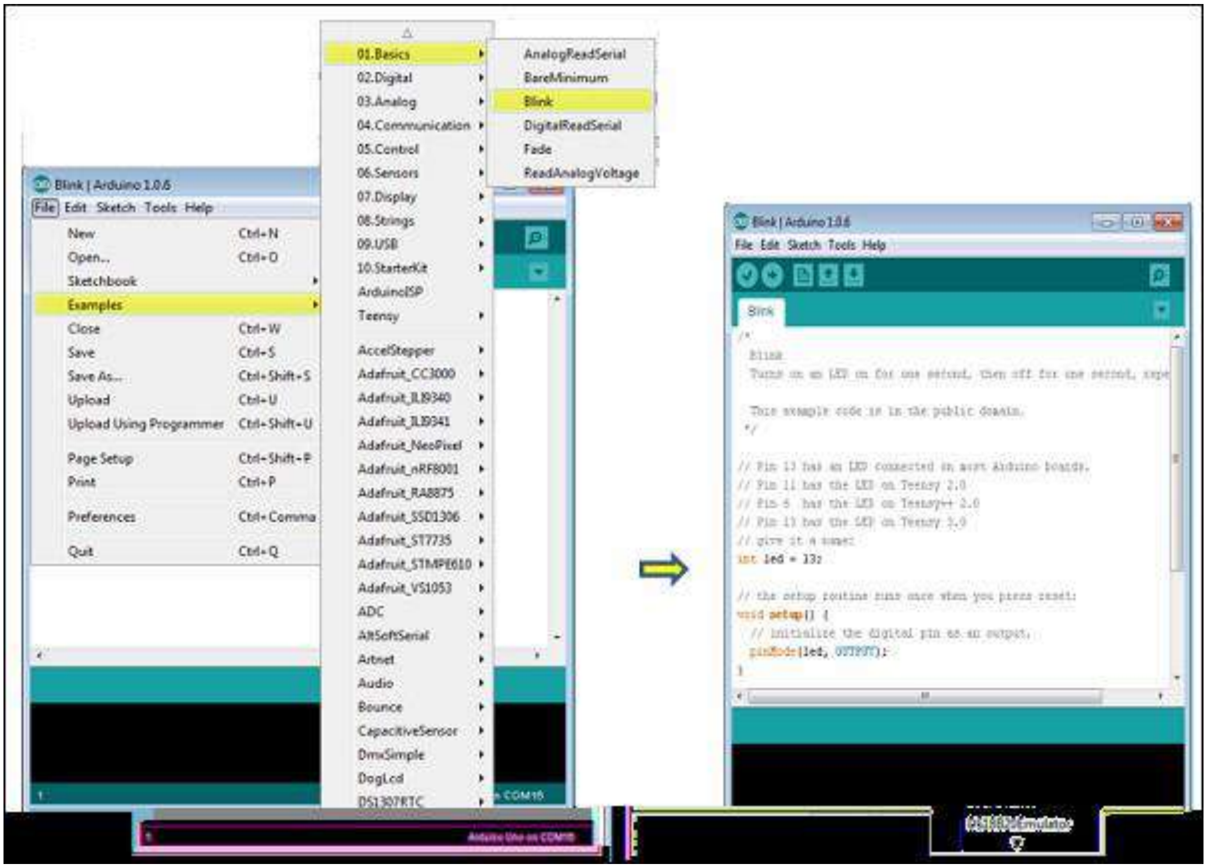
Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.



To open an existing project example, select File → Example → Basics → Blink.

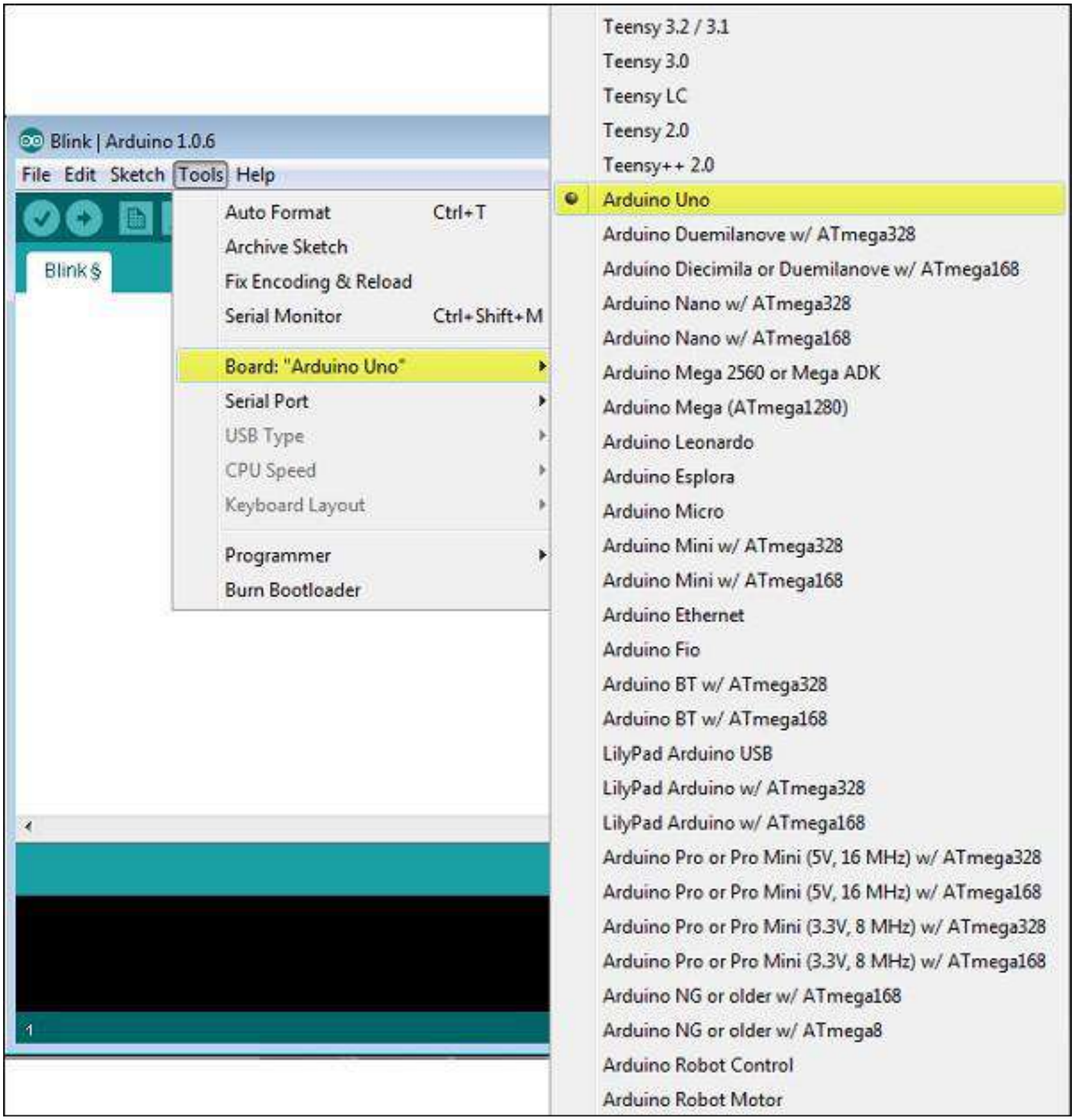


Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 – Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

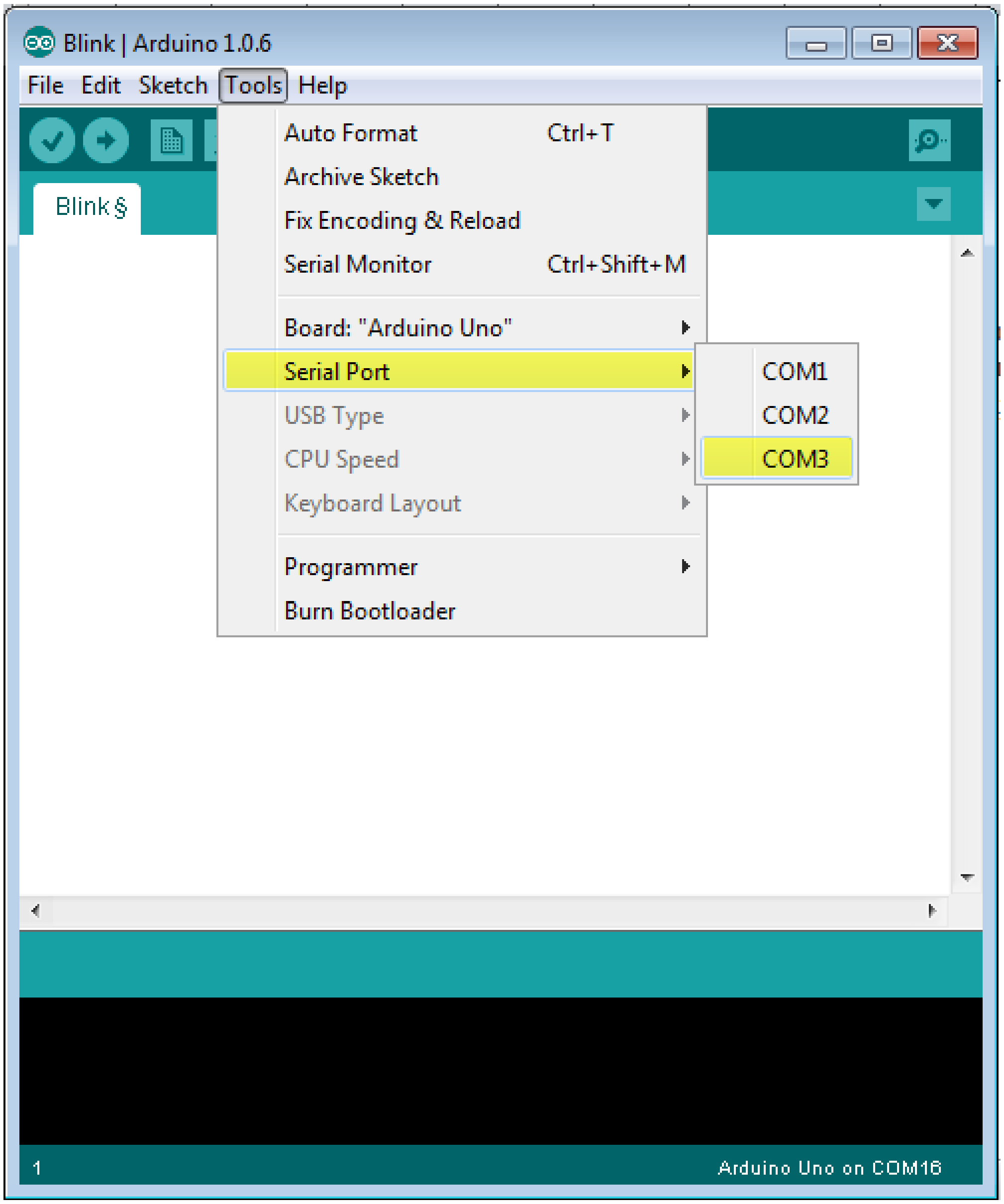
Go to Tools → Board and select your board.



Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

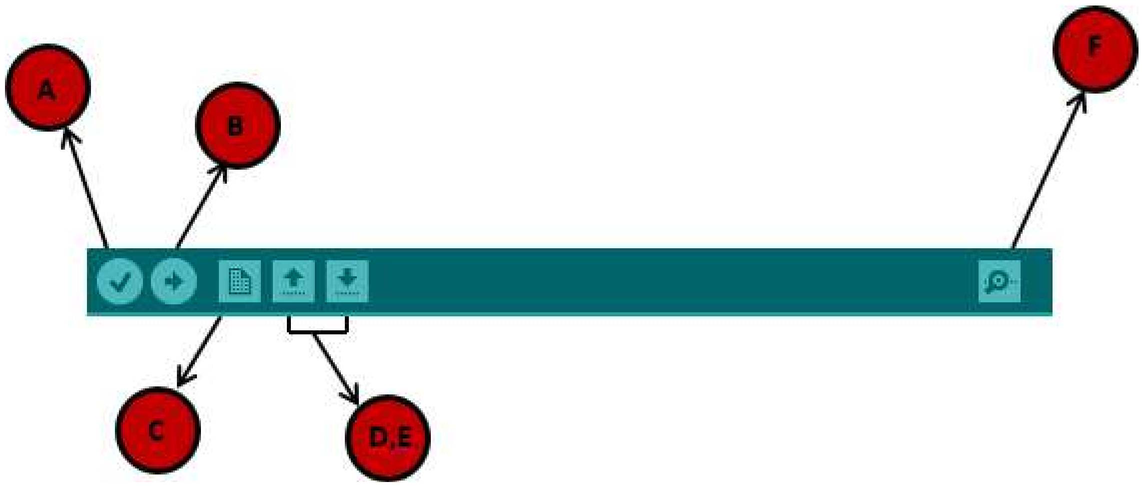
Step 7 – Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.



Step 8 – Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.



A − Used to check if there is any compilation error. **B**

− Used to upload a program to the Arduino board.**C**

− Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Connecting a Battery

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24 V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to solder the battery snap leads to a DC power plug and connect to the power jack on the board. A suitable plug is part number 28760 from www.jameco.com. Here is what this looks like.

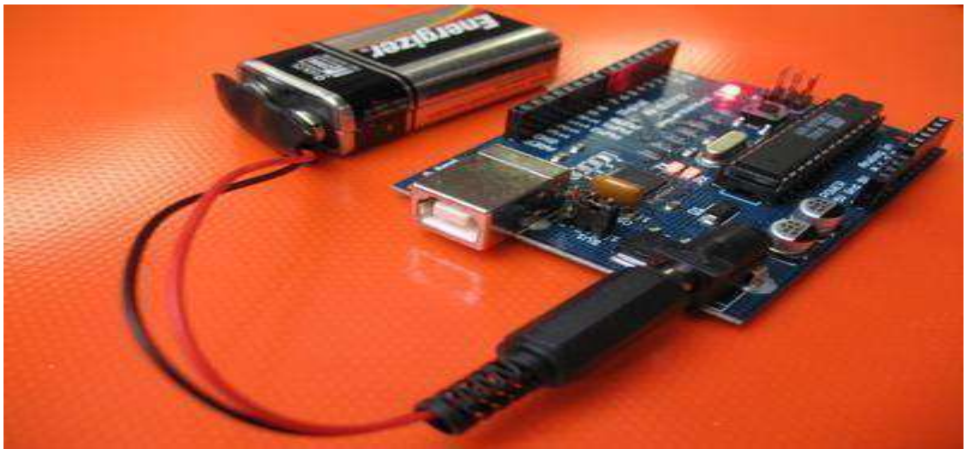


Fig.6.1: Arduino with battery

Disconnect your Arduino from the computer. Connect a 9 V battery to the Arduino power jack using the battery snap adapter. Confirm that the blinking program runs. This shows that you can power the Arduino from a battery and that the program you download runs without needing a connection to the host PC .

Moving On

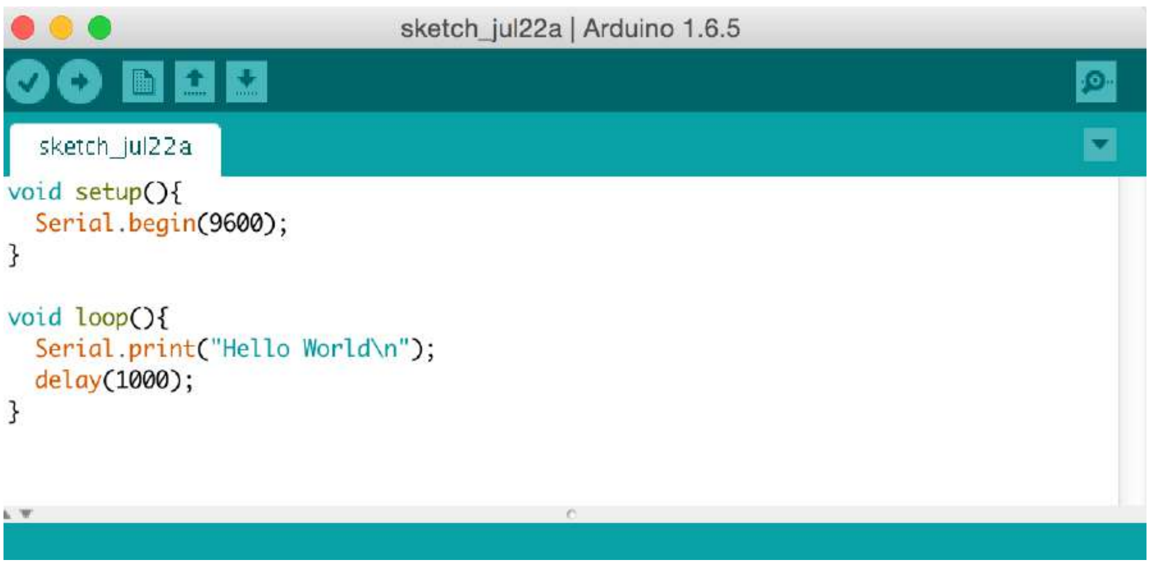
Connect your Arduino to the computer with the USB cable. You do not need the battery for now. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

Start the Arduino development environment. In Arduino-speak, programs are called “sketches”, but here we will just call them programs.

In the editing window that comes up, enter the following program, paying attention to where semi-colons appear at the end of command lines.

```
void setup()
```

```
{
  Serial.begin(9600);
  Serial.println("Hello World");
}
void loop() {}
```



6.2 MC Programming Language: Embedded C

This is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements. Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

6.3 Proteus:

Proteus: Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.

The designer can also develop 2D drawings for the product.

Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semiconductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Starting New Design

Step 1: Open ISIS software and select New design in File menu

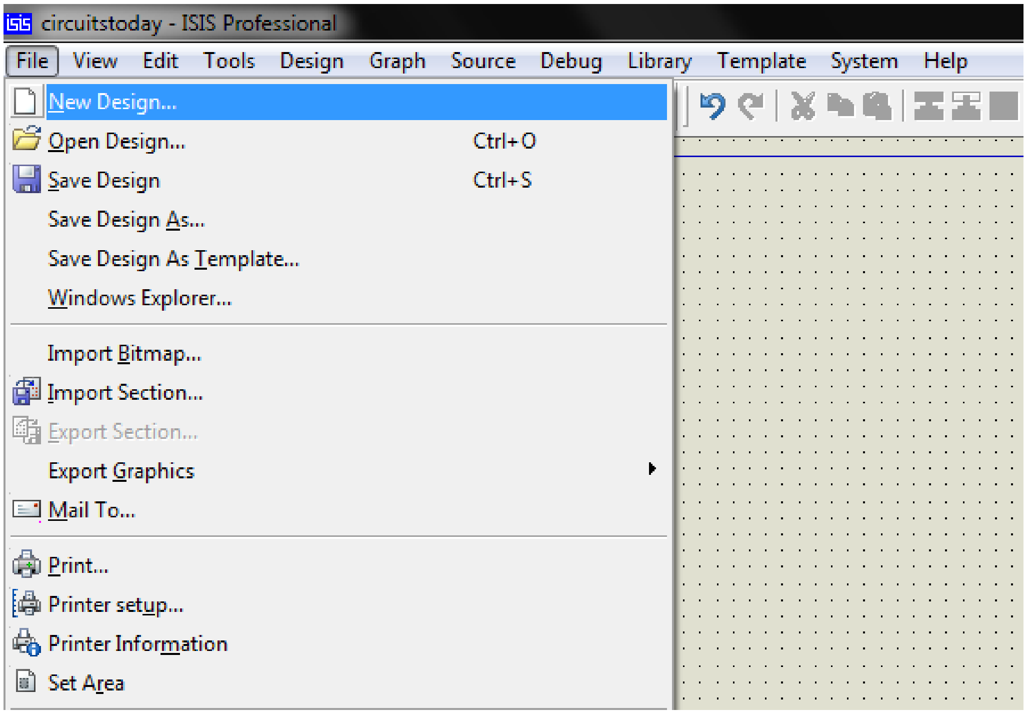


Fig Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

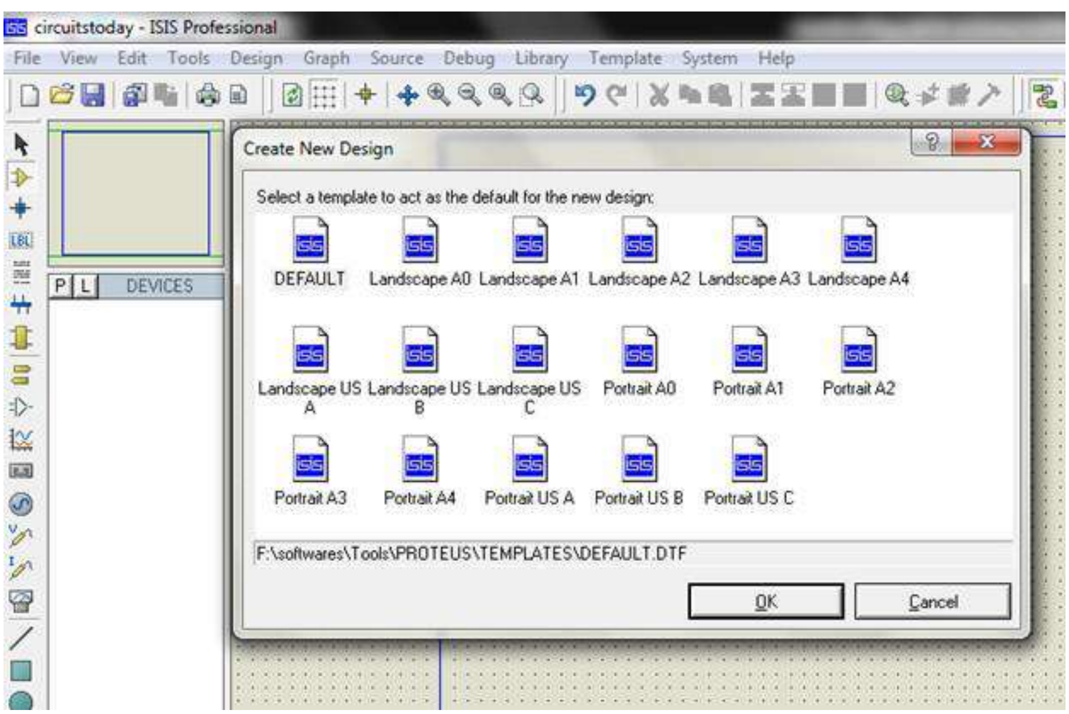


Fig Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

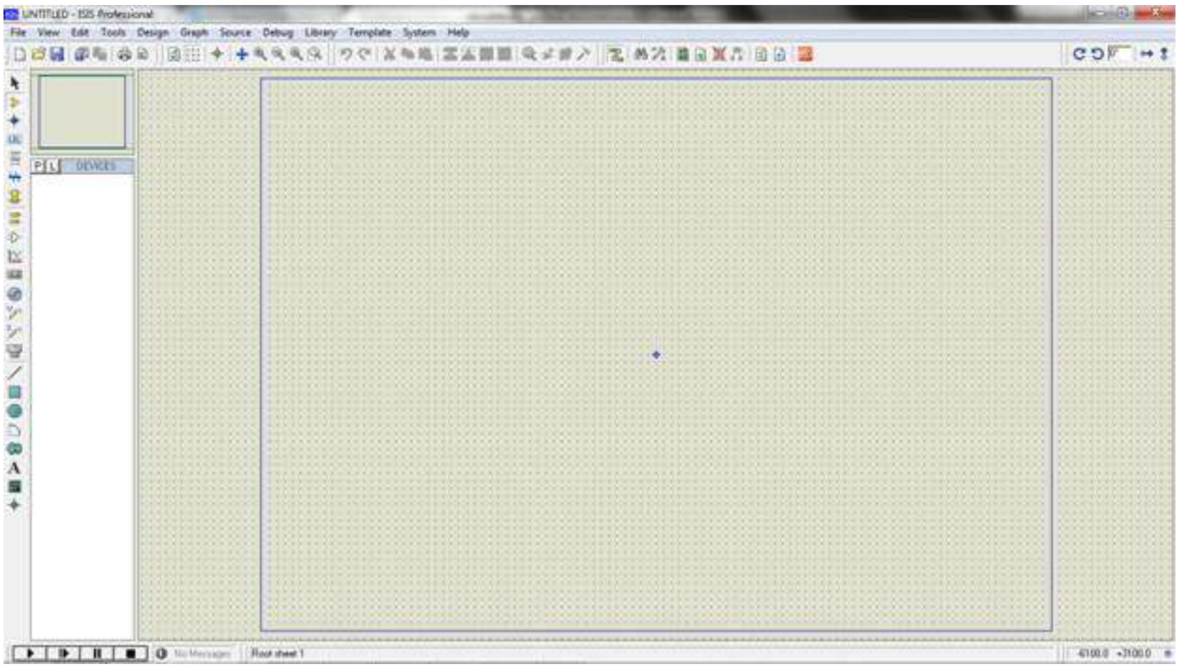


Fig Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

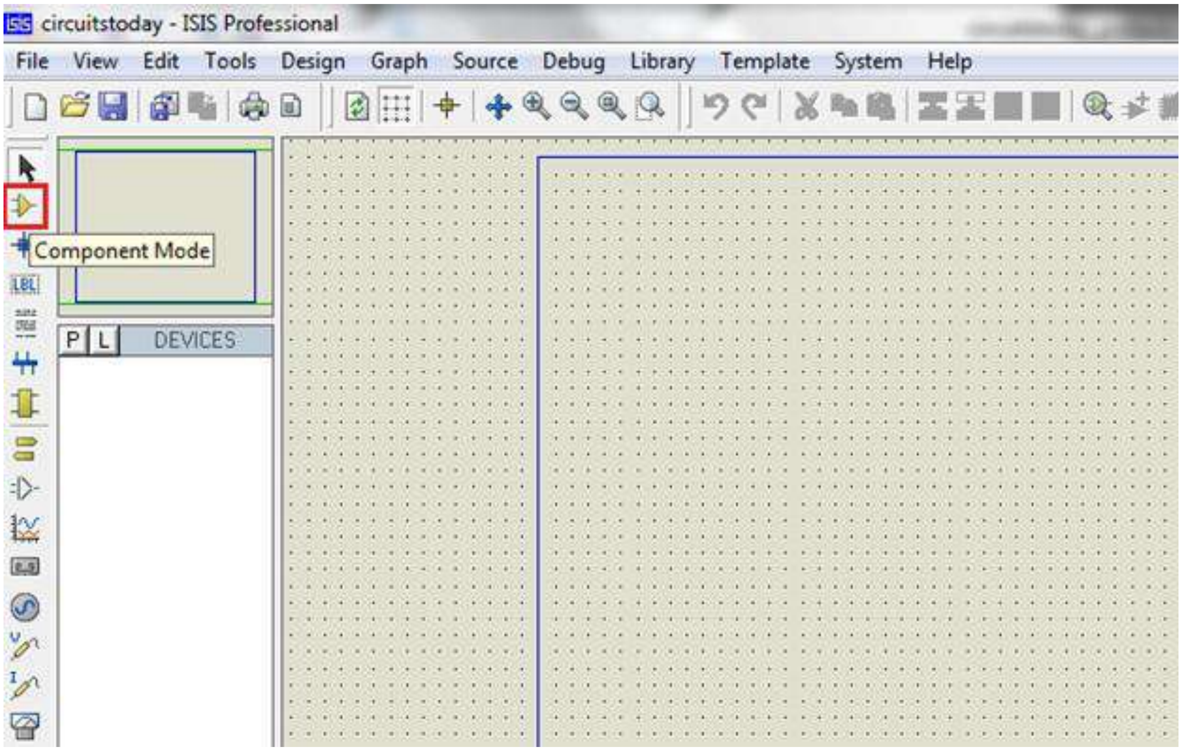


Fig Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

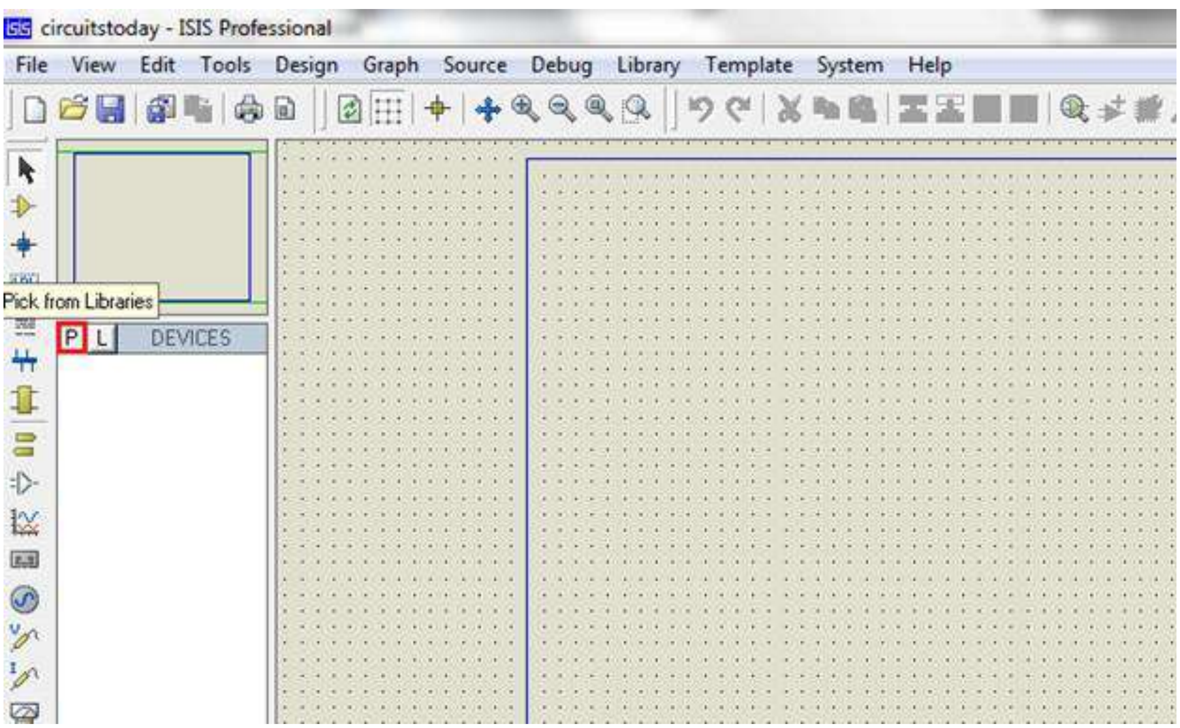


Fig Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

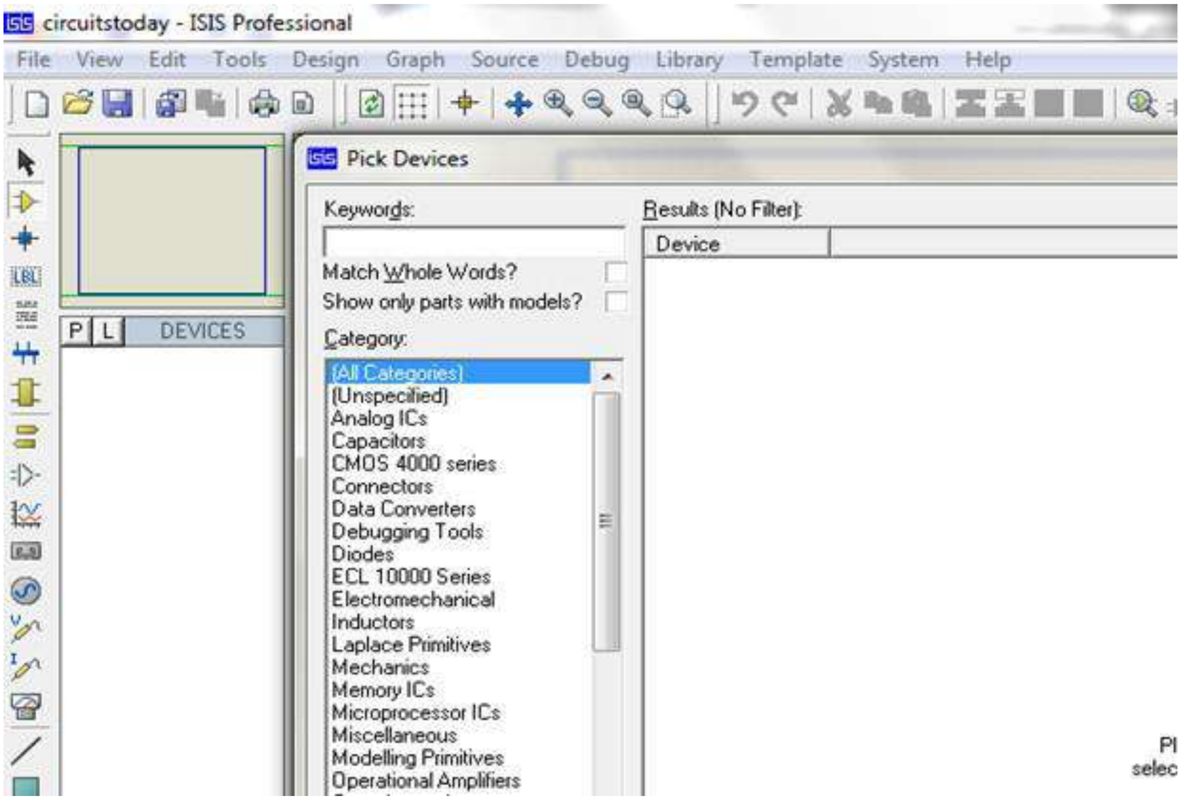


Fig Keywords Textbox

Example shows selection of push button. Select the components accordingly.

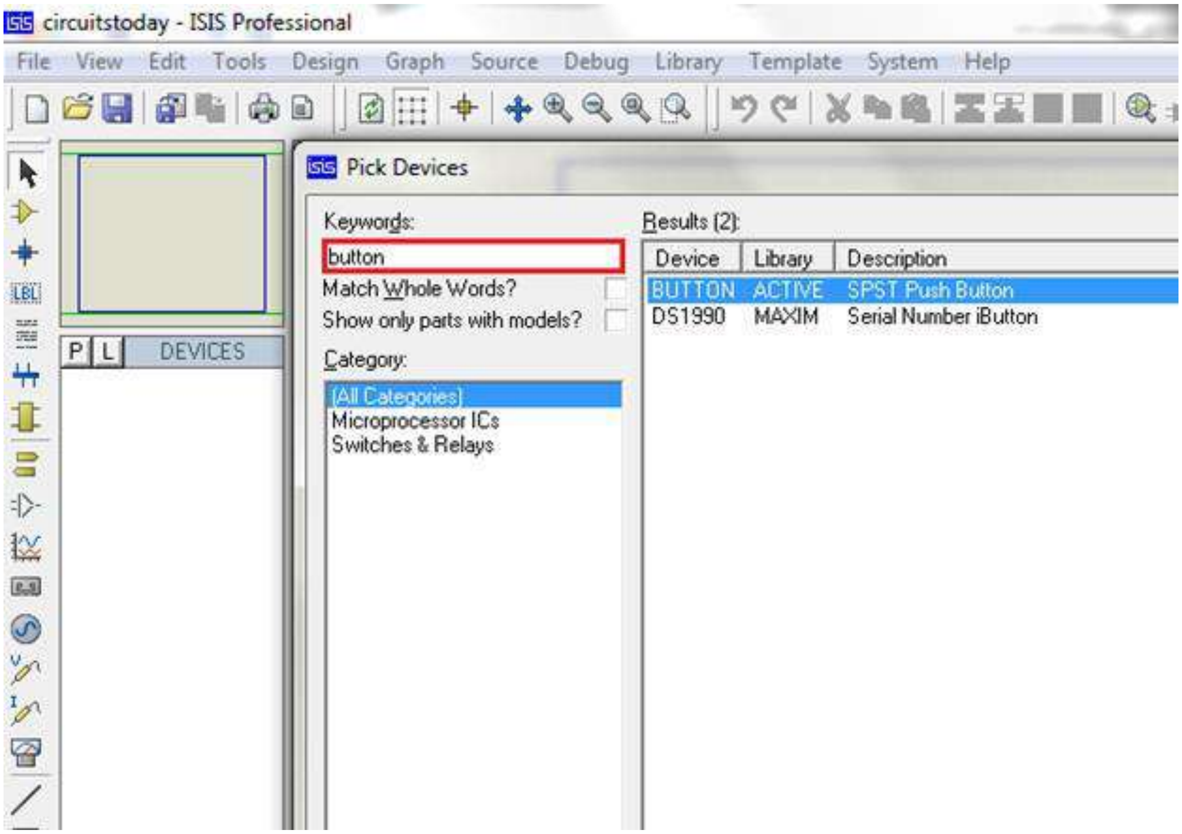


Fig Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

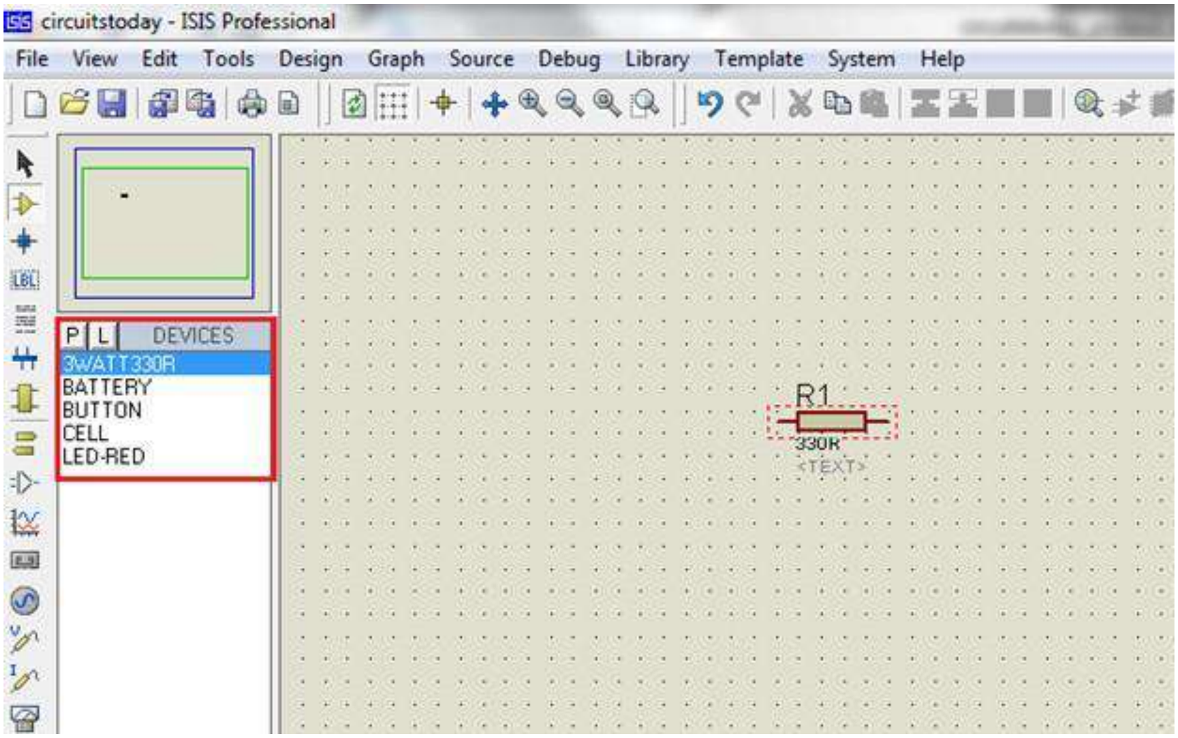


Fig Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

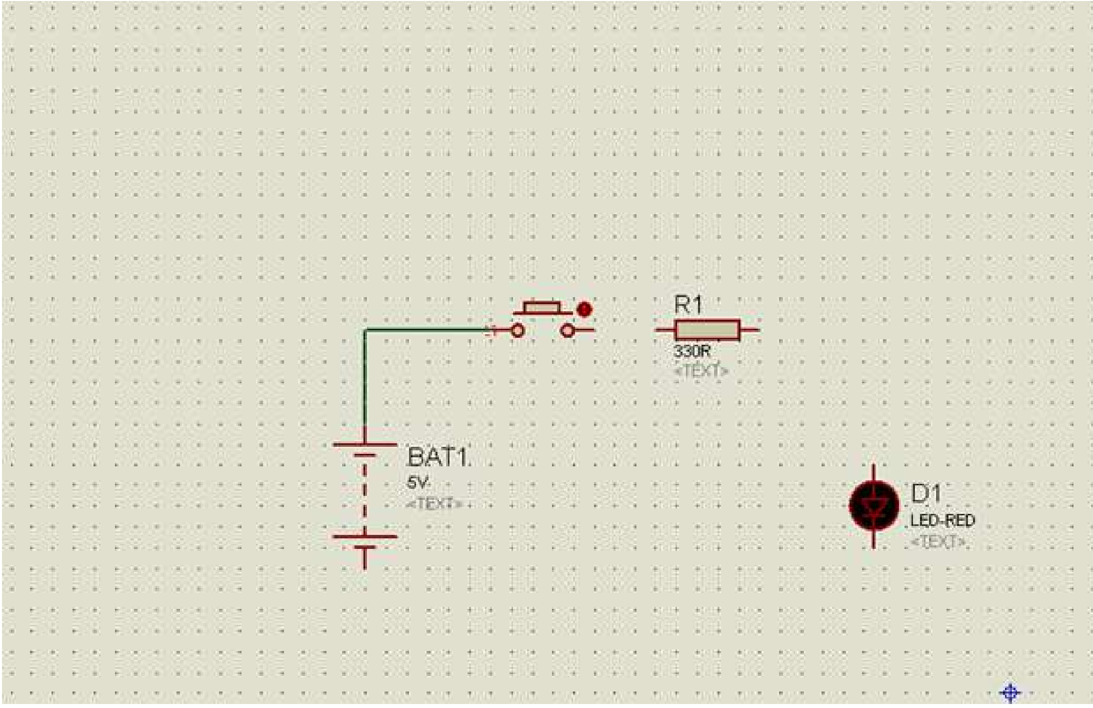


Fig Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

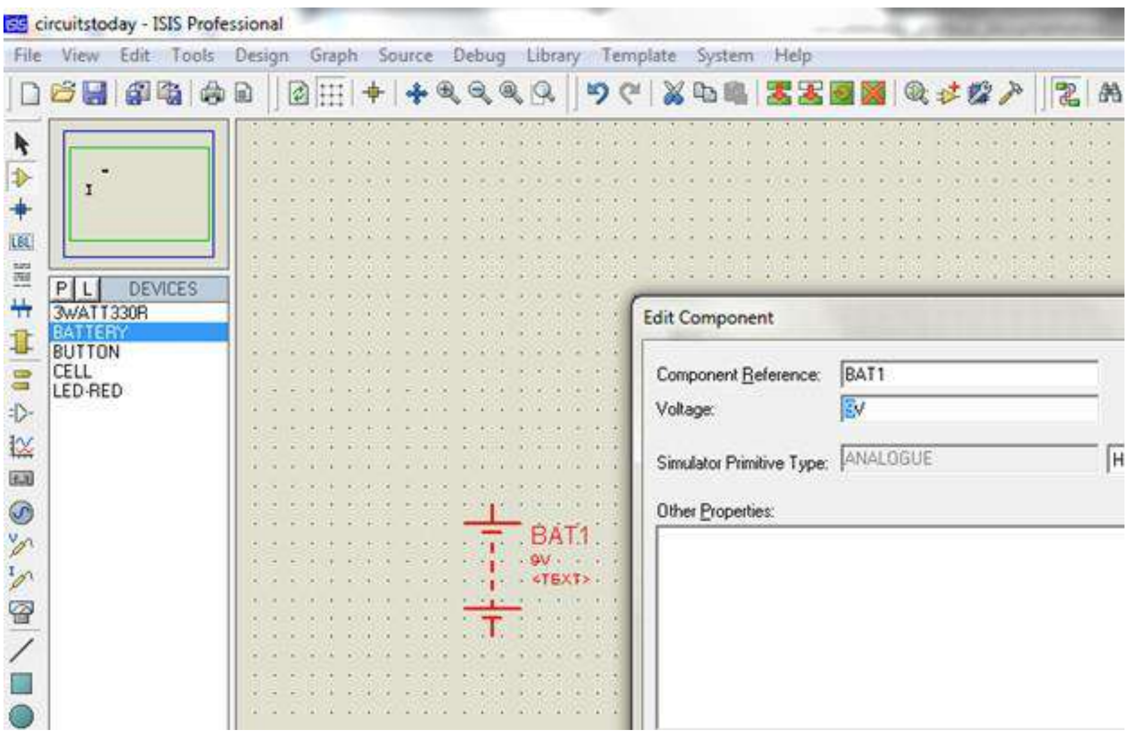


Fig Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

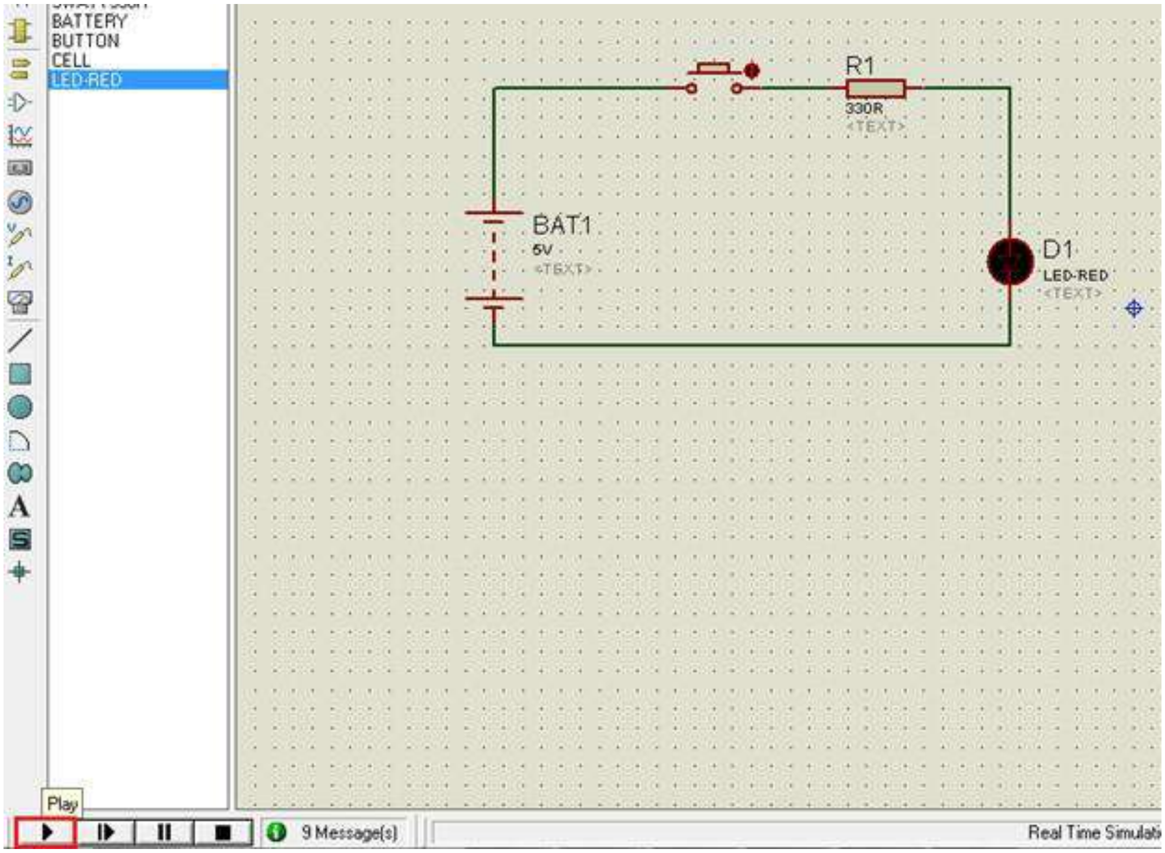


Fig Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

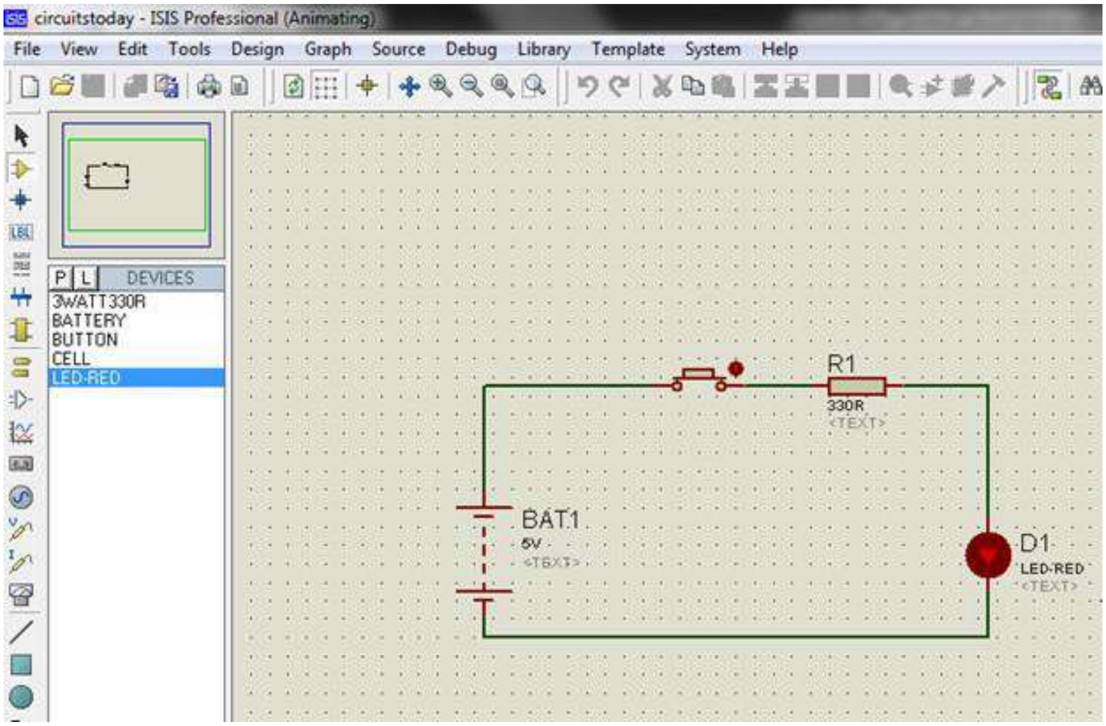


Fig Simulation Animating

Simulation can be stepped, paused or stopped at any time.

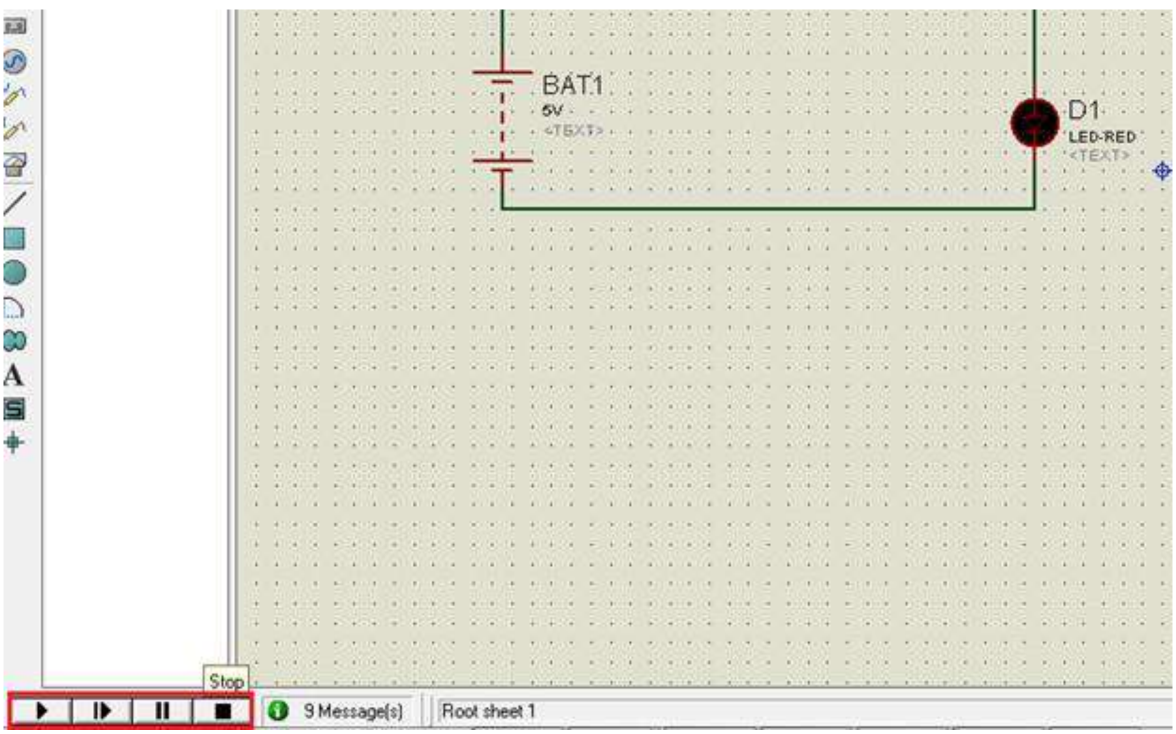


Fig Simulation Step-Pause-Stop Buttons

Project Code:

```
//#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);
const int trigPin = A5;
const int echoPin = A4;
int relay = 6;
int m1=2,m2=3,m3=4,m4=5;
int ir = A0;
void forward();
void st();
void left();
void right();

long duration;
int distance;
void setup() {

// Serial.begin(9600);
  pinMode(m1 , OUTPUT);
  pinMode(m2, OUTPUT);
  pinMode(m3, OUTPUT);
  pinMode(m4, OUTPUT);
  pinMode(relay, OUTPUT);

  pinMode(ir, INPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
```

```
 digitalWrite(m1,LOW);
 digitalWrite(m2,LOW);
 digitalWrite(m3,LOW);
 digitalWrite(m4,LOW);
 digitalWrite(relay,LOW);
 lcd.begin(16,2);
 lcd.clear();
 lcd.setCursor(0,0);
 lcd.print("Welcome");
 delay(500);
}


void loop() {
 int id = digitalRead(ir);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("D:");lcd.print(distance);delay(500);
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);
 digitalWrite(trigPin, HIGH);
 delayMicroseconds(10);
 digitalWrite(trigPin, LOW);
 duration = pulseIn(echoPin, HIGH);
 distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)

 if(distance>15)
{
forward();
}
```

```
//Serial.println("robo forward");
if(distance<15)
{
  st();
  delay(1000);
  back();
  delay(1000);
  left();
  delay(1000);
  st();
  delay(1000);
}
  if(id == LOW )
  {
     lcd.clear();lcd.setCursor(0,0);lcd.print("GRASS DETECTED");
     //motor1.write(0); //servo rotates at full speed to the right
   digitalWrite(relay,HIGH);
 delay(1000);
  }
else
{
 digitalWrite(relay,LOW);
 delay(1000);
  }
}

void forward()
{
  digitalWrite(m1,HIGH);
```

```
digitalWrite(m2,LOW);
digitalWrite(m3,HIGH);
digitalWrite(m4,LOW);
}
void st()
{
  digitalWrite(m1,LOW);
digitalWrite(m2,LOW);
digitalWrite(m3,LOW);
digitalWrite(m4,LOW);
}
void left()
{
 digitalWrite(m1,HIGH);
digitalWrite(m2,LOW);
digitalWrite(m3,LOW);
digitalWrite(m4,LOW);
  }
void back()
{
digitalWrite(m1,LOW);
digitalWrite(m2,HIGH);
digitalWrite(m3,LOW);
digitalWrite(m4,HIGH);
  }
```

CHAPTER-7

RESULT & CONCLUSION

7.1 APPLICATION OF THE PROJECT:

- Industries
- lawns.

7.2 ADVANTAGES OF THE PROJECT:

7.3 RESULT:

7.4 CONCLUSION:

The work done on solar grass cutter will come across the challenge of ecologically friendly production. The low cost of maintenance and operations possible due to no cost for fuelling. The automated machine i.e solar grass cutter has been technologically advanced for the use of houses and establishing that have grasslands where tractor driven mowers possibly will not be used. It is very simple for human beings, who practice this project work for further alterations. The automated grass cutter is light in weight and to place it occupies a very small area. As it makes use of a renewable energy sources for this reason there is a zero running price. It has the ability that can charge the battery while the machine is in workable condition. The automated machine is used to maintain the lawns are very clean and uniform in schools, playground, and public gardens.

7.5 FUTURE SCOPE:

Video cameras can be used to acquire pictures of the object being spotted. By fixing a metal detector to the robot, it can be send to battlefields, forests, coal mines, etc...to find out the metal objects. Temperature sensors are incorporated into the machine to get the extreme temperature of dangerous zones where human beings cannot go. As a result of attaching a smoke sensor to the machine, it can get the information related concentration of smoke or gases in the respective fields.The Size of the machine can be compact. The efficiency of the machine can be modified by increases the capacity of the battery. We can also place a bin to collect the grass and we can also configure the ultrasonic sensors depending on the distance. Additionally, sensors are integrated for perfect outcomes and upgraded robotics. Enhanced programs are used to update the machine and to perform different tasks.

REFERENCE

[1] International Journal of Advanced Research in Computer and Communication Engineering ISO 3297:2007
Certified Vol. 5, Issue 9, September 2016

[2] International Journal of Engineering Trends and Technology (IJETT) – Volume 32 Number 2- February 2016

[3] International Journal of Engineering Science and Computing, May 2017

[4] Sagar J. S. T. , M. S. Balamurugan and J. A. Vivek, "A wireless framework for automotive monitoring systems," in Indian Journal of Science and Technology, Vol 8(19), IPL0146, August 2015

[5] https://www.openhacks.com/uploadsproductos/rain_sensor_module.pdf

[6] http://designinformaticslab.github.io/productdesign_tutorial/2017/01/24/soilmoisture _sensor.html

[7] content://com.sec.android.app.sbrowser/readinglist/0721190655.mhtml

[8] https://www.hackster.io/techmirtz/using-16x2-lcd-with-arduino-d89028

[9] https://www.arduino.cc/en/Guide/ArduinoUnoWiF

[1] Ruben tharan Sivagurunathan Link esvaran Sivagurunathan, Jeremy Chia Jun Hao, Design and Fabrication of Low Cost Portable Lawn Mower,Scholars Journal of Engineering and Technology (SJET), 2017; 5(10):584-591.

[2] Venkatesh. k, Priyanka K, Sridhar R, Sakthivel A, Fabrication and Analysis of Lawn Mower, International Journal of Innovative Research in Science, Engineering and Technology, Vol. 4, Special Issue 6, May 2015.

[3] AkshayRamteke, YogeshBokde, Amit Waghmare, Ajay kumar Nagpur, NitinkumarPadghan, A Review on Design and Modification of Grass Cutting Machine, JETIR May 2020, Volume 7, Issue 5.

[4] Khurmi, R.S. and Gupta, J.K. (2003). Machine Design. Eurasa Publishing House, Ltd. New Delhi, India.

[5] Shigley. J.E. and Mischke.C.R (2001). Mechanical Engineering Design. Mc-Graw Hill Co. Inc. New York.

[6] Yong and Chow, S.H. (1991). Design and Construction of an Improved Domestic Lawn Mower. Project Report; Beijing Institute of Technology, China.

[7] Desa Ahmad, “A Width of Cut Analysis on the Performance of a Rotary Strip Tiller”, Pertanika Vol. 9 No. I, 1986.

[8] MohdIshammudin Bin MohdYunus, “Design and Development of Grass Cutting Machine Using DFMA Methodology”, University Teknikal Malaysia Melaka, 2007/2008.

[9] "Smart Solar Grass Cutter Robot for Grass Trimming" by Ashish kumarchaudhari, Yuvrajsahu, Pramod Kumar Sahu, Subhash Chandra verma

[10] "Design and Implementation of Automatic Lawn Cutter" by Pratik Patil, AshwiniBhosale, Prof.SheetalJagtap.

[11] IJIRST "Modification of Solar Grass Cutting Machine "by Praful P. Ulhe, Manish D. Inwate and Fried D. Wankhede Krishna kumar S. Dhakte

[12] Bravo, R., “Tired From Mowing the Lawn”, Journal of Pediatric Health Care, 24: 2010, 123–126.

[13] Scherer, E. “Humanoid Robots for Human Life Support”, Proceedings of IFAC Conference on Supplemental ways for improving International stability through automation 15-17 June 2006, Ed. P. Kopacek, 101 – 105, Elsevier.

[14] Arkin, E.M., Fekete, S.P., Mitchell, J.S.B. “The lawnmower problem”, Proceedings of the 5th Canadian Conference on Computational Geometry, 1993, 461-466.

[15] Pansire, D.G. “Self-propelled Self-guiding Lawn Mower.” U.S. Patent 4, 1980, 180,964.

A

MAJOR PROJECT REPORT

A HEIRARCHICAL IMAGE MATTING MODEL FOR BLOOD VESSEL SEGMENTATTION IN FUNDUS IMAGES

Submitted By

Ms. G. SOWMYA (16K81A0414)

Submitted in partial fulfillment of the requirement for the award of degree of

BACHLOR OF TECHNOLOGY

IN

Electronics And Communication Engineering

Under the guidance of

Mr. D. Prasad M.Tech.

Assistant Professor

DEPARTMENT OF ELECTRONICS & COMMUNICATIONS

St.MARTIN'S ENGINEERING COLLEGE

AN AUTONOMOUS INSTITUTE

Dhulapally, Secunderabad – 500 100

JUNE 2021

St.MARTIN'S ENGINEERING COLLEGE

UGC AUTONOMOUS

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

Dhulapally, Secunderabad-500 100

NBA &NAAC A+ Accredited

BONAFIDE CERTIFICATE

Certified that this project report titled " **A HEIRARCHICAL IMAGE MATTING MODEL FOR BLOOD VESSEL SEGMENTATION IN FUNDUS IMAGES** ", is a bonafide work of **G. SOWMYA (16K81A0414)** for partial fulfillment of the requirements for the award of degree of **BACHLOR OF TECHNOLOGY** in **ELECTRONICS AND COMMUNICATIONS** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Project Guide
Mr. D. Prasad
Assistant Professor
Dept. Of Electronics & Communications

Head Of the Department
Dr. B. Hari Krisnna
Dept. Of Electronics & Communications

Internal Examiner

External Examiner

Place:

Date:

DECLARATION

I, the student of **Bachelor of Technology** in Department of '**Electronics and Communications**' session: <2016 – 2021>, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled **A Heirarchical Image Matting Model For Blood Vessel Segmentation In Fundus Images.** is the outcome of my own bonafide work and is correct to the best of my knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

G. SOWMYA (16K81A0414)

// ACKNOWLEDGEMENT

While bringing out this project report to its final form, I came across a number of people whose contributions in various ways helped our field of research and they deserve a special thanks, It is a pleasure to convey our gratitude to all of them.

First and foremost, I would like to express our deep sense of gratitude and indebtedness to our C**ollege Management** for their kind support and permission to use facilities available in the institute.

I especially would like to express my deep sense of gratitude and indebtedness to **Dr. Santosh Kumar Patra, our beloved principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired us in innumerable ways.

I would like to express my gratitude to **Dr. B. Hari Krishna Head & Professor,** the Dept. of Electronics & Communications for his advice, supervision and the vital contribution as and when required during this research. I am proud to record that I had an opportunity to with an exponentially experienced professor like him. The time we spent with him will remain in our memories for years to come.

I would like to express my gratitude to our Dept. Coordinator **Mr. A. Anand Kumar,** for his involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time come.

I sincerely thank our project internal guide **Mr. D. Prasad Assistant Professor,** Department of ECE and our class project coordinator **Ms. P. Pushpa Assistant Professor,** Department of ECE for guidance and encouragement in carrying out this project.

I am highly grateful to our college level project coordinators. **Dr.Malyadri, Professor in IT & Dr. R. Santosh Kumar Associate Professor in CSE** for their advice, supervision and the vital contribution as and when required during this research work and for their cooperation that is difficult to express in words.

G.SOWMYA

(16K81A0414)

Abstract

A hierarchical image matting model is proposed to extract blood vessels from fundus images. More specifically, a hierarchical strategy utilizing the continuity and extendibility of retinal blood vessels is integrated into the image matting model for blood vessel segmentation. Normally the matting models require the user specified trimap, which separates the input image into three regions manually: the foreground, background and unknown regions. However, since creating a user specified trimap is a tedious and time-consuming task, region features of blood vessels are used to generate the trimap automatically in this paper. The proposed model has low computational complexity and outperforms many other state-of-art supervised and unsupervised methods in terms of accuracy, which achieves a vessel segmentation accuracy of 96.0%, 95.7% and 95.1% in an average time of 10.72s, 15.74s and 50.71s on images from three publicly available fundus image datasets DRIVE, STARE, and CHASE DB1, respectively.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Retinal blood vessels generally show a coarse to fine centrifugal distribution and appear as a wire mesh-like structure or tree-like structure. Their morphological features, such as length, width and branching, play an important role in diagnosis, screening, early detection and treatment of various cardiovascular and pharmacological diseases such as stroke, vein occlusions, diabetes and arteriosclerosis . The analysis of morphological features of retinal blood vessels can facilitate a timely detection and treatment of a disease when it is still in its early stage. Moreover, the analysis of retinal blood vessels can assist in evaluation of retinal image registration , the relationship between vessel tortuosity and hypertensive.

Image matting refers to the problem of accurately extracting a foreground object from an input image. Generally image matting includes two main steps. The first step is generating a user specified trimap. The trimap is a hand-drawn segmented image, which separates the input image into three regions: foreground (shown in white), background (shown in black) and unknown (shown in gray). The second step is applying the image matting model to extract the pixels belonging to the foreground object from the unknown regions, based on the samples of foreground and background pixels marked by the user. An exemplary result achieved .Image matting is very useful in many important applications, such as image (or video) segmentation, image editing, video production, new view synthesis, and film making. To the best of our knowledge, image matting has never been employed before to extract blood vessels from fundus images. The major reason is that for retinal blood vessel segmentation, generating a user specified trimap is a tedious and time consuming task. In other words, it is not appropriate to create a trimap manually for retinal blood vessel segmentation. In addition, a normal image matting model needs to be designed carefully to improve the performance of blood vessel segmentation. In order to overcome these problems, region features of blood vessels are applied to generate the trimap automatically. Then a hierarchical image matting model is proposed to extract the pixels of blood vessel from the unknown regions. More specifically, a hierarchical strategy utilizing the continuity and extendibility of retinal blood vessels is integrated into the image matting model for blood vessel segmentation. The proposed model is evaluated on three public available datasets DRIVE, STARE, and

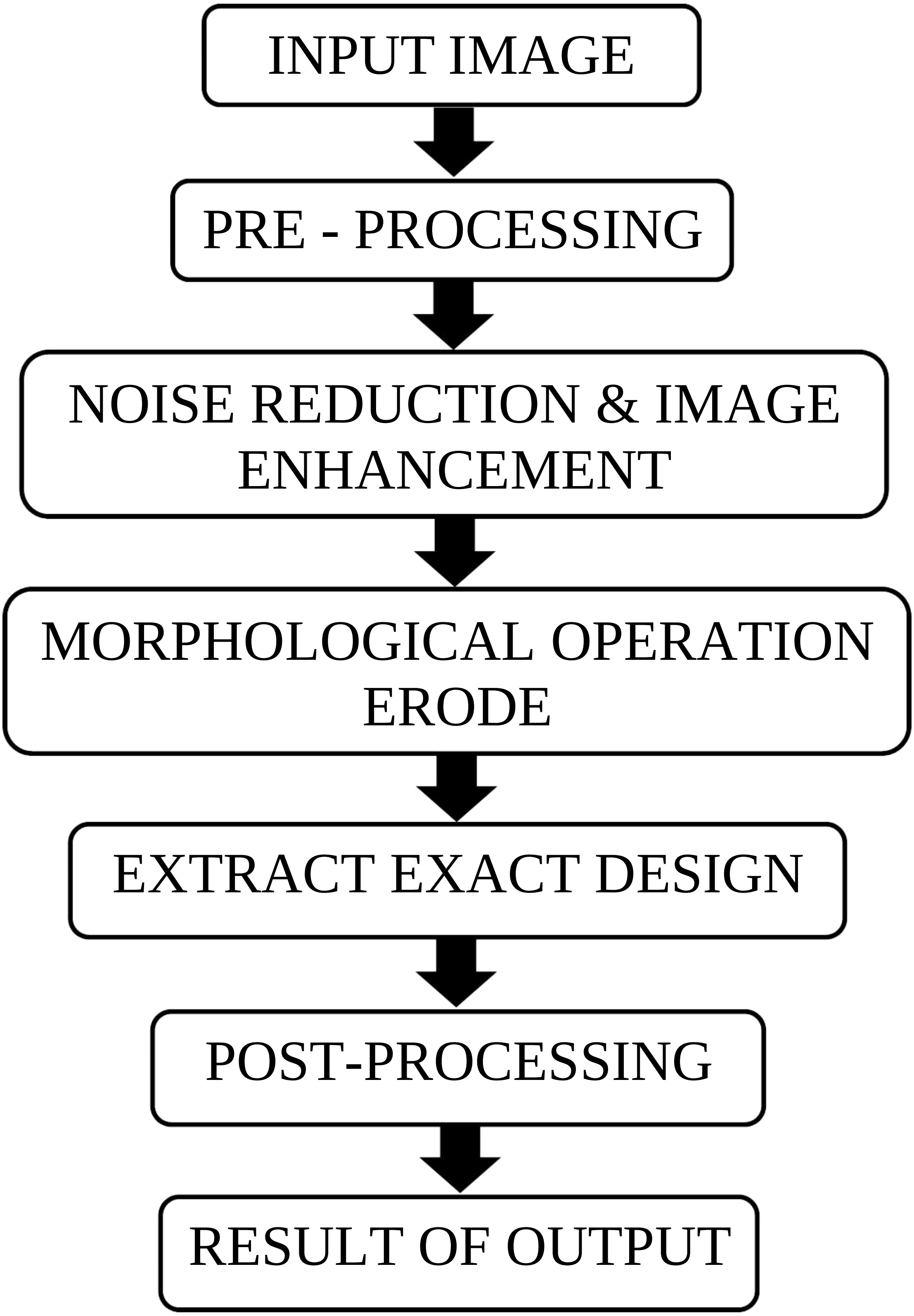
CHASE DB1, which have been widely used by other researchers to develop their own methods. The vessel segmentation performance demonstrates the efficiency and effectiveness of the proposed hierarchical image matting model.

Chapter 2

LITERATURE REVIEW

The segmentation of retinal blood vessels has been a heavily researched area in recent years. Broadly speaking, existing algorithms can be divided into supervised and unsupervised methods. In supervised methods, a number of different features are extracted from fundus images, and applied to train the effective classifiers with the purpose of extracting retinal blood vessels. In [10], Staal et al. extracts 27 features for each image pixel with ridge profiles, and performs feature selection by using sequential forward selection method to pick those pixels that result in better segmentation performance by a K-Nearest Neighbor (KNN) classifier. Soares et al. [11] introduces a feature-based Bayesian classifier with Gaussian mixtures for vessel segmentation, which uses the intensity information and Gabor wavelet transform responses to build a 7-D feature vector for each pixel. In [12], Lupascu et al. utilizes an AdaBoost classifier and a 41-D feature vector which includes information on the local intensity structure, spatial properties, and geometry at multiple scales. Marin et al. [13] constructs a 7-D vector composed of gray-level and moment invariantsbased features, and then trains a neural network (NN) for pixel classification. Roychowdhury et al. [14] extracts the major vessel from the fundus images and uses a Gaussian Mixture Model classifier for vessel segmentation with a set of 8 features, which are extracted based on pixel neighborhood and first and second-order gradient images. In [15], Liskowski et al. employs a deep neural network to extract vessel pixels from fundus images. In unsupervised methods, the researchers try to find inherent properties of retinal blood vessels that can be applied to extract vessel pixels from fundus images. The unsupervised methods can be further divided into multiscale approaches, matched filtering, vessel tracking, mathematical morphology and model based methods [9]. The multiscale approach introduced by [16] develops a vessel enhancement filter with the analysis of multiscale second order local structure of an image (Hessian), and obtains a vesselness measure by using the eigenvalues of the Hessian. The matched filtering method described by [17] employs different threshold probes to extract blood vessels from matched filter response images. In [18], the methodology based on vessel tracking applies a wave propagation and traceback mechanism to label each pixel the likelihood of belonging to vessels in angiography images. The mathematical morphology with the extraction of vessel centerlines [19] is also developed to find the morphological characteristics of retinal blood vessels. Model based methods generally use geometric deformable models [20], parametric deformable models [21], vessel profile models [22] and active contour models [23] for blood vessel segmentation.

FLOWCHART :


INPUT IMAGE
PRE - PROCESSING
NOISE REDUCTION & IMAGE ENHANCEMENT
MORPHOLOGICAL OPERATION ERODE
EXTRACT EXACT DESIGN
POST-PROCESSING
RESULT OF OUTPUT

CHAPTER 3

INTRODUCTION TO IMAGE PROCESSING

INTRODUCTION

3.1.IMAGE:

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Fig 3.1 General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

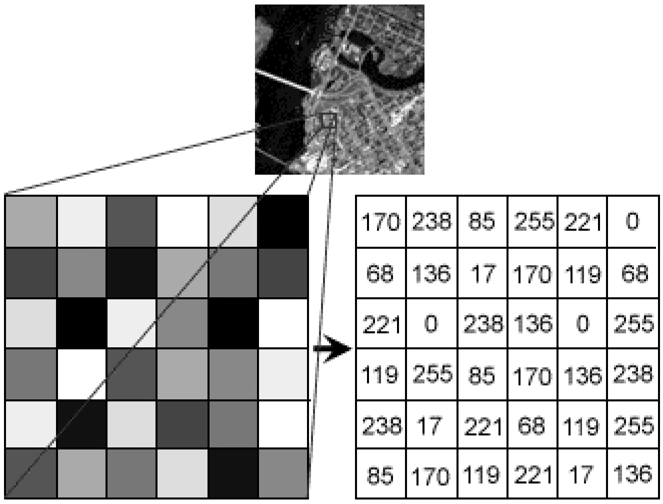


Fig 3.2 Image pixel

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

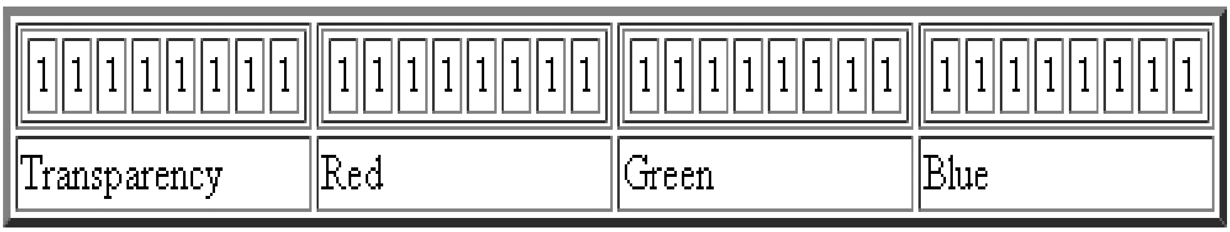


Fig 3.3. Transparency image

3.2 IMAGE FILE SIZES:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

3.3 IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

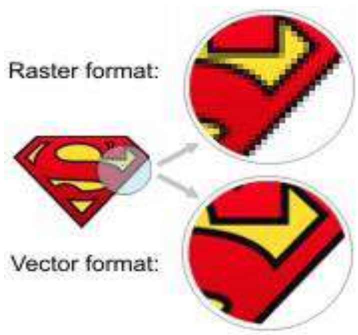


Fig3.4 Resolution image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

3.3.1 RASTER FORMATS:

These formats store images as bitmaps (also known as pixmaps)

- **JPEG/JFIF:**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

- **EXIF:**

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

- **TIFF:**

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

- **PNG:**

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

- **GIF:**

GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

- **BMP:**

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

3.3.2 VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM:**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

- **SVG:**

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

3.4 IMAGE PROCESSING:

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

3.5 FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:

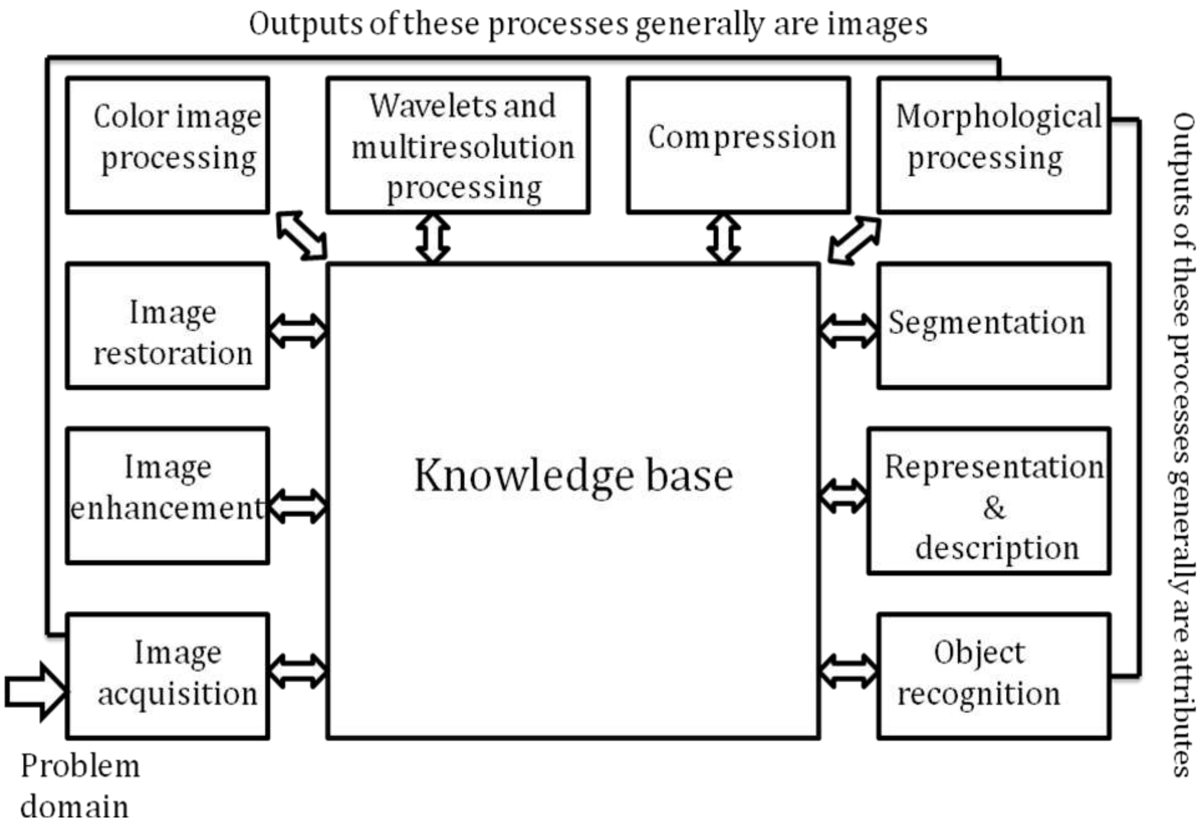


Fig 3.6 Image fundamental

3.5.1 Image Acquisition:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Fig 3.5.1 Digital camera image

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Fig 3.5.2 digital camera cell

3.5.2 Image Enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because "it looks better." It is important to keep in mind that enhancement is a very subjective area of image processing.

Fig 3.5.3 Image enhancement

3.5.3 Image restoration:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Fig 3.5.4 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, where as removal of image blur by applying a deblurring function is considered a restoration technique.

3.5.4 Color image processing:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Fig 3.5.5 Color & Gray scale image

3.5.5 Wavelets and multiresolution processing:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform based image processing since the late1950's, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

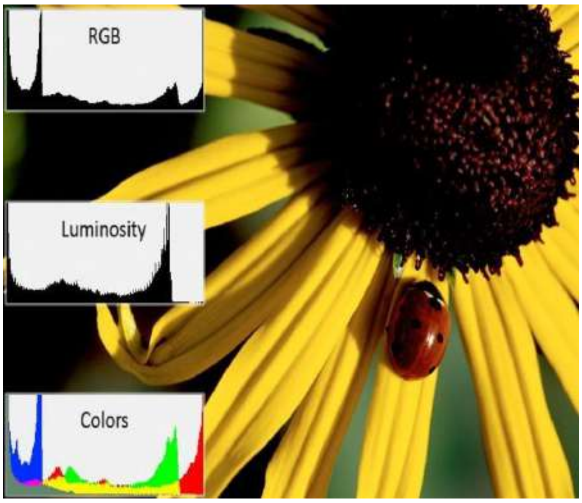


Fig 3.5.6rgb histogram image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called **Multiresolution** theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

3.5.6 Compression:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

3.5.7 Morphological processing:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Fig 3.5.7 blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

3.5.8 Segmentation:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

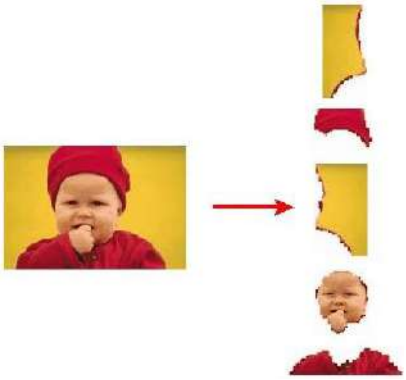


Fig 3.5.8 Image segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

3.5.9 Representation and description:

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

3.5.10 Object recognition:

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

3.5.11 Knowledgebase:

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an inter related to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

3.6 COMPONENTS OF AN IMAGE PROCESSING SYSTEM:

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

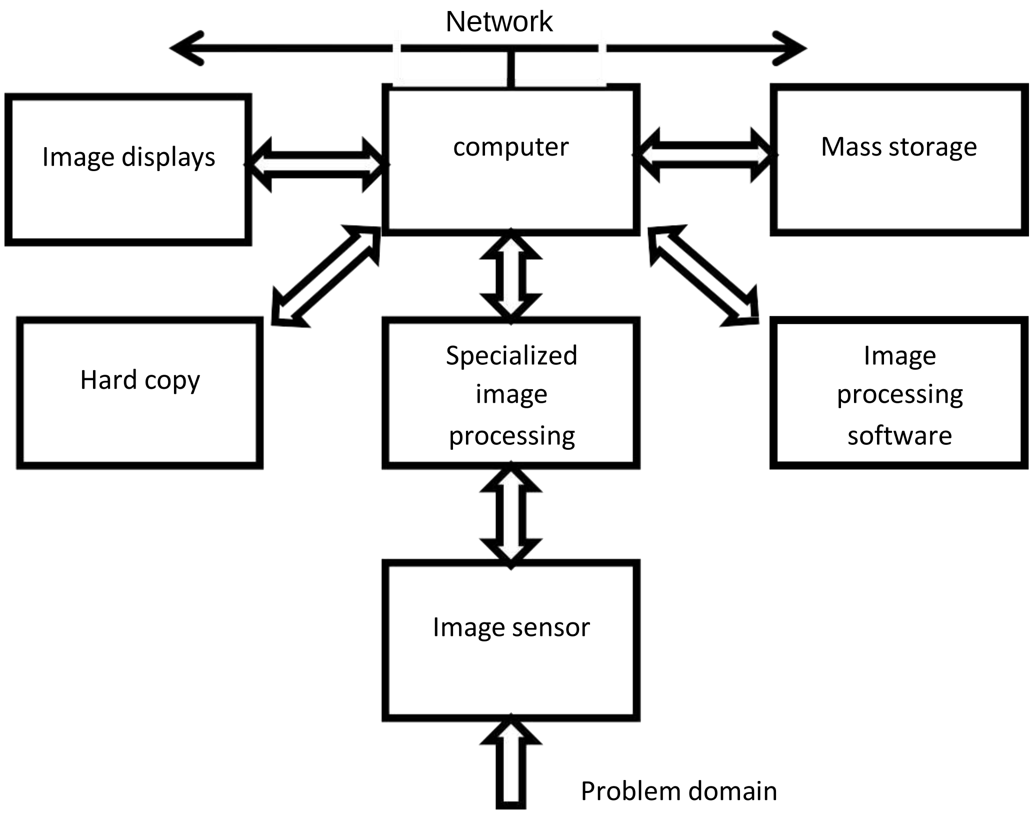


Fig 3.6 Component of image processing

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 1.24 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

- **Image sensors:**

With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

- **Specialized image processing hardware:**

  Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

- **Computer:**

  The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

- **Image processing software:**

  Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

- **Mass storage:**

  Mass storage capability is a must in image processing applications. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage forimage processing applications fall into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), M bytes (one million bytes), G bytes (meaning giga, or one billion, bytes), and T bytes (meaning tera, or one trillion, bytes)

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig. 1.24. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in "jukeboxes" are the usual media for archival applications.

- **Image displays:**

  Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are there requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

- **Hardcopy:**

  Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

- **Network:**

  Networking is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of optical fiber and other broadband technologies.

Color and texture are two low-level features widely used for image classification, indexing and retrieval. Color is usually represented as a histogram, which is a first order statistical measure that captures global distribution of color in an image One of the main drawbacks of the histogram- based approaches is that the spatial distribution and local variations in color are ignored. Local spatial variation of pixel intensity is commonly used to capture texture information in an image. Grayscale Co-occurrence Matrix (GCM) is a well-known method for texture extraction in the spatial domain. A GCM stores the number of pixel neighborhoods in an image that have a particular grayscale combination. Let I be an image and let p and Np respectively denote any arbitrary pixel and its neighbor in a given direction. If GL denotes the total number of quantized gray levels and gl denotes the individual gray levels, where, gl $\in$ {0, . . .,GL _ 1}, then each component of GCM can be written as follows:

$$\mathrm{gcm}(i, j) = \Pr((\mathrm{gl}_p, \mathrm{gl}_{N_p}) = (i, j))$$

gcm(i, j) is the number of times the gray level of a pixel p denoted by glp equals i, and the gray level of its neighbor Np denoted by glNp equals j, as a fraction of the total number of pixels in the image. Thus, it estimates the probability that the gray level of an arbitrary pixel in an image is i, and that of its neighbor is j. One GCM matrix is generated for each possible neighborhood direction, namely, 0, 45, 90 and 135.Average and range of 14 features like Angular Second Moment, Contrast, Correlation, etc., are generated by combining all the four matrices to get a total of 28 features. In the GCM approach for texture extraction, color information is completely lost since only pixel gray levels are considered.

To incorporate spatial information along with the color of image pixels, a feature called color correlogram has recently been proposed. It is a three dimensional matrix that represents the probability of finding pixels of any two given colors at a distance ‘d’ apart Auto correlogram is a variation of correlogram, which represents the probability of finding two pixels with the same color at a distance ‘d’ apart. This atproach can effectively represent color distribution in an image. However, correlogram features do not capture intensity variation Many image databases often contain both color as well as gray scale images. The color correlogram method does not constitute a good descriptor in such databases.

Another method called Color Co-occurrence Matrix (CCM) has been proposed to capture color variation in an image. CCM is represented as a three-dimensional matrix, where color pair of the pixels p and Np are captured in the first two dimensions of the matrix and the spatial distance ‘d’ between these two pixels is captured in the third dimension. This approach is a generalization of the color correlogram and reduces to the pure color correlogram for d = 1. CCM is generated using only the Hue plane of the HSV (Hue, Saturation and Intensity Value) color space. The Hue axis is quantized into HL number of levels. If individual hue values are denoted by hl, where $\mathrm{hl} \in \{0, \ldots, \mathrm{HL} - 1\}$, then each component of CCM can be written as follows:

$$\mathrm{ccm}(i, j) = \mathrm{Pr}((\mathrm{hl}_{\mathrm{p}}, \mathrm{hl}_{N_{\mathrm{p}}}) = (i, j))$$

Four matrices representing neighbors at angles 0, 90, 180 and 270 are considered. This approach was further extended by separating the diagonal and the non-diagonal components of CCM to generate a Modified Color Co-occurrence Matrix (MCCM). MCCM, thus, may be written as follows: MCCM = (CCMD;CCMND)

Here, CCMD and CCMND correspond to the diagonal and off-diagonal components of CCM. The main drawback of this approach is that, like correlogram, it also captures only color information and intensity information is completely ignored.

An alternative approach is to capture intensity variation as a texture feature from an image and combine it with color features like histograms using suitable weights . One of the challenges of this approach is to determine suitable weights since these are highly application-dependent. In certain applications like Content-based Image Retrieval (CBIR), weights are often estimated from relevance feedback given by users.

While relevance feedback is sometimes effective, it makes the process of image retrieval user-dependent and iterative. There is also no guarantee on the convergence of the weight-learning algorithms. In order to overcome these problems, researchers have tried to combine color and texture features together during extraction.

proposed two approaches for capturing color and intensity variations from an image using the LUV color space. In the Single-channel Co-occurrence Matrix (SCM), variations for each color channel, namely, L, U and V are considered independently. In the Multi channel Co-occurrence Matrix (MCM), variations are captured taking two channels at a time – UV, LU and LV. Since the LUV color space separates out chrominance (L and U) from luminance (V), SCM in effect, generates one GCM and two CCMs from each image independently. As a result, correlation between the color channels is lost

However, in MCM, the count of pair wise occurrences of the values of different channels of the color space is captured. Thus, each component of MCM can be written as follows:

mcmUV(i; j) = Pr((up; vNp) = (i; j))

mcmLU(i; j) = Pr((lp; uNp) = (i; j))

mcmLV(i; j) = Pr((lp; vNp) = (i; j))

Here, mcmUV(i, j) is the number of times the U chromaticity value of a pixel p denoted by up equals i, and the V chromaticity value of its neighbor Np denoted by vNp equals j, as a fraction of the total number of pixels in the image. Similarly, mcmLU(i, j) and mcmLV(i, j) are defined. One MCM matrix is generated for each of the four neighborhood directions, namely, 0, 45, 90 and 135.

Deng and Manjunath (2001)proposed a two-stage method called JSEG, which combines color and texture after image segmentation. In the first stage, colors are quantized to the required levels for differentiating between various regions of an image. Pixel values of the regions are then replaced by their quantized color levels to form a color map. Spatial variation of color levels between different regions in the map is viewed as a type of texture composition of the image.

Yu et al. (2002) suggested the use of color texture moments to represent both color and texture of an image. This approach is based on the calculation of

Local Fourier Transformation (LFT) coefficients. Eight templates equivalent to LFT are operated over an image to generate a characteristic map of the image. Each template is a 3 · 3 filter that considers eight neighbors of the current pixel for LFT calculation. First and second order moments of the characteristic map are then used to generate a set of features.

In this paper, we propose an integrated approach for capturing spatial variation of both color and intensity levels in the neighborhood of each pixel using the HSV color space. In contrast to the other methods, for each pixel and its neighbor, the amount of color and intensity variation between them is estimated using a weight function. Suitable constraints are satisfied while choosing the weight function for effectively relating visual perception of color and the HSV color space properties. The color and intensity variations are represented in a single composite feature known as Integrated Color and Intensity Co-occurrence Matrix (ICICM). While the existing schemes generally treat color and intensity separately, the proposed method provides a composite view to both color and intensity variations in the same feature. The main advantage of using ICICM is that it avoids the use of weights to combine individual color and texture features. We use ICICM feature in an image retrieval application from large image databases.

Early result on this work was reported in (Vadivel et al., 2004a). In the next section, we describe the proposed feature extraction technique after introducing some of the properties of the HSV color space. Choice of quantization levels for color and intensity axes, selection of parameter values and a brief overview of the image retrieval application

Integrated color and intensity co-occurrence matrix:

We propose to capture color and intensity variation around each pixel in a two-dimensional matrix called Integrated Color and Intensity Co-occurrence Matrix (ICICM). This is a generalization of the Grayscale Co-occurrence Matrix and the Color Co-occurrence Matrix techniques. For each pair of neighboring pixels, we consider

their contribution to both color perception as well as gray level perception to the human eye. Some of the useful properties of the HSV color space and their relationship to human color perception are utilized for extracting this feature. In the next sub-section, we briefly explain relevant properties of the HSV color space. In the subsequent subsection, we describe how the properties can be effectively used for generating ICICM.

HSV color space:

HSV Color space: Basically there are three properties or three dimensions of color that being hue, saturation and value HSV means Hue, Saturation and Value. It is important to look at because it describes the color based on three properties. It can create the full spectrum of colors by editing the HSV values. The first dimension is the Hue. Hue is the other name for the color or the complicated variation in the color. The quality of color as determined by its dominant wavelength. This Hue is broadly classified into three categories. They are primary Hue, Secondary Hue and Teritiary Hue. The first and the foremost is the primary Hue it consists of three colors they are red, yellow and blue. The secondary Hue is formed by the combination of the equal amount of colors of the primary Hue and the colors of the secondary Hue which was formed by the primary Hue are Orange, Green and violet. The remaining one is the teritiary Hue is formed by the combination of the primary Hue and the secondary Hue. The limitless number of colors are produced by mixing the colors of the primary Hue in different amounts. Saturation is the degree or the purity of color. Then the second dimension is the saturation. Saturation just gives the intensity to the colors. The saturation and intensity drops just by mixing the colors or by adding black to the color. By adding the white to the color in spite of more intense the color becomes lighter. Then finally the third dimension is the Value. The value is the brightness of the color. When the value is zero the color space is totally black with the increase in the color there is also increase in the brightness and shows the various colors. The value describes the contrast of the color. That means it describes the lightness and darkness of the color. As similar to the saturation this value consists of the tints and shades. Tints are the colors with the added white and shades are the colors with the added black.

Properties of the HSV color space:

Sensing of light from an image in the layers of human retina is a complex process with rod cells contributing to scotopic or dim-light vision and cone cells to photopic or bright-light vision (Gonzalez and Woods, 2002). At low levels of illumination, only the rod cells are excited so that only gray shades are perceived. As the illumination level increases, more and more cone cells are excited, resulting in increased color perception. Various color spaces have been introduced to represent and specify colors in a way suitable for storage, processing or transmission of color information in images. Out of these, HSV is one of the models that separate out the luminance component (Intensity) of a pixel color from its chrominance components (Hue and Saturation). Hue represents pure color, which is perceived when incident light is of sufficient illumination and contains a single wavelength. Saturation gives a measure of the degree by which a pure color is diluted by white light. For light with low illumination, corresponding intensity value in the HSV color space is also low.

The HSV color space can be represented as a Hexa cone, with the central vertical axis denoting the luminance component, I (often denoted by V for Intensity Value). Hue, is a chrominance component defined as an angle in the range [0,2p] relative to the red axis with red at angle 0, green at 2p/3, blue at 4p/3 and red again at 2p. Saturation, S, is the other chrominance component, measured as a radial distance from the central axis of the hexacone with value between 0 at the center to 1 at the outer surface. For zero saturation, as the intensity is increased, we move from black to white through various shades of gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the color represented by its hue. When saturation is near 0, all the pixels in an image look alike even though their hue values are different.

As we increase saturation towards 1, the colors get separated out and are visually perceived as the true colors represented by their hues. Low saturation implies presence of a large number of spectral components in the incident light, causing loss of color information even though the illumination level is sufficiently high. Thus, for low values of saturation or intensity, we can approximate a pixel color by a gray level while for higher saturation and intensity, the pixel color can be approximated by its hue. For low intensities, even for a high saturation, a pixel color is close to its gray value. Similarly, for low saturation even for a high value of intensity, a pixel is perceived as gray. We use these properties to estimate the degree by which a pixel contributes to color perception and gray level perception.

One possible way of capturing color perception of a pixel is to choose suitable thresholds on the intensity and saturation. If the saturation and the intensity are above their respective thresholds, we may consider the pixel to have color dominance; else, it has gray level dominance. However, such a hard thresholding does not properly capture color perception near the threshold values. This is due to the fact that there is no fixed level of illumination above which the cone cells get excited. Instead, there is a gradual transition from scotopic to photopic vision. Similarly, there is no fixed threshold for the saturation of cone cells that leads to loss of chromatic information at higher levels of illumination caused by color dilution. We, therefore, use suitable weights that vary smoothly with saturation and intensity to represent both color and gray scale perception for each pixel.

NON INTERVAL QUANTIZATION:

Due to the large range for each component by directly calculating the characteristics for the retrieval then the computation will be very difficult to ensure rapid retrieval. It is essential to quantify HSV space component to reduce computation and improve efficiency. At the same time, because the human eye to distinguish colors is limited, do not need to calculate all segments. Unequal interval quantization according the human color perception has been applied on H ,S ,V components.

Based on the color model of substantial analysis, we divide color into eight parts. Saturation and intensity is divided into three parts separately in accordance with the human eyes to distinguish. In accordance with the different colors and subjective color perception quantification, quantified hue(H), saturation(S) and value(V)

In accordance with the quantization level above, the H, S, V three-dimensional feature vector for different values of with different weights to form one dimensional feature vector and is given by the following equation:

$$G = Qs*Qv*H+Qv*s+V$$

Where Qs is quantized series of S and Qv is he quantized series of V. And now by setting Qs = Qv= 3, then G = 9H + 3S + V.

$$H = \begin{cases} 0 \text{ if } h \in [316,20] \\ 1 \text{ if } h \in [21,40] \\ 2 \text{ if } h \in [41,75] \\ 3 \text{ if } h \in [76,155] \\ 4 \text{ if } h \in [156,190] \\ 5 \text{ if } h \in [191,270] \\ 6 \text{ if } h \in [271,295] \\ 7 \text{ if } h \in [296,315] \end{cases} \qquad S = \begin{cases} 0 \text{ if } s \in [0,0.2) \\ 1 \text{ if } s \in [0.2,0.7) \\ 2 \text{ if } s \in [0.7,1) \end{cases} \qquad V = \begin{cases} 0 \text{ if } v \in [0,0.2) \\ 1 \text{ if } v \in [0.2,0.7) \\ 2 \text{ if } v \in [0.7,1) \end{cases}$$

In this way three component vector of the HSV from one dimensional vector, Which quantize the whole color space for the 72 kinds of the main colors. So we can handle 72 bins of one dimensional histogram. This qualification is effective in reducing the images by the effect of the light intensity, but also reducing the computational time and complexity.

IMAGE RETRIEVAL:

Image retrieval is nothing but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords or the descriptions to the images so that the retrieval can be performed. Manual image annotation is time consuming, expensive and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc..,

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others. one can today generate volumes of images with content as diverse as family get-togethers and national park visits. Low-cost storage and easy Web hosting has fueled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchable image data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real-world image search system designers.

As far as technological advances are concerned, growth in Content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real world implications, applications, and constraints of the technology. Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future.

An image retrieval system designed to serve a personal collection should focus on features such as personalization, flexibility of browsing, and display methodology. For example, Google’s Picasa system [Picasa 2004] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods. The notable few applications include an FPGA implementation of a color-histogram-based image retrieval system [Kotoulas and Andreadis 2003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 2003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 2002].

Discussion. Regardless of the nature of the collection, as the expected user-base grows, factors such as concurrent query support, efficient caching, and parallel and distributed processing of requests become critical. For future real-world image retrieval systems, both software and hardware approaches to address these issues are essential. More realistically, dedicated specialized servers, optimized memory and storage support, and highly parallelizable image search algorithms to exploit cluster computing powers are where the future of large-scale image search hardware support lies.

OVERVIEW OF TEXTURE:

We all know about the term Texture but for defining it is a hard time. One can differentiate the two different Textures by recognizing the similarities and differences. Commonly there are three ways for the usage of the Textures:

Based on the Textures the images can be segmented To differentiate between already segmented regions or to classify them.We can reproduce Textures by producing the descriptions. The texture can be analyzed in three different ways. They are Spectral, Structural and Statistical:

CHAPTER-4

DIGITAL IMAGE PROCESSING

Digital image processing

Background:

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An important characteristic underlying the design of image processing systems is the significant level of testing & experimentation that normally is required before arriving at an acceptable solution. This characteristic implies that the ability to formulate approaches &quickly prototype candidate solutions generally plays a major role in reducing the cost & time required to arrive at a viable system implementation.

What is DIP

An image may be defined as a two-dimensional function f(x, y), where x & y are spatial coordinates, & the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital image by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensor, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis& computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting

& somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low-level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low- level process is characterized by the fact that both its inputs & outputs are images.

Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images. Finally higher- level processing involves "Making sense" of an ensemble of recognized objects, as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

What is an image?

An image is represented as a two dimensional function f(x, y) where x and y are spatial co-ordinates and the amplitude of 'f' at any pair of coordinates (x, y) is called the intensity of the image at that point.

Gray scale image:

A grayscale image is a function I(xylem) of the two spatial coordinates of the image plane.

I(x, y) is the intensity of the image at the point (x, y) on the image plane.

I (xylem)takes non-negative values assume the image is bounded by arectangle[0, a] ×[0, b]I: [0, a] × [0, b] → [0, info)

Color image:

It can be represented by three functions, R (xylem)for red,G (xylem)for green *and*B (xylem)for blue.

An image may be continuous with respect to the x and y coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate’s values is calledsampling. Digitizing the amplitude values is called quantization.

Coordinate convention:

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image f(x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size M X N. The values of the coordinates (xylem) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates.

In many image processing books, the image origin is defined to be at (xylem)=(0,0).The next coordinate values along the first row of the image are (xylem)=(0,1).It is important to keep in mind that the notation (0,1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Following figure shows the coordinate convention. Note that x ranges from 0 to M-1 and y from 0 to N-1 in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (xylem) the toolbox uses the notation (race) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate topples, (alb), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentation refers to the coordinates. Less frequently the toolbox also employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refers to rows. This is the opposite of our use of variables x and y.

Image as Matrices:

The preceding discussion leads to the following representation for a digitized image function:

f (0,0)   f(0,1)   ………..   f(0,N-1)

f (1,0)   f(1,1)   …………   f(1,N-1)

f (xylem)=   .   .   .

.   .   .

f (M-1,0)   f(M-1,1).................f(M-1,N-1)

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements.A digital image can be represented naturally as a MATLAB matrix:

f (1,1)   f(1,2) .... f(1,N)

f (2,1)   f(2,2) .... f (2,N)

.   .   .

f =   .   .   .

f (M,1)   f(M,2).......... f(M,N)

Where f (1,1) = f(0,0) (note the use of a monoscope font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation f(p ,q) denotes the element located in row p and the column q. For example f(6,2) is the element in the sixth row and second column of the matrix f. Typically we use the letters M and N respectively to denote the number of rows and columns in a matrix. A 1xN matrix is called a row vector whereas an Mx1 matrix is called a column vector. A 1x1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using mono-scope characters. We use conventional Roman, italic notation such as f(x ,y), for mathematical expressions

Reading Images:

Images are read into the MATLAB environment using function imread whose syntax is

Imread (‘filename’)

Format name	Description	recognized extension
TIFF	Tagged Image File Format	.tif, .tiff
JPEG	Joint Photograph Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format	.gif
BMP	Windows Bitmap	.bmp
PNG	. Portable Network Graphics	.png
XWD	X Window Dump	xwd

.

Here filename is a spring containing the complete of the image file(including any applicable extension).For example the command line

>> f = imread (‘8. jpg’);

Reads the JPEG (above table) image chestxray into image array f. Note the use of single quotes (‘) to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output If a semicolon is not included. MATLAB displays the results of the operation(s) specified in that line. The prompt symbol (>>) designates the beginning of a command line, as it appears in the MATLAB command window.

Data Classes:

Although we work with integers coordinates the values of pixels themselves are not restricted to be integers in MATLAB. Table above list various data classes supported by MATLAB and IPT are representing pixels values. The first eight entries in the table are refers to as numeric data classes. The ninth entry is the char class and, as shown, the last entry is referred to as logical data class.

All numeric computations in MATLAB are done in double quantities, so this is also a frequent data class encounter in image processing applications. Class unit 8 also is encountered frequently, especially when reading data from storages devices, as 8 bit images are most common representations found in practice. These two data classes, classes logical, and, to a lesser degree, class unit 16 constitute the primary data classes on which we focus. Many ipt functions however support all the data classes listed in table. Data class double requires 8 bytes to represent a number uint8 and int 8 require one byte each, uint16 and int16 requires 2bytes and unit 32.

Name Description

Double	Double precision, floating point numbers are approximate
Unit 8	Unsigned 8 bit integers in the range [0,255] (1 byte per element)
Unit 16	Unsigned 16 bit integers in the range [0, 65535] (2 bytes per element)
Unit 32	Unsigned 32 bit integers in the range [0, 4294967295] (4 bytes per element).
Int 8	Signed 8_bit integers in the range [-128,127] (1 byte per element)
Int 16	Signed 16_byte integers in the range [32768, 32767] (2 bytes per element).
Int 32	Signed 32_byte integers in the range [-2147483648, 21474833647] (4 byte per element).

Single single _precision floating _point numbers with values

In the approximate range (4 bytes per elements)

Char characters (2 bytes per elements).

Logical values are 0 to 1 (1byte per element).

Int 32 and single required 4 bytes each. The char data class holds characters in Unicode representation. A character string is merely a 1*n array of characters logical array contains only the values 0 to 1,with each element being stored in memory using function logical or by using relational operators.

Image Types:

The toolbox supports four types of images:

1 .Intensity images;

2. Binary images;

3. Indexed images;

4. R G B images.

Most monochrome image processing operations are carried out using binary or intensity images, so our initial focus is on these two image types. Indexed and RGB colour images.

Intensity Images:

An intensity image is a data matrix whose values have been scaled to represent intentions. When the elements of an intensity image are of class unit8, or class unit 16, they have integer values in the range [0,255] and [0, 65535], respectively. If the image is of class double, the values are floating point numbers. Values of scaled, double intensity images are in the range [0, 1] by convention.

Binary Images:

Binary images have a very specific meaning in MATLAB.A binary image is a logical array 0s and1s.Thus, an array of 0s and 1s whose values are of data class, say unit8, is not considered as a binary image in MATLAB .A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, we create an array B using the statement.

B=logical (A)

If A contains elements other than 0s and 1s.Use of the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.

Using relational and logical operators also creates logical arrays.

To test if an array is logical we use the I logical function: islogical(c).

If c is a logical array, this function returns a 1.Otherwise returns a 0. Logical array can be converted to numeric arrays using the data class conversion functions.

Indexed Images:

An indexed image has two components:

A data matrix integer, x

A color map matrix, map

Matrix map is an m*3 arrays of class double containing floating point values in the range [0, 1].The length m of the map are equal to the number of colors it defines. Each row of map specifies the red, green and blue components of a single color. An indexed images uses “direct mapping” of pixel intensity values color map values. The color of each pixel is determined by using the corresponding value the integer matrix x as a pointer in to map. If x is of class double ,then all of its components with values less than or equal to 1 point to the first row in map, all components with value 2 point to the second row and so on. If x is of class units or unit 16, then all components value 0 point to the first row in map, all components with value 1 point to the second and so on.

RGB Image:

An RGB color image is an M*N*3 array of color pixels where each color pixel is triplet corresponding to the red, green and blue components of an RGB image, at a specific spatial location. An RGB image may be viewed as “stack” of three gray scale images that when fed in to the red, green and blue inputs of a color monitor

Produce a color image on the screen. Convention the three images forming an RGB color image are referred to as the red, green and blue components images. The data class of the components images determines their range of values. If an RGB image is of class double the range of values is [0, 1].

Similarly the range of values is [0,255] or [0, 65535].For RGB images of class units or unit 16 respectively. The number of bits use to represents the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8bit image, the corresponding RGB image is said to be 24 bits deep.

Generally, the number of bits in all component images is the same. In this case the number of possible color in an RGB image is (2^b) ^3, where b is a number of bits in each component image. For the 8bit case the number is 16,777,216 colors

CHAPTER 5

Software Introduction:

5.1. Introduction to MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

5.2 The MATLAB system:

The MATLAB system consists of five main parts

- **Development Environment**:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library**:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language**:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

- **Graphics:**

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

  This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

5.3 GRAPHICAL USER INTERFACE (GUI):

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

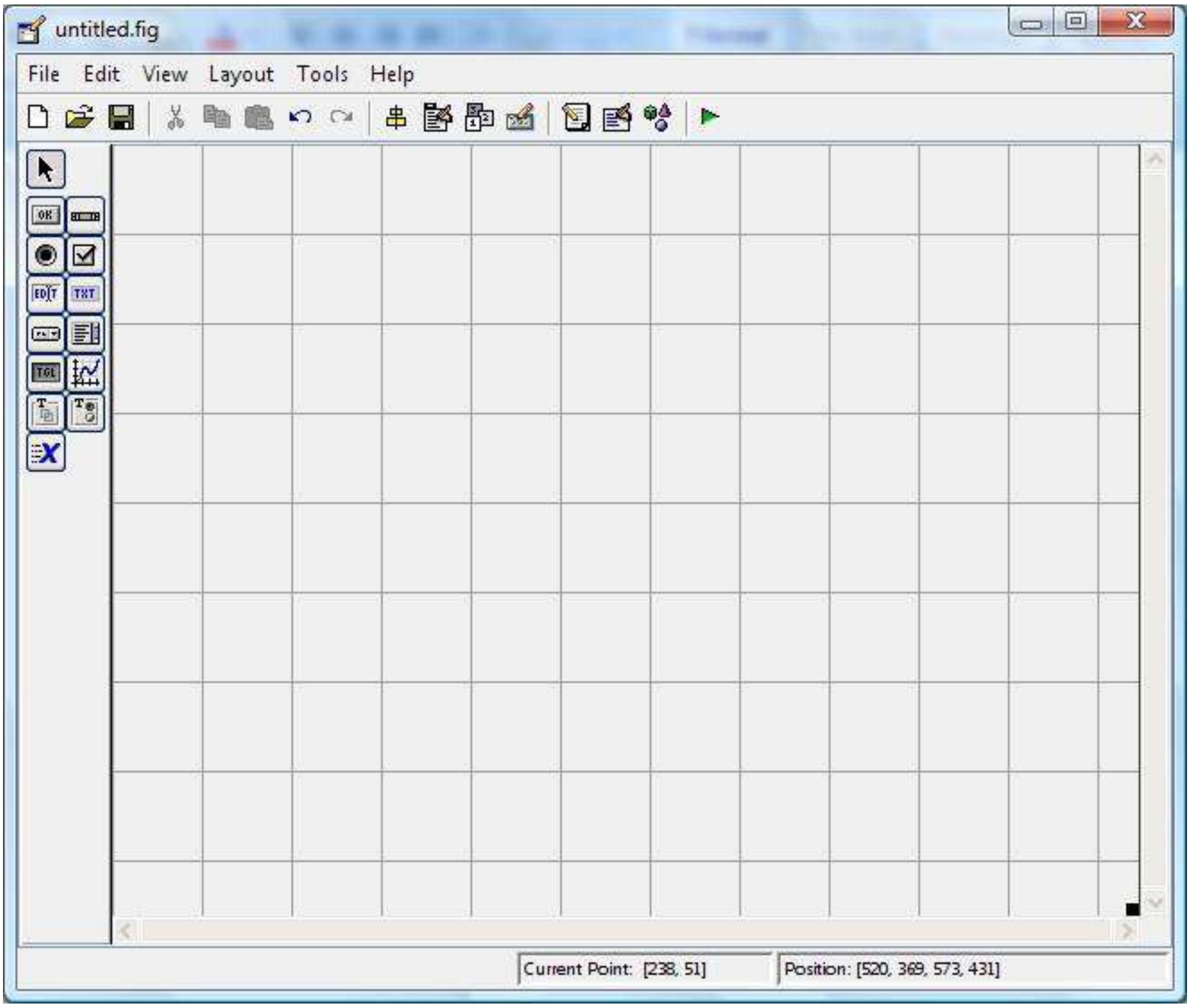
- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.
- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type

guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted,

GUIDE opens a new (i.e., blank) window.



A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

5.4 Getting Started

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator,

and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

5.4.1 Introduction - describes the components of the MATLAB system.

5.4.2 Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

5.4.3 Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

5.4.4 Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

5.4.5 Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

5.5 DEVELOPMENT ENVIRONMENT

5.5.1Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

5.5.2 Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.On a UNIX platform, to start MATLAB, type

Matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

5.5.3 Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

5.5.4 MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

5.5.5 Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window

- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacsmagik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type helpbrowser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to runMATLAB functions. The Editor/Debugger provides a graphical user interface for basic textediting, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

5.6 MANIPULATING MATRICES

5.6.1 Entering Matrices

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Dürer's matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]

MATLAB displays the matrix you just entered.

```
A =  16    3    2   13

      5   10   11    8

      9    6    7   12

      4   15   14    1
```

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

5.6.2 Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

5.6.3 Operators

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

5.6.4 Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.14159265...
I	Imaginary unit, $\sqrt{-1}$
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

5.7 GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

5.7.1 Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

5.7.2 GUI Development Environment

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of its

children (uicontrols and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the callbacks, which are defined as subfunctions. This M-file is referred to as the

application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

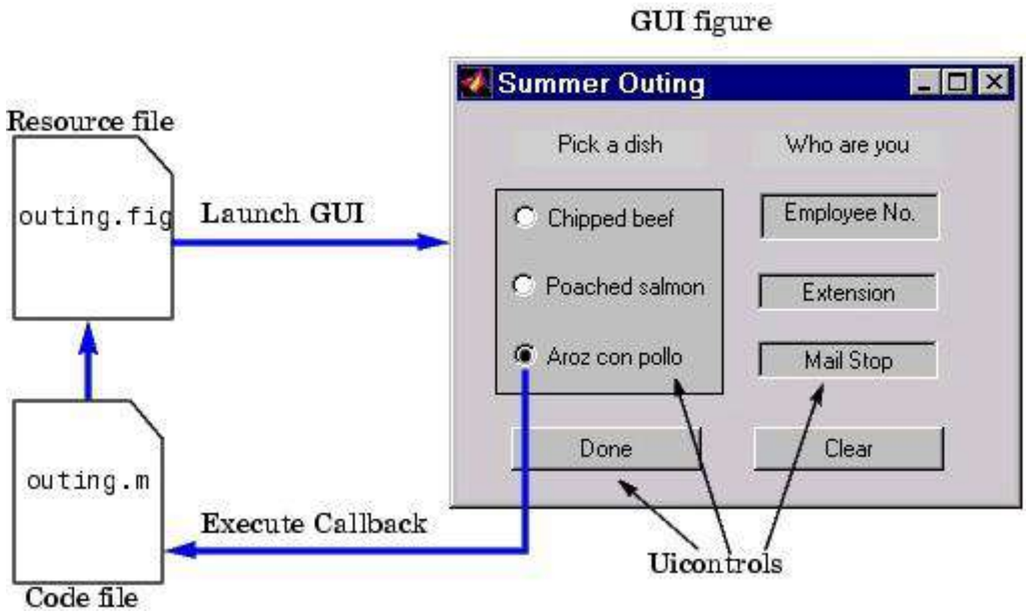


FIG 3.7.2 graphical user blocks

5.7.3 Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted sub-function prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

5.7.4 Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques

used in the application M-file.

Application Examples - a collection of examples that illustrate techniques

which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

5.7.5 User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- Listboxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)

button_state = get(h,'Value');

if button_state == get(h,'Max')

   % toggle button is pressed

elseif button_state == get(h,'Min')

   % toggle button is not pressed

end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);

a(:,:,2) = rand(16,128);
```

```
a(:,:,3) = rand(16,128);

set(h,'CData',a)
```

Radio Buttons

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)

set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)

off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];

mutual_exclude(off)

% Continue with callback
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)

if (get(h,'Value') == get(h,'Max'))

   % then checkbox is checked-takeapproriate action

else

   % checkbox is not checked-takeapproriate action

End
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)

user_string = get(h,'string');

% proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)

user_entry = str2double(get(h,'string'));

if isnan(user_entry)

errordlg('You must enter a numeric value','BadInput','modal')

end

% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menubar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menubar does not cause the editable text callback routine to execute.

This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

Listboxes differentiate between single and double clicks on an item and set the figure SelectionType property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections.In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure SelectionType property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

switch val

case 1

% The user selected the first item

case 2

% The user selected the second item

% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

string_list = get(h,'String');

selected_string = string_list{val}; % convert from cell array to string

% etc.
```

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uicontrolobjects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes ButtonDownFcn property to define the callback.

5.7.6 Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

```
axes(handles.axes1)
```

makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axes. See GUI with Multiple Axes for and example that uses two axes.

Figure

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control.

Chapter 6

RELATED WORK

we will first review some background knowledge of image matting, and then briefly introduce vessel enhancement filters used in our work.

A. Image Matting

As aforementioned, image matting aims to accurately extract the foreground given a trimap of an image. Specifically, the input image I(z)(z = (x, y)) is modeled as a linear combination of a foreground image F(z) and a background image B(z):

$$I(z) = \alpha_z F(z) + (1 - \alpha_z) B(z) \qquad (1)$$

where αz, called alpha matte, is the opacity of the foreground. αz ranges from 0 to 1. If αz is constrained to be either 0 or 1, then the matting problem becomes the segmentation problem, where each pixel belongs to either foreground or background.

After obtaining the user specified trimap, to infer the alpha matte in the unknown regions, Chuang et al. uses sets of Gaussian distribution to model the color distributions of the foreground and background images, and estimates the optimal alpha value with a maximum-likelihood criterion. In , Levin et al. derives an effective cost function from the assumption that the foreground and background colors are locally smooth, and employs this function to find the optimal alpha matte. Zheng et al. proposes a local learning based approach and a global learning based approach to perform image matting. In ,Kaiming et al. solves a large kernel matting Laplacian, and achieves a fast matting algorithm. In ,Shahrian and Rajan analyze the texture and color features of the image, and optimize an objective function containing the color and texture components to choose the best foreground and background pair for image matting. Shahrian et al. expands the sampling range of foreground and background regions, and collects a representative set of samples to estimate the alpha matte. In , Cho et al. presents an image matting method to extract alpha mattes across sub-images of a light field image.

B. Vessel Enhancement filters

Vessel enhancement filters plays an important role in retinal blood vessel segmentation . Here two effective filters used in our work are introduced.

1) Morphologically Reconstructed Filter: Morphologically reconstructed filter is an effective tool for blood vessel enhancement [31]. For each input fundus image I, the green channel image Ig is extracted firstly since Ig has the best vessel-background contrast [13]. Then the morphological tophat transformation is performed:

$$I_{th}^{\theta} = I_g^c - (I_g^c \circ S_e^{\theta}) \qquad (2)$$

where I c g is the complement image of Ig, I θ th represents the top-hat transformed image, Se is a structuring element for morphological opening ◦, and θ specifies the angle (in degrees) of the structuring element. The structuring element is of 1- pixel width and 21-pixels length, which approximately fits the diameter of the biggest vessels in the fundus images [31]. Since the morphological top-hat transformation transformation given in Equation (2) can only brighten blood vessels in one direction, the sum of top-hat transformation I θ th along each direction is performed in order to enhance the whole vessel image:

$$I_{mr} = \sum_{\theta \in A} I_{th}^{\theta} \qquad (3)$$

where Imr represents the enhanced vessel image using morphologically reconstructed filter, ”A” is the set of angles of the structuring element and defined as $\{x|0 < x < \pi \;\&\; x \bmod (\pi/12) = 0\}$.

2) Isotropic Undecimated Wavelet Filter: The isotropic undecimated wavelet filter has been used for blood vessel segmentation, and has a good performance on vessel enhancement [32]. Applied to a signal c0 = Ig (Ig is the input green channel image), scaling coefficients are computed by convolution with a filter h ↑j firstly:

$$c_{j+1} = c_j * h^{\uparrow j} \qquad (4)$$

where h0 = [1, 4, 6, 4, 1]/16 is derived form the cubic Bspline, h ↑j is the upsampled filter obtained by inserting 2 j −1 zeros between each pair of adjacent coefficients of h0. Wavelet coefficients are the difference between two adjacent sets of scaling coefficients, i.e.,

$$w_{j+1} = c_j - c_{j+1} \qquad (5)$$

Reconstruction of the original signal from all wavelet coefficients and the final set of scaling coefficients is straightforward, and requires only addition. The final enhanced vessel image is depicted as follows:

$$I_{iuw} = c_n + \sum_{j=1}^{n} w_j \qquad (6)$$

where Iiuw represents the enhanced vessel image using isotropic undecimated wavelet filter. In blood vessel segmentation, wavelet scales: 2 − 3 are selected according to [32].

Chapter 7

METHODOLOGY

The process of generating the trimap of an input fundus image automatically is introduced, followed by detailing the proposed hierarchical image matting model

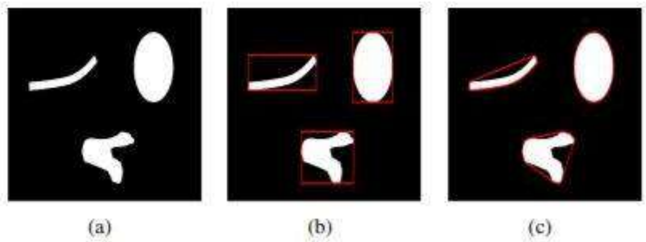


Fig. 2. An example to illustrate the bounding box and convex hull. (a) The exemplary image. (b) The image for the illustration of bounding box. The red boxes are the bounding boxes. (c) The image for the illustration of convex hull. The red polygons are the convex hulls.

A. Trimap Generation

Region features of blood vessels have been used for blood vessel segmentation and performed well on segmentation accuracy and computational efficiency [33]. In this paper, region features of blood vessels are applied to generate the trimap of the input fundus image automatically. The definitions of regions features are given as follows:

• Area indicates the actual number of pixels in the region.

• Bounding Box specifies the smallest rectangle containing the region. An example for the illustration of bounding box is shown in Fig.2(b).

• Extent is the ratio of pixels in the region to pixels in the total bounding box.

• VRatio is the ratio of the length to the width of the bounding box.

• Convex Hull specifies the smallest convex polygon that can contain the region. An example for the illustration of convex hull is shown in Fig.2(c).

• Solidity is the ratio of pixels in the region to pixels in the total convex hull.

The default threshold values of region features:Extent, VRatio, Solidity and their recommended ranges used in this work are reported in Table I. e1 and e2 are two threshold values of Extent features used in this work; r is the threshold value of V Ratio feature; s is the threshold value of Solidity feature. For Area feature, two threshold values: a1 = fi × 2 and a2 = fi×35 are used in this work. fi , called the internal factor, is calculated as d × max(h,w) min(h,w) , where d = 21 is approximately the diameter of the biggest vessels in fundus images [31], h and w are the height and width of the fundus image. The proposed model is not sensitive to above mentioned region features. In other words, these region features can be selected in a relatively large range without sacrificing the performance. In Section V(D)- ”Sensitivity analysis of threshold values of region features”, empirical study is conducted to verify the insensitivity of the proposed model to the threshold values of region features.

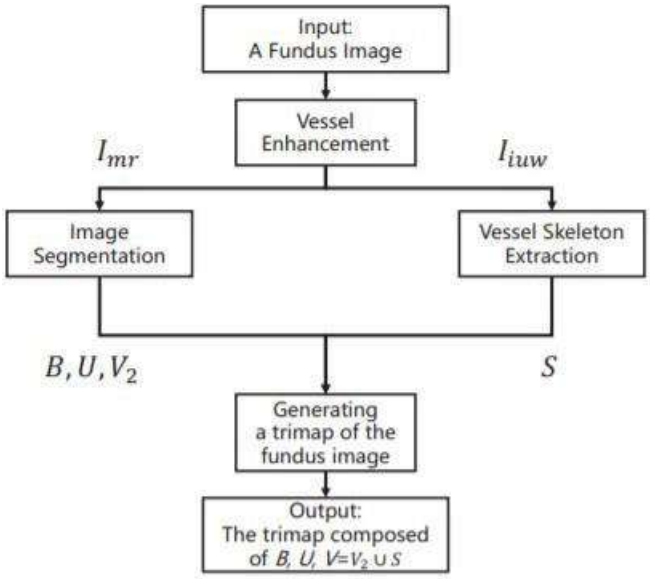


Fig. 3. The process of trimap generation. B represents the background regions; U represents the unknown regions; V2 represents the denoised preliminary vessel regions; S represents the skeleton of blood vessels; V represents the vessel regions.

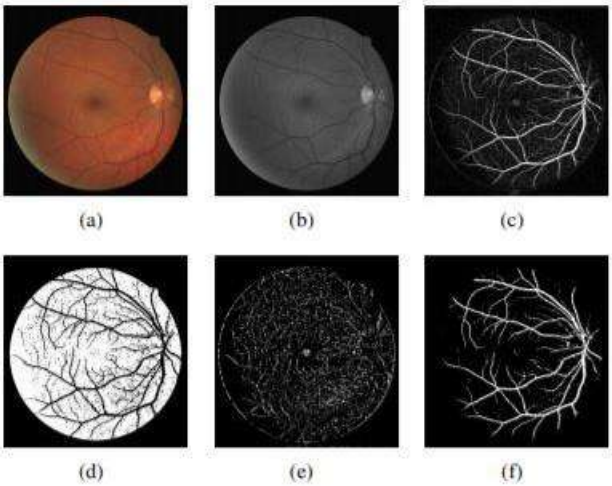


Fig. 4. Image segmentation. (a) The fundus image I. (b) The green channel of the fundus image Ig. (c) The enhanced vessel image Imr. (d) The background regions B. (e) The unknown regions U. (f) The denoised preliminary vessel regions V2

Creating the trimap of the input fundus image includes two main steps: 1) Image Segmentation and 2) Vessel Skeleton Extraction. The process of trimap generation is given in Fig.3.

1) Image Segmentation: The goal of image segmentation is to divide the input image into three regions: the vessel (foreground), background and unknown regions. Firstly the enhanced vessel image Imr is segmented into three regions: the background regions (B), unknown regions (U) and preliminary vessel regions (V1)

$$I_{mr} = \begin{cases} B & \text{if } 0 < I_{mr} < p_1 \\ U & \text{if } p_1 \leqslant I_{mr} < p_2 \\ V_1 & \text{if } p_2 \leqslant I_{mr} \end{cases} \qquad (7)$$

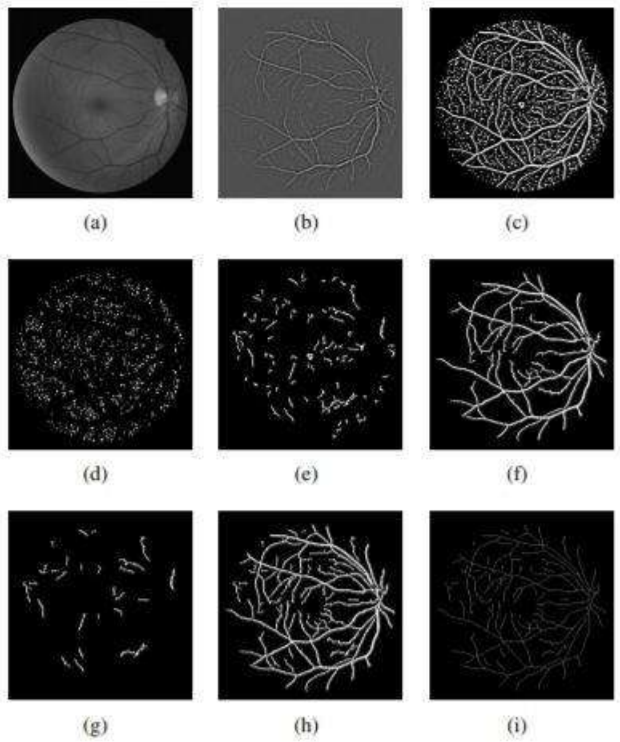


Fig. 5. Vessel skeleton extraction. (a) The green channel of the fundus image I_g. (b) The vessel enhanced image I_{iuw}. (c) The binary image T. (d) The background regions T_1. (e) The candidate regions T_2. (f) The vessel regions T_3. (g) T_4: The preserved regions in T_2. (h) The combined regions of T_3 and T_4. (i) The vessel skeletons S.

where p1 = 0.2 and p2 = 0.35 restrict the unknown region as thin as possible in order to achieve the better matting result [28], [34]. In order to remove the noise regions in V_1, the regions with Area > a1 in V_1 are extracted firstly (V ∗1). Then regions in V ∗ 1 whose Extent ≤ e1 && V Ratio ≤ r && Solidity ≥ s are abandoned, resulting in the denoised preliminary vessel regions V_2. An example of image segmentation is shown in Fig.4.

2) Vessel Skeleton Extraction: Vessel Skeleton Extraction aims to further distinguish the unknown regions and provide more information on blood vessels. In Section V(B)-"Vessel Segmentation Performance", the effectiveness of vessel skeleton extraction will be presented. Firstly, a binary image T is obtained by global thresholding the enhanced vessel image I_{iuw}.

$$T = \begin{cases} 1 & I_{iuw} > t \\ 0 & I_{iuw} \leqslant t \end{cases} \quad (8)$$

where $t = O_{tsu}(I_{iuw}) - \varepsilon$, ε is set as 0.03. Then T is divided into three regions-according to the Area feature

$$T = \begin{cases} T_1 & \text{if } 0 < Area < a_1 \\ T_2 & \text{if } a_1 \leq Area \leq a_2 \\ T_3 & \text{if } a_2 < Area \end{cases} \quad (9)$$

In vessel skeleton extraction, the regions in T_3 are preserved while the regions in T1 are abandoned. Then the regions in T_2 with Extent > e_2 && V Ratio ≤ r are preserved as T_4. Finally skeleton extraction [35] is performed on the combined regions of T_3 and T_4 in order to obtain the skeleton of blood

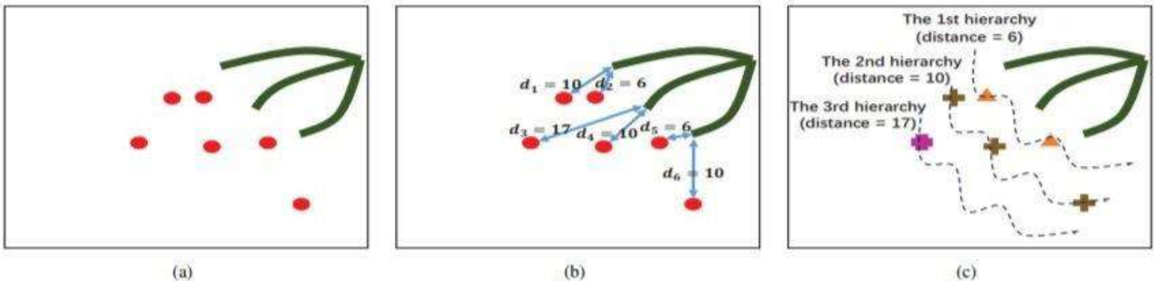


Fig. 6. An example to illustrate the process of initialization. (a) An exemplary image (green pixels represent vessel pixels, red pixels represent unknown pixels). (b) Calculating the closest distance for each unknown pixel (di means the closest distance for the ith unknown pixel). (c) Stratifying unknown pixels into different hierarchies.

vessels S. An example of vessel skeleton extraction is shown in Fig.5. After performing image segmentation and vessel skeleton extraction, the trimap of the input fundus image is generated (as shown in Fig.7(b)), which is composed of the background regions (B), unknown regions (U) and vessel (or foreground) regions ($V = V_2 \cup S$).

B. Hierarchical Image Matting

Model Hierarchical image matting model is proposed to label the pixels in the unknown regions as vessels or background in an incremental way. Specifically, after stratifying the pixels in unknown regions (called unknown pixels) into m hierarchies by a hierarchical strategy, let u j i indicates the ith unknown pixel in the jth hierarchy, the segmented vessel image Iv(u j i) is modeled as follows:

$$I_{mr} = \begin{cases} B & \text{if } 0 < I_{mr} < p_1 \\ U & \text{if } p_1 \leqslant I_{mr} < p_2 \\ V_1 & \text{if } p_2 \leqslant I_{mr} \end{cases} \tag{7}$$

where corre indicates the correlation function (depicted in Equation (13)). The implementation of the hierarchical image matting model consists of two main steps: Step 1 Stratifying the unknown pixels: Stratify pixels in the unknown regions into different hierarchies. Step 2 Hierarchical update: Assign new labels (V or B) to pixels in each hierarchy. The pseudo-code of implementing this model is given in Algorithm

Chapter 8

CONCLUSION

Image matting refers to the problem of accurately extracting a foreground object from an input image, which is very useful in many important applications. However, to the best of our knowledge, image matting has never been employed before to extract blood vessels from fundus image. The major reason is that for retinal blood vessel segmentation, generating a user specified trimap is a tedious and time-consuming task. In addition, a normal image matting model needs to be designed carefully to improve the performance of blood vessel segmentation. In order to overcome these problems, region features of blood vessels are applied to generate the trimap automatically. Then a hierarchical image matting model is proposed to extract the pixels of blood vessel from the unknown regions. More specifically, a hierarchical strategy utilizing the continuity and extendibility of retinal blood vessels is integrated into the image matting model for blood vessel segmentation. The proposed model is very efficient and effective in blood vessel segmentation, which achieves a segmentation accuracy of 96.0%, 95.7% and 95.1% on three public available datasets with an average time of 10.72s, 15.74s and 50.71s, respectively. The experimental results show that it is a very competitive model compared with many other segmentation approaches, and has a low computational time

REFERENCES

[1] J. J. Kanski and B. Bowling, Clinical Ophthalmology: A Systematic Approach. Elsevier Health Sciences, 2011.

[2] F. Zana and J.-C. Klein, “A multimodal registration algorithm of eye fundus images using vessels detection and hough transform,” IEEE Transactions on Medical Imaging, vol. 18, no. 5, pp. 419–428, 1999.

[3] M. Foracchia, E. Grisan, and A. Ruggeri, “Extraction and quantitative description of vessel features in hypertensive retinopathy fundus images,” in Book Abstracts 2nd International Workshop on Computer Assisted Fundus Image Analysis, vol. 6, 2001.

[4] C. Heneghan, J. Flynn, M. OKeefe, and M. Cahill, “Characterization of changes in blood vessel width and tortuosity in retinopathy of prematurity using image analysis,” Medical Image Analysis, vol. 6, no. 4, pp. 407–429, 2002.

[5] E. Grisan and A. Ruggeri, “A divide et impera strategy for automatic classification of retinal vessels into arteries and veins,” in Engineering in Medicine and Biology Society, 2003. Proceedings of The 25th Annual International Conference of The IEEE, vol. 1, 2003, pp. 890–893.

[6] K. Fritzsche, A. Can, H. Shen, C. Tsai, J. Turner, H. Tanenbuam, C. Stewart, B. Roysam, J. Suri, and S. Laxminarayan, “Automated model based segmentation, tracing and analysis of retinal vasculature from digital fundus images,” State-of-The-Art Angiography, Applications and Plaque Imaging Using MR, CT, Ultrasound and X-rays, pp. 225–298, 2003.

[7] C. Marino, M. G. Penedo, M. Penas, M. J. Carreira, and F. Gonzalez, ˜ “Personal authentication using digital retinal images,” Pattern Analysis and Applications, vol. 9, no. 1, pp. 21–33, 2006.

[8] A. Haddouche, M. Adel, M. Rasigni, J. Conrath, and S. Bourennane, “Detection of the foveal avascular zone on retinal angiograms using markov random fields,” Digital Signal Processing, vol. 20, no. 1, pp. 149–154, 2010.

[9] M. M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvara, A. R. Rudnicka, C. G. Owen, and S. A. Barman, “Blood vessel segmentation methodologies in retinal images–a survey,” Computer Methods and Programs In Biomedicine, vol. 108, no. 1, pp. 407–433, 2012.

[10] J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. van ` Ginneken, “Ridge-based vessel segmentation in color images of the retina,” IEEE Transactions on Medical Imaging, vol. 23, no. 4, pp. 501– 509, 2004.

[11] J. V. Soares, J. J. Leandro, R. M. Cesar, H. F. Jelinek, and M. J. Cree, “Retinal vessel segmentation using the 2-d gabor wavelet and supervised classification,” IEEE Transactions on Medical Imaging, vol. 25, no. 9, pp. 1214–1222, 2006.

PROJECT REPORT

On

BREAST CANCER DETECTION FROM HISTOPATHOLOGICAL IMAGES USING DEEP LEARNING

Submitted by

1)Manasseh john wesley (17K81A04K4)

2)J.Sunith (17K81A04L0)

3)v.Sai Yashwanth Varda (17K81A04P8)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATIONS ENGINEERING

Under The Guidance of

Mr.K.Balasubramanyam

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATIONS ENGINEERING

ST.MARTIN'S ENGINEERING COLLEGE

(An Autonomous Institute)

Dhulapally, Secunderabad – 500 100

JUNE 2021

BONAFIDE CERTIFICATE

This is to certify that the project entitled Breast cancer detection, is being submitted by 1. **Manasseh john wesley (17K81A04K4), 2.J.Sunith(17K81A04L0), 3.V.Sai yashwanth varda (17K81A04P8)** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONCS AND COMMUNICATIONS ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Signature	Head of the Department
MR.K.BALASUBRAMANYAM	**Dr.B.HARI KRISHNA**
Department of ECE	Department of ECE

Internal Examiner External Examiner

Place:

Date:

TUESDAY, 15 JUNE 2021

INTERNSHIP CERTIFICATE

THIS IS TO CERTIFY THAT **GADI MANASSEH JOHN WESLEY** WITH ROLL NO.**17K81A04K4, JADAV SUNITH** WITH ROLL NO.**17K81A04L0**, **VENKATA SAI YASHWANTH VARDA** WITH ROLL NO.**17K81A04P8,** OF B. TECH – IV YEAR, **ELECTRONICS AND COMMUNICATIONS ENGINEERING DEPARTMENT** OF **ST. MARTIN'S ENGINEERING COLLEGE**, KOMPALLY, SECUNDERABAD HAVE COMPLETED ONE MONTH INTERNSHIP PROGRAM AT **LASYA IT SOLUTION PVT. LTD, KOMPALLY.**

DURING THE PERIOD, THEY HAVE SUCCESSFULLY COMPLETED MAJOR PROJECT TITLED "**BREAST CANCER DETECTION FROM HISTOPATHOLOGICAL IMAGES USING DEEP LEARNING**" AT OUR DEVELOPMENT CENTER, KOMPALLY.

WE WISH THEM SUCCESS IN THEIR FUTURE ENDEVOUR

Lasya IT Solutions Pvt. Ltd. Kompally-500014.

ORUGANTI VENKAT
DIRECTOR
TRAININGS & PLACEMENTS
LASYA IT SOLUTIONS PVT LTD

Lasya IT Solutions Pvt Ltd, Behind Cine Planet, Kompally, Medchal Road, Secunderabad 500014
Email : contact@lasyainfotech.com, ov@lasyainfotech.com
Website : www.lasyainfotech.com | contact: 7330666881/82/83/84/86

DECLARATION

We, the student of **Bachelor of Technology** in Department of ELECTRONICS AND COMMUNICATIONS ENGINEERING', session: <2017 – 2021>, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad, hereby declare that work presented in this Project Work entitled "BREAST CANCER DETECTION FROM HISTOPATHALOGICAL IMAGES USING DEEP LEARNING" is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

Manasseh John wesley	**J.Sunith**	**V.Sai yashwanth varda**
(17K81A04K4)	**(17K81A04L0)**	**(17K81A04P8)**

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragements and guidance have crowded effects with success.

First and foremost, we would like to express my deep sense of gratitude and indebtedness to **our College Management** for their kind support and permission to use the facilities available in the Institute.

We especially would like to express my deep sense of gratitude and indebtedness to **Dr.P.Santosh Kumar Patra**, **our beloved Principal, St. Martin's Engineering College** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

We would like to express our gratitude **to Dr.B.Hari krishna** sir, **Head &Professor**,Dept. of Electronics and Communication Engineering for his advice, supervision, and the vital contribution as and when required during this research. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him/her. The time spent with him will remain in my memory for years to come.

We would like to express our gratitude to **our dept. project coordinator Ms.P.Pushpa** Ma'am for her involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come.

We sincerely thank our project internal guide **Mr.K.Balasubramanyam Assistant professor**, Department of ECE and our class project coordinator **Ms.P.Pushpa Assistant professor**, Department of ECE for guidance and encouragement in carrying out this project work.

Finally, we are highly grateful and obliged to our college level Project guide **Mr.K.Balasubramanyam Assistant professor** Department of ECE for his advice, supervision, and the vital contribution as and when required during this research work and for their support and co- operation that is difficult to express in words.

1. Manasseh John wesley (17K81A04K4)
2. J.Sunith (17K81A04L0)
3 V.Sai yashwanth varda (17K81A04P8)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Glossary Of commonly Used Terms

Mias - The mammographic image analysis society

Mri - magnetic resonance imaging

Ct - computed tomography

Exif - Exchangeable image file format

Tiff -(Tagged Image File Format

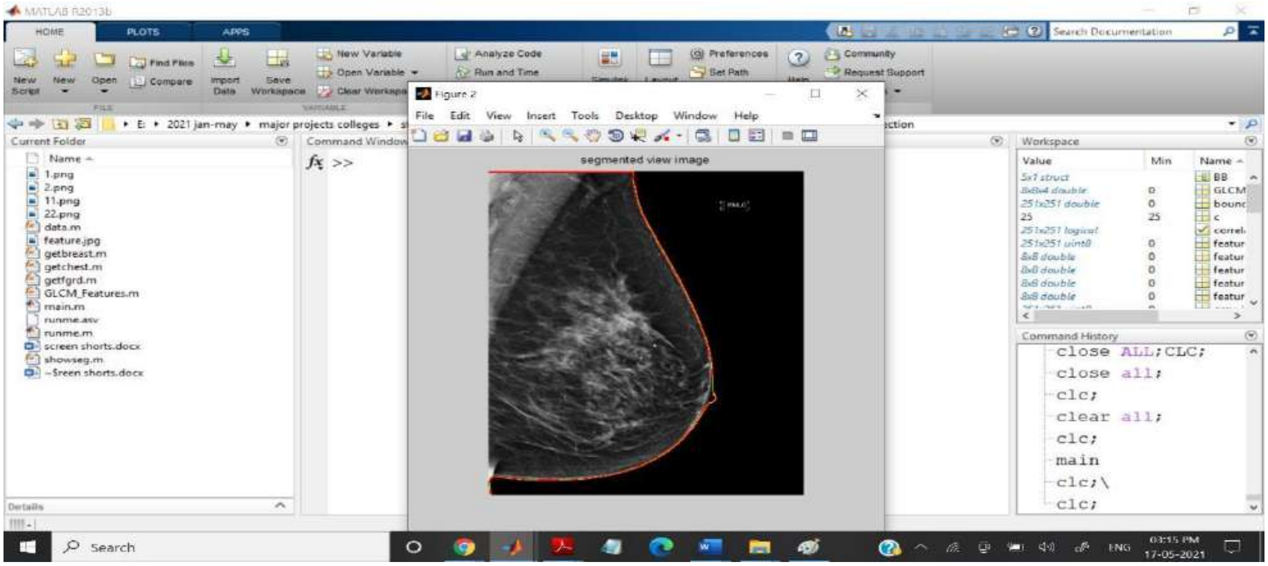
Jpeg - JPEG (Joint Photographic Experts Group)

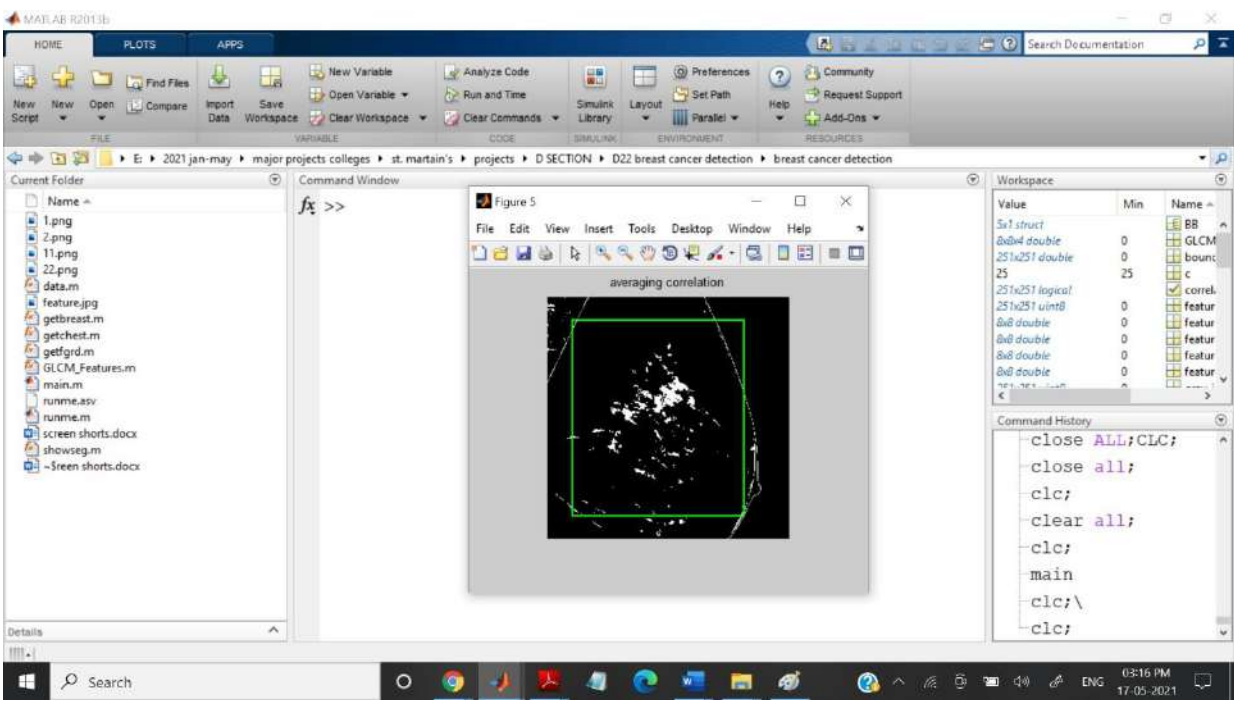
// Abstract

Cancer is the most dangerous disease in the world, it's mainly effective for women. So, our prime target must be curing cancer through scientific investigation and the second main target should be early detection of cancer because the early detection of cancer can be helpful for removing the cancer completely. After reviewing 41 papers we found that several techniques are available for cancer detection. In this paper we proposed Deep Learning algorithm convolutional neural network for diagnosed breast cancer using Mammography MIAS database. The paper shows how we can use deep learning technology for diagnosis of breast cancer using MIAS Dataset. Because deep learning techniques are almost used for high task objective Computer Vision, Image processing, Medical Diagnosis, Neural Language Processing. But in this paper, we are applying deep learning technology on the MIAS Database and we have seen that it is very beneficial for us to diagnose breast cancer with accuracy 98%. This paper is divided into three parts: first we have to collect a dataset and apply a pre-processing algorithm for scaled and filtered data, then we split the dataset for training and testing purposes and generate some graphs for visualization data. I last implemented a model on a training dataset and achieved accuracy 98%. So, we have seen that deep learning technology is a good way to diagnose breast cancer with the MIAS Dataset. This database provides 200 images and 12 features in the dataset. In this paper we have used 12 features for diagnosing breast cancer that we have got after preprocessing. But before the train model we have applied some preprocessing algorithms like Watershed Segmentation, Colour based segmentation and Adaptive Mean Filter For scaled dataset then applied model and achieved accuracy. In this paper we also compare deep learning algorithms with other machine learning and see our proposed system is proved best from others machine learning algorithms.

Keyboards–Deep Learning, Convolutional Neural Network,Neural Network, Random Forest, Support Vector Machine, MachineLearning,MIASDataset.

LIST OF OUTPUT SCREENS:





CHAPTER 1

INTRODUCTION

Breast Cancer is the most common female cancer worldwide. Almost 25% of all cancers with an estimated 1.67 million new cancer cases diagnosed in 2012. Breast cancer is the most common type of cancer among women and its incidence is increasing day by days. The life time risk of developing breast cancer in women is approximately 1/8 in USA, 1/ 12 in Europe, 1/40 in Asia (WHO 2008). As per the cancerindia.org one woman dies of cervical cancer every 8 minutes in India, around 2.5 million number of people living with cancer in India and over 7 lakh new cancer patients registered in every year. India also breast cancer is the most common cancer in women and account for 27% of all cancers. As per the previous study 1,44,937 new cases registered and deaths 70,218 in 2012. But in India the incidence rates begin to rise in the early thirties and peak at ages 50-60 years. According to the World Health Organization breast cancer was responsible for 502,000 deaths in 2005 alone and 1,301,867 of new cases of breast cancer resisted. Health case authorities continuously doing efforts to overcome this merciless disease in which one of the efforts is screening. By screening the breast cancer can be detected in early stages and thus the treatment can be more effective. Many other methods also available such as mammography, ultrasound, CT and MRI. Mammography is the most widely used screening method. In our proposed method we are using Break His[A1] dataset that provided Biopsy Images generated by Mammography method

1.1 IMAGE:

An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Fig 1 General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

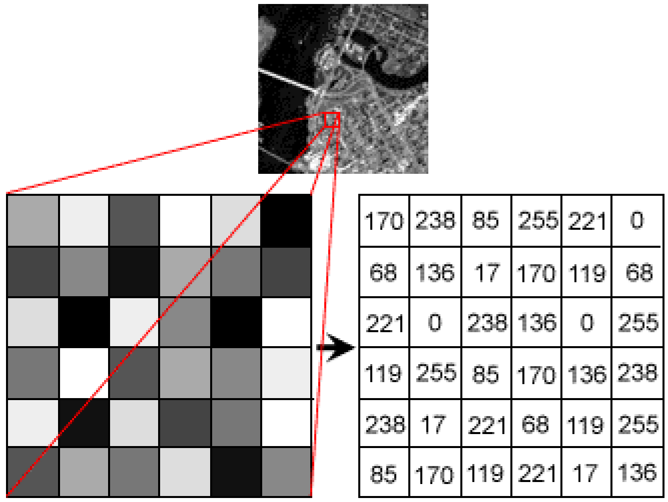


Fig 1.1 Image pixel

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.

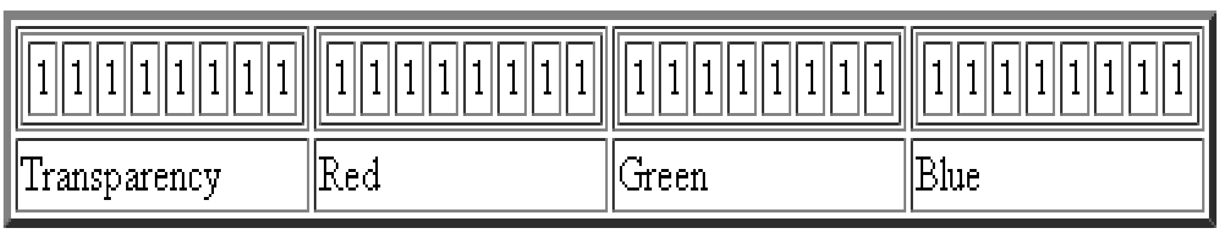


Fig1.2 Transparency image

1.2 IMAGE FILE SIZES:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

1.3 IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

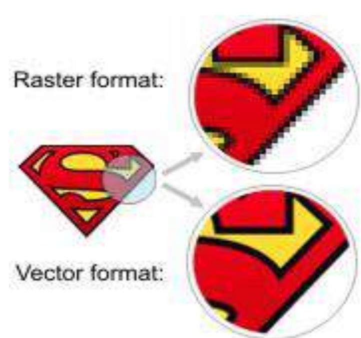


Fig1.3 Resolution image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

1.3.1 RASTER FORMATS:

These formats store images as bitmaps (also known as pixmaps)

- **JPEG/JFIF:**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

- **EXIF:**

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

- **TIFF:**

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

PNG:

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lossy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

- **GIF:**

GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

- **BMP:**

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

1.3.2 VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM:**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

- **SVG:**

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

1.4 image processing:

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, mis-connect ions, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

1.5 fundamental steps in digital image processing:

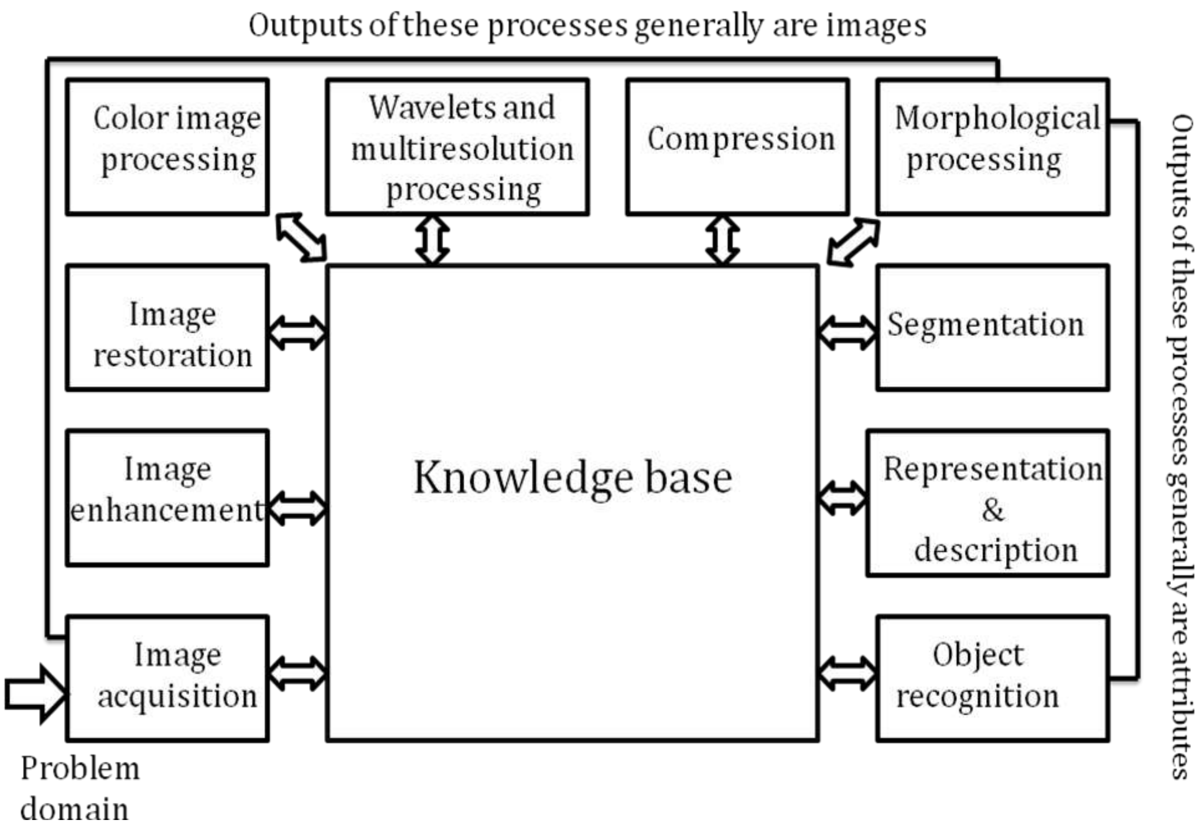


Fig 1.5 Image fundamental

1.5.1 Image Acquisition:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

Fig 1.5.1 Digital camera image

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.

Fig 1.5.2 digital camera cell

1.5.2 Image Enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because “it looks better.” It is important to keep in mind that enhancement is a very subjective area of image processing.

Fig 1.5.3 Image enhancement

1.5.3 Image restoration:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Fig 1.5.4 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a "good" enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, whereas removal of image blur by applying a deblurring function is considered a restoration technique.

1.5.4 Color image processing:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.

Fig 1.5.5 Color & Gray scale image

1.5.5 Wavelets and multiresolution processing:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform based image processing since the late1950's, a more recent transformation, called the wavelet transform, and is now making it even easier to compress, transmit, and analyze many images. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small values, called Wavelets, of varying frequency and limited duration.

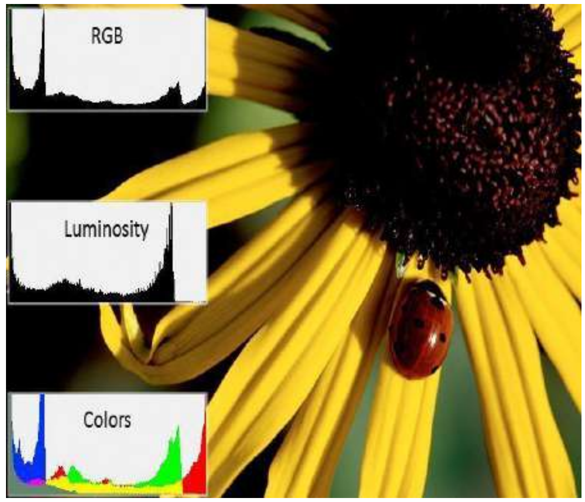


Fig 1.5.6 rgb histogram image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called **Multiresolution** theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

1.5.6 Compression:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

1.5.7 Morphological processing:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image.

Fig 1.5.7 blur to deblur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

1.5.8 Segmentation:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

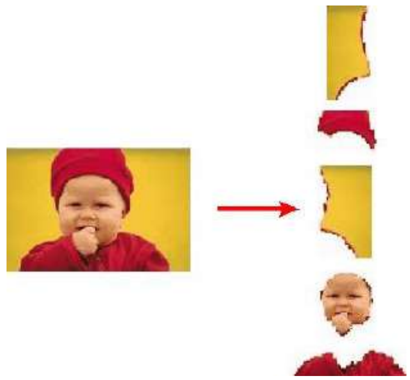


Fig 1.5.8 Image segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

1.5.9 Representation and description:

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

1.5.10 Object recognition:

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

1.5.11 Knowledgebase:

Knowledge about a problem domain is coded into image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image when the information of interests is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an inter related to list of all major possible defects in a materials inspection problem or an image data base containing high resolution satellite images of a region in connection with change deletion application. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. The system must be endowed with the knowledge to recognize the significance of the location of the string with respect to other components of an address field. This knowledge glides not only the operation of each module, but it also aids in feedback operations between modules through the knowledge base. We implemented preprocessing techniques using MATLAB.

1.6 components of an image processing system:

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 1.24 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

- **Image sensors:**

With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

- **Specialized image processing hardware:**

Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an

ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

- **Computer:**

The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

- **Image processing software:**

Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

- **Mass storage:**

Mass storage capability is a must in image processing applications. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage forimage processing applications fall into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), Gbytes (meaning giga, or one billion, bytes), and Tbytes (meaning tera, or one trillion, bytes)

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig. 1.24. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in “jukeboxes” are the usual media for archival applications.

- **Image displays:**
- Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are there requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

- **Hardcopy:**

Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

- **Network:**

Networking is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of optical fiber and other broadband technologies.

Color and texture are two low-level features widely used for image classification, indexing and retrieval. Color is usually represented as a histogram, which is a first order statistical measure that captures global distribution of color in an image One of the main drawbacks of the histogram- based approaches is that the spatial distribution and local variations in color are ignored. Local spatial variation of pixel intensity is commonly used to capture texture information in an image. Grayscale Co-occurrence Matrix (GCM) is a well-known method for texture extraction in the spatial domain. A GCM stores the number of pixel neighborhoods in an image that have a particular grayscale combination. Let I be an image and let p and Np respectively denote any arbitrary pixel and its neighbor in a given direction. If GL denotes the total number of quantized gray levels and gl denotes the individual gray levels, where, gl $\in$ {0, . . .,GL _ 1}, then each component of GCM can be written as follows:

$$\mathrm{gcm}(i, j) = \mathrm{Pr}((\mathrm{gl}_{\mathrm{p}}, \mathrm{gl}_{N_{\mathrm{p}}}) = (i, j))$$

gcm(i, j) is the number of times the gray level of a pixel p denoted by glp equals i, and the gray level of its neighbor Np denoted by glNp equals j, as a fraction of the total number of pixels in the image. Thus, it estimates the probability that the gray level of an arbitrary pixel in an image is i, and that of its neighbor is j. One GCM matrix is generated for each possible neighborhood direction, namely, 0, 45, 90 and 135.Average and range of 14 features like Angular Second Moment, Contrast, Correlation, etc., are generated by combining all the four matrices to get a total of 28 features. In the GCM approach for texture extraction, color information is completely lost since only pixel gray levels are considered.

To incorporate spatial information along with the color of image pixels, a feature called color correlogram has recently been proposed. It is a three-dimensional matrix that represents the probability of finding pixels of any two given colors at a distance ‘d’ apart Auto correlogram is a variation of correlogram, which represents the probability of finding two pixels with the same color at a distance ‘d’ apart. This approach can effectively represent color distribution in an image. However, correlogram features do not capture intensity variation Many image databases often contain both color as well as gray scale images. The color correlogram method does not constitute a good descriptor in such databases.

Another method called Color Co-occurrence Matrix (CCM) has been proposed to capture color variation in an image. CCM is represented as a three-dimensional matrix, where color pair of the pixels p and Np are captured in the first two dimensions of the matrix and the spatial distance ‘d’ between these two pixels is captured in the third dimension. This approach is a generalization of the color correlogram and reduces to the pure color correlogram for d = 1. CCM is generated using only the Hue plane of the HSV (Hue, Saturation and Intensity Value) color space. The Hue axis is quantized into HL number of levels. If individual hue values are denoted by hl, where $\mathrm{hl} \in \{0, \ldots, \mathrm{HL}-1\}$, then each component of CCM can be written as follows:

$$\mathrm{ccm}(i,j) = \Pr((\mathrm{hl}_{\mathrm{p}}, \mathrm{hl}_{N_{\mathrm{p}}}) = (i,j))$$

Four matrices representing neighbors at angles 0, 90, 180 and 270 are considered. This approach was further extended by separating the diagonal and the non-diagonal components of CCM to generate a Modified Color Co-occurrence Matrix (MCCM). MCCM, thus, may be written as follows: MCCM = (CCMD;CCMND)

Here, CCMD and CCMND correspond to the diagonal and off-diagonal components of CCM. The main drawback of this approach is that, like correlogram, it also captures only color information and intensity information is completely ignored.

An alternative approach is to capture intensity variation as a texture feature from an image and combine it with color features like histograms using suitable weights. One of the challenges of this approach is to determine suitable weights since these are highly application-dependent. In certain applications like Content-based Image Retrieval (CBIR), weights are often estimated from relevance feedback given by users.

While relevance feedback is sometimes effective, it makes the process of image retrieval user-dependent and iterative. There is also no guarantee on the convergence of the weight-learning algorithms. In order to overcome these problems, researchers have tried to combine color and texture features together during extraction.

proposed two approaches for capturing color and intensity variations from an image using the LUV color space. In the Single-channel Co-occurrence Matrix (SCM), variations for each color channel, namely, L, U and V are considered independently. In the Multi-channel Co-occurrence Matrix (MCM), variations are captured taking two channels at a time – UV, LU and LV. Since the LUV color space separates out chrominance (L and U) from luminance (V), SCM in effect, generates one GCM and two CCMs from However, in MCM, the count of pairwise occurrences of the values of different channels of the color space is captured. Thus, each component of MCM can be written as follows:

mcmUV(i; j) = Pr((up; vNp) = (i; j))

mcmLU(i; j) = Pr((lp; uNp) = (i; j))

mcmLV(i; j) = Pr((lp; vNp) = (i; j))

Here, mcmUV(i, j) is the number of times the U chromaticity value of a pixel p denoted by up equals i, and the V chromaticity value of its neighbor Np denoted by vNp equals j, as a fraction of the total number of pixels in the image. Similarly, mcmLU(i, j) and mcmLV(i, j) are defined. One MCM matrix is generated for each of the four neighborhood directions, namely, 0, 45, 90 and 135.

Deng and Manjunath (2001) proposed a two-stage method called JSEG, which combines color and texture after image segmentation. In the first stage, colors are quantized to the required levels for differentiating between various regions of an image. Pixel values of the regions are then replaced by their quantized color levels to form a color map. Spatial variation of color levels between different regions in the map is viewed as a type of texture composition of the image.

Yu et al. (2002) suggested the use of color texture moments to represent both color and texture of an image. This approach is based on the calculation of

Local Fourier Transformation (LFT) coefficients. Eight templates equivalent to LFT are operated over an image to generate a characteristic map of the image. Each template is a 3 · 3 filter that considers eight neighbors of the current pixel for LFT calculation. First and second order moments of the characteristic map are then used to generate a set of features.

In this paper, we propose an integrated approach for capturing spatial variation of both color and intensity levels in the neighborhood of each pixel using the HSV color space. In contrast to the other methods, for each pixel and its neighbor, the amount of color and intensity variation between them is estimated using a weight function. Suitable constraints are satisfied while choosing the weight function for effectively relating visual perception of color and the HSV color space properties. The color and intensity variations are represented in a single composite feature known as Integrated Color and Intensity Co-occurrence Matrix (ICICM). While the existing schemes generally treat color and intensity separately, the proposed method provides a composite view to both color and intensity variations in the same feature. The main advantage of using ICICM is that it avoids the use of weights to combine individual color and texture features. We use the ICICM feature in an image retrieval application from large image databases.

Early results on this work was reported in (Vadivel et al., 2004a). In the next section, we describe the proposed feature extraction technique after introducing some of the properties of the HSV color space. Choice of quantization levels for color and intensity axes, selection of parameter values and a brief overview of the image retrieval application

Integrated color and intensity co-occurrence matrix:

We propose to capture color and intensity variation around each pixel in a two-dimensional matrix called Integrated Color and Intensity Co-occurrence Matrix (ICICM). This is a generalization of the Grayscale Co-occurrence Matrix and the Color Co-occurrence Matrix techniques. For each pair of neighboring pixels, we consider

their contribution to both color perception as well as gray level perception to the human eye. Some of the useful properties of the HSV color space and their relationship to human color perception are utilized for extracting this feature. In the next sub-section, we briefly explain relevant properties of the HSV color space. In the subsequent subsection, we describe how the properties can be effectively used for generating ICICM.

HSV color space:

HSV Color space: Basically, there are three properties or three dimensions of color that being hue, saturation and value HSV means Hue, Saturation and Value. It is important to look at because it describes the color based on three properties. It can create the full spectrum of colors by editing the HSV values. The first dimension is the Hue. Hue is the other name for the color or the complicated variation in the color. The quality of color as determined by its dominant wavelength. This Hue is broadly classified into three categories. They are primary Hue, Secondary Hue and Teritiary Hue. The first and the foremost is the primary Hue it consists of three colors they are red, yellow and blue. The secondary Hue is formed by the combination of the equal amount of colors of the primary Hue and the colors of the secondary Hue which was formed by the primary Hue are Orange, Green and violet. The remaining one is the teritiary Hue is formed by the combination of the primary Hue and the secondary Hue. The limitless number of colors are produced by mixing the colors of the primary Hue in different amounts. Saturation is the degree or the purity of color. Then the second dimension is the saturation. Saturation just gives the intensity to the colors. The saturation and intensity drops just by mixing the colors or by adding black to the color. By adding the white to the color in spite of more intense the color becomes lighter. Then finally the third dimension is the Value. The value is the brightness of the color. When the value is zero the color space is totally black with the increase in the color there is also increase in the brightness and shows the various colors. The value describes the contrast of the color. That means it describes the lightness and darkness of the color. As similar to the saturation this value consists of the tints and shades. Tints are the colors with the added white and shades are the colors with the added black.

Properties of the HSV color space:

Sensing of light from an image in the layers of human retina is a complex process with rod cells contributing to scotopic or dim-light vision and cone cells to photopic or bright-light vision (Gonzalez and Woods, 2002). At low levels of illumination, only the rod cells are excited so that only gray shades are perceived. As the illumination level increases, more and more cone cells are excited, resulting in increased color perception. Various color spaces have been introduced to represent and specify colors in a way suitable for storage, processing or transmission of color information in images. Out of these, HSV is one of the models that separate out the luminance component (Intensity) of a pixel color from its chrominance components (Hue and Saturation). Hue represents pure color, which is perceived when incident light is of sufficient illumination and contains a single wavelength. Saturation gives a measure of the degree by which a pure color is diluted by white light. For light with low illumination, corresponding intensity value in the HSV color space is also low.

The HSV color space can be represented as a Hexa cone, with the central vertical axis denoting the luminance component, I (often denoted by V for Intensity Value). Hue, is a chrominance component defined as an angle in the range [0,2p] relative to the red axis with red at angle 0, green at 2p/3, blue at 4p/3 and red again at 2p. Saturation, S, is the other chrominance component, measured as a radial distance from the central axis of the hexacone with value between 0 at the center to 1 at the outer surface. For zero saturation, as the intensity is increased, we move from black to white through various shades of gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the purest form of the color represented by its hue. When saturation is near 0, all the pixels in an image look alike even though their hue values are different.

As we increase saturation towards 1, the colors get separated out and are visually perceived as the true colors represented by their hues. Low saturation implies presence of a large number of spectral components in the incident light, causing loss of color information even though the illumination level is sufficiently high. Thus, for low values of saturation or intensity, we can approximate a pixel color by a gray level while for higher saturation and intensity, the pixel color can be approximated by its hue. For low intensities, even for a high saturation, a pixel color is close to its gray value. Similarly, for low saturation even for a high value of intensity, a pixel is perceived as gray. We use these properties to estimate the degree by which a pixel contributes to color perception and gray level perception.

One possible way of capturing color perception of a pixel is to choose suitable thresholds on the intensity and saturation. If the saturation and the intensity are above their respective thresholds, we may consider the pixel to have color dominance; else, it has gray level dominance. However, such a hard thresholding does not properly capture color perception near the threshold values. This is due to the fact that there is no fixed level of illumination above which the cone cells get excited. Instead, there is a gradual transition from scotopic to photopic vision. Similarly, there is no fixed threshold for the saturation of cone cells that leads to loss of chromatic information at higher levels of illumination caused by color dilution. We, therefore, use suitable weights that vary smoothly with saturation and intensity to represent both color and gray scale perception for each pixel.

Non interval quantization:

Due to the large range for each component by directly calculating the characteristics for the retrieval then the computation will be very difficult to ensure rapid retrieval. It is essential to quantify HSV space component to reduce computation and improve efficiency. At the same time, because the human eye to distinguish colors is limited, do not need to calculate all segments. Unequal interval quantization according the human color perception has been applied on H , S ,V components.

Based on the color model of substantial analysis, we divide color into eight parts. Saturation and intensity is divided into three parts separately in accordance with the human eyes to distinguish. In accordance with the different colors and subjective color perception quantification, quantified hue(H), saturation(S) and value(V)

In accordance with the quantization level above, the H, S, V three-dimensional feature vector for different values of with different weights to form one dimensional feature vector and is given by the following equation:

$G = Qs*Qv*H+Qv*s+V$

Where Qs is the quantized series of S and Qv is the quantized series of V. And now by setting Qs = Qv = 3, Then $G = 9H+3S+V$

In this way three component vector of the HSV from one dimensional vector, Which quantize the whole color space for the 72 kinds of the main colors. So we can handle 72 bins of one dimensional histogram. This qualification is effective in reducing the images by the effect of the light intensity, but also reducing the computational time and complexity.

Image retrieval:

Image retrieval is nothing but a computer system used for browsing searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval use some method of adding metadata by captioning, Keywords or the descriptions to the images so that the retrieval can be performed. Manual image annotation is time consuming, expensive and laborious. For addressing this there has been a large amount of research done on automatic image annotation. It is crucial to understand the scope and nature of the image data in order to determine the complexity of the image search system design. The design is also largely dependent on the factors. And some of the factors include archives, Domain specific collection, Enterprise collection, Personal collection and web etc..,

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others. one can today generate volumes of images with content as diverse as family get-togethers and national park visits. Low-cost storage and easy Web hosting has fueled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchable image data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real-world image search system designers.

As far as technological advances are concerned, growth in Content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real world implications, applications, and constraints of the technology. Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future.

An image retrieval system designed to serve a personal collection should focus on features such as personalization, flexibility of browsing, and display methodology. For example, Google's Picasa system [Picasa 2004] provides a chronological display of images taking a user on a journey down memory lane. Domain specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals. A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods. The notable few applications include an FPGA implementation of a color-histogram-based image retrieval system [Kotoulas and Andreadis 2003], an FPGA implementation for sub image retrieval within an image database [Nakano and Takamichi 2003], and a method for efficient retrieval in a network of imaging devices [Woodrow and Heinzelman 2002].

Discussion. Regardless of the nature of the collection, as the expected user-base grows, factors such as concurrent query support, efficient caching, and parallel and distributed processing of requests become critical. For future real-world image retrieval systems, both software and hardware approaches to address these issues are essential. More realistically, dedicated specialized servers, optimized memory and storage support, and highly parallelizable image search algorithms to exploit cluster computing powers are where the future of large-scale image search hardware support lies.

Overview of texture:

We all know about the term Texture but for defining it is a hard time. One can differentiate the two different Textures by recognizing the similarities and differences. Commonly there are three ways for the usage of the Textures:

Based on the Textures the images can be segmented To differentiate between already segmented regions or to classify them.We can reproduce Textures by producing the descriptions. The texture can be analyzed in three different ways. They are Spectral, Structural and Statistical:

CHAPTER 2

LITERATURE REVIEW

Cancer is the most important disease which must be taken into consideration. The early detection of cancer can be helpful in curing the disease completely. There are some techniques available in the literature for the detection of cancer. Many researchers have contributed their idea in the detection of cancer.

Ahmad M. Sarhan [5] suggests that cancer classification is based on microarray gene expression data using DCT and ANN. He mainly deals about, a stomach cancer detection system based on Artificial Neural Network (ANN) [94,111 - 112], and the Discrete Cosine Transform (DCT). His model uses DCT to obtain classification features from stomach microarrays. The obtained features from the DCT coefficients are then employed to an ANN for classification (tumor or non tumor). Using the Stanford Medical Database he has studied a gene expression within a single sample or to evaluate gene expressions inside two tissue models, such as in tumor and non tumor tissues. Thus, a strong model for stomach cancer detection using microarrays is presented by the author. The system involves a feature extraction stage and an ANN classification stage. The feature extraction stage employs the 2D DCT to condense the input microarray. Low frequency parts of the DCT array comprises of most of the energy/information of the input microarray. These parts were, used as typical features and were obtained using a windowing approach. This work also examines through simulations, optimal parameters such as the optimal number of DCT coefficients/features and the best ANN structure for the detection of stomach cancer. His method produced a success rate of 99.7%. The sensitivity, specificity, and correctness of the system were found to be equal to 99.2%, 100%, and 99.66% correspondingly. Experimental tests on the SMD Database achieved 99.7% of recognition accuracy using only100 DCT coefficients, with a simple 2-layer ANN structure and low computational cost.

Woten et al., [120] proposed a numerical examination for the enhancement ANN in detection of breast cancer using a planar broadband antenna and a three-region breast technique. In this proposed technique, a Modified Four point antennas are used for constructing several wave polarizations. The result of wave polarization on statistical detection is fully described in this approach by the author. Mass spectrometry-based proteomics provides a significant approach for the efficient diagnosis of different diseases. However, there are some issues in the mass spectral data such as huge volume, data complexity and the presence of noise which make the investigation of the proteomic pattern very tough

Xu et al., [123] presented a neural network based approach for proteomic pattern analysis for prostate cancer screening. His approach consists of three stages namely feature selection depending on statistical significant test, classification by a Radial Basis Function Neural Network (RBFNN), a Probabilistic Neural Network (PNN), and ultimately results in optimization via ROC analysis. His approach has high sensitivity (97.1%) and specificity (96.8%) when combined with prostatic biopsy and is expected to help in early detection of prostate cancer

Fooladi et al., [32] proposed that the sensitivity to the induction of chromosomal damage by ionizing Gama Exposure is higher in breast cancer patients than in normal healthy controls. The Gamma effect in each person's lymphocytes and the comparison among two groups is examined, seventy two hours after blood example culturing, by exposing the examples to Gamma rays and then they are produced. The exposure of gamma rays cause abnormality in chromosomes. The database used in this approach includes chromosome breakage in seven chromosome groups and age of patients. In this technique Principle Component Analysis (PCA) is used for feature selection stage. Then Artificial Neural Networks (ANN) is used for classification of normal cases from abnormal cases.

Cheng et al., [22] proposed a novel texture analysis approach based on fuzzy co-occurrence matrix concept. This approach is used to handle early and exact breast cancer diagnosis by examining the microscope-slide biopsy images. A novel feature extraction algorithm is used to extract the features from the digitized images, and then the extracted features are given as input to a multilayer back-propagation neural network to categorize the images into three risk groups. The performances of the conventional cancer diagnosis methods and the proposed algorithm are evaluated and it is found that, this approach has higher performance compared to the existing methods. Cheng’s technique has wide applications in the areas of pattern recognition and image processing.

Shukla et al., [105] proposed a novel technique to simulate a Knowledge Based System for diagnosis of Breast Cancer using Soft Computing tools like Artificial Neural Networks (ANNs) and Neuro Fuzzy Systems

A vital early sign of breast cancer are Clusters of microcalcifications in mammograms. Songyang Yu Ling Guan [107] described a Computer-Aided Diagnosis (CAD) system for the automatic detection of clustered microcalcifications in digitized mammograms.

Architectural distortion is a frequently missed sign of breast cancer. Banik et al., [9] examines the detection of architectural distortion, in mammograms of interval-cancer cases taken prior to the diagnosis of breast cancer, using Gabor filters, phase portrait analysis, fractal dimension, and texture analysis. These approaches were applied to identify initial candidates for sites of architectural deformation in prior mammograms of interval-cancer and also normal cases. A total of 4212 Regions of Interest (ROIs) were routinely obtained from 106 prior mammograms of 56 interval-cancer scenarios, which includes 262 ROIs connected to architectural distortion, and from 52 prior mammograms of 13 normal cases. For every ROI, the fractal dimension and Haralick's texture features were calculated. Feature selection was obtained through stepwise logistic regression and in terms of the area under the Receiver Operating characteristics (ROC) Curve (AUC). The significant results obtained, in terms of AUC, are 0.75 with the Bayesian classifier, 0.71 with Fisher Linear Discriminant Analysis, and 0.76 with an Artificial Neural Network (ANN) based on Radial Basis Functions (RBF). The performance evaluation of the techniques with free-response receiver operating characteristics denoted a sensitivity of 0.80 at 10.5 false positives per image.

Hany Ayad Bastawrous Fukumoto et al., [43] proposed a CAD approach used for the detection of Ground Glass Opacity (GGO) nodules in chest CT images. Gabor filter on the CT image is used in this approach in order to enhance the detection process. Then some morphological process including threshold process and labeling to extract the objects having high intensity values are performed. Then, to observe these objects several feature analysis is utilized which determines that which of them are possible to be cancer candidates. A template matching among the potential cancer candidates and some Gaussian reference approaches is carried out by following the feature analysis, to find the similarity among them. The approach was applied on 715 slices containing 25 GGO nodules and achieved detection sensitivity of 92% with False Positive (FP) rate of 0.76 FP/slice. Ultimately, ANN is used to minimize the number of FP findings. The FP rate decreases to 0.25 FP/slice after applying ANN but at the expense of decreasing the detection sensitivity to 84%.

Land et al., [67] proposed an approach based on a modified form of Fogel's evolutionary programming approach for evolving neural networks for the detection of breast cancer using fine needle aspirate data. Data visualization and pre-processing description portrays the gentle and cruel raw data in graphical interpretative form. Moreover, it portrays a symmetrized dot pattern of the same data which may be used to corroborate the classification provided by the network. These developed architectures regularly achieved a classification accuracy of greater than 96% while, together, achieving a much smaller type II error. These results were attained with different data sets using the same architecture, and were also attained with the same data set over a family of developed architectures.

Mini et al., [77] presented a new approach to the problem of computeraided analysis of digital mammograms for breast cancer detection. This approach classified mammograms into normal and abnormal. The structures in mammograms formed by normal glandular tissue of diverse density were removed using a Wavelet Transform (WT) based local average subtraction. Then the linear markings produced by the normal connective tissue were identified and eliminated. Any irregularity that was present in the mammogram was hence improved in the residual image, which caused the choice concerning the normality of the mammogram much easier. Statistical descriptors based on high-order statistics obtained from the residual image were applied to a Probabilistic Neural Network (PNN) for classification. A recognition score of 71% was obtained by using the Mammographic Image Analysis Society (MIAS) database.

Baoyu Zheng et al., [10] proposed a novel Multistage Neural Network (MSNN) for locating and classifying micro-calcification in digital mammography. MSNN was trained using Back Propagation (BP) with Kalman Filtering (KF). A novel nonlinear decision approach to enhance the performance of the classification was proposed. From the experimental observation, it is clearly observed that the sensitivity of this classification/detection is 100% with the false positive detection rate of less than 1 Micro-Calcification Clusters (MCCs) per image. The proposed approaches were automatic or operator independent and offer realistic image processing time as needed for breast cancer screening programs.

Woten et al., [120] described the applicability of Artificial Neural Networks (ANNs) to breast cancer detection. A simplified sample of the breast containing a tumor was used to decide the scattering of electromagnetic waves in the microwave band. ANN was trained with this data which was then tested on new data to forecast the presence or absence of a tumor. [Synthetic variation was added to the data to realistic sample the breast.

Herold et al., [27] compares unsupervised and supervised gene selection methods. Recent mechanism learning approaches depend on matrix disintegration methods similar to Independent Component Analysis (ICA) offer innovative and well-organized investigation tools which are explored presently to evaluate gene expression outline. These tentative characteristic extraction methods gave instructive expression modes which provided indication of fundamental regulatory procedures. The gene which exhibited the strong behavior was taken for the classification of the tissue samples under examination. To verify this result, it was compared with supervised gene selection techniques which depended on numerical scores or support vector. This technique was used in macrophages loaded/de-loaded with chemically customized low density lipids.

A new nonparametric gene selection method for classification of microarray data was suggested by Lihua Ye et al., [69]. Selection of the gene was an essential step of gene appearance data investigation. An innovative nonparametric technique, Gene Selection for Multiclass (GSM), was implemented, which chose genes based on the measure of the large inter-class variation and the small intra-class variation. Using the default training and testing sets on two publicly available datasets, leukemia (two classes) and SRBCT (four classes), the method had been evaluated and compared with three relative methods, F-test, SAM and chi.

Dinc et al., [99] put forth a gene classification technique using appropriate feature selection method and Fukunaga-Koontz transform kernel. In his approach, a new algorithm related with feature selection method mostly implemented in data mining, machine learning and pattern recognition fields was proposed. Traditional Fukunaga-Koontz transform was broaden to a binary kernel classifier. This method used the cDNA microarrays to evaluate 11,000 gene expression profiles in 60 individual cancer cell lines which are utilized in a medicine discovery screen by the National Cancer Institute and Diffuse huge B-cell lymphoma data including 62 cells and higher than 4,000 genes. This approach used two stage algorithm applied on NCI60 and LYM dataset was compared with other characteristic selection models in detail.

Boyang Li et al., [13] presented a gene classification method using an improved SVM classifier with soft decision boundary. SVM classifier provided a well-organized technique to deal with a variety of categorization difficulties, because it depended on the margin maximization and arithmetical methods. Gene data was very different from other categorization data in many ways. One gene may have numerous different functions, so some gene may have more than one functional label. Since some kind of solid boundaries were generally used to categorize the data randomly in the majority of the traditional method, they were not legitimate for the data with a common part among the classes. An additional representative difficulty in gene data was data inequality that means the size of one class is greatly higher than other classes, which was the most important reason for causing the excursion of division boundary. The system defined a kind of confidence degree based on the decision values of the samples. The boundary was a categorization boundary that depends on confidence degree of data. Numerical approaches and curve fitting algorithms of SVM was utilized to categorize multi- label gene data and also deals with data inequality

Lee et al., [68] suggested a gene ranking method using text-mining for the identification of disease related genes. For the recognition of significant genes participated in particular diseases, microarray gene appearance profiles had been widely used to prioritize candidate genes. The author proposed a new gene ranking process that utilized gene-gene relations extracted from literature along with gene expression scores obtained from microarrays. Here the gene-gene relations were extracted by taking a hybrid method which was a combination of syntactic analysis and co-occurrence based approaches. Specifically, the author performed the syntactic parsing on the text and then, within each clause of the parsed sentence, the co-occurred gene names were considered to be mutually related. Both the gene network derived from the gene-gene relations obtained in the above way and the gene expression scores were given as the inputs to the Gene-Rank algorithm. For the evaluation of his approach, many experiments were conducted with the publicly available prostate cancer data. The results show that this method was superior in the precision and the recall to the original Gene-Rank which employs the genegene relations built from gene ontology annotations. Furthermore, this hybrid approach to the gene-gene relation extraction produces better prioritization of truly disease-related genes in top ranks than the existing popular cooccurrence approach.

A novel gene ranking algorithm based on random subspace method was presented by Ruichu Cai et al., [95]. Gene selection is to select the gene which contains the maximum information from the whole gene set. It's a significant preprocessing procedure for the discriminant examination of microarray information, because many of the genes were unrelated or redundant to the discriminant difficulty. The gene selection difficulty was considered as a gene ranking trouble and a Random Subspace Method based Gene Ranking (RSM-GR) algorithm was developed. In RSM-

GR, firstly subsets of the genes are randomly generated, then Support Vector Machines are respectively trained on each subset and thus produce the important factor of each gene; finally, the importance of each gene obtained from these randomly selected subsets was combined to constitute its final importance. Experiments on two public datasets show that RSM-GR obtains gene sets leading to more precise classification results than other gene selection methods, and it demanded less computational time. RSM-GR could also deal better with datasets with a large number of genes and a big number of genes to be selected.

Hero [3] put forth a gene selection and ranking with microarray data. Over many years, an explosion in the quantity of genomic data available to biomedical researchers due to advances in biotechnology. For instance, by utilizing gene microarrays, it was very easy to investigate a person's gene expression profile more than 30,000 genes of the person genome. Signals obtained from gene microarray experimentations can be associated to genetic features underlying disease, improvement, and aging in a population. This had significantly speeded up the gene detection. However, the enormous scale and investigational variability of genomic data made removal of biologically important genetic information was very challenging. One of the biggest disputes was to recognize the affected genes that were participated in that specific disease based on a gene microarray research. The authors illustrated multi criterion approaches that are proposed for this gene selection and ranking difficulty.

Shaik et al., [52] proposed an empirical CDF approach to estimate the significance of gene ranking for finding differentially expressed genes. It recommended an approach for finding importance of gene ranking. The microarray data usually had a huge number of genes that were not differentially conveyed across several circumstances. In microarray examination, it was a general practice to first reject these genes as uninformative, depends on some filtering condition. This filtering method resulted the information loss as the uninformative genes, may be used to assemble an empirical distribution of genes under the null hypothesis. The distribution of the non-differentially expressed genes is difficult and considered as a combination of distributions. The null hypothesis is that the gene is not differentially expressed. The importance of the differentially expressed genes can be approximated by using the empirical distribution function of the huge number of insignificant genes. The proposed approach was very effective, fewer computation intensive and implemented on microarray datasets of any sample size.

Yang Feng et al., [126] suggested a technique for improving robustness of gene ranking by re-sampling and permutation based score correction and normalization Feature ranking was one of the commonly used feature selection approach, which ranks features through their individual importance. Traditional feature ranking criteria were apt to produce inconsistent ranking results even with light perturbations in training samples when applied to high dimensional and small-sized gene expression data. A widely used strategy for solving the inconsistencies was the multi-criterion combination. But one problem encountered in combining multiple criteria was the score normalization. In his work, problems in existing methods are first analyzed, and a new gene importance transformation algorithm is then proposed. Experimental studies on three popular gene expression datasets showed that the multi-criterion combination based on the proposed score correction and normalization produced gene rankings with enhanced robustness.

Chen Liao et al., [20] presented a gene selection for cancer classification using Wilcoxon rank sum test and Support Vector Machine. Gene selection was an important difficulty in microarray data processing. A novel gene selection approach derived from Wilcoxon rank sum test and Support Vector Machine (SVM) was developed. Wilcoxon rank sum test was utilized to choose a subset is done first. Next step was to train and test the selected gene with the use of SVM classifier with linear kernel independently and genes with elevated testing accuracy rates are selected to form the last reduced gene subset. Leave-One-Out Cross Validation (LOOCV) categorization outputs on two datasets namely ALL/AML leukemia and breast cancer to exhibit the implemented method can obtain good result with final reduced subset. The resulted genes were listed and their expression levels were outlined to illustrate that the selected genes can produce clear separation between two classes.

Zhipeng Cai et al., [131] proposed the gene clustering to identify discriminatory genes with higher classification accuracy. A particular DNA microarray evaluated thousands to tens of thousands of gene expression levels, but investigational datasets usually contains very lower arrays of genes, most probably in tens to hundreds, taken over a selection of tissue samples. The biological understanding of these data concentrates on recognizing subsets of induced genes that could be used to differentiate numerous categories of tissue, to provide investigational evidence for connections between a subset of genes and the tissue pathology. Many methods were available to recognize discriminatory gene subsets which can be used and then it was ranked based on the classification accuracy. But the large dimensionality of the gene expression space when combined with comparatively smaller amount tissue samples which would definitely produce the dimensionality difficulty gene subsets that were too large to provide convincing evidence for any plausible causal connection between that gene subset and the tissue pathology. The author developed an innovative gene selection approach called Clustered Gene Selection (CGS). When this method was combined with the existing approaches will recognize gene subsets that overcome the dimensionality difficulty and enhance the classification accuracy. Observations on many real datasets proves that CGS can recognize many more cancer related genes and evidently enhance the classification accuracy, compared with three other non-CGS based gene selection methods.

A statistical method for ranking differentially expressed genes was recommended by Broberg [85]. Current methods have evaluated gene selection approaches by utilizing ROC curves calculated by simulation. But, no effort had been made to evaluate selection accuracy as a function of population parameters. In particular, simulation investigations are limited unavoidably when the multiplicity was considered, as only a small subset of conditions can possibly be explored. The authors summarized a technique for predicting an optimal test statistic with which to rank genes with respect to differential expression. A test of this approach demonstrated that it permitted generation of top gene lists that give a small number of false positives and a small amount of false negatives. Evaluation of the false-negative as well as the false-positive rate illustrated the main focus of this approach.

Identifying significant genes from microarray data was presented by Han-Yu Chuang et al., [44]. Microarray method was a latest improvement in investigational molecular biology which can produce quantitative expression magnitudes for large number of genes in a single, cellular mRNA sample. All these gene expression magnitudes outline a collective profile of the sample, which could be utilized to distinguish samples from dissimilar classes such as tissue types or treatments. In the gene expression profile data gathered in a specific evaluation, most likely only few genes would be differentially expressed among the classes, whereas many other genes had similar expression levels. Selecting a set of informative differential genes obtained from these large set of data was significant for microarray data analysis. The authors illustrate a framework for selecting informative genes which was called as Ranking And Combination Analysis (RAC). This method provided the combination of many existing better informative gene selection approaches. To evaluate this method they conducted many experiments using three data sets and six existing feature selection methods. The observations demonstrated that the RAC framework was a strong and effective method to categorize informative gene for microarray data. In many cases, the combined approach on two selecting techniques provided a better performance compared to the efficiency of the individual technique. Significantly, the combined approach outperformed each of the individual feature selection approach when considering all three data sets together. All these experimentation result showed that RCA was an effective and useful approach for the microarray gene expression analysis.

Kai-Bo Duan et al., [62] presented a multiple SVM-RFE for gene selection in cancer classification with expression data. It suggested an innovative feature selection method that employs a backward elimination method similar to that utilized in Support Vector Machine Recursive Feature Elimination (SVM-RFE). The proposed method was different from the SVMRFE method, as it evaluated the feature ranking score from a numerical examination of weight vectors of numerous linear Support Vector Machines trained on subsamples of the original training data in all the steps. This approach was tested on many gene expression datasets for cancer categorization. The outputs demonstrate that the implemented feature selection approach selects improved gene subsets compared to the original SVM-RFE and increases the classification accuracy. A Gene Ontology-based resemblance evaluation indicates that the selected subsets are functionally diverse, additionally validating this proposed gene selection approach. This examination also recommends that, for gene expression-based cancer classification, standard test error from several divisions of training and test sets can be suggested as an indication of performance quality.

Support vectors based correlation coefficient for gene and sample selection in cancer classification was recommended by Mundra et al., [84]. Correlation was an extensively used filter criterion for gene selection in cancer classification. However, it used all the training samples in ranking, which might not be equally significant for the classification. Using support vectors, the author demonstrated that classical correlation coefficient based gene selection was biased because of the sample points were away from classification margin. To remove such bias, this approach used only the support vectors for computation of correlation coefficient and proposed a backward elimination based SVcc-RFE algorithm. The method was tested on numerous benchmark cancer gene expression datasets and the outputs show enhancement in classification performance compared to other state-of-the-art methods.

Jin-Hyuk Hong Herold et al., [57] put forth a cancer classification with incremental gene selection based on DNA microarray data. Gene selection was significant issue for cancer classification. Filter and wrapper techniques are extensively used for gene selection, where the filter technique was hard to measure the relationship among the genes and the wrapper technique required lots of computation. The author proposed an innovative method called gene boosting which was used to select appropriate gene subsets by combining filter and wrapper approaches. This approach continuously selects a set of top-ranked informative genes using a filtering technique regarding to a chronological training dataset constructed according to the classification result for the original training dataset. Experiments conducted on few microarray benchmark datasets result showed that this technique was very efficient in predicting an appropriate gene subset. Aggressive performance was obtained with fewer genes in a reasonable time. This also led to the detection of some genes selected frequently as useful features

Runxuan Zhang et al., [97] developed Extreme Learning Machine (ELM) which was used for directing multicategory classification problems in the cancer diagnosis area. ELM evades problems such as local minima improper learning rate and overfitting. It had calculated the multicategory classification performance of ELM on three standard microarray data sets for cancer diagnosis, that is, the GCM data set, the Lung data set, and the Lymphoma data set. The results reveal that ELM created comparable or recovered classification accuracies with condensed training time and implementation complexity contrast to Artificial Neural Networks methods such as conventional back-propagation ANN, Linder's SANN, and Support Vector Machine methods like Support Vector Machine – One Versus One (SVM-OVO) and Support Vector Machine – One Versus All (SVM-OVA). ELM also attained better correctness for classification of individual categories.

A new Mutual Information (MI)-based feature-selection method to solve the so-called large p and small n problem experienced in a microarray gene expression-based data is presented by Huang et al., [48]. Firstly, a gridbased feature clustering algorithm was initiated to remove redundant features. A large gene set was then highly condensed in a very capable way. As a result, the computational effectiveness of the entire feature-selection process was largely improved. Secondly, MI was directly anticipated using quadratic MI mutually with Parzen window density estimators. This approach was capable of delivering dependable results even when only a small pattern set was accessible. Also, a novel MI-based criterion was planned to evade the greatly redundant selection results in a systematic way. Finally, credited to the direct evaluation of MI, the suitable selected feature subsets can be reasonably determined.

Extracting biologically pertinent information from DNA microarrays was a very imperative task for drug development and test, function annotation, and cancer diagnosis. Numerous clustering methods had been planned for the study of gene expression data, however when analyzing the huge and heterogeneous collections of gene expression data, conventional clustering algorithms cannot frequently create a satisfactory solution. Biclustering algorithm had been presented by Wen-Hui Yang et al., [118] as an alternative approach to standard clustering techniques to identify local structures from gene expression data set. The patterns gave clues on the most important biological processes connected with different physiological states. Unlike from existing bicluster patterns, it first introduces a further general pattern: correlated bicluster, which had instinctive biological interpretation. Subsequently, it planned a new transform technique based on singular value decomposition so that recognizing correlated-bicluster problem from gene expression matrix was changed into two global clustering problems. To produce δ-corBiclusters capably the Mixed-Clustering algorithm and the Lift algorithm are developed. The biclusters acquired using the method from gene expression data sets of many human organs and the yeast Saccharomyces cerevisiae demonstrates obvious biological meanings

Class discovery from gene expression data was a significant job for cancer diagnosis. In this work, Zhiwen Yu et al., [132] presented a new framework for class discovery. The novel framework incorporates the perturbation technique, the cluster ensemble approach, and the cluster validity index. Specially, it first produced a set of perturbed datasets from the original microarray data. Second, the Neural Gas, which provided as the basic clustering algorithm, was pertained to attain the partitions from the original dataset and the perturbed datasets. At last, a new cluster validity index called Disagreement/Agreement (DA) Index (DAI) was planned to recognize the number of classes in the dataset by considering the difference among the partition obtained from the original dataset and the partitions obtained from the perturbed datasets

Huang et al., [47] focused on effective identification of informative genes. It examined gene selection models from the perspective of optimization theory. As a result, a novel strategy was planned to change conventional search engines. Moreover, over fitting was possible to take place in microarray data because of their small sample set, a point injection technique was generated to deal with the problem of over fitting. The proposed strategies had been calculated on three kinds of cancer diagnosis.

Kai-Bo Duan et al., [62] proposed a new feature selection method that used a backward elimination procedure similar to that implemented in Support Vector Machine Recursive Feature Elimination (SVM-RFE). Not like the SVM-RFE method, at every step, the proposed approach calculated the feature ranking score from a statistical analysis of weight vectors of several linear SVMs trained on subsamples of the original training data. The proposed method was experienced on four gene expression datasets for cancer classification. A Gene Ontology-based comparison assessment showed that the selected subsets are functionally varied, further validating the gene selection method. This investigation suggested that, for gene expression-based cancer classification, average test error from several partitions of training and test sets could be recommended as a mention of presentation quality.

CHAPTER 3

SOFTWARE INTRODUCTION:

3.1. Introduction to MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

3.2 The MATLAB system:

The MATLAB system consists of five main parts

- **Development Environment**:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library**:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigenvalues, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language**:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both “programming in the small” to rapidly create quick and dirty throw-away programs, and “programming in the large” to create large and complex application programs.

- **Graphics:**

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

3.3 graphical user interface (gui):

MATLAB's Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.)and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension .fig, called a FIG-file that contains a complete graphical description of all the function's GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.

- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type

guide filename

Where filename is the name of an existing FIG-file on the current path. If filename is omitted,

GUIDE opens a new (i.e., blank) window.

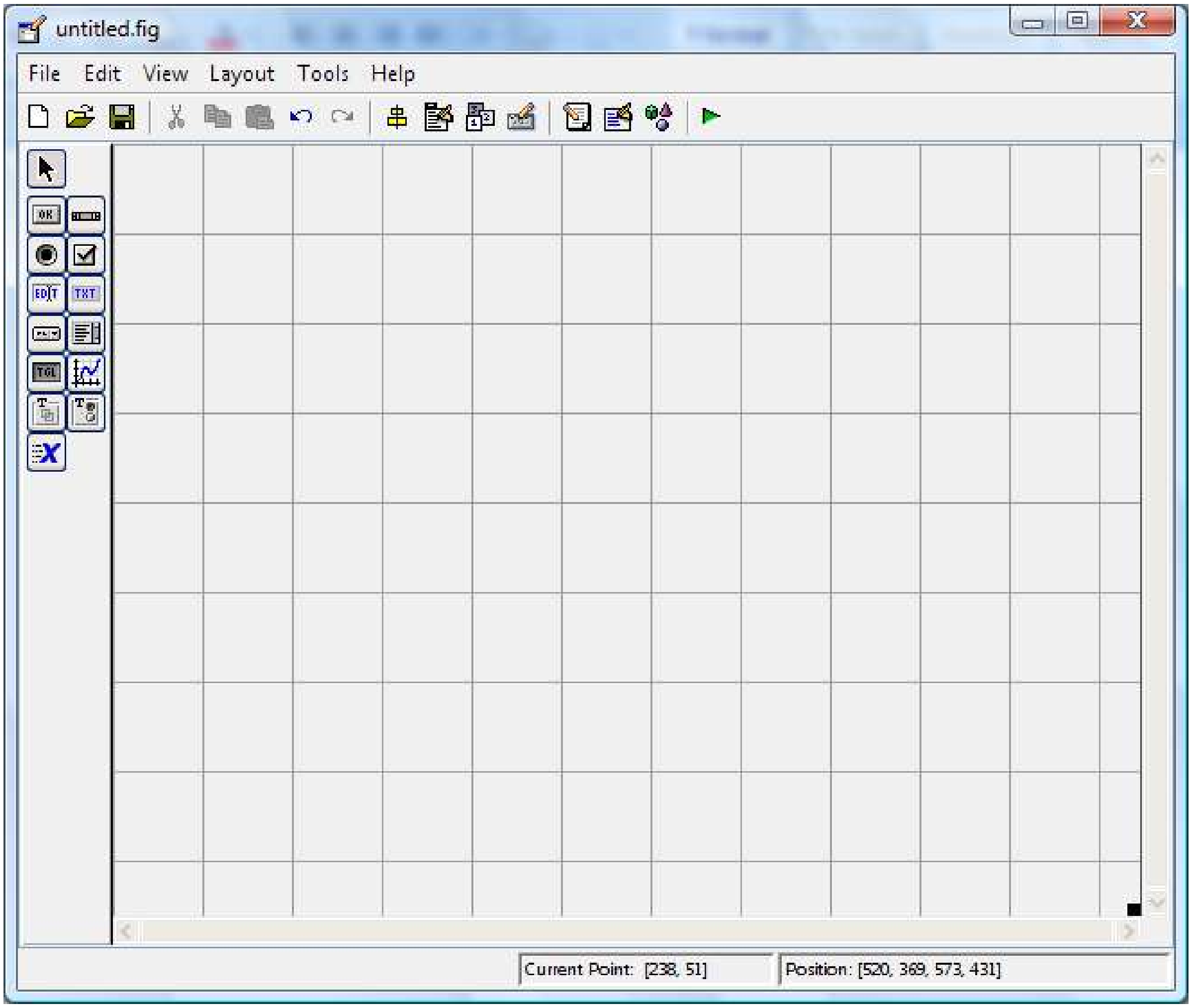


Fig-3.0 Graphical interface

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

3.4 getting started

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

3.4.1 Introduction - describes the components of the MATLAB system.

3.4.2 Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

3.4.3 Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

3.4.4 Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

3.4.5 Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

3.5 development environment

3.5.1 Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

3.5.2 Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

3.5.3 Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

3.5.4 MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

3.5.5 Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type helpbrowser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to runMATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

3.6 manipulating matrices

3.6.1 Entering Matrices

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Dürer's matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]

MATLAB displays the matrix you just entered.

A =

16 3 2 13

5 10 11 8

9 6 7 12

4 15 14 1

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

3.6.2 Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

3.6.3 Operators

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction

*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

3.6.4 Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.14159265...
I	Imaginary unit, $\sqrt{-1}$
I	Same as i

Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

3.7 GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

3.7.1 Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

3.7.2 GUI Development Environment

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of its

children (uicontrols and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the

callbacks, which are defined as subfunctions. This M-file is referred to as the

application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

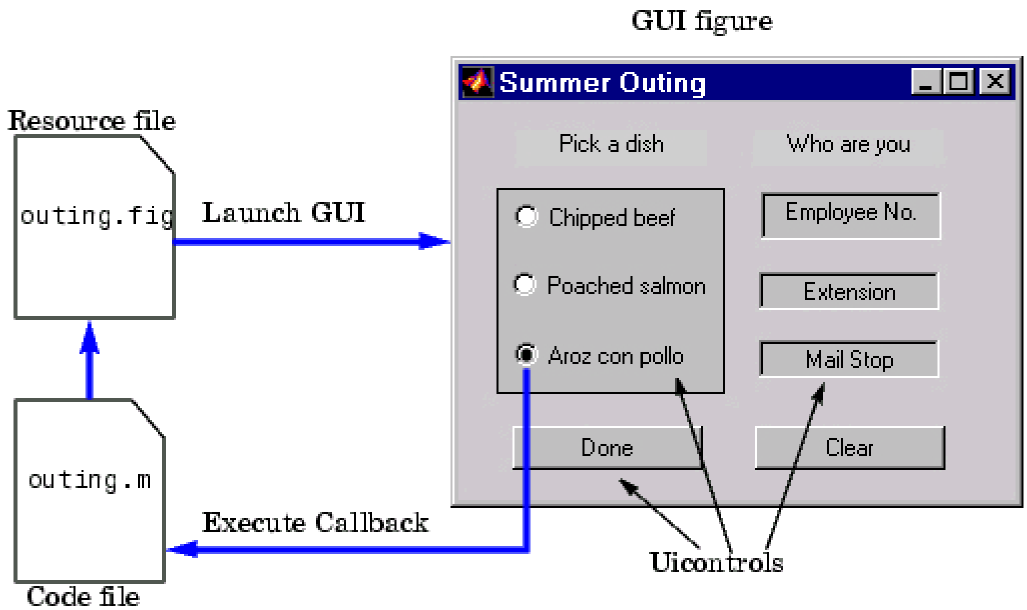


FIG 3.7.2 graphical user blocks

3.7.3 Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

3.7.4 Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques used in the application M-file.

Application Examples - a collection of examples that illustrate techniques which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

3.7.5 User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- Listboxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)

button_state = get(h,'Value');

if button_state == get(h,'Max')

% toggle button is pressed

elseif button_state == get(h,'Min')

% toggle button is not pressed
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);

a(:,:,2) = rand(16,128);

a(:,:,3) = rand(16,128);

set(h,'CData',a)
```

Radio Buttons

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)

set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)

off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];

mutual_exclude(off)

% Continue with callback

.

.

.
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)

if (get(h,'Value') == get(h,'Max'))

% then checkbox is checked-take approriate action

else

% checkbox is not checked-take approriate action

end
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)

user_string = get(h,'string');
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)

user_entry = str2double(get(h,'string'));

if isnan(user_entry)

errordlg('You must enter a numeric value','Bad Input','modal')

end

% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menubar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menubar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

Listboxes differentiate between single and double clicks on an item and set the figure SelectionType property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure SelectionType property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either:

Set the list box Callback property to the empty string ('') and remove the callback subfunction from the application M-file. Leave the callback subfunction stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

switch val

case 1

% The user selected the first item

case 2

% The user selected the second item
```

% etc.

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

string_list = get(h,'String');

selected_string = string_list{val}; % convert from cell array to string

% etc.
```

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is

grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes Button DownFcn property to define the callback.

3.7.6 Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axe

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

axes(handles.axes1)

makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axes. See GUI with Multiple Axes for and example that uses two axes.

Figure

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control.

CHAPTER 4
PROPOSED METHOD

Flow diagram of proposed work

The Fig1 shows the process flow diagram or proposed work. In this paper we used MIAS Dataset that publicly available for download and used. In the second step we pre-processed the dataset. In the pre-processing we have used three different methods for scaled and pre-process dataset. After that applied Deep Learning Neural Network and compute the accuracy. We have seen CNN achieved 98% accuracy

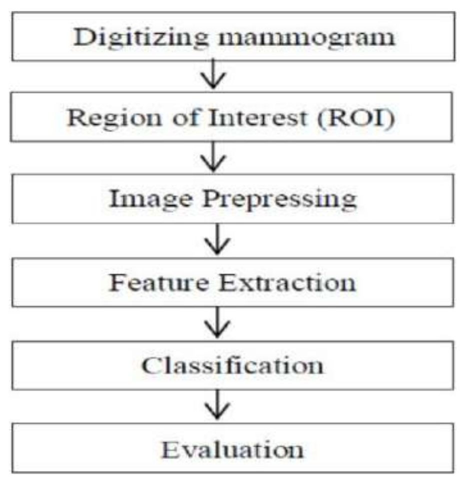


Fig:4 Process Flow Diagram

Methodology

In this paper we have used convolutional neural network for diagnosis breast cancer and also implemented same dataset on other machine learning algorithm such as Neural Network, Support Vector Machine, Random Forest. All paper consists main two parts: predict models, Preprocessing data. In this paper we have used MIAS Dataset that are publicly available for researcher this database is generated from Histopathology images that having 200images of dataset. In the experiment part first, we have pre-processed the dataset and extracted useful feature using histogram and watershed model then implement some encoder and found final training dataset. For the trained the dataset we have used deep learning model convolutional neural network algorithm and achieved 98% accuracy

Pre-processing data

As you know that data pre-processing is a data mining technique that used for filter data in a usable format. Because the real-world dataset almost available in different format. Its not available as per our requirement so its must be fitters the dataset in a understandable format Data preprocessing is a proven method of resolving such issues. Data pre-processing convert the dataset in to usable format for pre-processing we have used standardization method to preprocess the MIAS dataset.

Marker Controlled Watershed Segmentation

The watershed transforms find “Catchment basins” and “watershed ridge lines” in an image by treating it as a surface where light pixels are high and dark pixels are low. Segmentation using the watershed transform works better it we can identify or mark foreground object and background locations

Texture Segmentation Using Texture Filters

In this paper used entropy for creating texture images. Entropy is standard function that return an array value in which each output pixel contains the entropy value of the 9x9 pixel around the corresponding pixel in the input image. The following are some most used filters.

Median Filter

We have used median filter in this paper 12 features that we have used for model trained

TABLE I. TOTAL NUMBER OF SELECTED FEATURES

Sr. No	Attributes
1	Mean
2	Standard Deviation
3	Kurtosis
4	skewness
5	Entropy
6	Energy
7	Contrast
8	Correlation
9	Homogeneity
10	Concavity Mean
11	Symmetry Mean
12	Class

Fig-4.1

We have tried different 2 segmentation algorithm on same images.

Model implementation

In this stage we have implemented Deep Learning Neural Network algorithm on MIAS data set. Neural network in work on a human biological method. In which we have to pass input and get output. But in these two layer some hidden layer are work and some additional process must be added before calculate final output. These additional processes are added unit of bias, add some of additional hidden layer, calculate some activation function then the final output generated. In this paper we have used following parameters for calculating and trained model.

TABLE II. USED PARAMETERS IN CNN MODEL

Number of input	**12**
Number of Neurons	12
Activation function	Sigmoid
Number of Epochs	30

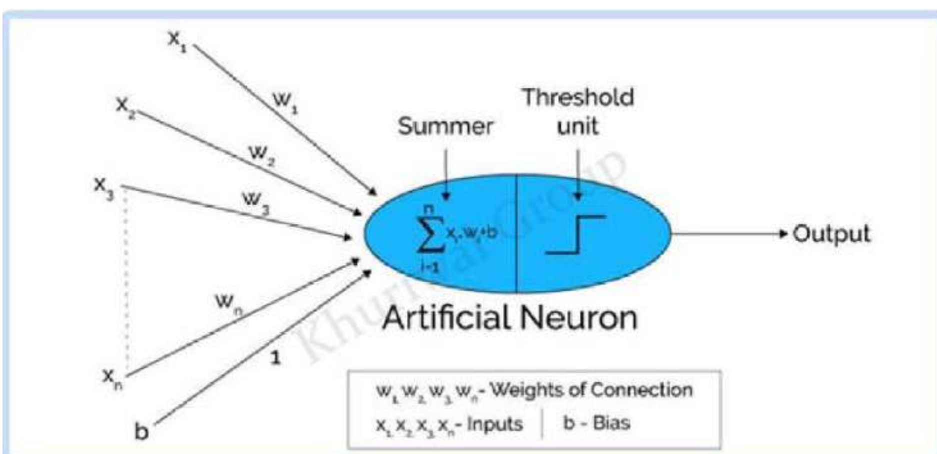


Fig. Shows Neural Network Functionality

When we want to use perceptron bias b = threshold then perceptron rule rewritten. Fig show the basic functionality of the neural network. In which after the fixed position the neurons are fired that is called threshold condition.

TABLE III. CNN MODEL SUMMARY

Layer Name	Type	Output Shape	Parameters
Convolutional Layer	Dense	12	496
Hidden Layer	Dense	8	136
Fully Connected Layer	Dense	1	9
Total	params:	641	
Trainable	params:	641	
Non-trainable	params:	0	

Fig 4.2

In this paper I have create the four-layer convolutional model in which first is input layer. Convolutional layer, hidden layer and fully connected layers. The total number of neurons is 12 for first input layers and 8 neurons for hidden layers and final 1 neurons in output layers. The following step are briefly describing the experimental in this research.

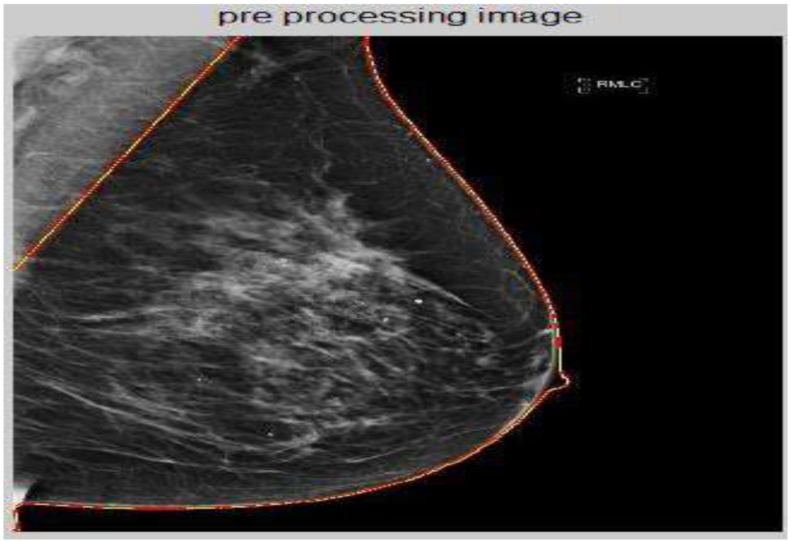
Step 1: the region of interest was extracted from the abnormal images depending on the images.

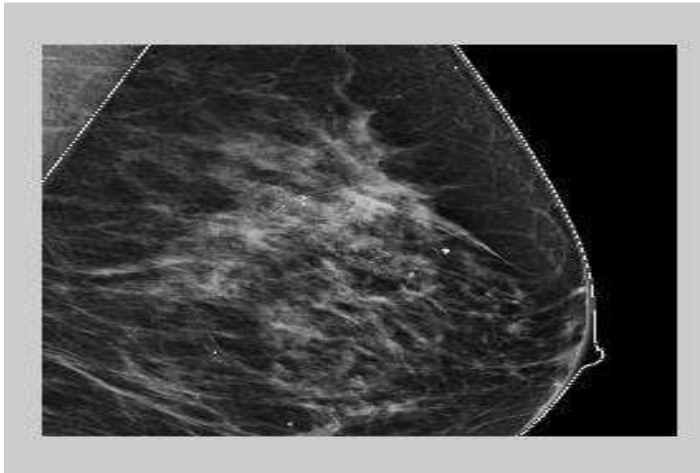
Step 2: for improve image quality we have used median filter and histogram equalization method.

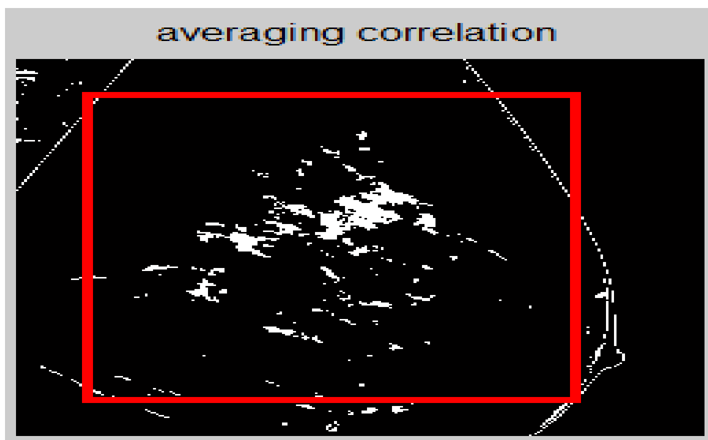
Step 3: for classification image we have used deep learning algorithm convolutional neural network.

CHAPTER 5

RESULTS






averaging correlation

CHAPTER 6

CONCLUSION AND FUTURE WORK

As we know that deep learning method convolutional neural network mostly used for image dataset classification that why we used convolutional neural network in this paper. After the implementation this paper we have achieved 98% accuracy. As we mention this paper worked on only 12 features only. In future we will try with new features and also try with the real images dataset so that we can achieved best result and accuracy for diagnosis the cancer. In feature we will also try this method on different type in cancer not only for breast cancer.

Chapter 7

REFERENCES

[1] Abdullah-Al Nahid, Aaron Mikaelian and Yinan Kong, Histopathological breast-image classification with restricted

Boltzmann machine along with backpropagation, Biomedical Research Volume 29, Issue 10, (2018).

[2] [24] Mohamad Mahmoud Al Rahhal, "Breast Cancer Classification in Histopathological Images using Convolutional Neural Network"

International Journal of Advanced Computer Science and Applications (IJACSA), 9(3), 2018.

[3] M. H. Yap et al., "Automated Breast Ultrasound Lesions Detection Using Convolutional Neural Networks," in IEEE Journal of

Biomedical and Health Informatics, vol. 22, no. 4, pp. 1218-1226, July 2018.

[4] Araújo T, Aresta G, Castro E, Rouco J, Aguiar P, Eloy C, et al. (2017) Classification of breast cancer histology images using

Convolutional Neural Networks. PLoS ONE 12(6): e0177544. https://doi.org/10.1371/journal.pone.0177544.

[5] H. Song, A. Men and Z. Jiang, "Breast tumor detection using empirical mode decomposition features," in IEEE Access.

[6] M. M. Islam, H. Iqbal, M. R. Haque and M. K. Hasan, "Prediction of breast cancer using support vector machine and K-Nearest

neighbors," 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Dhaka, 2017, pp. 226-229.

[7] R. D. Ghongade and D. G. Wakde, "Detection and classification of breast cancer from digital mammograms using RF and RF-ELM

algorithm," 2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech), Kolkata,

2017, pp. 1-6.

[8] S. Nayak and D. Gope, "Comparison of supervised learning algorithms for RF-based breast cancer detection," 2017 Computing

and Electromagnetics International Workshop (CEM), Barcelona, 2017, pp. 13-14.

[9] Y. Yang, P. A. Fasching and V. Tresp, "Predictive Modeling of Therapy Decisions in Metastatic Breast Cancer with Recurrent Neural

Network Encoder and Multinomial Hierarchical Regression Decoder," 2017 IEEE International Conference on Healthcare

Informatics (ICHI), Park City, UT, 2017, pp. 46-55.

[10] Y. Tsehay et al., "Biopsy-guided learning with deep convolutional neural networks for Prostate Cancer detection on multiparametric

MRI," 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), Melbourne, VIC, 2017, pp. 642-645.

[11] A. Alzubaidi, G. Cosma, D. Brown and A. G. Pockley, "Breast Cancer Diagnosis Using a Hybrid Genetic Algorithm for Feature

Selection Based on Mutual Information," 2016 International Conference on Interactive Technologies and Games (ITAG),

Notthingham, 2016, pp. 70-76.

[12] A. I. Pritom, M. A. R. Munshi, S. A. Sabab and S. Shihab, "Predicting breast cancer recurrence using effective classification and feature

selection technique," 2016 19th International Conference on Computer and Information Technology (ICCIT), Dhaka, 2016, pp. 310-314.

[13] A. Osareh and B. Shadgar, "Machine learning techniques to diagnose breast cancer," 2010 5th International Symposium on Health Informatics and Bioinformatics, Antalya, 2010, pp. 114-120.

[14] A. Qasem et al., "Breast cancer mass localization based on machine learning," 2014 IEEE 10th International Colloquium on Signal Processing and its Applications, Kuala Lumpur, 2014, pp. 31-36.

[15] B. M. Abed et al., "A hybrid classification algorithm approach for breast cancer diagnosis," 2016 IEEE Industrial Electronics and Applications Conference (IEACon), Kota Kinabalu, 2016, pp. 269- 274.

[16] B. M. Gayathri and C. P. Sumathi, "Comparative study of relevance vector machine with various machine learning techniques used for

detecting breast cancer," 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC),

Chennai, 2016, pp. 1-5.

[17] C. Deng and M. Perkowski, "A Novel Weighted Hierarchical Adaptive Voting Ensemble Machine Learning Method for Breast

Cancer Detection," 2015 IEEE International Symposium on Multiple- Valued Logic, Waterloo, ON, 2015, pp. 115-120.

[18] D. Bazazeh and R. Shubair, "Comparative study of machine learning

algorithms for breast cancer detection and diagnosis," 2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), Ras Al Khaimah, 2016, pp. 1-4.

[19] D. T. Saleh, A. Attia and O. Shaker, "Studying combined breast cancer biomarkers using machine learning techniques," 2016 IEEE

14th International Symposium on Applied Machine Intelligence and Informatics (SAMI), Herlany, 2016, pp. 247-251.

[20] H. R. Mhaske and D. A. Phalke, "Melanoma skin cancer detection and classification based on supervised and unsupervised learning," 2013 International conference on Circuits, Controls and Communications (CCUBE), Bengaluru, 2013, pp. 1-5. [3]

Bibliography

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 2nd Edition
by Aurélien Géron

2.Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms Paperback – 7 July 2017 by Nikhil Buduma

3. Deep Learning for Coders with fastai and PyTorch by Jeremy Howard, Sylvain Gugger
Released July 2020
Publisher(s): O'Reilly Media, Inc.
ISBN: 9781492045526

4. Breast Cancer Prediction Using Deep Learning and Machine Learning Techniques
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3558786

5. https://link.springer.com/article/10.1007/s42979-020-00305-w

6. Deep Learning to Improve Breast Cancer Detection on Screening Mammography
Scientific Reports volume 9, Article number: 12495 (2019) Cite this article

7. Explainable Deep Learning in Breast Cancer Prediction,
https://towardsdatascience.com/explainable-deep-learning-in-breast-cancer-prediction
ae36c638d2a4